



UNIVERSIDAD ESAN
FACULTAD DE INGENIERÍA
INGENIERÍA DE TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

Desarrollo de un videojuego para mejorar el nivel de comprensión lectora
en estudiantes de primaria

Tesis para obtener el Título de Ingeniero de Tecnologías de Información y Sistemas que
presenta:

Alvaro Rodrigo Machuca Breña

Asesor: Deyby Maycol Huamanchahua Canchanya

Lima, julio del 2021

Esta tesis denominada:

DESARROLLO DE UN VIDEOJUEGO PARA MEJORAR EL NIVEL DE
COMPRENSIÓN LECTORA EN ESTUDIANTES DE PRIMARIA

ha sido aprobada.



.....
Eber Joseph Ballón Álvarez (Jurado Presidente)



.....
Marks Arturo Calderón Niquin (Jurado)



.....
Wilfredo Mamani Ticona (Jurado)

DESARROLLO DE UN VIDEOJUEGO PARA MEJORAR EL NIVEL DE
COMPRENSIÓN LECTORA EN ESTUDIANTES DE PRIMARIA

Dedicatoria:

A mis padres que me dieron la vida, por ser siempre mi mejor ejemplo de esfuerzo, superación y sacrificio y por permitirme asimilar los conocimientos necesarios para poder ayudar a otras personas y hacer de este mundo un lugar mejor.

A mis hermanos Ricardo y Andrea, que siempre han sido mi apoyo incondicional, aquellos que siempre estuvieron a mi lado brindándome su ayuda y cariño incondicional en todo momento y siempre me motivaron para ser mejor.

A mis amigos Ángelo, Augusto, Raúl, Junior, GianJairo y mi mejor amigo Alexander a quienes considero como una segunda familia y con los cuales he compartido muy buenos momentos y que siempre me han apoyado en cualquier dificultad que he presentado.

A mis compañeros de trabajo que siempre me apoyaron con buenos consejos para el desarrollo del presente proyecto de tesis y fueron muy flexibles y comprensivos en los momentos que más los necesitaba especialmente agradezco a mis jefes William y Braulio que no son solo buenas personas sino excelentes profesionales que confiaron en mí dándome la oportunidad de aplicar los conocimientos adquiridos en mi vida universitaria y me ayudaron a dar mis primeros pasos en el mundo laboral.

Finalmente, quiero dedicar este trabajo con mucho cariño al señor Raúl Andonayre y a la señora Maruja Pasco que me vieron crecer desde que era un bebé y que partieron de este mundo pero que estoy seguro de que se encuentran en un lugar mucho mejor y a su vez estoy seguro de que están felices y orgullosos del profesional en el que me he convertido. Que Dios los tenga en su gloria siempre.

Alvaro Machuca Breña

ÍNDICE GENERAL

RESUMEN	1
ABSTRACT.....	2
INTRODUCCIÓN	3
CAPITULO I: PLANTEAMIENTO DEL PROBLEMA	6
1.1 Descripción de la realidad problemática.....	6
1.2 Formulación del problema	17
1.2.1 Problema General	17
1.2.2 Problemas específicos	17
1.3 Objetivos de la investigación	17
1.3.1 Objetivo General	17
1.3.2 Objetivos específicos.....	17
1.4 Variables e Hipótesis	18
1.4.1 Variables.....	18
1.4.2 Hipótesis General	18
1.4.3 Hipótesis específicas	18
1.5 Justificación de la investigación	19
1.5.1 Justificación teórica.....	19
1.5.2 Justificación práctica	20
1.5.3 Justificación Metodológica	21
1.6 Delimitación del estudio	23
1.6.1 Espacial	23
1.6.2 Temporal	23
1.6.3 Conceptual.....	24
CAPITULO II: MARCO TEÓRICO	26
2.1 Antecedentes de la investigación	26
2.2 Bases Teóricas	41
2.2.1 Videojuegos.....	41
2.2.1.1 Unity: Motor gráfico para el desarrollo de videojuegos.....	42
2.2.2 Metodologías para el desarrollo de videojuegos	43
2.2.2.1 Matriz de evaluación de metodologías de software.....	43
2.2.2.2 Metodología en cascada.....	46
2.2.2.2.1 Comunicación	46
2.2.2.2.2 Planeación	46
2.2.2.2.3 Modelado	46

2.2.2.2.4 Desarrollo.....	46
2.2.2.2.5 Despliegue.....	47
2.2.3 Aprendizaje basado en videojuegos	47
2.2.4 Videojuego como herramienta de aprendizaje en el aula.....	49
2.2.4.1 Beneficios para el docente	49
2.2.5 La comprensión lectora	49
2.2.5.1 Niveles de la comprensión lectora.....	50
2.2.5.2 Importancia de la enseñanza de comprensión de los textos	50
2.2.5.3 La Lectura Comprensiva como tema del aprendizaje de textos	51
2.2.6 Nueva metodología de enseñanza bajo la pandemia de Covid-19	51
2.2.7 Estrategias para mejorar la comprensión lectora.....	53
2.2.8 Realidad Virtual	53
2.2.8.1 Definición y características	53
2.2.8.2 Tipos de realidad virtual	54
2.2.9 Base de Datos	54
2.2.9.1 Definición	55
2.2.9.2 Bases de datos No relacionales.....	56
2.2.9.2.1 Firebase Database	57
2.2.10 Hosting o Alojamiento Web.....	58
2.2.10.1 Node.js.....	59
2.2.11 Deep Learning	60
2.2.11.1 Redes Neuronales	60
2.2.11.1.1 El Perceptrón.....	61
2.2.11.1.2 Función de activación	63
2.2.11.1.3 Multilayer Perceptron (MLP)	65
2.2.11.1.4 Algoritmo de Back-Propagation	66
2.2.11.1.5 Ejemplo aplicativo	73
2.2.12 Machine Learning	83
2.2.12.1 Aprendizaje por refuerzo	85
2.2.12.1.1 Ejemplo aplicativo	87
2.2.12.1.2 Deep Reinforcement Learning.....	88
2.2.12.1.2.1 Proximal Policy Optimization (PPO).....	91
2.2.13 Técnica estadística para el análisis de resultados.....	96
2.2.13.1 Formulación de las hipótesis	96
2.2.13.2 P- valor	96

2.2.13.3 Nivel de significancia	96
2.2.13.4 Decisión de aceptación o rechazo de la hipótesis.....	97
2.2.13.5 Prueba T para dos muestras independientes	97
2.2.13.6 Metodología para la obtención del estadístico T de Student	97
2.3 Marco Conceptual o Contexto de Investigación	100
2.3.1 Marco Conceptual	100
2.3.2 Contexto de Investigación.....	101
CAPITULO III: METODOLOGÍA DE LA INVESTIGACIÓN	104
3.1 Diseño de Investigación.....	104
3.1.1 Diseño.....	104
3.1.2 Tipo	105
3.1.3 Enfoque	105
3.1.4 Población y muestra	106
3.1.4.1 Justificación de división en grupos de control y experimental.....	107
3.1.5 Operacionalización de las variables	109
3.2 Instrumentos de medida	111
3.2.1 Pretest y postest.....	111
3.2.2 Encuesta de satisfacción.....	112
3.3 Técnica de recolección de datos	113
3.3.1 Entrevista.....	113
3.4 Criterio de selección de la metodología para el desarrollo del videojuego	113
3.4.1 Población de metodologías de desarrollo de software	114
3.4.2 Metodologías utilizadas en papers de videojuegos	114
3.4.3 Resultados de matriz de evaluación de metodologías	114
3.5 Criterio de selección del tipo de realidad virtual a utilizar	116
3.6 Criterio de selección de base de datos a utilizar	117
3.7 Criterio de selección de técnica de Inteligencia Artificial	118
3.7.1 Selección del tipo de red neuronal	119
3.8 Criterio de selección de técnica estadística para el análisis de resultados.....	120
3.9 Etapas del proyecto de investigación.....	123
3.9.1 Desarrollo del videojuego	123
3.9.2 Protocolo de aprendizaje	126
3.9.3 Entrenamiento de los estudiantes	127
3.9.4 Presentación de resultados	128
3.10 Metodología de implementación de la solución	130

3.10.1 Metodología en cascada	130
3.10.1.1 Comunicación	131
3.10.1.1.1 Actividad 1: Recopilación de requerimientos del docente	131
3.10.1.1.2 Actividad 2: Clasificación de requerimientos	132
3.10.1.2 Planeación	133
3.10.1.2.1 Actividad 1: Establecimiento de recursos de trabajo	133
3.10.1.2.2 Actividad 2: Desarrollo de cronograma de trabajo	134
3.10.1.3 Modelado	135
3.10.1.3.1 Actividad 1: Desarrollo de casos de uso	135
3.10.1.3.2 Actividad 2: Desarrollo de bosquejos del prototipo	135
3.10.1.3.3 Actividad 3: Desarrollo de la arquitectura de las escenas	136
3.10.1.3.4 Actividad 4: Desarrollo de secuencia de Gameplay	136
3.10.1.3.5 Actividad 5: Descargas de modelos en 3D	136
3.10.1.4 Desarrollo	139
3.10.1.4.1 Actividad 1: Desarrollo de las escenas del videojuego	139
3.10.1.4.2 Actividad 2: Implementación de tecnologías	139
3.10.1.4.3 Actividad 3: Implementación de técnica de Inteligencia Artificial ...	140
3.10.1.4.4 Actividad 4: Implementación de técnicas de aprendizaje	141
3.10.1.4.5 Actividad 5: Pruebas de funcionalidad individual y general	141
3.10.1.5 Despliegue	143
3.10.1.5.1 Actividad 1: Entrega del producto desarrollado	143
3.10.1.5.2 Actividad 2: Retroalimentación del proyecto	143
3.11 Descripción general del prototipo	145
3.12 Implementación de realidad virtual no inmersiva	148
3.13 Implementación de la base de datos en tiempo real de Firebase	149
3.13.1 Conexión con Unity	149
3.14 Implementación de Deep Reinforcement Learning (DRL)	152
3.14.1 Creación del agente inteligente	153
3.14.2 Creación del ambiente de entrenamiento	156
3.14.3 Arquitectura y componentes del Multilayer Perceptrón	156
3.14.3.1 Configuración del Multilayer Perceptrón desde Unity	159
3.14.3.2 Diseño y selección de mejor arquitectura de red	159
3.14.3.3 Interpretación de las salidas de la red sobre el agente	163
3.14.3.4 Aplicación del aprendizaje por refuerzo	164
3.14.3.4.1 Otorgamiento de premios y castigos al agente	164

3.14.4 Desarrollo de experimentos.....	164
3.14.4.1 Modelos de DRL utilizados y aplicación del algoritmo de PPO.....	165
3.14.4.2 Explicación de Hiperparámetros y rangos sugeridos por Unity.....	169
3.14.4.3 Archivo Trainer_config.yaml y ejecución de modelo de DRL.....	171
3.15 Implementación del servicio de Hosting.....	174
3.16 Implementación de técnicas para el aprendizaje de la comprensión lectora.....	175
3.16.1 Narración de historia y oraciones dentro del videojuego.....	175
3.16.2 Adaptación de textos educativos dentro del videojuego.....	176
3.16.2.1 Configuración del contenido para cada grado.....	176
3.16.3 Sección de estrategias de lectura.....	177
3.17 Aplicación del aprendizaje basado en videojuegos.....	178
3.17.1 Desarrollo del protocolo de aprendizaje.....	178
3.18 Metodología para la medición de los resultados.....	179
3.18.1 Medición de resultados - Metodología aplicada al videojuego.....	179
3.18.2 Medición de resultados – Experimentos de DRL.....	180
3.18.2.1 Resultados generales de los modelos de DRL desarrollados.....	181
3.18.2.2 Resultados del mejor modelo de DRL.....	185
3.18.2.2.1 Resultados del entrenamiento del agente por episodio.....	185
3.18.2.2.2 Tabla de entradas y salidas de la red.....	186
3.18.2.2.3 Matriz de pesos.....	186
3.18.3 Medición de resultados - Entrenamiento de los estudiantes.....	186
3.18.3.1 Análisis cuantitativo de las evaluaciones por cada grado.....	187
3.18.3.1.1 Resumen de resultados generales de las evaluaciones.....	188
3.18.3.2 Análisis cualitativo de la encuesta de satisfacción.....	188
3.19 Cronograma de actividades.....	189
CAPITULO IV: DESARROLLO DE LA SOLUCIÓN.....	191
4.1 Determinación y evaluación de las alternativas de solución.....	191
4.1.1 Propuesta de solución.....	193
4.2 Desarrollo del videojuego.....	194
4.2.1 Fase de comunicación.....	194
4.2.2 Fase de planeación.....	196
4.2.3 Fase de modelado.....	198
4.2.3.1 Creación y especificación de casos de uso.....	198
4.2.3.2 Creación de bosquejos y arquitectura de las escenas.....	206
4.2.3.3 Secuencia de Gameplay del videojuego.....	210

4.2.3.4 Modelos en 3D descargados	212
4.2.4 Fase de desarrollo.....	214
4.2.4.1 Desarrollo de las escenas	214
4.2.4.2 Implementación de realidad virtual no inmersiva	219
4.2.4.3 Implementación de la base de datos de Firebase	220
4.2.4.4 Implementación de Deep Reinforcement Learning.....	223
4.2.4.4.1 Desarrollo de experimentos	228
4.2.4.5 Implementación de técnicas de aprendizaje	233
4.2.4.6 Clasificación de contenido para cada grado	235
4.2.4.7 Implementación del servicio de hosting	237
4.2.4.8 Pruebas de funcionalidad.....	238
4.2.5 Fase de despliegue.....	240
4.2.5.1 Retroalimentación del proyecto realizado	240
4.3 Protocolo de aprendizaje.....	241
4.4 Entrenamiento de los estudiantes.....	244
CAPITULO V: ANÁLISIS Y DISCUSIÓN DE RESULTADOS	245
5.1 Medición de resultados – Metodología aplicada al videojuego.....	245
5.2 Medición de Resultados – Experimentos de DRL.....	249
5.2.1 Resultados generales de los modelos de DRL desarrollados	250
5.2.2 Resultados del mejor modelo de DRL seleccionado.....	263
5.2.2.1 Resultados del entrenamiento del agente por episodio.....	263
5.2.2.2 Tabla de entradas y salidas de la red	267
5.2.2.3 Matriz de pesos	268
5.3 Medición de Resultados – Entrenamiento de los estudiantes	270
5.3.1 Análisis cuantitativo descriptivo de las evaluaciones	270
5.3.1.1 Análisis cuantitativo descriptivo – Primer Grado.....	271
5.3.1.2 Análisis cuantitativo descriptivo – Segundo Grado	275
5.3.1.3 Análisis cuantitativo descriptivo – Tercer Grado	278
5.3.1.4 Análisis cuantitativo descriptivo – Cuarto Grado.....	282
5.3.1.5 Análisis cuantitativo descriptivo – Quinto Grado	285
5.3.1.6 Análisis cuantitativo descriptivo – Sexto Grado	289
5.3.2 Análisis cuantitativo inferencial de las evaluaciones.....	293
5.3.2.1 Ejemplo del análisis cuantitativo inferencial aplicado al primer grado.....	293
5.3.2.2 Resumen de los resultados obtenidos para cada grado.....	299
5.3.3 Análisis cualitativo de la encuesta de satisfacción.....	302

5.4 Discusión de los resultados	307
CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES	315
6.1 Conclusiones	315
6.2 Recomendaciones	321
Referencias.....	325
Anexos	331
Anexo 1. Organigrama de la I.E.P Colegio “Mater Christi”	331
Anexo 2. Material educativo para la enseñanza de las clases de lectura	332
Anexo 3. Clases de lectura antes de la pandemia del Covid-19.....	333
Anexo 4. Desarrollo de ejercicios de comprensión lectora	334
Anexo 5. Encuesta de videojuego de Moreno, Vahos, y Mazo (2019).....	335
Anexo 6. Instrumento de validez de encuesta de videojuego	336
Anexo 7. Matriz de consistencia	337

LISTA DE FIGURAS

Figura N°1: Primeros y últimos puestos de la prueba PISA 2012.....	7
Figura N°2: Resultados en las 3 áreas de la prueba PISA 2015 en países de América.	8
Figura N°3: Puntajes de las pruebas PISA del 2009,2012 y 2015 en el Perú.....	9
Figura N°4: Puntajes de las pruebas PISA 2015 y 2018 en el Perú.....	10
Figura N°5: Videojuegos en la actualidad.	12
Figura N°6: Escena de videojuego FIFA 2020.	13
Figura N°7: PlayApartTogether: Una medida preventiva ante la actual pandemia.	14
Figura N°8: Escena de videojuego Call of Duty 4 : Modern Warface	15
Figura N°9: Tendencia de artículos relacionados a videojuegos entre el 2013 y 2018.	26
Figura N°10: Escenas del videojuego desarrollado.	28
Figura N°11: Fases del proyecto desarrollado	29
Figura N°12: Utilización del videojuego en las aulas de clases.	29
Figura N°13: Diagrama de casos de uso y diagrama de clases.....	31
Figura N°14: Visualización de videojuego en la web.....	31
Figura N°15: Metodología en cascada aplicada a videojuegos.	32
Figura N°16: Visualización del videojuego terminado.....	33
Figura N°17: Login del videojuego (izq.) y Registro de alumnos (Der.).	34
Figura N°18: Metodología Scrum aplicada a videojuegos.	35
Figura N°19: Ambiente desarrollado para el entrenamiento del agente.	37
Figura N°20: Implementación de aprendizaje por refuerzo profundo usando ML-Agents.	38
Figura N°21: Porcentaje de participación de Unity en el mercado.....	43
Figura N°22: Matriz de evaluación de metodologías.	45
Figura N°23: Modelo de cascada.....	47
Figura N°24: Educación a distancia en el Perú por la pandemia del Covid-19.....	52
Figura N°25: Modelo entidad-relación.	56
Figura N°26: Servicio de Hosting Web.	59
Figura N°27: Modelo de un perceptrón.	61
Figura N°28: Modelo del perceptrón incluyendo el bias.	63
Figura N°29: Función Sigmoide.	64
Figura N°30: Función RELU.	65
Figura N°31: Gráfico del flujo de salida para una neurona j.	66
Figura N°32: Gráfico del flujo de salida para una neurona k conectada a una neurona j.....	70

Figura N°33: Estructura de red para ejemplo aplicativo.....	74
Figura N°34: Valores de los inputs, Outputs y pesos de la red.	74
Figura N°35: Representación visual de regla de la cadena en la capa de salida.....	78
Figura N°36: Representación visual de regla de la cadena en la capa oculta.	81
Figura N°37: Tipos de Machine Learning.	85
Figura N°38: Flujo de entrenamiento en el Aprendizaje por refuerzo.....	86
Figura N°39: Aprendizaje por refuerzo aplicado a videojuegos.....	87
Figura N°40: Ejemplo aplicativo del aprendizaje por refuerzo.	87
Figura N°41: Flujo de trabajo de DRL aplicado a videojuegos.....	88
Figura N°42: Proceso de decisión de Markov.	89
Figura N°43: Flujo de trabajo para los grupos de control y experimental.....	109
Figura N°44: Secuencia de actividades para implementar los instrumentos de medición. ...	111
Figura N°45: Resultados de la matriz de evaluación de metodologías.....	115
Figura N°46: Selección de prueba estadística a utilizar.	122
Figura N°47: Diagrama de flujo para el desarrollo de la primera etapa.	125
Figura N°48: Secuencia de actividades para el desarrollo del protocolo de aprendizaje.	127
Figura N°49: Secuencia de actividades para el desarrollo del periodo de entrenamiento.	128
Figura N°50: Secuencia de actividades para el análisis de las evaluaciones a los estudiantes.	130
Figura N°51: Metodología en cascada aplicado al presente proyecto de investigación.	131
Figura N°52: Resumen gráfico de la metodología para la implementación de la solución...	144
Figura N°53: Descripción general del prototipo.....	146
Figura N°54: Recomendaciones para niños para prevenir el Covid-19.....	148
Figura N°55: Configuración del Input Manager para configuración del teclado.	149
Figura N°56: Flujo de la conexión entre la base de datos de Firebase y Unity.	152
Figura N°57: Flujo de trabajo para la aplicación de DRL en el prototipo.....	153
Figura N°58: Arquitectura general de la red neuronal a utilizar.....	157
Figura N°59: Configuración de la red neuronal a través de Unity.	159
Figura N°60: Funcionamiento de capas oculta en red Multilayer Perceptrón.....	161
Figura N°61: Lógica de entrenamiento y selección de la mejor arquitectura de red.....	162
Figura N°62: Estructura interna del Archivo Trainer_config.yaml.	172
Figura N°63: Modelo de DRL entrenado.	173
Figura N°64: Utilización del modelo entrenado dentro del Agente.	174
Figura N°65: Ejemplo de los resultados del Tensorboard para los experimentos de DRL. ..	185

Figura N°66: Resumen de las actividades para el análisis cuantitativo descriptivo.	187
Figura N°67: Secuencia de actividades para aplicar la prueba T para muestras independientes.	188
Figura N°68: Cronograma general para el desarrollo del proyecto de investigación.	190
Figura N°69: Recolección de requerimientos del docente.	195
Figura N°70: Diagrama de casos de uso del videojuego.	198
Figura N°71: Especificación de caso de uso CUN01.	200
Figura N°72: Especificación de caso de uso CUN02.	201
Figura N°73: Especificación de caso de uso CUN03.	202
Figura N°74: Especificación de caso de uso CUN04.	203
Figura N°75: Especificación de caso de uso CUN05.	204
Figura N°76: Especificación de caso de uso CUN06.	205
Figura N°77: Bosquejo inicial del Login del videojuego.	206
Figura N°78: Bosquejo inicial del registro de los alumnos.	206
Figura N°79: Bosquejo inicial del menú principal del videojuego.	207
Figura N°80: Bosquejo inicial de los niveles del videojuego.	207
Figura N°81: Arquitectura del Login del videojuego.	208
Figura N°82: Arquitectura de escena de Registro de alumnos.	208
Figura N°83: Arquitectura del menú general del videojuego	209
Figura N°84: Arquitectura de un nivel del videojuego.	210
Figura N°85: Secuencia de Gameplay desarrollada.	211
Figura N°86: Personajes enemigos del videojuego.	212
Figura N°87: Terrenos en 3D del videojuego.	213
Figura N°88: Modelo de armas en 3D usados en el videojuego.	214
Figura N°89: Escena construida para el Login de videojuego.	214
Figura N°90: Escena construida para el registro de los estudiantes.	215
Figura N°91: Escena construida para el menú principal del videojuego.	216
Figura N°92: Escena construida para visualizar las instrucciones del videojuego.	216
Figura N°93: Nivel del videojuego y visualización del score y barra de vidas.	217
Figura N°94: Pausa desarrollada para el videojuego.	218
Figura N°95: Créditos finales (Izq.), Recomendaciones prevenir el Covid-19 (Der.).	218
Figura N°96: Importación de Standard Assets para implementar un personaje en primera persona.	219
Figura N°97: Implementación de un personaje en primera persona en el videojuego.	220

Figura N°98: Creación de una nueva base de datos.....	221
Figura N°99: Creación de la clase Usuario.....	222
Figura N°100: Funciones de EnviarPorPost() y RecibirDeBD().....	223
Figura N°101: Descarga de la librería ML-Agents de Unity.....	224
Figura N°102: Entradas a la red neuronal.....	225
Figura N°103: Otorgamiento de refuerzos positivos y negativos al agente.....	226
Figura N°104: Configuración de la red neuronal desde Unity para el entrenamiento del agente.....	227
Figura N°105: Ambiente de entrenamiento construido para el entrenamiento del agente. ...	228
Figura N°106: Modelos de DRL insertados en el archivo Trainer_config.yaml.....	229
Figura N°107: Entrenamiento de agentes inteligentes.....	230
Figura N°108: Implementación del mejor modelo entrenado al agente.....	231
Figura N°109: Asignación del personaje controlado por el jugador al objetivo del agente....	232
Figura N°110: Agente inteligente implementado en el videojuego desarrollado.....	232
Figura N°111: Narración de historia (izq.) y oraciones dentro de cada nivel (der.).....	233
Figura N°112: Aparición de lecturas (izq.) y preguntas de opción múltiple (der.)	234
Figura N°113: Sección de estrategias de lectura dentro del videojuego.....	235
Figura N°114: Búsqueda de Tag correspondiente para redireccionar a escenas específicas.	236
Figura N°115: Archivos creados por Unity en la exportación.....	237
Figura N°116: Reproducción del videojuego final en la computadora.	238
Figura N°117: Prueba de envío de información a la base de datos.	239
Figura N°118: Verificación de almacenamiento de información en la base de datos.	239
Figura N°119: Entrenamiento de estudiantes - Primer grado (izq.), Segundo grado (centro), Tercer grado (Der.).....	244
Figura N°120: Entrenamiento de estudiantes - Cuarto grado (izq.), Quinto grado (centro), Sexto grado (Der.).....	244
Figura N°121: Resultados de los modelos de DRL desarrollados – Cumulative Reward.....	250
Figura N°122: Resultados de los modelos de DRL desarrollados – Episode Length.....	251
Figura N°123: Resultados de los modelos de DRL desarrollados – IsTraining.....	252
Figura N°124: Resultados de los modelos de DRL desarrollados – PolicyLoss.....	253
Figura N°125: Resultados de los modelos de DRL desarrollados – ValueLoss.....	254
Figura N°126: Resultados de los modelos de DRL desarrollados – Entropy.....	256
Figura N°127: Resultados de los modelos de DRL desarrollados – Extrinsic Reward.....	257

Figura N°128: Resultados de los modelos de DRL desarrollados – Extrinsic Value Estimate.	258
Figura N°129: Resultados de los modelos de DRL desarrollados – Learning Rate.	259
Figura N°130: Resultados del análisis cualitativo – Experiencia de uso del videojuego.	303

LISTA DE TABLAS

Tabla 1: Resumen de mejora porcentual con respecto a las 2 últimas pruebas PISA	10
Tabla 2: Resumen de puntajes por área y puestos de Perú en las pruebas PISA.....	11
Tabla 3: Acciones a realizar por el agente del videojuego	36
Tabla 4:Resumen de antecedentes de videojuegos en un contexto educacional	39
Tabla 5:Resumen de metodologías aplicadas, nivel de significancia establecido y mejora significativa obtenida en los antecedentes de videojuegos en un contexto educacional	40
Tabla 6: Ponderaciones para la elección de la metodología adecuada	44
Tabla 7:Beneficios de la aplicación del aprendizaje basado en videojuegos.....	48
Tabla 8:Niveles de la comprensión lectora.....	50
Tabla 9: Población y muestra del proyecto de investigación.....	107
Tabla 10: Operacionalización de la variable dependiente e independiente	110
Tabla 11: Estructura general de pretest y postest.....	112
Tabla 12:Resumen de tipo de realidad virtual aplicada en los trabajos de investigación.....	116
Tabla 13:Operaciones aplicadas a una base de datos.....	118
Tabla 14:Resumen de técnica de Inteligencia Artificial aplicada en los trabajos de investigación	119
Tabla 15:Resumen de tipo de arquitectura de red aplicada en los trabajos de investigación	119
Tabla 16: Arquitecturas de red utilizados en videojuegos tradicionales de mercado.	120
Tabla 17: Resumen de técnica estadística utilizada en los trabajos de investigación.....	121
Tabla 18: Resumen de actividades a realizar en la etapa de comunicación.....	133
Tabla 19: Resumen de actividades a realizar en la etapa de planeación.....	134
Tabla 20: Resumen de actividades a realizar en la etapa de modelado	138
Tabla 21: Resumen de actividades a realizar en la etapa de desarrollo	142
Tabla 22: Resumen de actividades a realizar en la etapa de despliegue	143
Tabla 23: Descripción de componentes de la red neuronal	157
Tabla 24: Arquitecturas de Multilayer Perceptrón para cada modelo de DRL.....	160
Tabla 25: Interpretación de las salidas de la red para el desplazamiento del agente	163
Tabla 26: Hiperparámetros con valores constantes para los 7 modelos de DRL.....	166
Tabla 27: Hiperparámetros con valores variables para los 7 modelos de DRL.....	168
Tabla 28: Especificaciones técnicas del equipo a utilizar para el desarrollo del prototipo ...	197
Tabla 29: Participantes del proyecto de investigación.....	197
Tabla 30: Descripción general de los casos de uso del videojuego	199

Tabla 31: Protocolo de aprendizaje desarrollado para el entrenamiento de los estudiantes de primaria.....	243
Tabla 32: Resultados de la metodología en cascada – Fase de Comunicación	245
Tabla 33: Resultados de la metodología en cascada – Fase de Planeación	246
Tabla 34: Resultados de la metodología en cascada – Fase de Modelado.....	247
Tabla 35: Resultados de la metodología en cascada – Fase de Desarrollo.....	249
Tabla 36: Resultados generales del entrenamiento de los modelos de Deep Reinforcement Learning	261
Tabla 37: Resultados del mejor modelo de DRL por episodio de entrenamiento	264
Tabla 38: Entradas y salidas iniciales de la red neuronal durante el entrenamiento.....	268
Tabla 39: Matriz de pesos iniciales antes del entrenamiento.....	269
Tabla 40: Matriz de pesos de la capa de salida antes del entrenamiento.....	269
Tabla 41: Matriz de pesos actualizados al final el entrenamiento	270
Tabla 42: Matriz de pesos de la capa de salida después del entrenamiento	270
Tabla 43: Resumen estadístico del pretest para el primer grado	271
Tabla 44: Resumen estadístico del postest para el primer grado	271
Tabla 45: Resultados de indicadores del pretest para el grupo de control – Primer grado....	272
Tabla 46: Resultados de indicadores del pretest para el grupo experimental – Primer grado	273
Tabla 47: Resultados de indicadores del postest para el grupo de control – Primer grado ...	273
Tabla 48: Resultados de indicadores del postest para el grupo experimental – Primer grado	274
Tabla 49: Resumen estadístico del pretest para el segundo grado.....	275
Tabla 50: Resumen estadístico del postest para el segundo grado	275
Tabla 51: Resultados de indicadores del pretest para el grupo de control – Segundo grado.	276
Tabla 52: Resultados de indicadores del pretest para el grupo experimental – Segundo grado	276
Tabla 53: Resultados de indicadores del postest para el grupo de control – Segundo grado	277
Tabla 54: Resultados de indicadores del postest para el grupo experimental – Segundo grado	278
Tabla 55: Resumen estadístico del pretest para el tercer grado	278
Tabla 56: Resumen estadístico del postest para el tercer grado.....	279
Tabla 57: Resultados de indicadores del pretest para el grupo de control – Tercer grado	279
Tabla 58: Resultados de indicadores del pretest para el grupo experimental – Tercer grado	280

Tabla 59: Resultados de indicadores del postest para el grupo de control – Tercer grado....	281
Tabla 60: Resultados de indicadores del postest para el grupo experimental – Tercer grado	281
Tabla 61: Resumen estadístico del pretest para el cuarto grado	282
Tabla 62: Resumen estadístico del postest para el cuarto grado.....	283
Tabla 63: Resultados de indicadores del pretest para el grupo de control – Cuarto grado....	283
Tabla 64: Resultados de indicadores del pretest para el grupo experimental – Cuarto grado	284
Tabla 65: Resultados de indicadores del postest para el grupo de control – Cuarto grado ...	284
Tabla 66: Resultados de indicadores del postest para el grupo experimental – Cuarto grado	285
Tabla 67: Resumen estadístico del pretest para el quinto grado	286
Tabla 68: Resumen estadístico del postest para el quinto grado	286
Tabla 69: Resultados de indicadores del pretest para el grupo de control – Quinto grado....	287
Tabla 70: Resultados de indicadores del pretest para el grupo experimental – Quinto grado	287
Tabla 71: Resultados de indicadores del postest para el grupo de control – Quinto grado ...	288
Tabla 72: Resultados de indicadores del postest para el grupo experimental – Quinto grado	289
Tabla 73: Resumen estadístico del pretest para el sexto grado.....	289
Tabla 74: Resumen estadístico del postest para el sexto grado	290
Tabla 75: Resultados de indicadores del pretest para el grupo de control – Sexto grado.....	290
Tabla 76: Resultados de indicadores del pretest para el grupo experimental – Sexto grado.	291
Tabla 77: Resultados de indicadores del postest para el grupo de control – Sexto grado	291
Tabla 78: Resultados de indicadores del postest para el grupo experimental – Sexto grado	292
Tabla 79: Prueba de normalidad de Shapiro-Wilk – Pretest de Primer Grado	294
Tabla 80: Tamaño, media y varianza – Pretest del Primer Grado	295
Tabla 81: Prueba de Levene – Pretest del Primer Grado	295
Tabla 82: Prueba T para muestras independientes – Pretest del Primer Grado	296
Tabla 83: Prueba de normalidad de Shapiro-Wilk – Postest de Primer Grado.....	297
Tabla 84: Tamaño, media y varianza – Postest del Primer Grado.....	298
Tabla 85: Prueba de Levene – Postest del Primer Grado.....	298
Tabla 86: Prueba T para muestras independientes – Postest del Primer Grado.....	299

Tabla 87: Resumen de los resultados del análisis estadístico inferencial para cada grado de educación primaria.....	301
Tabla 88: Satisfacción encontrada de los estudiantes de primaria por uso del videojuego ...	302
Tabla 89: Resumen de las mejoras significativas encontradas en el nivel comprensión lectora de los estudiantes de primaria ordenados de mayor a menor.....	315
Tabla 90: Impacto en funcionalidad del sistema de las tecnologías y técnicas de Inteligencia Artificial implementadas.....	319

LISTA DE ECUACIONES

Ecuación (1): Suma-producto de las entradas de la red neuronal por sus pesos.....	61
Ecuación (2): Cálculo de la salida de una neurona	62
Ecuación (3): Valor de una neurona adicionando el <i>bias</i>	62
Ecuación (4): Resultado de la sumatoria de las entradas y pesos incluyendo el <i>bias</i>	62
Ecuación (5): Cálculo para obtener el valor de una neurona en la capa oculta	62
Ecuación (6): Cálculo para obtener el valor de una neurona en la capa de salida.....	62
Ecuación (7): Valor de una nueva entrada x_0 a la red neuronal	63
Ecuación (8): Peso establecido para la nueva entrada a la red	63
Ecuación (9): Función de activación Sigmoide	64
Ecuación (10): Función de activación RELU	64
Ecuación (11): Cálculo del valor de una neurona en un MLP.....	67
Ecuación (12): Cálculo del valor de una neurona j en la capa de salida.....	67
Ecuación (13): Aplicación de la regla de la cadena en un MLP	67
Ecuación (14): Factor de sensibilidad para el peso sináptico	67
Ecuación (15): Valor del segundo componente de la derivada parcial.....	67
Ecuación (16): Derivada de la salida de la red con respecto a $v_j(n)$	67
Ecuación (17): Derivada de la salida de una neurona j con respecto a $w_{ji}(n)$	67
Ecuación (18): Reformulación de la regla de la cadena para un MLP	68
Ecuación (19): Corrección aplicada a $w_{ji}(n)$	68
Ecuación (20): Resultado de la expresión $\Delta w_{ji}(n)$	68
Ecuación (21): Cálculo de la gradiente para la salida de una neurona j	68
Ecuación (22): Cálculo del error para una neurona en la capa de salida	69
Ecuación (23): Reformulación de la gradiente local para una neurona j	70
Ecuación (24): Cálculo del valor $\mathcal{E}(n)$	70
Ecuación (25): Derivada de la expresión $\mathcal{E}(n)$ con respecto a la señal de salida $y_j(n)$	70
Ecuación (26): Reformulación de la expresión $\partial \mathcal{E}(n)/\partial y_j(n)$	71
Ecuación (27): Resultado de la expresión $e_k(n)$	71
Ecuación (28): Representación de la expresión $\partial e_k(n)/\partial v_k(n)$	71
Ecuación (29): Resultado de la expresión $v_k(n)$	71
Ecuación (30): Derivada de la expresión $v_k(n)$ con respecto a $y_j(n)$	71
Ecuación (31): Resultado de la derivada de $\mathcal{E}(n)$ con respecto a $y_j(n)$	71

Ecuación (32): Expresión final de la gradiente para una neurona j en la capa oculta	71
Ecuación (33): Corrección aplicada al peso que conecta la neurona i con la neurona j	72
Ecuación (34): Cálculo de la señal de salida de una neurona	72
Ecuación (35): Representación del valor de una neurona j	72
Ecuación (36): Salida de una neurona en la primera capa oculta de la red	73
Ecuación (37): Resultante de una neurona en la capa de salida de una red.....	73
Ecuación (38): Función de activación logística	75
Ecuación (39): Cálculo del error total de la red	77
Ecuación (40): Cálculo de la recompensa esperada de un agente inteligente.....	89
Ecuación (41): Cálculo de la recompensa acumulada	90
Ecuación (42): Función de <i>State-Value</i>	90
Ecuación (43): Función de <i>Action-Value</i>	91
Ecuación (44): Método <i>Actor-Critic</i>	91
Ecuación (45): Reformulación del método <i>Actor-Critic</i>	92
Ecuación (46): Cálculo del <i>Value Loss</i>	92
Ecuación (47): Cálculo del <i>Policy Loss</i>	92
Ecuación (48): Cálculo del <i>Policy Loss</i> incluyendo la entropía	93
Ecuación (49): Cálculo del valor de la entropía.....	93
Ecuación (50): Estimador de gradiente común \hat{g}	93
Ecuación (51): Función objetivo acotada (<i>Clipped surrogate Objective Function</i>).....	94
Ecuación (52): Cálculo del ratio entre la antigua y nueva <i>Policy</i>	94
Ecuación (53): Expresión final del algoritmo de <i>Proximal Policy Optimization</i>	95
Ecuación (54): Decisión de rechazo de la hipótesis nula.....	97
Ecuación (55): Decisión de aceptación de la hipótesis nula	97
Ecuación (56): Cálculo del tamaño, media y varianza por cada muestra	98
Ecuación (57): Cálculo de la diferencia de medias.....	99
Ecuación (58): Cálculo para la obtención de la varianza común.....	99
Ecuación (59): Cálculo del error estándar de la diferencia de medias.....	99
Ecuación (60): Cálculo para la obtención del estadístico T de student	100
Ecuación (61): Cálculo para la obtención de los grados de libertad.....	100
Ecuación (62): Función de activación – Tangente hiperbólica.....	157
Ecuación (63): Cálculo para la normalización de datos.....	158
Ecuación (64): Actualización de la tasa de aprendizaje.....	183

RESUMEN

Las participaciones de Perú en las pruebas PISA han demostrado que el país posee un bajo nivel principalmente en el área de lectura quedando en la mayoría de las ocasiones en los últimos puestos. Por otro lado, el mercado de los videojuegos ha crecido de forma exponencial incrementándose en 50% el número de jugadores peruanos solamente en el primer semestre del 2020 con la aparición del coronavirus. La presente investigación consistió desarrollar un videojuego para mejorar el nivel de comprensión lectora en estudiantes de primaria. Los participantes fueron 112 estudiantes de 1ro a 6to grado de primaria con un rango de edad de 5 a 11 años. Para el desarrollo del videojuego se utilizó la metodología en cascada que incluye las fases de comunicación, planeación, modelado, desarrollo y despliegue. Los resultados revelaron que los estudiantes que utilizaron el videojuego demostraron una mejora significativa de 1.77 puntos (1er grado), 1.45 puntos (2do grado), 1.06 puntos (3er grado), 1.17 puntos (4to grado), 1.34 puntos (5to grado) y 1.17 puntos (6to grado) en las evaluaciones realizadas. Asimismo, se evidenció que más del 80% de los estudiantes tuvieron una mejora en los niveles literal e inferencial de comprensión lectora.

Palabras clave: Videojuego, aprendizaje basado en videojuegos, metodología en cascada, comprensión lectora.

ABSTRACT

Peru's participations in the PISA tests have shown that the country has a low level, mainly in reading, remaining in the last positions in most cases. On the other hand, the video game market has grown exponentially, with the number of Peruvian players increasing by 50% only in the first half of 2020 with the appearance of the coronavirus. This project consisted of the development of a video game to improve the level of reading comprehension in primary school students. The participants were 112 students from 1st to 6th grade of primary school with an age range of 5 to 11 years. For the development of the video game, the Waterfall model was used, which includes the requirements phase, planning, design, development, and deployment phases. The results revealed that the students who played the video game showed a significant improvement of 1.77 points (1st grade), 1.45 points (2nd grade), 1.06 points (3rd grade), 1.17 points (4th grade), 1.34 points (5th grade) and 1.17 points (6th grade) in the evaluations. Likewise, it was evidenced that more than 80% of the students had an improvement mainly in the literal and inferential levels of reading comprehension.

Keywords: Video game, Game-Based Learning, Waterfall model, Reading comprehension.

INTRODUCCIÓN

El objetivo central del presente proyecto de investigación es desarrollar un videojuego que permita mejorar el nivel de comprensión lectora en estudiantes de educación primaria. Para realizar dicho objetivo se ha desarrollado un videojuego aplicando la metodología en cascada y para llevar a cabo el entrenamiento de los estudiantes cada grado fue dividido en un grupo de control y experimental para que el primero siga la metodología tradicional de enseñanza y el segundo aprenda a través del uso del videojuego y las clases tradicionales de tal modo que a través de las evaluaciones de un pretest y postest se pueda determinar si existe una mejora significativa en el nivel de comprensión lectora del grupo experimental con respecto al grupo de control.

Las participaciones del Perú en las pruebas PISA, en donde se evalúan las áreas de matemáticas, ciencias y comprensión lectora no han sido alentadoras. El Perú en la mayoría de las pruebas PISA ha quedado en último y penúltimo puesto principalmente en el área de lectura. Si bien es cierto que ha habido un avance con el pasar de los años con respecto al Perú y su participación en las pruebas PISA, se ha observado una mejora mucho más lenta con respecto al área de comprensión lectora.

Por otro lado, el mercado de los videojuegos ha tenido un avance muy acelerado en los últimos años creciendo a nivel mundial en el 2019 en un 9.6% y a nivel Latinoamérica en un 11.1%. Con la llegada del nuevo coronavirus, la OMS recomendó el uso de los videojuegos como una medida preventiva para que la gente evite salir de sus casas y puedan estar protegidos de este nuevo virus, en función a esto último, el porcentaje de jugadores de videojuegos en el Perú se incrementó en un 50% solamente en la primera mitad del 2020.

En el primer capítulo, se realizará una descripción del problema del bajo nivel de comprensión lectora que poseen los estudiantes de educación básica del Perú a través de los resultados obtenidos en las pruebas PISA. De igual modo se hará una explicación del crecimiento que ha tenido el mercado de los videojuegos en los últimos años. Habiendo planteado ambas variables, se procederá a formular el problema central, así como el objetivo central y específicos del presente proyecto de investigación.

En el segundo capítulo se realizará la presentación de los principales trabajos de investigación existentes en el mundo en donde se utilizan videojuegos con el objetivo de mejorar una determinada habilidad en los estudiantes o para el aprendizaje de un determinado tema. Seguidamente, se procederán a explicar las principales teorías detrás de un videojuego, así como las principales tecnologías que existen para su desarrollo. Para darle a un mayor grado de realismo al videojuego se abordará el tema de Inteligencia Artificial y particularmente la técnica de *Deep Reinforcement Learning (DRL)* con el objetivo de entrenar a un agente inteligente que pueda perseguir al jugador principal. De igual modo, se presentarán las principales teorías para el aprendizaje de la comprensión lectora, así como los niveles que posee y la nueva metodología virtual de enseñanza bajo la pandemia del Covid-19. Este capítulo finaliza con la explicación de las consecuencias que ha traído la aparición de este nuevo coronavirus en el sector de educación y las medidas establecidas para afrontar esta nueva normalidad en el ámbito educativo.

En el tercer capítulo, se iniciará planteando el diseño de la investigación, así como los instrumentos utilizados para poder recabar y analizar los resultados obtenidos. Además, se realizará una evaluación y selección de las tecnologías más adecuadas para poder desarrollar el videojuego en función a los antecedentes de investigación presentados en el capítulo anterior. Por otro lado, se presentarán las etapas del presente proyecto de investigación, así como la explicación de cómo se desarrolló cada una. De igual modo, se presentará las principales actividades, tareas y entregables desarrollados para la creación del videojuego, así como la implementación de las técnicas de aprendizaje para que los alumnos de primaria mejoren su nivel de comprensión lectora. Adicionalmente, se presentará la metodología para poder realizar la medición de los resultados con respecto a la metodología implementada, modelos de DRL desarrollados y en relación con las evaluaciones completadas por los estudiantes de primaria. Este capítulo finaliza con la presentación del cronograma de trabajo desarrollado para poder desarrollar el presente proyecto.

En relación con el cuarto capítulo, se presentará el desarrollo del videojuego siguiendo las fases establecidas de la metodología en cascada. También se realizará la implementación de las principales tecnologías que fueron seleccionadas del capítulo anterior. Por último, se presentará el protocolo de aprendizaje desarrollado que los estudiantes del grupo experimental siguieron para poder realizar el entrenamiento correspondiente a través del uso del videojuego.

En el quinto capítulo se hará la presentación de los resultados obtenidos con respecto a cada fase de la metodología en cascada, con respecto al entrenamiento de los modelos de DRL y los resultados obtenidos del pretest y posttest de tal modo de que se puede detectar si hubo una mejora significativa por parte de los estudiantes de primaria que utilizaron el videojuego.

Finalmente, en el sexto y último capítulo se presentarán las conclusiones obtenidas de proyecto de investigación desarrollado de tal forma de poder aceptar o rechazar la hipótesis general planteada. Por último, se presentarán algunas recomendaciones en función a lo desarrollado durante este proyecto de investigación.

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

En el presente capítulo se hará una descripción del problema del bajo nivel de comprensión de textos que poseen los estudiantes de educación básica en el Perú. Para realizar lo anterior, se hará un análisis de los resultados de la prueba PISA en el Perú en las áreas de matemáticas, ciencias y comprensión lectora, concentrando el esfuerzo en el área de comprensión lectora dado que ha tenido un menor crecimiento en los últimos años. Por otro lado, se hará una descripción del crecimiento del mercado con respecto a videojuegos y el potencial que este tiene como una herramienta educativa en un salón de clases. Una vez analizado el problema de la comprensión lectora en el Perú desde el punto de vista de los resultados en las pruebas PISA y el potencial de los videojuegos como una herramienta educativa, se procederá a formular tanto el problema general como los objetivos que se pretenderá conseguir en este proyecto. Seguidamente, se hará una descripción general de las variables a utilizar, así como la formulación de la hipótesis general e hipótesis específicas. Finalmente, el presente capítulo concluirá con la justificación y delimitación del presente proyecto.

1.1 Descripción de la realidad problemática

La comprensión lectora es una de las capacidades más importantes que un estudiante debe tener. No solamente por el hecho de poder entender las ideas que una lectura quiere transmitir, sino porque ese alumno podrá obtener mejores oportunidades tanto en el aspecto educativo como profesional si comprende lo que lee. Según la Organización para la Cooperación y el Desarrollo Económicos (OECD, por sus siglas en inglés), la comprensión lectora permite comprender, usar, reflexionar e involucrarse en la lectura de textos escritos, con el fin de alcanzar metas propias, desarrollar el conocimiento y potencial personal para poder participar plenamente dentro de la sociedad (OECD, 2015). Cada 3 años, el Programa para la Evaluación Internacional de Estudiantes (PISA, por sus siglas en inglés) realiza una prueba, también conocida como prueba PISA, a alrededor de 82 países del mundo pertenecientes a la OCDE de los cuales 10 de ellos pertenecen a América Latina, para determinar el grado que tienen los alumnos de dichos países en las áreas de ciencias, matemáticas y lectura. Debido a que no es miembro de la OCDE, el Perú ha participado de manera voluntaria en la prueba PISA en el año 2001, 2009, 2012, 2015 y 2018. No obstante, los resultados obtenidos no han sido muy favorables. En la prueba PISA realizada en el año 2001, Perú salió en el último lugar de 43 países participantes (28 de ellos de la OCDE), tanto en matemáticas, ciencias y lectura (Correo, 2010). En la prueba PISA realizada en 2009, donde esta vez había 65 países inscritos (30 de ellos de la OCDE) el Perú quedó en el puesto 62 en lectura, 60 en matemática y 63 en ciencias

(Correo, 2010). En la prueba PISA realizada en el año 2012, en el cual se evaluaron a 65 países, el Perú quedó en el último lugar ubicándonos muy por debajo del promedio. Esto último demuestra que hubo un retroceso debido a que en la anterior prueba PISA del 2001 el Perú había salido del último lugar. En la Figura N°1, se observan los países ubicados en los primeros y los últimos puestos de la prueba PISA 2012 junto con la puntuación en cada área.

Preocupantes resultados de la PISA 2012

PAÍSES CON ALTA CALIFICACIÓN	PUESTO	DESEMPEÑO EN					
		MATEMÁTICA	COMPRESIÓN LECTORA	EN CIENCIAS			
	1	Shanghái	613	Shanghái	570	Shanghái	580
	2	Singapur	573	Hong Kong	545	Hong Kong	555
	3	Hong Kong	561	Singapur	542	Singapur	551
	4	Taipéi	560	Japón	538	Japón	547
	5	Corea	554	Corea	536	Finlandia	545
	PROMEDIO OECD		494	496	501		
PAÍSES CON BAJA CALIFICACIÓN	61	Jordania	386	Argentina	396	Túnez	398
	62	Colombia	376	Albania	394	Albania	397
	63	Qatar	376	Kazajistán	393	Qatar	384
	64	Indonesia	375	Qatar	388	Indonesia	382
	65	Perú	368	Perú	384	Perú	373

Figura N°1: Primeros y últimos puestos de la prueba PISA 2012.

Fuente: MINEDU. (2013). *¿Cómo mejorar la Comprensión Lectora de nuestros estudiantes?*

Sin embargo, en la prueba PISA realizada en el 2015, en el cual esta vez fueron 72 los países evaluados, el Perú logró salir del último lugar ubicándose en el puesto 64. En América Latina el Perú ha estado ubicado en el penúltimo lugar, superando solo a República Dominicana, país que en el 2015 se sometió a su primera evaluación PISA (MINEDU, 2013). En las áreas de lectura y ciencia seguimos por debajo de países como Chile, Uruguay, Costa Rica, Colombia, México y Brasil (MINEDU, 2013). La comparación de los resultados obtenidos de la prueba PISA del 2015 en las 3 diferentes áreas en los países América, se presentan en la Figura N°2.

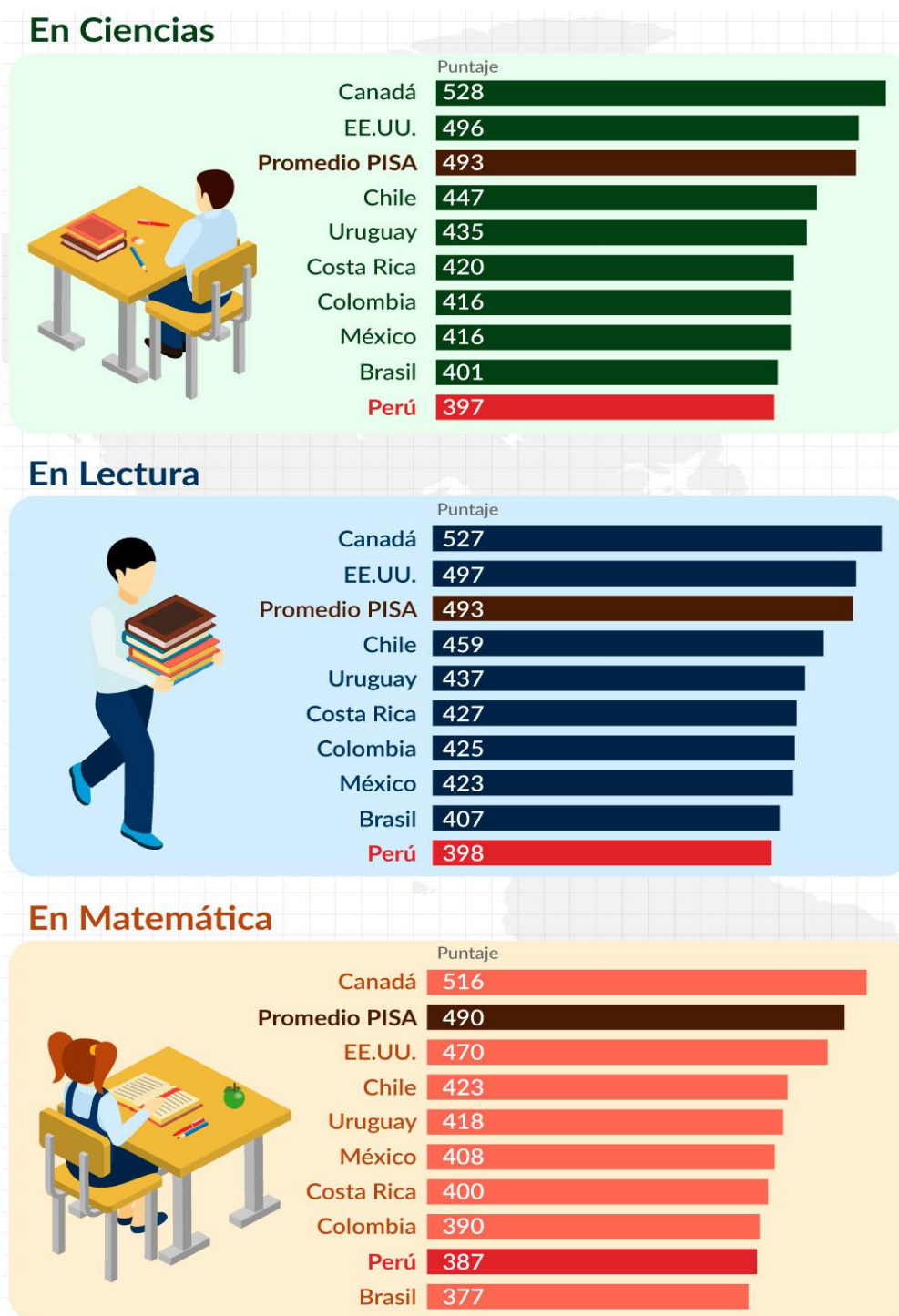


Figura N°2: Resultados en las 3 áreas de la prueba PISA 2015 en países de América.
Fuente: MINEDU. (2015). *El Perú en PISA 2015. Informe nacional de resultados.* (p.15)

Si bien es cierto que ha habido un avance en las áreas de ciencias, matemáticas y lectura, el Perú sigue estando en los últimos lugares. Además, de las 3 áreas que la prueba PISA evalúa, hay una mayor deficiencia de aprendizaje en el área de comprensión lectora. La anterior afirmación se sustenta mediante la comparación de las puntuaciones de cada área utilizando las

2 pruebas PISA que corresponden a los años 2012 y 2015. Como se muestra en la Figura N°3, en el área de Ciencias, en la prueba PISA del año 2012, el Perú obtuvo una puntuación de 373 y en la prueba PISA del año 2015 obtuvo una puntuación de 397. Es decir, que en los últimos 3 años se logró un incremento de 24 puntos o, en otras palabras, una mejora de 6.43 %. Posteriormente, en el área de Matemática, en la prueba PISA del año 2012, obtuvo una puntuación de 368 y en la del año 2015 pasó a tener 387. Es decir, hubo una mejora en esta área de alrededor de 5.16%. No obstante, en el área de comprensión lectora en el año 2012 el Perú obtuvo una puntuación de 384 y en la prueba PISA del año 2015 obtuvo un puntaje de 398. Es decir, que solamente incrementó en 3 años alrededor de 3.64%.

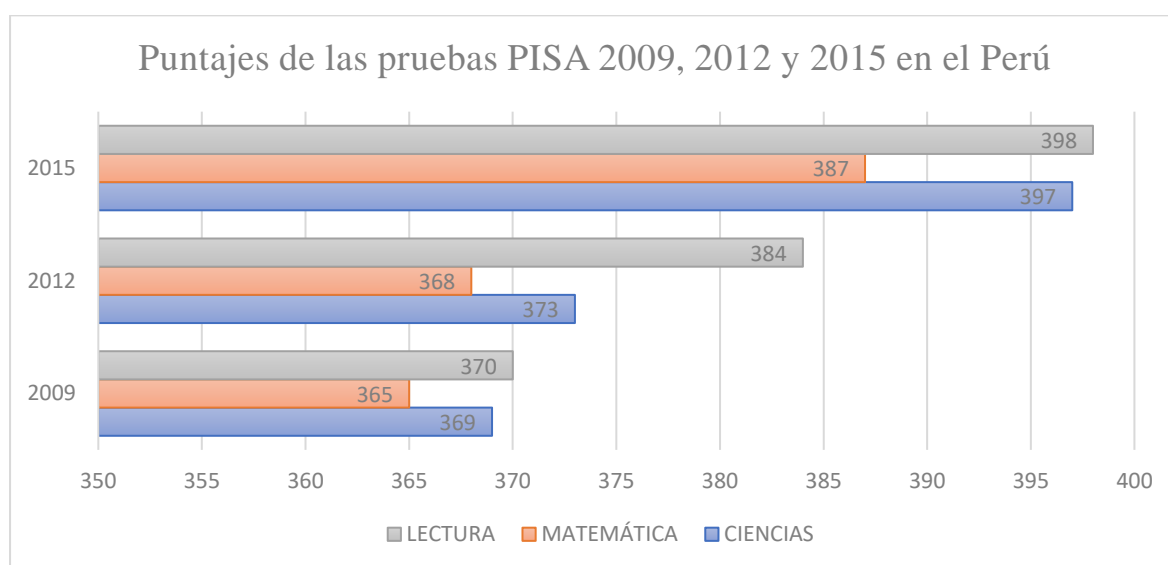


Figura N°3: Puntajes de las pruebas PISA del 2009,2012 y 2015 en el Perú.

Fuente: MINEDU. (2018). *Resultados de la evaluación internacional PISA 2018*.

Finalmente, en la reciente prueba PISA del 2018, el Perú quedó en el puesto número 64 de 77 países que participaron. En esta última prueba obtuvo en el área de Matemáticas un total de 400 puntos, en el área de ciencias 404 puntos y el área de comprensión lectora un total de 401 puntos. Haciendo una comparación de puntajes con la prueba PISA 2015, tal cual como se observa en la Figura N°4, se vio un incremento en el área de matemáticas de un 3.35%, en el área de ciencias se obtuvo un incremento de 1.76% y el área de comprensión lectora se obtuvo un incremento de 0.75%. Se puede apreciar que de las comparaciones de los resultados de las pruebas PISA 2015 con 2012 y los resultados de la prueba PISA 2018 con la del 2015, se observa que, de las 3 áreas, la que tiene un menor incremento es el área de comprensión lectora.

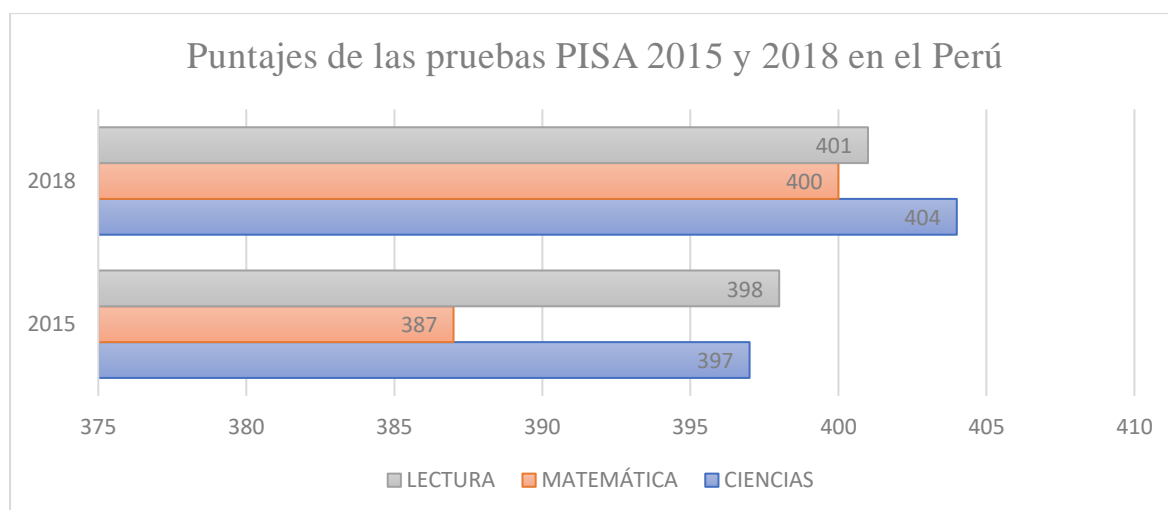


Figura N°4: Puntajes de las pruebas PISA 2015 y 2018 en el Perú.

Fuente: MINEDU. (2018). *Resultados de la evaluación internacional PISA 2018*.

En la Tabla 1, se puede apreciar a modo de resumen la mejora porcentual que ha habido en la Prueba Pisa 2015 con respecto a la del 2012 y en la Prueba Pisa 2018 con respecto a la del 2015 demostrando, como se había mencionado anteriormente, que en el área de lectura hay un crecimiento más lento a comparación con las otras 2 áreas.

Tabla 1: Resumen de mejora porcentual con respecto a las 2 últimas pruebas PISA

Áreas	Prueba PISA 2015	Prueba PISA 2018
	Mejora con respecto al 2012	Mejora con respecto al 2015
Matemática	5.16%	3.35%
Ciencias	6.43%	1.76%
Lectura	3.64%	0.75%

Fuente: Elaboración Propia.

Además, en la Tabla 2, se puede ver un resumen de las puntuaciones y los puestos en donde estado ubicado en Perú desde su primera participación en la prueba PISA hasta la actualidad.

Tabla 2: Resumen de puntajes por área y puestos de Perú en las pruebas PISA

Año	Lectura	Matemática	Ciencias	Posición
2001	327	292	333	43° (Último lugar)
2009	370	365	369	63° (Penúltimo lugar)
2012	384	368	373	65° (Último lugar)
2015	398	387	397	64° de 72
2018	401	400	404	64° de 77

Fuente: MINEDU. (2018). *Resultados de la evaluación internacional PISA 2018*.

Como se había mencionado, el área de lectura ha sido la que ha tenido un menor crecimiento con respecto a las otras 2 áreas en los puntajes de los últimos años por lo que es importante concentrar los esfuerzos en mejorar esta área en particular. Además, mejorando esta área se puede obtener una mejora en las otras 2 debido a que los alumnos podrían entender de una mejor manera los enunciados de los problemas de matemática o de ciencias.

Es preciso resaltar que esta prueba PISA se les toma a alumnos que tienen alrededor de 15 años y que se encuentran en educación secundaria. No obstante, la comprensión lectora es un área que se empieza a enseñar desde primaria y es necesario enfocar en brindar una sólida base en este último grado académico mencionado para que el niño cuando pase a grados superiores pueda afrontar de una manera adecuada los textos que tienen un grado de complejidad mucho mayor y a la par pueda resolver exitosamente pruebas exteriores.

Por otro lado, según Global Games Market Report, se registró en el año 2019 alrededor de 2.500 millones de gamers en todo el mundo generando un crecimiento de 152.100 millones de dólares lo que equivale a un crecimiento de 9.6% más con respecto al año 2018 (AETecno, 2019). Con respecto a América Latina, esta ha sido una de las regiones que ha tenido un crecimiento constante de 11.1% aproximadamente con respecto a la industria de videojuegos. En el año 2018, en América Latina, la industria generó un total de 5.000 millones de dólares y en el año 2019 generó 6 millones de dólares adicionales aproximadamente (AETecno, 2019). Tal cual como se muestra en la Figura N°5, los videojuegos modernos cuentan con una mayor calidad en los gráficos, así como sonidos mucho más realistas haciendo que los usuarios se involucren mucho más en ese nuevo mundo y creando nuevos nichos en el mercado.



Figura N°5: Videojuegos en la actualidad.

Fuente: Alonso, J. (2001). *Videojuegos 3D*. (p.30). Recuperado de:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/50201/2/Videojuegos%203D.pdf>.

Con respecto al uso de dispositivos para jugar videojuegos, la consola tiene un mayor incremento con respecto al resto con un 13.4% anual con videojuegos clásicos como FIFA tal cual como se muestra en la Figura N°6. En segundo lugar, se tienen a los juegos móviles con un crecimiento de 10.2% anualmente. Finalmente, el último lugar los videojuegos para PC con un crecimiento de 4% anual (AETecno, 2019).

No obstante, los videojuegos de PC, a pesar de su bajo crecimiento, siguen dando de qué hablar dado a la popularidad de los e-sports (AETecno, 2019). Con esto último, las empresas han buscado satisfacer a la demanda de usuarios de todo tipo en el mundo, desarrollando equipos más avanzados para brindar una mejor experiencia dentro del videojuego (AETecno, 2019).



Figura N°6: Escena de videojuego FIFA 2020.

Fuente: Needleman, S. (2020). *La pandemia aumenta las ganancias de la industria de los videojuegos.* (p.5)

Con el surgimiento del Covid-19, la OMS aconsejó en el primer trimestre del 2020, quedarse en sus casas para jugar videojuegos como una medida para evitar que las personas salgan de sus casas y logren protegerse de este peligroso virus altamente contagioso (Maulee, 2020). Como respuesta a la sugerencia de la OMS, las empresas de desarrollo de videojuegos han promovido la campañas y sugerencias que la OMS realiza para mantener informadas a las personas sobre el avance de este nuevo virus y las medidas sanitarias para evitar contraer esta peligrosa enfermedad. Esta campaña por parte de la OMS y más de 20 de las mejores empresas de desarrollo de videojuegos a nivel mundial se dio a conocer en Twitter con el hashtag PlayApartTogether (Business Wire, 2020). A través de esta iniciativa, tal cual como se muestra en la Figura N°7, se busca proteger a las personas, mantenerlas informadas y que a su vez puedan divertirse en la comodidad de sus casas. Además, las compañías de videojuegos realizan eventos especiales y dan recompensas en sus juegos más populares con el objetivo de que sus usuarios apliquen y compartan la información de los cuidados ante esta situación de pandemia (Business Wire, 2020).

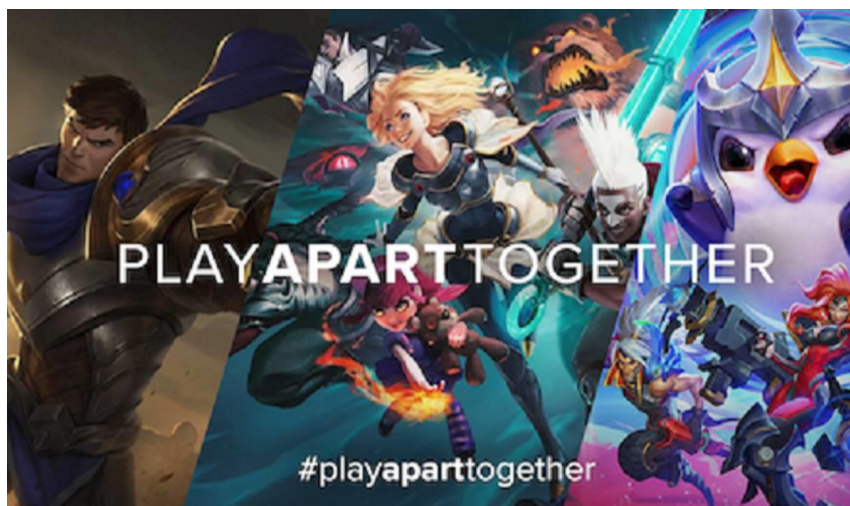


Figura N°7: PlayApartTogether: Una medida preventiva ante la actual pandemia.

Fuente: Ruete, B. (2020). *La industria se une a la iniciativa #PlayApartTogether contra el coronavirus.* (p.4)

Con respecto al pronóstico del mercado de videojuegos, se espera que, para finales del 2020, los ingresos sean de 159.3 mil millones de dólares con tendencia al alza hacia el año 2023 (Maulee, 2020). Esto es debido a que se han unido al mundo de los videojuegos gran cantidad de usuarios y los otros usuarios que ya llevan tiempo jugando decidan comprar nuevos videojuegos para mantenerse en sus casas y preservar su salud (Maulee, 2020).

Además, con respecto a los torneos de videojuegos que se desarrollan a nivel mundial, así como videojuegos que se juegan online en consolas tal cual como se muestra en la Figura N°8, ha habido un incremento considerable de los asistentes a los mismos haciendo que haya un incremento de los participantes de un 74% con respecto a las participaciones del año anterior (Maulee, 2020). En el 2019, hubo 6.3 millones de espectadores en los torneos de videojuegos y en 2020 hubo un crecimiento bastante elevado siendo la cantidad de espectadores de 11 millones de usuarios (Maulee, 2020).



Figura N°8: Escena de videojuego Call of Duty 4 : Modern Warfare

Fuente: Agoros, P. (2013). *Call Of Duty: Ghosts past-gen multiplayer review (PS3)*. Recuperado de <http://legacyfuture.com/gaming/call-duty-ghosts-review-ps3xbox-360/>

La mayoría de las empresas de desarrollo de videojuegos ha tenido un aumento de utilidades por la pandemia generada por el Covid-19. Un ejemplo de lo anterior mencionado es la empresa Microsoft que ha obtenido en el segundo trimestre del año 2020 un incremento en sus ganancias del 65% por el contenido de su consola de Xbox (Needleman, 2020). Otra empresa, Electronic Arts, reportó que en el segundo trimestre del 2020 reportó un incremento manifestó que sus ventas en ese periodo de tiempo fueron las mejores luego de 38 años de historia desarrollando videojuegos para los usuarios (Needleman, 2020).

En el Perú, se estima que el mercado de videojuegos ascenderá a un 20% anual generando 147 millones de dólares para el año 2023 (G&M News, 2020). Existe un gran entusiasmo por el mercado de videojuegos en Perú dado que en el 2018 58 millones de dólares por lo que se tienen altas expectativas de que este mercado pueda ir en incremento en los próximos años. Con respecto a cantidad de jugadores, el Perú se ubica en el puesto 49 a nivel mundial con alrededor de 8 millones de jugadores en consolas, celulares y computadoras (G&M News, 2020).

Adicionalmente, en el trabajo de investigación realizado por la empresa GFK y la Asociación Peruana de Deportes Electrónicos y Videojuegos, se desarrolló un análisis de los jugadores gamers peruanos, llegando a resultados interesantes dentro de los cuales se encuentra que el 78% de hombres en el Perú se consideran “gamers” y en el caso de las mujeres solamente el 22% (G&M News, 2020). Además, también se llegó a la conclusión de que el 76% de los

hombres juega un videojuego al menos una vez por semana y con respecto a las mujeres un 56% de estas se dedica a jugar por lo menos un videojuego a la semana (G&M News, 2020). Con respecto al estado civil, el 65% de mujeres que juega videojuego son solteras y alrededor del 18% son casadas. Con respecto a los hombres, el 49% de estos son casados los que juegan videojuegos y el 31% de los hombres son solteros. Adicionalmente, con respecto al dispositivo de preferencia para jugar videojuegos, los hombres prefieren jugar en PlayStation con un 29%, en smartphone con un 24% y en la computadora con un 23% (G&M News, 2020). Por el lado de las mujeres, ellas prefieren jugar en smartphones con un 38%, prefieren la PlayStation con un 20% y en computadora 12% (G&M News, 2020). Un dato adicional interesante con respecto al uso de videojuegos en plena cuarentena, los peruanos incrementaron su interacción en videojuegos en línea entre 50% y 500%.

Diversos autores han afirmado que un videojuego podría convertirse en una herramienta educativa muy potente dado que posibilita el aprendizaje de nuevos conocimientos, así como la mejora de alguna habilidad. La anterior afirmación se sustenta por el hecho de que los videojuegos sumergen a los usuarios dentro de un mundo diferente el cual posee sus propias reglas y leyes lo que obliga a los usuarios a adaptarse a este nuevo mundo interactuando con otros personajes y pasando por los diversos niveles que el videojuego posee (López C. , 2015).

Hoy en día, mientras se lidia con la aparición del nuevo coronavirus, las instituciones educativas deben adaptar sus metodologías tradicionales de enseñanza con el fin de que sus estudiantes puedan seguir adquiriendo conocimientos de la manera más didáctica posible de tal forma que puedan enfrentar este mundo que es cada vez más globalizado y competitivo (Roncancio et al., 2017).

Los docentes, en busca de hacer sus clases mucho más didácticas, están utilizando los videojuegos dentro de sus clases y esto es debido a que diversas investigaciones apuntan a que un videojuego aporta un mayor involucramiento por parte de las personas, los vuelve muchos más competitivos y colaborativos (Roncancio et al., 2017). Diversos trabajos de investigación realizados a lo largo de los últimos años afirman que los videojuegos traen diversos beneficios tanto para el alumno como el docente convirtiendo a un videojuego en una potente herramienta educativa (Roncancio et al., 2017).

Una explicación mucho más profunda de las principales investigaciones que se han desarrollado con respecto a videojuegos dentro de un salón de clases con el objetivo de mejorar

alguna determinada habilidad o adquirir ciertos conocimientos, así como los principales resultados que se obtuvieron, serán presentados en el próximo capítulo.

1.2 Formulación del problema

Habiendo presentado anteriormente el problema de la comprensión lectora en el Perú analizando los resultados de las pruebas PISA, así como el crecimiento del uso de los videojuegos tanto a nivel mundial, así como a nivel nacional y su uso potencial como una herramienta educativa dentro de las aulas de clases, se presenta a continuación la formulación del problema central y los problemas específicos.

1.2.1 Problema General

P.G: ¿De qué manera es posible desarrollar un videojuego que mejore el nivel de comprensión lectora en estudiantes de primaria?

1.2.2 Problemas específicos

P.E 1: ¿Qué tecnologías y técnicas de Inteligencia Artificial deben de integrarse para el desarrollo funcional del videojuego?

P.E 2: ¿De qué forma se puede obtener el porcentaje de aceptación del prototipo desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria?

P.E 3: ¿Cómo validar estadísticamente la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención?

P.E 4: ¿Cómo se puede generar un procedimiento pedagógico y tecnológico para complementar las clases de comprensión lectora mediante el uso del videojuego?

1.3 Objetivos de la investigación

A continuación, se presentarán el objetivo central y específicos de esta investigación.

1.3.1 Objetivo General

O.G: Desarrollar un videojuego que permita mejorar el nivel de comprensión lectora en estudiantes de primaria.

1.3.2 Objetivos específicos

O.E 1: Identificar las tecnologías y técnicas de Inteligencia Artificial que se integran para el desarrollo funcional del videojuego.

O.E 2: Obtener el porcentaje de aceptación del sistema desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria.

O.E 3: Validar estadísticamente la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención.

O.E 4: Generar un procedimiento pedagógico y tecnológico para complementar las clases de comprensión lectora mediante el uso del videojuego.

1.4 Variables e Hipótesis

A continuación, se presentará las variables involucradas e hipótesis general del presente proyecto de investigación en función de la problemática y los objetivos planteados.

1.4.1 Variables

En este apartado, se presentan las variables que serán objeto de estudio en el presente trabajo de investigación.

Variable Independiente: Uso de videojuego desarrollado.

Variable dependiente: Nivel de comprensión lectora de estudiantes.

1.4.2 Hipótesis General

En este presente trabajo de investigación, se pretende comprobar que la utilización de un videojuego por parte de estudiantes de primaria posibilita que exista una mejora en su nivel de comprensión lectora. La hipótesis general del presente proyecto de investigación es la siguiente:

H.G: El uso del videojuego desarrollado posibilita la mejora del nivel de comprensión lectora en estudiantes de primaria.

1.4.3 Hipótesis específicas

Las hipótesis específicas del proyecto se presentan a continuación:

H.E 1: La integración de las tecnologías y técnicas de Inteligencia Artificial implementadas posibilitarán la correcta funcionalidad del videojuego.

H.E 2: El porcentaje de aceptación obtenido del sistema desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria será mayor a 50%.

H.E 3: La validación estadística permitirá detectar la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención.

H.E 4: El procedimiento pedagógico y tecnológico desarrollado permitirá complementar las clases de comprensión lectora mediante el uso del videojuego.

1.5 Justificación de la investigación

A continuación, se presentarán la justificación teórica, práctica y metodológica para la elaboración de este proyecto.

1.5.1 Justificación teórica

Los videojuegos están siendo utilizados en los salones de clase para la enseñanza de un determinado tema o mejora de cierta habilidad en estudiantes. Esto es debido a que se ha comprobado que estos traen diversos beneficios a los alumnos como la cooperación y competencia. Pero también, los videojuegos permiten recuperar sensaciones como la motivación y superación que habían sido, en algunos casos, perdidas por parte de los alumnos debido a las metodologías tan monótonas que se aplican en las aulas de los colegios (Roncancio et al., 2017).

Además, los autores afirman que los videojuegos a diferencia de los juegos cotidianos poseen muchos más beneficios de los cuales se pueden aprovechar para desarrollo del aprendizaje. Los principales atributos que hacen a un videojuego una herramienta potencial dentro de un ambiente educativo es que aquel posee imágenes, videos, textos para narrarte historias, para la comunicación entre personajes, música, ambientes virtuales, entre otros. Dado lo anterior mencionado, los autores plantean que el videojuego no solamente puede servir como una herramienta para mantener a los alumnos más motivados dentro de las clases, sino que su utilización podría mejorar distintas habilidades que poseen los estudiantes (Tonatiuh, 2017).

Del mismo modo, se han hecho comparaciones entre las metodologías tradicionales que utilizan los profesores actualmente y las nuevas metodologías que se proponen al combinar las clases tradicionales y el uso del videojuego mediante una distribución de tiempos entre las clases y el videojuego demostrando que estas nuevas metodologías aplicando el aprendizaje a través de un videojuego suelen ser más efectivas que las metodologías tradicionales haciendo que los alumnos puedan mejorar de una manera mucho más rápida y eficaz sus habilidades psicomotrices y metacognitivas (Tonatiuh, 2017).

Desde un punto de vista teórico, mediante este trabajo se pretende demostrar que el uso de un videojuego dentro de las clases de los estudiantes contribuye a la mejora del nivel de comprensión lectora en estudiantes de primaria. Se buscará aprovechar los diversos recursos

que ofrece el videojuego descritos anteriormente como la utilización de textos dentro del propio videojuego, narración de historias, entre otros. Con los resultados obtenidos de este proyecto de investigación, se espera que estos puedan contribuir en el campo del uso de videojuegos utilizados en un ambiente educacional como una evidencia adicional a los trabajos de investigación ya existentes en el mundo y que sirva además como una apreciación por parte de profesores y alumnos de que efectivamente un videojuego permite el desarrollo y mejora de ciertas habilidades que poseen los escolares.

Asimismo, mediante la realización de este proyecto se espera que sirva como una reflexión sobre la utilización de tecnologías dentro de los salones de clases con el objetivo de hacer éstas más entretenidas y didácticas para que los alumnos puedan aprender de la manera más eficaz posible. La idea es poder aprovechar el avance de las tecnologías de los últimos años y las mejoras que ha habido en los videojuegos tanto en los propios motores de videojuegos como en las gráficas que poseen de tal modo que se pueda resolver un problema en particular.

Del mismo modo, este proyecto servirá como aporte a los trabajos de investigación desarrollados en Perú, dado que son muy escasos, en donde se desarrolle un videojuego con el objetivo de poder ejercitar una determinada habilidad de los estudiantes especialmente de la comprensión lectora dado que, como se ha visto anteriormente, los resultados de las pruebas PISA no son alentadores. Y a su vez que este proyecto posibilite la realización de otros tipos de videojuegos para mejorar otras habilidades que sean necesarias mejorar en los estudiantes para que puedan volverse alumnos mucho más competentes, que puedan reflexionar sobre determinados problemas y situaciones para poder encontrar soluciones óptimas a problemas tanto planteados en las propias clases como problemas cotidianos y de este modo se puedan formar no solo buenos profesionales sino mejores personas.

1.5.2 Justificación práctica

El desarrollo de este proyecto de investigación ha sido en medio de una situación muy difícil que se encuentra viviendo el mundo actualmente. Las consecuencias de la llegada del Covid-19 han sido tanto económicas como a nivel educacional. Se calcula que alrededor de 60,2 millones de docentes en todo el mundo se vieron afectados por el surgimiento de este nuevo virus (Lincapié, 2020), dado que una de las principales medidas para frenar el avance del número de contagiados es la educación a distancia a través de clases virtuales.

Los docentes tuvieron que cambiar radicalmente sus métodos tradicionales de enseñanza, así como sus materiales de trabajo para que estos se puedan adaptar a un aprendizaje a distancia

(Lincapié, 2020). Asimismo, los padres han tenido un mayor grado de involucramiento dentro de las clases a distancia dado que los estudiantes especialmente de primaria en sus primeros años de formación no tienen un conocimiento regular de las tecnologías para poder utilizar una laptop o celular de manera independiente en clases (Naslund, 2020).

Además, los padres deben de grabar a sus hijos realizando las tareas encomendadas por los profesores, así como ayudarlos revisando las clases para poder desarrollar las actividades planteadas de manera adecuada (Lincapié, 2020).

No obstante, no todos los padres cuentan con el tiempo suficiente para apoyar en todas las actividades que se les suele dejar a sus hijos del mismo modo que hay también padres en pobreza o pobreza extrema que no cuentan con conocimiento suficiente para poder apoyar a sus hijos en las labores de la escuela (Naslund, 2020).

El desarrollo de este proyecto de investigación pretende la utilización de un videojuego dentro de un salón de clase virtual que no solamente permita mejorar el nivel de los estudiantes de primaria, sino que además sirva como una herramienta adicional para el docente, para que este pueda tener nuevas formas de poder enseñar dentro de las clases virtuales dado que cambiar de una forma tan rápida y repentina de una metodología de clases presencial a una virtual no es sencillo.

Del mismo modo, el desarrollo de este proyecto es una ayuda también para los padres de familia dado que los estudiantes estarán usando una herramienta educativa y entretenida que no solamente permita a los niños distraerse divirtiéndose dentro de sus propias casa y a la vez cuidándose de no salir a las calles por el contagio de un virus altamente contagioso sino que aparte que los estudiantes se divierten jugando el videojuego van ejercitando su nivel de comprensión lectora de tal modo que la diversión y el aprendizaje puedan ir de la mano con el objetivo de que los estudiantes puedan entender de una mejor manera lo que leen.

1.5.3 Justificación Metodológica

El aprendizaje basado en videojuegos ha surgido en respuesta a las metodologías tradicionales de enseñanza de los centros educativos (Morales, 2009). Para realizar aquel aprendizaje se detecta en primer lugar un tema en particular que se desea que los alumnos conozcan o que les resulta difícil aprender. En respuesta a esta necesidad, se plantea el desarrollo de un videojuego especial que permita ayudar a que los alumnos puedan aprender cierto tópico de la mano con la teoría que tradicionalmente se enseña en las clases (Morales, 2009).

Es preciso mencionar, que los videojuegos desarrollados para un ambiente educativo o mejor conocidos como videojuegos educacionales no son iguales a los videojuegos de mercado dado que estos últimos se desarrollan con fines de entretenimiento y que de alguna manera mejoran indirectamente habilidades en las personas como el trabajo en equipo y concentración pero que no son diseñados para resolver un problema educacional en particular (Morales, 2009).

El presente trabajo de investigación pretende recopilar las necesidades de un cliente que en este caso son el o los profesores que se encargan de enseñar el curso de comprensión lectora de tal modo que esas necesidades puedan convertirse en requerimientos y en función a una metodología de desarrollo de software se pueda desarrollar un videojuego que permita solucionar el problema del bajo nivel de comprensión lectora.

Asimismo, la correcta elección de la metodología adecuada para realización de un producto de software que, en este caso particular es un videojuego, contribuye al éxito del proyecto (Romero, 2020). Las empresas de desarrollo de software se enfocan principalmente en dos principales factores para desarrollar productos exitosos de software: obtener el financiamiento adecuado para la realización de proyecto y la elección de la metodología adecuada para llevar a cabo el desarrollo del producto esperado (Méndez, 2006).

Además, una adecuada selección de la metodología para el desarrollo de software no solamente permite desarrollar el producto de manera adecuada, sino que posibilita la obtención de los mejores resultados posibles, así como un conocimiento de las fases que conlleva cada etapa de la metodología y las actividades que se desarrollan en cada una de ellas (Romero, 2020).

Del mismo modo, se ha detectado que en diversos proyectos de desarrollo de software se plantea la utilización de una metodología para llevar a cabo el producto. No obstante, no se llega a realizar un análisis previo de qué metodología es la más adecuada para poder desarrollar el proyecto trayendo como consecuencias resultados no deseados como retraso en los tiempos de ejecución, incremento innecesario en el costo del proyecto y desconocimiento de las fases de la metodología seleccionada por parte de los participantes (Méndez, 2006).

Mediante el desarrollo de este proyecto se seguirá una serie de procedimientos para la elección de la metodología óptima y el desarrollo del videojuego. Se planteará un conjunto de las metodologías más utilizadas en el desarrollo de software, pasando por la reducción de esa población de metodologías en función a los trabajos desarrollados en donde se utilice un

videojuego desarrollado para ser utilizado con fines educacionales y finalmente seccionar la mejor metodología a través de una matriz de evaluación con el objetivo de trabajar con la metodología que mejor se adapte a las necesidades del cliente y a los recursos con los que se cuenta.

Con el desarrollo del presente trabajo de investigación, se espera que mediante el uso de un adecuado criterio de selección metodológica pueda ayudar a promover el correcto uso de las distintas evaluaciones de metodología existentes de tal manera que la utilización de estas evaluaciones metodológicas puedan volverse común y así minimizar el error que se comete al seleccionar una metodología que tal vez no sea la que mejor se ajusta al proyecto que se espera realizar.

1.6 Delimitación del estudio

En este apartado se presentarán la delimitación espacial, temporal y conceptual del presente trabajo de investigación.

1.6.1 Espacial

Con respecto a la delimitación espacial, de manera tradicional este tipo de proyecto se utilizaría dentro de los salones de clase de un colegio particular. No obstante, dada la coyuntura de la pandemia del Covid-19 y siguiendo las medidas implementadas por el gobierno peruano de la educación a distancia, este proyecto se utilizará dentro de las clases virtuales de los estudiantes. Los alumnos tendrán acceso a el videojuego desarrollado a través de la computadora o laptop de sus hogares.

1.6.2 Temporal

El presente trabajo de investigación tiene la duración de un año y 5 meses partiendo desde noviembre del 2019 hasta marzo del 2021. Para la realización del proyecto del videojuego, se inició con la recopilación de los requerimientos por parte de los profesores. Luego de recopilar los requerimientos de los maestros, se procedió a elaborar los diseños necesarios para la elaboración del prototipo para finalmente construir el videojuego. El aprendizaje de los alumnos a través del videojuego se dará por un periodo de 5 semanas partiendo desde octubre del 2020 hasta mediados de noviembre de ese mismo año. La evaluación de los resultados obtenidos, así como la presentación de las conclusiones y recomendaciones del proyecto se realizarán en el primer trimestre del 2021.

Es preciso resaltar que se van a respetar durante todo el desarrollo del presente proyecto el protocolo establecido tanto en el colegio a desarrollar el videojuego como las medidas que vayan disponiendo el gobierno con respecto a los cuidados que deben adoptar tanto profesores como alumnado para la prevenir el contagio del Covid-19.

1.6.3 Conceptual

En este presente trabajo de investigación de busca desarrollar un videojuego educativo que pueda ser utilizado dentro de los salones de clase virtuales de un colegio en particular y que permita, de la mano con las clases tradicionales, mejorar significativamente el nivel de comprensión lectora de los estudiantes participantes.

Para realizar lo anterior mencionado, se van a combinan dos principales teorías que corresponden a las dos variables de estudio que son el uso del videojuego y el nivel de comprensión lectura de los alumnos. Se presentarán diversos trabajos como antecedentes de investigación y que a su vez sirvan de base para la realización del videojuego. No se harán comparaciones de los resultados obtenidos en este proyecto con respecto a los antecedentes ya mencionados dado que el lugar, los alumnos y el contexto donde se desarrollan los trabajos son unos diferentes de otros. Los estudiantes de un determinado país pueden tener conocimientos más avanzados con respecto al nivel de comprensión lectora o pueden aprender mucho más rápido debido a las tecnologías que contaba el colegio, las metodologías de enseñanza que utilicen, entre otros aspectos.

Con respecto a la variable independiente, se especificarán las principales definiciones que existen para definir lo que es un videojuego, la metodología tradicional que se suelen utilizar para desarrollarlos y las tecnologías existentes para crear videojuegos actualmente.

En relación con la variable de comprensión lectora, se detallarán los principales niveles que existen dentro de la comprensión lectora y qué caracteriza a cada uno de ellos. Además, se explicará la nueva metodología bajo la pandemia del Covid-19 que han empezado a adoptar los colegios para enseñar a los estudiantes a leer y qué estrategias son las más recomendables utilizar por parte de los maestros para que sus estudiantes puedan aprender de una mejor manera a entender lo que leen. Además, se detallarán las medidas ha adoptado el gobierno peruano para que los alumnos puedan seguir asimilando nuevos conocimientos.

Adicionalmente, se abordarán conceptos sobre las teorías del aprendizaje existentes para la adquisición de nuevos conocimientos y cómo estas metodologías se pueden adaptar para el desarrollo del videojuego. Del mismo modo, se abordará los conceptos que se encuentran

dentro del aprendizaje basado en videojuegos dado que, como se dijo anteriormente, es una nueva tendencia que se están aplicando en diversos centros educativos para mejorar la enseñanza en sus estudiantes.

Finalmente, se abordarán técnicas estadísticas para poder analizar los resultados de la manera más objetiva posible y determinar si en verdad el uso del videojuego combinado con las clases tradicionales permite una mejora significativa en el nivel de comprensión lectora en estudiantes de primaria.

CAPITULO II: MARCO TEÓRICO

En este capítulo se presentarán los diferentes trabajos relacionados con el uso del videojuego en un ambiente educativo que servirán como referencias para el desarrollo del presente proyecto. Por otro lado, se explicarán las bases teóricas en donde se detallarán los principales conceptos a utilizar, las tecnologías aplicadas a videojuegos, las técnicas de Inteligencia Artificial en videojuegos y la técnica estadística utilizada para poder analizar e interpretar de manera adecuada los resultados obtenidos. Finalmente, se mencionará y se explicará el sector al cual va dirigido la solución propuesta y cómo ha reaccionado este sector ante la pandemia ocasionada por el Covid-19.

2.1 Antecedentes de la investigación

En los últimos años, los videojuegos han sido estudiados en diversos campos dentro de los cuales se encuentran el campo de la educación, Inteligencia Artificial, psicología, salud, entre otros (García et al., 2019). En la Figura N°9, se puede apreciar la tendencia de publicaciones de videojuegos en el campo educacional entre los años 2013 y 2018.

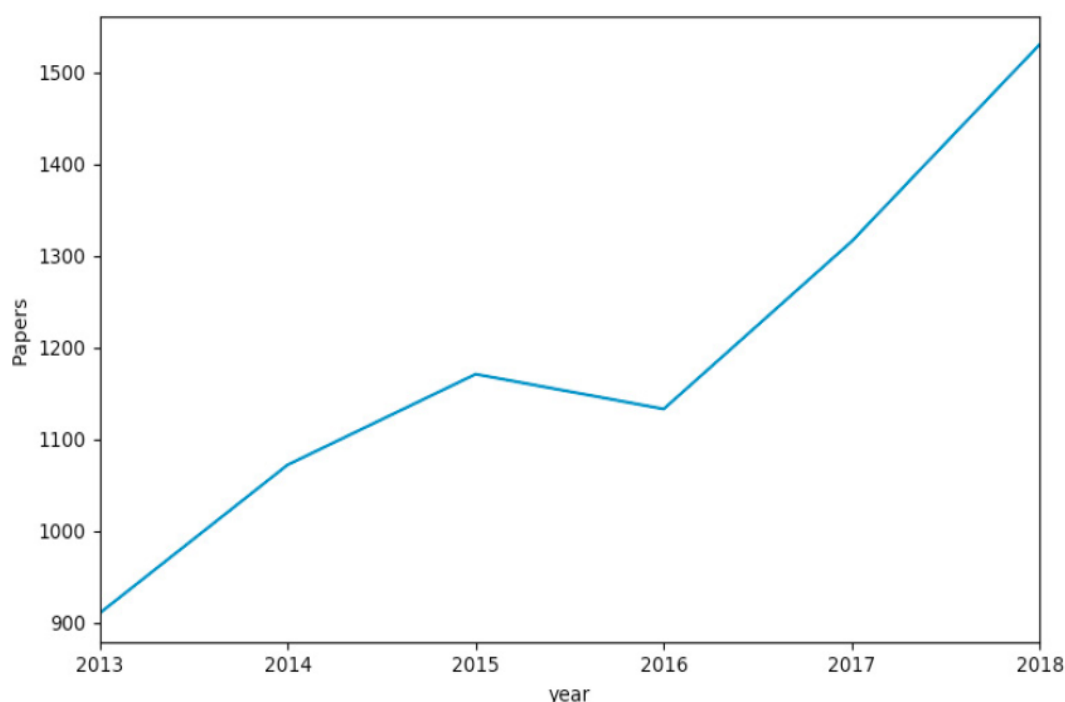


Figura N°9: Tendencia de artículos relacionados a videojuegos entre el 2013 y 2018.

Fuente: García, P., Mora, A., Castillo, P., y Pérez, I. (2019). *A bibliometric study of the research area of videogames using Dimensions.ai database*. *Procedia Computer Science*, 162, 737-744. (p.3)

En la Figura N°9 se puede apreciar que las publicaciones relacionadas a videojuegos han ido creciendo en más de 1000 artículos a partir del 2013 y que en los últimos 5 años ha habido una creciente ola de trabajos de investigación relacionados a videojuegos en donde cada año se desarrollan y publican alrededor de 200 artículos nuevos. En los siguientes párrafos se detallan los trabajos de videojuegos aplicados a un contexto educacional que servirán de base para el planteamiento de las teorías que se desarrollarán a lo largo del presente capítulo.

Un primer trabajo corresponde a Didehbani, Allen, Kandalajt, y Krawczyk (2016) en donde se desarrolló un videojuego con el objetivo de realizar una capacitación en cognición social para mejorar las habilidades sociales en niños. El diseño de investigación fue cuasiexperimental con una prueba antes y después.

La muestra estuvo conformada por 30 niños y adolescentes de entre 7 y 16 años. Dentro de este grupo se encontraron 26 hombres y 4 mujeres. Para que los participantes pudieran usar este videojuego de realidad virtual, se elaboró un protocolo de aprendizaje definido en 2 sesiones por semana durante 5 semanas. El videojuego consistía en recrear escenas de la vida cotidiana en donde el jugador debía de controlar a su propio avatar y usando gafas de realidad virtual.

Dentro de las escenas que contenía el videojuego solían representarse situaciones cotidianas de la vida diaria como por ejemplo pedir la hora a un desconocido, relajarse con los amigos, hacer fila para recoger el almuerzo en el colegio, responder a preguntas en un salón de clases, consolar a un amigo, entre otras cosas. Antes de iniciar con cada escena, el juego le informaba al jugador la escena que iba a aparecer y en la situación en la cual se iba a encontrar. Además, el juego le brindaba pistas de posibles frases que podría usar el jugador con el objetivo de que pueda usarlas en el ambiente virtual al interactuar con otros personajes. El videojuego desarrollado se presenta en la Figura N°10.

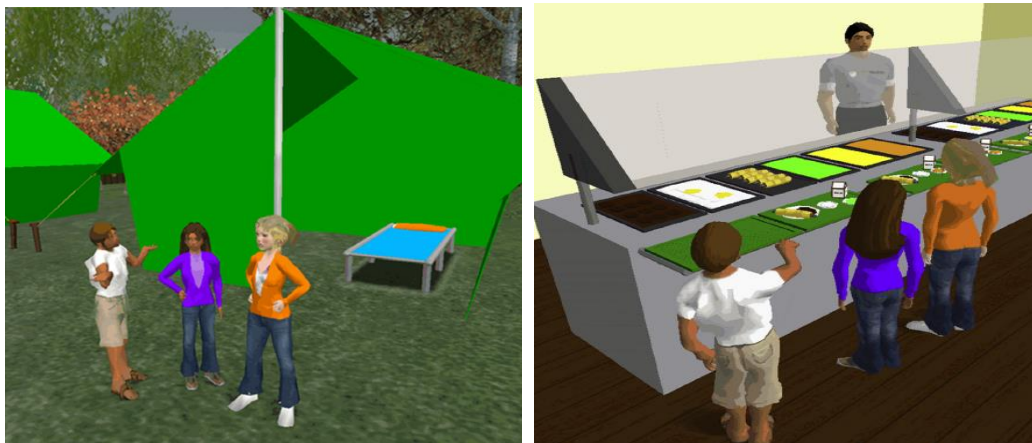


Figura N°10: Escenas del videojuego desarrollado.

Fuente: Didehbani, N., Allen, T., Kandalaft, M., y Krawczyk, D. (2016). *Virtual Reality Social Cognition Training for children with high functioning autism. Computers in Human Behavior*, 703-711. (p.5)

Para el análisis estadísticos de los 2 test realizados, se utilizó la técnica de muestras relacionadas con el estadístico T de Student y un nivel de significancia de 0.02. Dentro de los principales resultados, se obtuvo que los niños mejoraron su nivel de reconocimiento de afecto e interacción social en 3.5 puntos. Este estudio entrenó la cognición social mediante el uso de intercambios sociales del día a día mientras que el niño, a través de la manipulación de un personaje en 3D, interactuaba con otros.

Un segundo trabajo corresponde a Cheng y Tsaib (2019) en donde se desarrolló un videojuego utilizando un HMD para las clases de Ciencias Sociales con el propósito de ayudar a los estudiantes en las excursiones a campos como parte de la enseñanza dentro del propio curso. El diseño del estudio fue cuasiexperimental con un análisis tipo cuantitativo-cualitativo.

El estudio tuvo 4 principales fases, en primer lugar el desarrollo propio del videojuego recolectando los requerimientos de los profesores fue la primera fase, la segunda fase consistió en desarrollar un protocolo de aprendizaje. El objetivo de este último es poder dividir el tiempo tradicional de las clases de lectura y combinarlo con el uso del videojuego de tal modo que se cree una nueva metodología de enseñanza mediante la cual los alumnos puedan aprender. La tercera fase consiste en el entrenamiento propio de los alumnos en donde se aplican ya el protocolo de aprendizaje utilizando el videojuego ya desarrollado. Finalmente, la última fase consiste en la presentación de resultados logrados mediante el aprendizaje de los niños a través del videojuego desarrollado utilizando el dispositivo HMD. El resumen de las fases del proyecto se presentan en la Figura N°11.

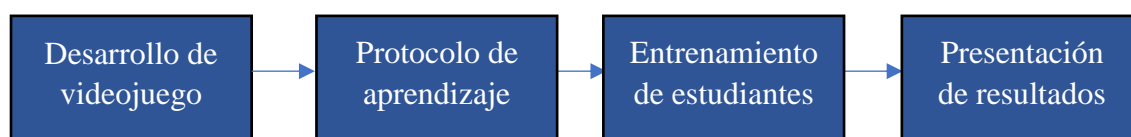


Figura N°11: Fases del proyecto desarrollado

Fuente: Elaboración Propia

El estudio contó con una muestra de 24 estudiantes de primaria en donde se les invitó a una excursión perteneciente a un campamento de verano de 2 semanas de tal manera que ellos usarían el videojuego a través del HMD durante su viaje. La edad de los participantes era de 9 y 12 años. Los participantes fueron 15 niños y 9 niñas.

Una vez desarrollado el videojuego, se procedió a desarrollar el protocolo de aprendizaje. Este protocolo consistió en 10 sesiones equivalentes a 2 semanas. Cada sesión usarían los niños el videojuego por un período de 15 minutos (Cheng y Tsaib, 2019). Las 2 primeras sesiones fueron de reconocimiento de materiales y pruebas con respecto al videojuego para que los niños se sientan familiarizados con los componentes que iban a utilizar. La tercera fase consistió con el entrenamiento de los estudiantes a través del videojuego desarrollado y del correcto seguimiento del protocolo por parte de los niños en las aulas de clase, se presenta en la Figura N°12.



Figura N°12: Utilización del videojuego en las aulas de clases.

Fuente: Cheng, K., y Tsaib, C. (2019). *A case study of immersive virtual field trips in an elementary classroom: Students' learning experience and teacher-student interaction behaviors*. *Computers & Education*, 140. (p.7)

La última fase consistió en la presentación de los resultados. Para el análisis de estos, se utilizó una evaluación de tipo pretest y postest. El nivel de significancia fue de 0.05 para el análisis cualitativo tanto del pre como del postest. Para tener conocimiento sobre si la utilización de la realidad virtual dentro de sus clases eran agradables, se les tomó a los

participantes una encuesta de satisfacción en donde redactaban su experiencia con el videojuego (Cheng y Tsaib, 2019).

Dentro de los principales resultados, se obtuvo que el uso del videojuego fue útil para mejorar las creencias motivacionales de los estudiantes del curso de ciencias sociales mejorando en el posttest el grupo experimental en 0.96 más con respecto al grupo de control y especialmente se comprobó que el videojuego desarrollado reducía la ansiedad de los participantes antes de las evaluaciones (Cheng y Tsaib, 2019). Por otro lado, con respecto al análisis cualitativo, el 90% de los participantes expresaron su placer con respecto al aprendizaje a través del uso de la realidad virtual.

Un tercer trabajo corresponde a Layona, Yulianto, y Tunardi (2018) en donde se desarrolló un videojuego con el objetivo de mejorar el aprendizaje del cuerpo humano. El estudio menciona que tradicionalmente, el aprendizaje del cuerpo humano se da a través de libros y maniqués lo que hace que esta actividad sea un tanto monótona y sin diversión para la persona que desea aprender (Layona et al., 2018).

Para desarrollar el videojuego se utilizó la metodología en cascada en donde se aplicaron de manera secuencial de sus 5 fases: comunicación, planeación, diseño, desarrollo y despliegue (Layona et al., 2018). Para la comunicación se recopilaron los requerimientos tanto de profesores como de alumnado y clasificarlos como funcionales y no funcionales para asignarles una prioridad de desarrollo. En la parte de planeación, se verificaron contar con los recursos necesarios para llevar a cabo el proyecto y realizar un cronograma de trabajo para desarrollar cada actividad que involucra la fase de desarrollo. En la fase de modelado, se realizaron casos de uso y descripción de casos de uso para entender el comportamiento del sistema tal cual como se muestra en la Figura N°13.

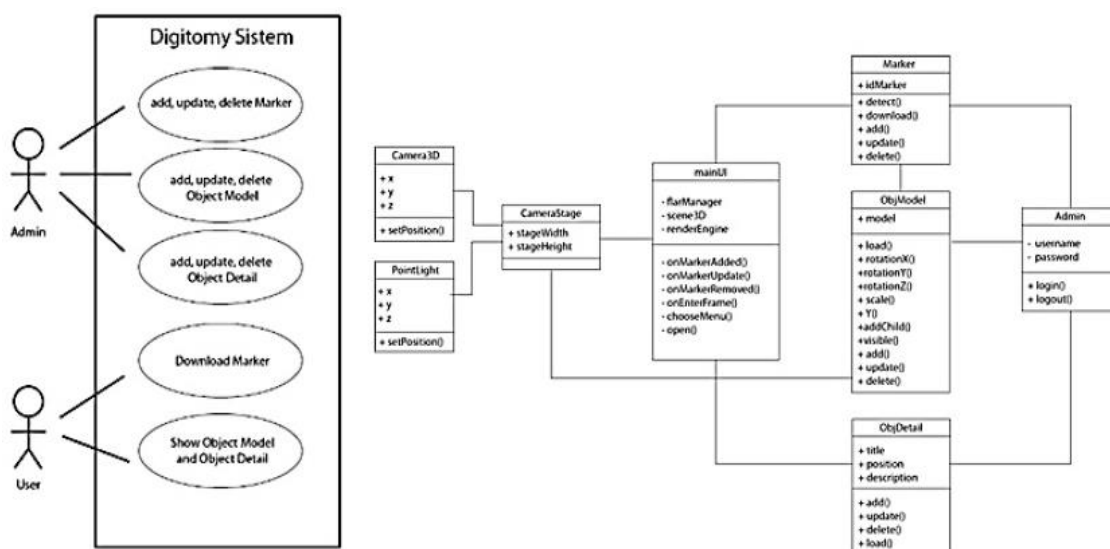


Figura N°13: Diagrama de casos de uso y diagrama de clases.

Fuente: Layona, R., Yulianto, B., y Tunardi, Y. (2018). *Web based Augmented Reality for Body Anatomy Learning. Computer Science*, 135,457-464. (p.4)

Para el desarrollo del videojuego, se construyeron las escenas diseñadas en la fase de modelado. Por último, se realizaron las pruebas de funcionalidad tanto individual como una ejecución del videojuego completo para determinar el correcto funcionamiento de las escenas. El videojuego fue instalado en un servidor publico de tal manera que los estudiantes puedan acceder a través de su computadora. Una imagen del videojuego funcional se presenta en la Figura N°14.



Figura N°14: Visualización de videojuego en la web

Fuente: Layona, R., Yulianto, B., y Tunardi, Y. (2018). *Web based Augmented Reality for Body Anatomy Learning. Computer Science*, 135,457-464. (p.5)

El resumen de las actividades realizadas al utilizar la metodología en cascada se puede apreciar en la Figura N°15.

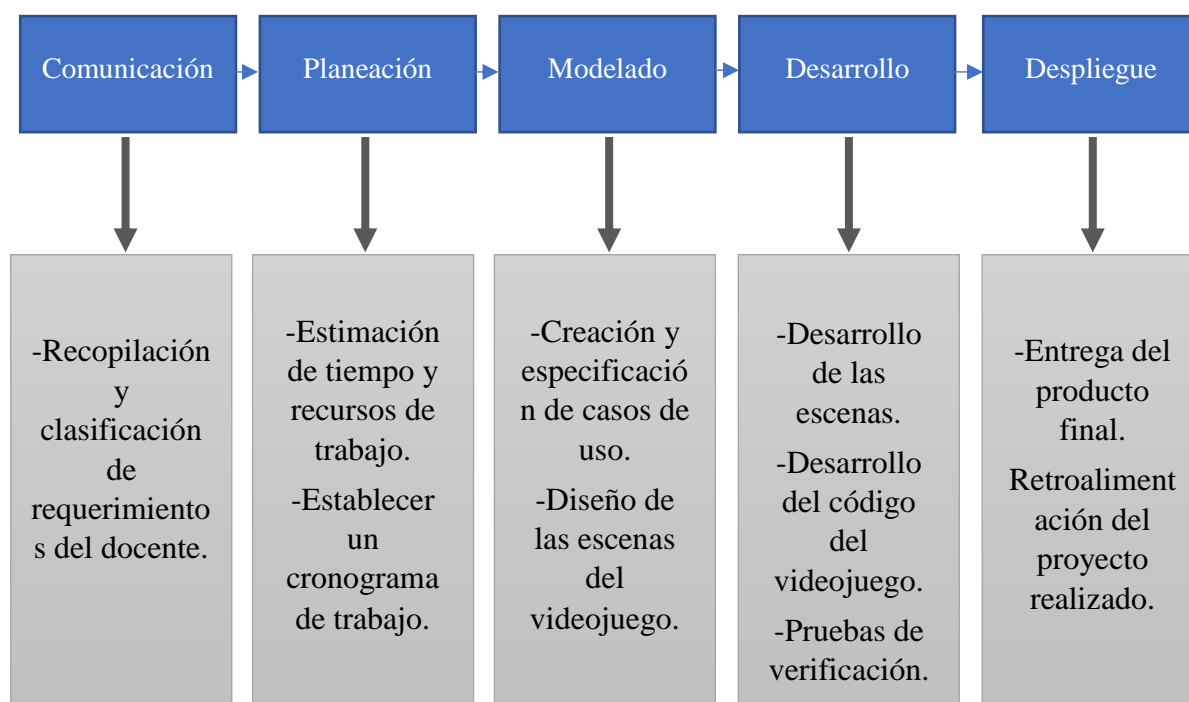


Figura N°15: Metodología en cascada aplicada a videojuegos.

Fuente: Elaboración Propia.

Para medir la efectividad del videojuego, se tomó a los participantes un cuestionario antes y uno después. Se aplicó un análisis estadístico con un nivel de significancia de 0.05 y el principal resultado es que los niños mejoraron sus conocimientos con respecto al conocimiento del cuerpo humano en 1.30 puntos. Por otro lado, los participantes afirmaron sentirse mucho más motivados por aprender acerca del cuerpo humano. Además, brindaron algunas recomendaciones para hacer del juego mucho más entretenido.

Un cuarto trabajo corresponde a Moreno, Vahos, y Mazo (2019) en donde se desarrolló un videojuego con la finalidad de enseñar a los estudiantes el cuidado del agua. En el trabajo se defiende la elección de este tema ya que el agua es un recurso sumamente importante y la enseñanza del cuidado de este recurso posibilita el desarrollo de buenos hábitos con el objetivo de preservar mejor el agua.

El videojuego fue desarrollado a través de la metodología en cascada donde se incluye la recopilación de requerimientos por parte de los profesores, realizar el diseño de cómo sería el videojuego a través de varios bosquejos, realizar la construcción del propio videojuego, realizar las pruebas de funcionalidad y finalmente, implementarlo en el aula de trabajo.

El prototipo se desarrolló con el objetivo de que los estudiantes controlaran un avatar y caminaran por diversos lugares que contengan agua como parques con piletas, lagos, ríos, entre

otros. De tal forma que en videojuego se vaya mostrando qué es lo que sucede cuando las actividades humanas contaminan el agua. Al final de cada sección, se presentan un conjunto de preguntas como retroalimentación de lo aprendido. El resultado del videojuego final construido se aprecia en la Figura N°16.



Figura N°16: Visualización del videojuego terminado.

Fuente: Moreno, J., Vahos, S., y Mazo, C. (2019). *Videojuego para la enseñanza del cuidado del agua*. *TecnoLógicas*, 22(45), 59-72. (p.5)

La muestra estuvo conformada por un grupo de control de 33 estudiantes y un grupo experimental de 32 participantes (Moreno et al., 2019). El diseño de la investigación fue cuasiexperimental con un análisis de tipo cuantitativo-cualitativo y con una medición de tipo pre y postest (Moreno et al., 2019). Además, para medir el grado de satisfacción del videojuego, se tomó una encuesta elaborado por los propios autores que consistían en 4 preguntas, 1 de ellas con escala de likert y el resto eran preguntas abiertas con el objetivo de recoger las diversas opiniones de los estudiantes (Moreno et al., 2019).

El análisis cuantitativo se dividió en 2 secciones: análisis cuantitativo descriptivo y análisis inferencial. El objetivo del primero era mostrar resultados de manera gráfica y hacer una breve descripción de los mismos (Moreno et al., 2019). El objetivo del segundo era validar o rechazar la hipótesis planteada por los investigadores. Para realizar dicha tarea se estableció un nivel de significancia de 0.05.

Como resultados del trabajo de investigación, se obtuvieron que analizando el postest de ambos grupos, el conjunto conformado por los estudiantes que usaron el videojuego mejoraron

significativamente en 0.24 puntos con respecto a los estudiantes que utilizaron la metodología tradicional de enseñanza.

Un quinto trabajo corresponde a Castro (2015) cuyo trabajo consistió en diseñar y desarrollar un videojuego educativo para ejercitar la comprensión lectora del lenguaje español lengua materna (L1) de estudiantes de primer ingreso de la Universidad del Norte.

Las escenas del videojuego constan de un formulario de Login y Registro tal cual como se muestra en la Figura N°17. El videojuego consiste en la manipulación de un personaje en 3D que va recorriendo los salones de un castillo embrujado. En cada salón deberá recoger 3 objetos y responder las preguntas de comprensión lectora que cada objeto le plantea.

La muestra estuvo compuesta por 78 estudiantes de ingeniería Eléctrica, Electrónica, de Sistemas y Educación Preescolar de la asignatura Competencias Comunicativas 1 (en la que se estudian técnicas de comprensión lectora) (Castro, 2015). Dentro de los instrumentos para evaluar la comprensión lectora, se utilizó instrumentos validados por expertos y, además, se usaron los exámenes propios del curso.

The figure shows two side-by-side screenshots of a video game's interface. The left screenshot, titled 'Iniciar sesión', features a red background with white text. It includes input fields for 'Usuario:' and 'Contraseña:', a grey 'Ingresar' button, and a red 'Registrarse' link at the bottom. The right screenshot, titled 'Registrarse', has a dark red background with white text. It includes input fields for 'Nombres:', 'Apellidos:', 'Usuario:', 'Contraseña:', and 'Confirmar contraseña:', and a dark red 'Registrarme' button at the bottom.

Figura N°17: Login del videojuego (izq.) y Registro de alumnos (Der.).

Fuente: Castro, A. D. (2015). *Maximum Consequentia: un videojuego educativo para ejercitar la comprensión lectora. Innovación educativa*, 25,191-206. (p.195)

Dentro de los principales resultados, están que los estudiantes lograron un incremento en su comprensión lectora en 3.86 puntos adicionales usando el videojuego con un nivel de significancia de 0.05. Seguidamente, afirmaron a través de un cuestionario es posible jugar y aprender a la vez usando el videojuego. Además, los estudiantes dieron la recomendación de que el videojuego sea compatible con celulares o Tablet con el objetivo de que puedan entretenerse en cualquier lugar.

Un sexto trabajo corresponde a Huerta (2017) en donde se desarrollo un videojuego en una plataforma web con la finalidad de que estudiantes de educación primaria puedan mejorar su nivel de comprensión lectora. Para ayudar a los estudiantes a comprender mejor el texto, se intrdujo un apartadado de vocabulario para que el niño pueda entender el significado de una palabra que no entendia y que estaba presente en el texto que estaba leyendo (Huerta, 2017). Seguidamente, el niño recibia ejercicios que estaban presentes de manera virtual en el sistema para que el niño pueda poner en practica tanto su comprensión con la lectura como las nuevas palabras aprendidas.

Para el desarrollo del prototipo se utilizó la metodología SCRUM tal cual como se muestra en la Figura N°18 dado que se contó con un equipo multidisciplinario y la idea era realizar el videojuego por partes de tal modo que se iban dando entregables a los profesores para que puedan ya ser utilizados por los alumnos.

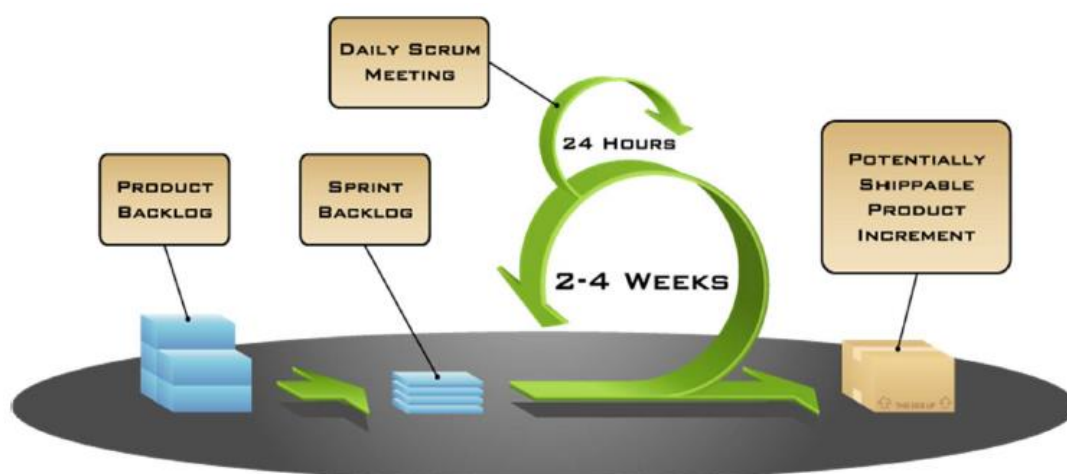


Figura N°18: Metodología Scrum aplicada a videojuegos.

Fuente: Huerta, R. (2017). *Plataforma web para el apoyo en la mejora de la comprensión lectora en los alumnos de educación básica*. Lima. (p.20)

La muestra consistió en un grupo de control conformado por 39 niños (sin intervención con el sistema) y un grupo experimental conformado con 49 niños (con intervención del sistema). Además, se tomaron dos exámenes a los largo de todo el proyecto: un examen inicial antes de la intervención y otro al finalizar con el experimento.

Dentro de los resultados, se pudo observar que el grupo experimental tuvo una mejora en su nivel de comprensión lectora en 1.04 puntos estableciendo un nivel de significancia de 0.05 observando que las notas de las pruebas de salida eran mayor a las notas de la prueba de entrada a comparación con el grupo de control.

Adicionalmente, a los trabajos de investigación presentados en donde se ha detallado el uso de los videojuegos dentro de un salón de clases para la mejora de una determinada habilidad, se ha realizado una búsqueda de trabajos relacionados a Inteligencia Artificial en videojuegos de tal modo que el videojuego a desarrollar pueda ser mucho más atractivo para los usuarios. La idea es recabar algunos trabajos de investigación en donde se aplique Inteligencia Artificial de tal modo que se puedan recopilar las técnicas utilizadas y adaptar esa técnica al videojuego a desarrollar. A continuación se presentarán los trabajos de investigación recopilados donde aplican Inteligencia Artificial a videojuegos.

Un séptimo trabajo corresponde a Expósito (2018) en donde se desarrolló un videojuego llamado “*DungeonBrain*” utilizando Inteligencia Artificial de tal modo que el agente inteligente implementado pueda destruir a los enemigos dentro del videojuego.

Tal cual como se muestra en la Tabla 3, el agente tenía que entender 5 principales acciones, se le otorgaba una recompensa cada vez que realizaba las acciones de manera adecuada y se le penalizaba si es que las acciones eran incorrectas.

Tabla 3: Acciones a realizar por el agente del videojuego

Valor	Acción
0	Pararse
1	Seguir al jugador
2	Volver a la posición inicial
3	Atacar al jugador
4	Usar beneficio

Fuente: Expósito, P. (2018). *Inteligencia Artificial de Comportamiento de Agentes no jugadores en un RPG clásico con Unity y TensorFlow*. (p.39)

Para el modelo creado, la red neuronal tuvo 1024 capas ocultas con 3 neuronas por cada capa oculta y tres épocas para el entrenamiento. Por cada acción ejecutada por la red neuronal se otorgaba una recompensa al agente de tal manera que el agente iba entendiendo la acción a realizar.

Uno de las conclusiones de este trabajo de investigación para la implementación de un agente dentro de un videojuego es que al incluir demasiadas acciones a la red neuronal origina que el entrenamiento se relentece. Otra conclusión importante es con respecto al tamaño de las recompensas y castigos en el agente. Lo que se recomienda es que los números para cada refuerzo sean pequeños de tal modo que no entorpezca el entrenamiento dado que podría llevar al agente a realizar acciones erróneas (Expósito, 2018).

Un octavo trabajo corresponde a Ropilo (2019) en donde se desarrolló una Inteligencia Artificial para un videojuego para que este agente recolecte la mayor cantidad de monedas bonus mientras iba esquivando los obstáculos del videojuego.

La escena entrenada para el funcionamiento el agente constó de 4 paredes con obstaculos en ella y el agente representado por un *sprite* debe de acumular la mayor cantidad de monedas mientras esquiva los obstáculos de la propia escena de entrenamiento tal cual como se presenta en la Figura N°19.

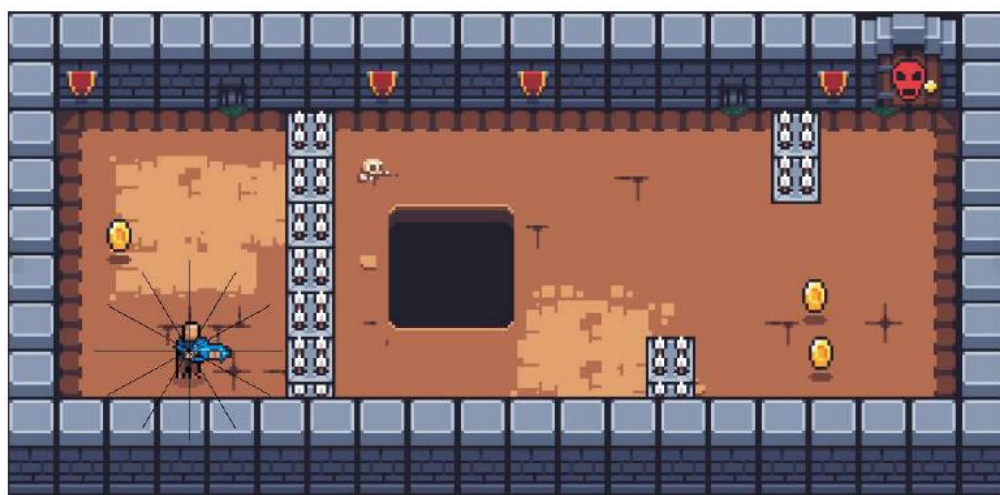


Figura N°19: Ambiente desarrollado para el entrenamiento del agente.

Fuente: Ropilo, T. (2019). *Teaching a machine learning agent to survive in a 2D Top-Down Environment*. (p.13)

Las acciones para realizar por el agente son de desplazamiento hacia las 4 direcciones (arriba, abajo, izquierda y derecha). Las recompensas para el agente eran positivas si es que aquel esquivaba los obstáculos de fuego, si evitaba las paredes con púas y evitaba caerse a un pozo. Las recompensas negativas se daban por cumplir las restricciones anteriores y además si cruzaba la puerta de salida sin haber recolectado todas las monedas (Ropilo, 2019).

En los resultados del entrenamiento, se observó que el agente evadía los obstáculos y evitaba los ataques de fuego. La recolección de las monedas fue lo que más tiempo llevó en aprender dado que inicialmente el agente daba vueltas en círculos hasta detectar las monedas y no se llegaba a desplazar correctamente (Ropilo, 2019). Además, se obtuvo una recompensa promedio de 0.8 y que el agente luego de 450 000 iteraciones empezó a entender las acciones que se le habían encomendado.

Una novena investigación corresponde a Adamsson (2018) en donde se entrenó a un agente mediante *Deep Reinforcement Learning* utilizando la librería ML-Agents de Unity para que

pueda esquivar disparos dentro de una escena de tal modo que entre más disparos acumule, más puntaje obtiene.

Para desarrollar el *environment* y crear al agente inteligente, se utilizó como plataforma de videojuegos a Unity con la librería de ML-Agents que proporciona los componentes necesarios para implementar un multilayer perceptron.

Para la implementación del aprendizaje por refuerzo profundo para la creación de agentes inteligentes utilizando la librería de ML-Agents de Unity, se siguieron 3 etapas. La primera etapa consistió en el desarrollo del ambiente donde se entrenó al agente inteligente. La segunda etapa consistió en el entrenamiento del propio agente, esta etapa se dio a través del establecimiento de un modelo de *Deep Reinforcement learning* el cual contiene hiperparámetros pertenecientes al algoritmo de *Proximal Policy Optimization* y para la red neuronal. El último paso consistió la implementación del agente dentro del propio videojuego desarrollado. El resumen de las tres etapas para la implementación de *Deep Reinforcement Learning*, se presenta en la Figura N°20.

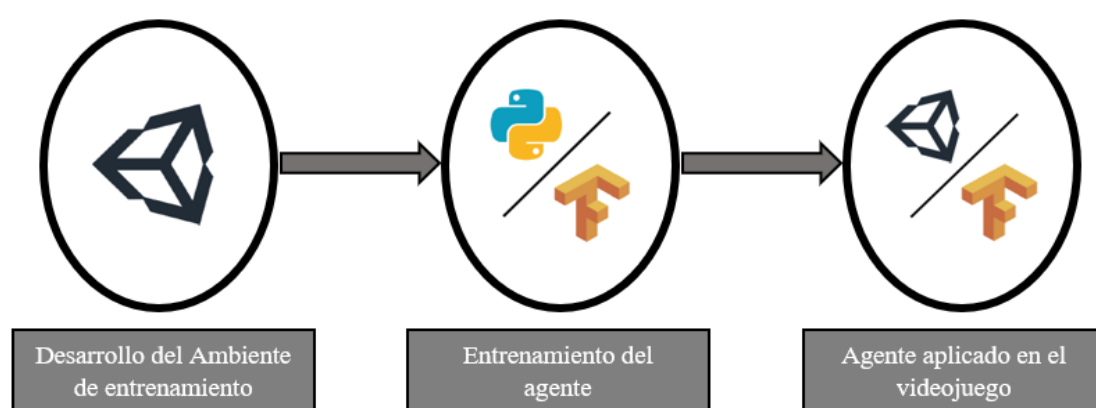


Figura N°20: Implementación de aprendizaje por refuerzo profundo usando ML-Agents.
Fuente: Elaboración Propia.

Los resultados obtenidos mostraron que se obtuvo, luego 2.5 millones de iteraciones, se obtuvo una recompensa acumulada de -35 aproximadamente. Las conclusiones del proyecto demostraron que es de vital importancia desarrollar un ambiente de entrenamiento que se asemeje al real de tal modo que el agente puede aplicar la tarea aprendida de la mejor manera posible (Adamsson, 2018). Por otro lado, la configuración de los valores para hiperparámetro puede influir en los resultados del entrenamiento así como la cantidad de acciones que se le asigne al agente de tal modo que se recomendó no asignarle al agente una variedad de acciones a aprender dado que puede ocasionar un lento entrenamiento y un mayor esfuerzo computacional (Adamsson, 2018).

A modo de resumen, con respecto a los trabajos de investigación donde se desarrollan videojuegos en un contexto educacional, se presenta la Tabla 4 donde se hacen mención a los autores que desarrollaron cada trabajo de investigación, la base de datos a la que se hizo consulta para extraer cada artículo, el tipo de evaluación que se hizo que tal cual como se ha visto, consta de un pre y post test en la mayoría de casos y adicionalmente se suele agregar una encuesta de satisfacción para saber si el videojuego fue del agrado o no de los usuarios. Además, se presenta una columna con el tipo de diseño para cada trabajo de investigación, el tipo de prueba estadística utilizada para el análisis de información y la cantidad total de participantes divididos en grupo de control y grupo experimental.

Tabla 4: Resumen de antecedentes de videojuegos en un contexto educacional

Autor	Base de datos utilizada	Evaluación	Diseño	Análisis estadístico	Muestra
Didehbani et al. (2016)	Scopus	-Pretest y Postest.	Cuasiexperimental	Prueba T de student	N = 30
Cheng y Tsaib (2019)	Scopus	-Pretest y Postest.	Cuasiexperimental	Prueba T de student	N = 24
Layona, Yulianto, y Tunardi (2018)	Scopus	-Pretest y Postest.	Cuasiexperimental	Prueba T de Student	N = 48
Moreno, Vahos, y Mazo (2019)	Google Scholar	-Pretest y Postest. -Encuesta de satisfacción.	Cuasiexperimental	Prueba T de student	N = 55 GC: 33 GE:32
Castro (2015)	Google Scholar	Pretest y Postest.	Experimental	Prueba T de student	N = 78

Huerta (2017)	Google Scholar	Pretest y Posttest	Experimental	Promedio, moda, desviación estandar.	N = 88 GC: 39 GE:49
------------------	----------------	-----------------------	--------------	---	---------------------------

Nota. N = Número total de la muestra; GC= Grupo de control; GE= Grupo Experimental.
Fuente: Elaboración Propia

Adicionalmente, en la Tabla 5, se presentan las principales metodologías utilizadas en los trabajos de desarrollo de videojuegos para fines educacionales, el nivel de significancia establecido para realizar el análisis estadístico y la mejora significativa que hubo por cada trabajo.

Tabla 5: Resumen de metodologías aplicadas, nivel de significancia establecido y mejora significativa obtenida en los antecedentes de videojuegos en un contexto educacional

Autor	Metodología	Nivel de significancia	Mejora significativa
Didehbani et al. (2016)	Metodología Mixta (Educativa-VR)	0.02	3.05 puntos
Cheng y Tsaib (2019)	Metodología Mixta (Educativa-VR)	0.05	0.96 puntos
Layona, Yulianto, y Tunardi (2018)	Metodología en cascada	0.05	1.30 puntos
Moreno, Vahos, y Mazo (2019)	Metodología en cascada	0.05	0.24 puntos
Castro (2015)	Propuesta de metodología de desarrollo de videojuego	0.05	3.86 puntos

Huerta (2017)	Metodología SCRUM	0.05	1.04 puntos
---------------	----------------------	------	-------------

Fuente: Elaboración Propia

De la Tabla 5 es preciso resaltar que diversos trabajos aplican una metodología de trabajo mixta que combina el uso de un videojuego de realidad virtual semi inmersiva o inmersiva total con la metodología tradicional de enseñanza.

En esta metodología en particular, no existen pasos formales y definidos dado que las actividades a realizar se establecen en mutuo acuerdo de las partes interesadas : la parte de los desarrolladores que construyen el videojuego y la parte de los profesores, psicólogos o los educadores que estén participando del proyecto. Esto se dá de esta manera dado que al utilizar una tecnología emergente como la realidad virtual especialmente si es semi inmersiva o inmersiva total, requiere la utilización de equipos tecnológicos por parte de los alumnos por lo que se debe establecer bien las bases del uso de este equipo así como el uso de los mismos durante el periodo de entrenamiento (Geronazzo et al., 2019).

2.2 Bases Teóricas

En la presente sección se explicará la definición de los videojuegos, así como los tipos de videojuegos que existen. Seguidamente, se abordarán las principales metodologías existentes para el desarrollo de videojuegos, así como las tecnologías y técnicas de Inteligencia Artificial utilizadas para el desarrollo de videojuegos y finalmente se realizará la explicación de la técnica estadística aplicada para el análisis de los resultados.

2.2.1 Videojuegos

Diversos autores han dado su propia definición de lo que es un videojuego. Por ejemplo, para Frasca (2001) un videojuego es “cualquier software de entretenimiento por computadora, usando una plataforma electrónica y la participación de uno o muchos jugadores en un entorno físico o de red” (Eguia et al., 2012, p. 5). Además, según Zyda (2005), un videojuego es “una prueba mental, llevaba a cabo frente a una computadora de acuerdo con reglas cuyo objetivo es la diversión o esparcimiento” (Eguia et al., 2012, p. 5). Finalmente, en Juul (2005) se afirma que un videojuego es “un juego usando una computadora y un visor de video. Puede ser un computador, un teléfono móvil o una consola de juegos” (Eguia et al., 2012, p. 5). A modo de resumen, se puede decir que un videojuego es cualquier software destinado al entretenimiento

el cual es usado a través de dispositivos electrónicos, generalmente en computadoras o teléfonos.

Para clasificar los videojuegos, se hará hincapié en la clasificación propuesta en Martín et al. (1995) citado en García (2009), donde se hace una clasificación de los videojuegos en función a su contenido en siete ramas:

- **Arcade**, en los que el jugador, a través de un personaje, ha de superar obstáculos de dificultad creciente, matar a sus enemigos y recoger ítems necesarios para continuar en el juego.
- **Aventura**, bajo la premisa de conseguir un objetivo determinado en un ambiente de incertidumbre y peligro, en el que el jugador debe superar distintas pruebas de resolución de problemas, enigmas y luchas.
- **Estrategia**, en los que el jugador debe controlar muchas variables para conseguir un fin concreto.
- **De rol**, que siguen los mismos parámetros que los tipos de juegos de mesa con este mismo nombre.
- **Simuladores**, que reproducen situaciones o actividades complejas como por ejemplo, los simuladores de vuelo, los de conducción de vehículos o los de realización de deportes concretos.
- **Educativos**, Juegos en los que prima la finalidad educativa, sobre todo.
- **Juegos de mesa**, que se asemejan a los juegos de mesa tradicionales. (p. 123)

2.2.1.1 Unity: Motor gráfico para el desarrollo de videojuegos

El motor gráfico de Unity sirve para la creación de videojuegos en 2D o 3D, cualquier tipo de material interactivo, así como animaciones en 3D (Alonso J. , 2001). Una de las características que diferencia a Unity de otros motores es que éste permite ejecutar aplicaciones desde un navegador web. Además, permite crear aplicaciones para Windows, MAC, Android y recientemente, con las últimas actualizaciones, se pueden crear aplicaciones tanto como para Play Station 3 como para Xbox 360 (Alonso J. , 2001).

Unity posee el 45% de participación en el mercado de motores gráfico para videojuegos, tres veces más que su competidor más próximo que es Unreal Engine. Además, hay alrededor de 5 millones de desarrolladores de Unity y alrededor de 2.400 millones de dispositivos móviles utilizan esta plataforma (Nanalyze, 2017). Lo anterior mencionado, se aprecia en la Figura N°21.

Global game engine market share

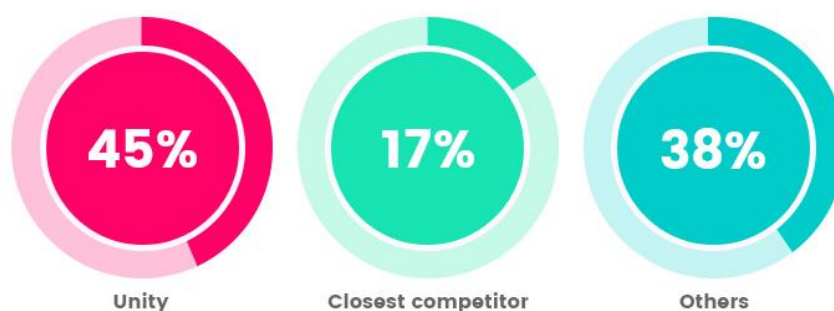


Figura N°21: Porcentaje de participación de Unity en el mercado.

Fuente: Quora (2016). *How many game programmers use Unity?* Recuperado de: <https://www.quora.com/How-many-game-programmers-use-Unity>

2.2.2 Metodologías para el desarrollo de videojuegos

El objetivo de establecer una metodología para desarrollar un videojuego es tener planificado las fases que un equipo debe de seguir para llegar a desarrollar el producto final que desea el cliente. Es preciso mencionar que no existe una metodología única para desarrollar videojuegos, sino que esta se tiene que amoldar en función a la naturaleza del proyecto, los requerimientos del cliente, el tiempo, el costo establecido, entre otros aspectos (Drake, 2008). Establecer una secuencia o fases para desarrollar videojuegos en un proyecto permite mejorar la calidad del producto final, disminuir errores en el desarrollo, mejorar el entendimiento de las tareas o actividades de cada participante y generar información adecuada para obtener una retroalimentación de todo el proceso en general.

2.2.2.1 Matriz de evaluación de metodologías de software

En Méndez (2006) se desarrolla una propuesta para seleccionar la metodología más adecuada para el desarrollo de un proyecto de software. Para elaborar la matriz y seleccionar la metodología mas adecuada se hizo una investigación en diversos artículos para determinar las variables más importantes a la hora de seleccionar una metodología (Méndez, 2006). Además, se hicieron entrevistas y se elaboraron cuestionarios a personas encargadas de dirigir proyectos de software y a personas que intervienen las distintas fases del proyecto. La propuesta para la elección de la metodología adecuada contiene 13 variables o factores de éxito. Además de establecer los factores de éxito, se establecieron las escalas de poderación que se debían seguir. Las escalas van del 1 al 4 y su significado se encuentran presentes en la Tabla 6.

Tabla 6: Ponderaciones para la elección de la metodología adecuada

Valor	Descripción
1	Debilidad Grave
2	Debilidad Menor
3	Fortaleza Menor
4	Fortaleza Importante

Fuente: Méndez, E. (2006). Modelo de evaluación de metodologías para el desarrollo de software. Caracas:Universidad Católica Andres Bello.

Por último se obtuvo un cuadro general, en donde el encargado de escoger la metodología adecuada deberá escoger la puntuación que le corresponde a cada metodología por cada factor de éxito y multiplicarlo por el eso correspondiente a cada factor obteniéndose así un resultado ponderado por cada factor (Méndez, 2006). Al final, se deberá sumar todos los resultados ponderados por cada factor en la metodología correspondiente y hacer una comparación entre el total de una con el total de la otra. La metodología que se escogerá será aquella que tenga un resultado ponderado mayor (Méndez, 2006). La matriz general para hacer las comparaciones entre metodologías de software en donde se visualiza cada factor de éxito, así como sus ponderaciones para cada factor, se visualiza en la Figura N°22.

<i>Factor clave de éxito</i>	<i>Ponderación</i>	<i>Metodología 1</i>		<i>Metodología 2</i>	
		<i>Puntuación</i>	<i>Resultado Ponderado</i>	<i>Puntuación</i>	<i>Resultado Ponderado</i>
La metodología debe ajustarse a los objetivos	0.080				
La metodología debe cubrir el ciclo entero del desarrollo de software	0.086				
La metodología debe integrar las distintas fases del ciclo de desarrollo	0.086				
La metodología debe incluir la realización de validaciones	0.080				
La metodología debe soportar la determinación de la exactitud del sistema a través del ciclo de desarrollo	0.075				
La metodología debe ser la base de una comunicación efectiva	0.078				
La metodología debe funcionar en un entorno dinámico orientado al usuario	0.064				
La metodología debe especificar claramente los responsables de resultados	0.078				
La metodología debe emplearse en un entorno amplio de proyectos de software	0.083				
La metodología se debe de poder enseñar	0.075				
La metodología debe estar soportada por herramientas CASE	0.070				
La metodología debe soportar la eventual evolución del sistema	0.075				
La metodología debe contener actividades conducentes a mejorar el proceso de desarrollo de software	0.072				
TOTAL					

Figura N°22: Matriz de evaluación de metodologías.

Fuente: Méndez, E. (2006). *Modelo de evaluación de metodologías para el desarrollo de software*. Caracas: Universidad Católica Andrés Bello.

2.2.2.2 Metodología en cascada

El modelo en cascada es una metodología usada en el desarrollo de software que consta de 5 fases que se siguen de manera secuencial (Pressman, 2010). Este modelo se suele utilizar cuando los requerimientos para el desarrollo del software están bien definidos y el cliente está seguro de que estos serán estables a lo largo del proyecto (Pressman, 2010). El modelo en cascada consta de 5 fases: Comunicación, Planeación, Modelado, Desarrollo y Despliegue. Estas serán explicadas en las siguientes sub secciones.

2.2.2.2.1 Comunicación

Con esta etapa se inicia el proyecto y es quizá la más importante de todas debido a que se hace la recolección de los requerimientos del cliente y que da como resultados todas las características que el videojuego final contendrá. Es preciso mencionar que en esta etapa se debe comprender las funcionalidades que el videojuego debe poseer, pero sin entrar a detalles muy técnicos (Pressman, 2010).

2.2.2.2.2 Planeación

En esta etapa, se hace estimación de todos los recursos a utilizar, así como la elaboración de todas las actividades técnicas que se desarrollarán por cada etapa (Pressman, 2010). Como resultado de esta etapa, se obtiene un cronograma general de trabajo con el objetivo de dar seguimiento a todas las tareas a desarrollar.

2.2.2.2.3 Modelado

En la fase de modelado, se hace un diseño de lo que se va a construir. Se juntan los requerimientos del cliente para llegar a un bosquejo de las escenas que tendrá el producto final, un diseño de los personajes, módulos, entre otros. Además de elaborar el diseño principal del videojuego, también se elaboran los detalles con respecto a la arquitectura que tendrá al final, la secuencia de *GamePlay* que seguirá, arquitectura de módulos e interfaces que contendrá, entre otros (Pressman, 2010). Como resultado final, se obtienen los detalles del comportamiento que poseerá el videojuego a desarrollar en función a los requerimientos otorgados en la fase de comunicación.

2.2.2.2.4 Desarrollo

En la etapa de desarrollo, el diseño se traduce a un programa utilizando ya uno o varios lenguajes de programación y que cumple con las necesidades del cliente. Normalmente, en esta etapa se obtiene una primera versión del videojuego, la cual se van mejorando conforme se detecten errores en las ejecuciones (Ionos, 2019). En esta fase, todos los diseños de la interfaz,

módulos correspondientes y modelos de relaciones se convierten en módulos reales que funcionan por separado (Ryte Wiki, 2019). Además, en esta fase se realizan las pruebas para saber si cada componente está funcionando de manera correcta. Las pruebas son un conjunto de actividades que verifican que el sistema posee un determinado requerimiento del cliente (Pressman, 2010). De encontrarse un error, este debe ser corregido de inmediato para que no afecte el comportamiento general del sistema (Ionos, 2019).

2.2.2.2.5 Despliegue

En esta última fase, se procede a entregar e instalar el videojuego final sabiendo que todos sus componentes funcionan de manera adecuada. Una vez, instalado, se debe hacer un plan para su respectivo mantenimiento. Es esta última etapa, es preciso hacer un plan de seguimiento para detectar los posibles errores que el videojuego pueda tener y para evitar tener algún tipo de problema con el cliente (Domínguez, 2017). Finalmente, se obtiene una retroalimentación se todo lo elaborado en el proyecto para saber los errores que hubo, cómo se solucionaron y en qué se puede mejorar. A modo de resumen, en la Figura N°23, se muestra el modelo de la metodología en cascada.

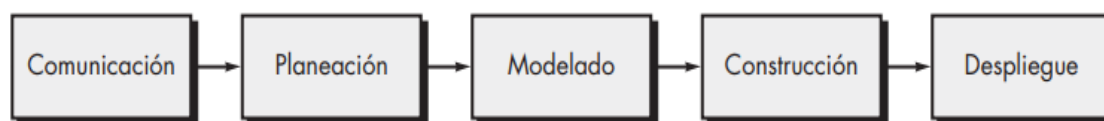


Figura N°23: Modelo de cascada.

Fuente: Pressman, R. (2010). *Ingeniería de software. Un enfoque práctico*. México, D.F.: The McGraw-Hill. (p.34)

2.2.3 Aprendizaje basado en videojuegos

El aprendizaje basado en videojuegos consiste en usar un videojuego con el fin de poder enseñar de una mejor manera a las personas a través de la mejora del proceso de aprendizaje (Trends E., 2016). Normalmente, se suele utilizar en el ámbito de la educación con el fin de que estudiantes se involucren más con las clases impartidas por un profesor y para que este último pueda utilizar nuevas herramientas (en este caso tecnológicas) con el fin de poder hacer de las clases mucho más dinámicas y entretenidas.

En la Tabla 7, se puede visualizar los beneficios de la aplicación de esta nueva tendencia para la mejora del aprendizaje en estudiantes.

Tabla 7: Beneficios de la aplicación del aprendizaje basado en videojuegos

COGNITIVOS	MOTIVACIONALES	EMOCIONALES	SOCIALES
Mejora la concentración.	Incrementa el compromiso en las actividades propias.	Promueve la relajación.	Mejora la comunicación con otros jugadores.
Mejora la inteligencia viso espacial.	Ayuda a tener un mejor desempeño en el trabajo en equipo.	Mejora el estado de ánimo.	Reduce los sentimientos de hostilidad entre todos los miembros del equipo.
Ayuda a mejorar la atención.	Enseña a controlar las frustraciones con el fin de lograr el objetivo final.	Evita la ansiedad.	Incrementa la capacidad de liderar equipos.
Ayuda a discriminar la información relevante de la imprescindible.	Enseña a los jugadores a ser persistentes.	Incrementa el sentimiento de alegría luego de lograr pasar un nivel con un alto grado de dificultad.	Ayuda a poder organizar las ideas de otros participantes para crear una estrategia.
Permite mejorar el rendimiento de los individuos.	Contribuye al desarrollo de la automotivación.	Contribuye a la mejora de la autoestima.	Ayuda a tener un mejor pensamiento crítico al estar expuesto a determinadas circunstancias.

Fuente: Granic, I., Lobel, A., y Engels, R. (2014). *The Benefits of Playing Video Games*.

2.2.4 Videojuego como herramienta de aprendizaje en el aula

Es preciso mencionar que la utilización de los videojuegos en las aulas de clases parte de una necesidad de poner desarrollar las clases de una mejor manera de modo que los niños puedan divertirse mientras aprenden y puedan retener por mucho tiempo lo aprendido (Morales, 2009). Las aulas de los colegios poseen reglas de comportamiento, normas que los alumnos deben de seguir para poder desarrollar sus actividades diarios. Partiendo de ese punto, se puede decir que un videojuego posee características similares ya que son un conjunto pequeño de reglas, de tareas que los niños deben de desarrollar con el objetivo de poder pasar las metas (Morales, 2009).

2.2.4.1 Beneficios para el docente

Se ha mencionado anteriormente los múltiples beneficios que trae el uso de los videojuegos en los jugadores, pero también la utilización de los videojuegos en las aulas de clase da mayores ventajas al profesor para poder mejorar la calidad en enseñanza dentro del salón. Según Morales (2009), Estas serán son las siguientes:

- Los videojuegos son ya conocidos por los niños ya que la mayoría de ellos habrá jugado con alguno en alguna ocasión. Por lo que el profesor no tiene que explicar a grandes rasgos el uso de los videojuegos a los niños.
- Los videojuegos dentro del aula como herramienta educativa pueden servir para poder enseñar algún tema específico que a los niños les resulte complicado entender de manera tradicional.
- La utilización de videojuegos en el aula contribuye al desarrollo social entre los propios alumnos disminuyendo las diferencias entre ellos ya que todo el salón deberá trabajar como equipo con el objetivo de poder alcanzar las metas del videojuego y superar los niveles.

2.2.5 La comprensión lectora

Para Bormuth, Manning y Pearson (1970) la comprensión lectora es “el conjunto de habilidades cognitivas que permiten al sujeto adquirir y exhibir una información obtenida a partir de la lectura del lenguaje impreso” (Gordillo y Del Pilar (2009), p. 97). Además, para Peronard y Gómez (1991) la comprensión lectora es el resultado de la interacción entre el texto y el lector en donde este último, a través de los conocimientos que posee trata de dar un significado a lo leído para obtener una interpretación adecuada del contenido (Gordillo y Del Pilar (2009), p. 97).

2.2.5.1 Niveles de la comprensión lectora

Strang (1965), Jenkinson (1976) y Smith (1989) citados en Gordillo y Del Pilar (2009), afirman que existen 3 niveles diferentes de comprensión lectora los cuales se presentan en la Tabla 8.

Tabla 8: Niveles de la comprensión lectora

Nivel Literal	Nivel Inferencial	Nivel Crítico
Se centra en la información que se encuentra en el texto de manera explícita como el orden de la narración de los hechos, los personajes que intervienen en la lectura, entre otros (Gordillo y Del Pilar, 2009).	Permite al lector reconocer asociaciones presentes dentro del texto para poder deducir de lo que se encuentra de manera implícito dentro del texto (Gordillo y Del Pilar, 2009).	El lector elabora argumentos sólidos con respecto a lo leído, puede aceptar o rechazar partiendo de la elaboración de una justificación adecuada (Gordillo y Del Pilar, 2009).

Fuente: Gordillo, A., y Del Pilar, M. (2009). *Los niveles de comprensión lectora: hacia una enunciación investigativa y reflexiva para mejorar la comprensión lectora en estudiantes universitarios*.

2.2.5.2 Importancia de la enseñanza de comprensión de los textos

En Camargo, Montenegro, Maldonado, y Magzul (2016) se presentan las razones por las cuales es importante la enseñanza de la comprensión de textos en el aula las cuáles se presentan a continuación:

- **La lectura es el principal medio para desarrollar el lenguaje:** Los niños a través de las lecturas van asimilando el vocabulario que se encuentra en ella (Camargo et al., 2016). Además, el niño va entendiendo las estructuras de las oraciones a través de los ejemplos que leen en las lecturas y pone en práctica generalmente esta habilidad cuando ellos escriben sus propios cuentos.
- **La lectura estimula la imaginación creadora:** Los niños van representándose mentalmente los personajes que van leyendo. Ellos tratan de poder comprender el contexto en donde se origina la lectura, cómo es el mundo donde viven los personajes, entre otras cosas. Este ejercicio posibilita la estimulación de la imaginación de los niños para que ellos puedan entender mejor lo que leen.

- **La lectura expande la memoria:** En las comunidades, por ejemplo, se suelen relatar cuentos o refranes de manera oral a los niños de la comunidad. Esto posibilita poder hacer un esfuerzo adicional para poder acumular información en la memoria y para poder utilizar está a largo plazo. Los refranes, acertijos, entre otros, son medios que ayudan a los niños a poder retener mejor la información y ayuda a preservar las creencias y culturas populares que vienen dadas de generación en generación (Camargo et al., 2016).
- **La lectura desarrolla el sentido crítico:** El lector que lee a diversos autores tiene la posibilidad de escuchar distintos puntos de vista con el objetivo de poder mejorar sus propios pensamientos, de poder debatir las ideas que plantean con el objetivo de poder pensar de manera más adecuada, teniendo conocimientos mucho más sólidos con respecto a su realidad que lo rodea (Camargo et al., 2016).

2.2.5.3 La Lectura Comprensiva como tema del aprendizaje de textos

La lectura comprensiva es uno de los principales temas que se suelen abordar dentro del curso de los cursos de comprensión lectora ya que el objetivo de este tema es que las personas aprendan a leer lo que leen (Matesanz, 2014). Comprender es capturar los contenidos que uno va leyendo a través las lecturas. Los lectores que aprenden a leer comprensivamente, asimilar la información que posee la lectura, luego la procesar y la convierten en nuevo conocimiento para ellos, el cual puede servir para la elaboración de nuevos conocimientos (Javier, 2013).

2.2.6 Nueva metodología de enseñanza bajo la pandemia de Covid-19

Desde mediados de marzo del 2020, con la aparición del nuevo coronavirus, las escuela y centros de trabajo han tenido que realizar sus actividades de manera virtual.

En la mayoría de los colegios particulares del Perú, los colegios han adoptado una modalidad 100% virtual con sus alumnos. Las principales aplicaciones que se utilizan para la realización de las clases son Zoom y Google Meet. Los alumnos suelen interactuar con el profesor a través de una web cam mientras van explicando con diapositivas en reemplazo de las pizarras (Mueller y Taj, 2020).

Las tareas que tradicionalmente se realizaban de manera de manera presencial, ahora se suelen realizar a través de evidencia como fotos y videos para que los profesores tengan sustento de que los alumnos realizan las actividades encomendadas.

Con respecto a los colegios estatales y pensando además en escolares que no cuentan con recursos en sus propias casas para acceder a una computadora o laptop, el gobierno peruano ha

adoptado a la educación televisiva de manera momentánea para que los alumnos puedan seguir adquiriendo conocimientos. Lo que se transmite en televisión va desde grabaciones de clases presenciales, caricaturas educativas, grabaciones especiales por parte de profesores, entre otros.

En el caso de Perú, se ha elaborado un plan de estudios para niños de todos los grados de tal modo que se pueda seguir una enseñanza para todos por igual. Las lecciones que se brindan corresponden a los cursos que se ven de manera tradicional como historia, personal social, lectura, finanzas, entre otros (Mueller y Taj, 2020).

A esta modalidad de enseñanza desarrollada en el Perú, se le ha dado el nombre de “Aprendo en casa”. El cual es un trabajo conjunto por parte de canales de televisión, radios y redes sociales para que los jóvenes tanto de primaria como secundaria puedan seguir aprendiendo (Mueller y Taj, 2020).

A través de las redes sociales del Ministerio de Educación, se suben los temas que se verán en el transcurso de la semana tanto en la tv como en las radios (Mueller y Taj, 2020). En el caso de los estudiantes de primaria, se ha sugerido una mayor participación de los padres de familia para la realización de actividades (Mueller y Taj, 2020). En la Figura N°24, se puede apreciar un ejemplo de cómo los estudiantes aprenden a través de la modalidad de educación a distancia durante la pandemia del Covid-19.

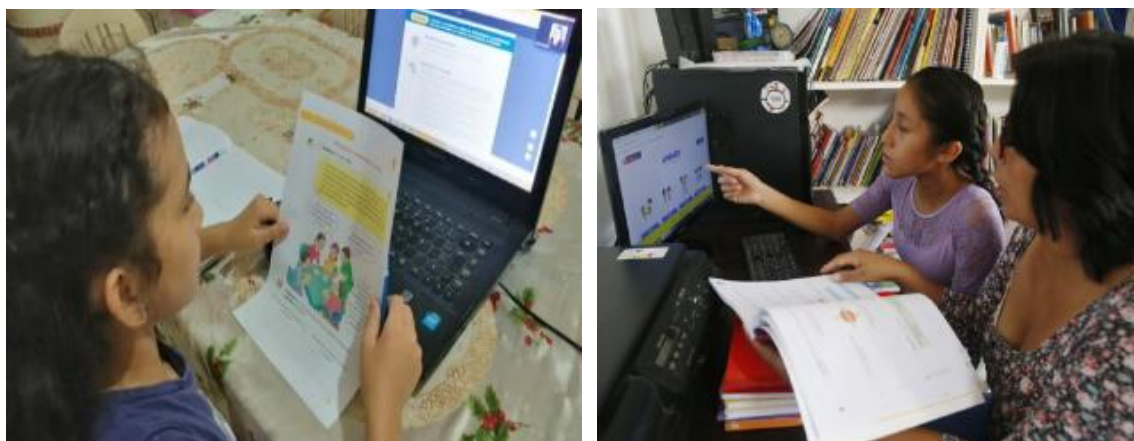


Figura N°24: Educación a distancia en el Perú por la pandemia del Covid-19.

Fuente: Richter, A., y Smarrelli, G. (2020). *Hacia un “nuevo normal” en la educación de niños y adolescentes: riesgos, retos y oportunidades*. Recuperado de: <http://hacerperu.pe/hacia-un-nuevo-normal-en-la-educacion-de-ninos-y-adolescentes-riesgos-retos-y-oportunidades/>

2.2.7 Estrategias para mejorar la comprensión lectora

En Roa (2020) se proponen una serie de técnicas que pueden ayudar a los lectores a entender mejor lo que leen. Algunas estrategias planteadas son las siguientes:

- Empezar a leer despacio si se siente que la lectura es difícil.
- Leer más rápido si se está buscando un dato importante.
- Detenerse en cada fragmento para pensar en lo leído.
- Leer una línea dos o más veces si es necesario memorizar la información.
- Leer varias veces un párrafo para entender las palabras difíciles.
- Si la lectura es nueva, realizar una lectura rápida para saber de qué puede tratar.
- Detenerse cada 4 o 5 líneas para reflexionar si se está entendiendo lo que se va leyendo.
- Al terminar el texto, realizarse preguntas para saber si se ha entendido lo que se ha leído.
- Leer prestando atención y sin pensar en cosas secundarias.
- Leer por lo menos 15 minutos al día con el objetivo de obtener un hábito de lectura.
- Si el cansancio aparece, descansar un momento y retomar la lectura luego.
- Si se encuentra una palabra que no se conoce, esta se debe de buscar en el diccionario.
- Poner en práctica las palabras nuevas que van aprendiendo ya que así se quedarán mejor grabadas en la memoria.

2.2.8 Realidad Virtual

En este apartado se explicará acerca de la tecnología emergente llamada Realidad Virtual. Se ahondará en sus características, los tipos de realidad virtual que existen, la arquitectura que posee todo sistema de realidad virtual y los principales dispositivos utilizados para sumergirse en ambientes virtuales.

2.2.8.1 Definición y características

Según Rivera (2018), la realidad virtual es una tecnología que crea otro mundo pero es virtual y a través de un ordenador u otro dispositivo se puede acceder a él, estando conscientes de que esa realidad no existe. La realidad virtual “supone la inmersión en la simulación digital de un mundo en el que el usuario puede manipular los objetos e interactuar con el ambiente” (Trends E. , 2017, p. 6). En esta última definición, se utiliza la palabra “inmersión” la cual va muy ligada a lo que es realidad virtual. La inmersión es “la capacidad de la percepción humana que permite a las personas sentirse dentro de un dispositivo de realidad virtual como si se estuviera físicamente en otro lugar” (Barambones, 2016, p.69).

Otegui (2017) sostiene que la realidad virtual posee 3 características principales. Estos son los siguientes:

1. **Inmersión:** Los usuarios se desprenden del ambiente real que los rodea y se sumergen por completo en un ambiente virtual. El grado de inmersión dependerá del tipo de periféricos que se utilice para tener contacto con el mundo virtual (Otegui, 2017).
2. **Interacción:** Al desprenderse del mundo real, el usuario empieza a interactuar con este nuevo mundo virtual que lo rodea y empieza a recibir estímulos por parte de este (Otegui, 2017). El nivel de interacción con el ambiente virtual dependerá de que tan sumergido se encuentre el usuario en el mundo virtual.
3. **Imaginación:** El usuario siente la nueva realidad que lo rodea, dependerá del propio usuario que tan consciente se encuentre de que el mundo que lo rodea no existe para poder sentir el ambiente virtual de manera completa o parcial (Otegui, 2017).

2.2.8.2 Tipos de realidad virtual

La realidad virtual suele clasificarse en función al grado de inmersión. Rivera (2018) afirma que existen 3 tipos de realidad virtual, los cuales se explicarán a continuación:

1. **Realidad virtual No inmersiva:** Este tipo de realidad virtual se proyecta a través de una pantalla, controlada a través de un teclado, mouse u otros componentes que permiten tener contacto con otros objetos en tiempo real. Este tipo de realidad es la más común y se puede observar en videojuegos en computadoras o Smartphone sin la necesidad de utilizar algún equipo especial.
2. **Realidad virtual Semi Inmersiva:** Este tipo de realidad virtual se caracteriza por usar 3 pantallas en la pared y en ocasiones se utiliza gafas especiales para realidad virtual. Esto último se hace con el objeto de tener una visión más cercana al mundo virtual. No obstante, el nivel de inmersión no es elevado sino intermedio.
3. **Realidad virtual inmersiva:** Este tipo de realidad tiene una inmersión total en el mundo virtual y permite tener una respuesta sensorial. Normalmente, se utilizan diversos dispositivos como gafas, trajes, cascos, entre otros, para poder llegar a este nivel elevado de inmersión.

2.2.9 Base de Datos

En esta sección, se explicará la definición del concepto de base de datos, así como su clasificación (relacionales y no relacionales) y sus principales características que las

diferencian. Además, se describirá una base de datos por cada tipo para poder observar el funcionamiento particular de cada una.

2.2.9.1 Definición

Una base de datos es una agrupación de datos que poseen una estructura definida y representan a entidades y las relaciones que existen entre estas (Camps et al., 2005). En términos más simples, una base de datos es un repositorio en donde se pueden almacenar grandes cantidades de información de manera ordenada y que posteriormente puede ser consultada a través de uno o muchos usuarios (Pérez D. , 2007). Estas bases de datos poseen columnas ,en donde se almacena un elemento de información (dni, nombres, apellidos,etc) , y de filas que representa cada registro de información (Pérez D. , 2007). Las principales características de las bases de datos son las siguientes:

- Mínima redundancia entre los datos.
- Realización de consultas complejas a la fuente de datos.
- Brinda seguridad a las tablas creadas.
- Brinda la oportunidad de creación de archivos de respaldo y recuperación.

Para realizar el diseño de una base de datos, se suelen diseñar un diagrama o modelo de entidad-relación tal cual como se presenta en la Figura N°25. Este diagrama permite presentar las entidades que serían pertinentes dentro de un sistema de información así como la relación que pueden existir entre estas entidades de tal modo que se pueda tener un mejor entendimiento de los datos así como del flujo de la información (Pérez D. , 2007). Las relaciones entre las entidades pueden darse principalmente de tres formas diferentes:

- **Relación de uno a uno:** Una instancia de la entidad A se relaciona con una instancia de entidad B.
- **Relación de uno a muchos:** Cada instancia de A, se relaciona con muchas instancias de la entidad B.
- **Relación de muchos a muchos:** Cualquier instancia dentro de la entidad A, se relaciona con cualquier instancia presente en la entidad B.

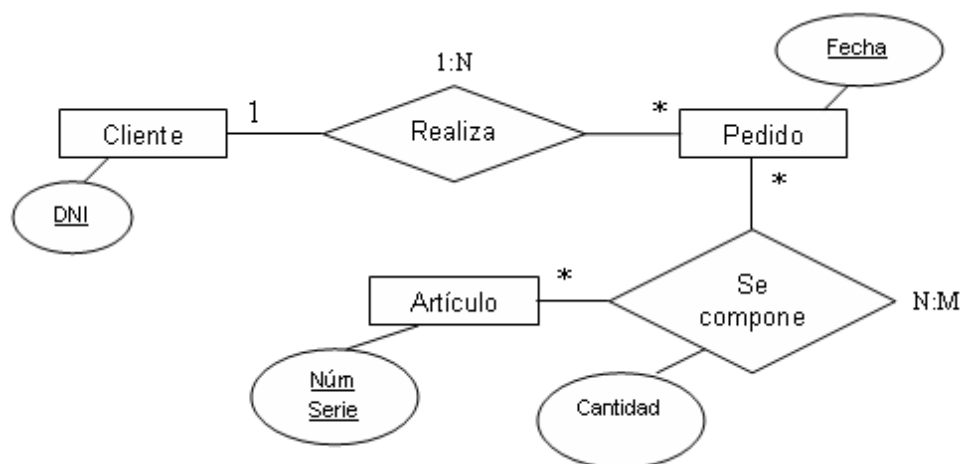


Figura N°25: Modelo entidad-relación.

Fuente: Pérez, D. (2007). ¿Qué son las bases de datos? Recuperado de: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

2.2.9.2 Bases de datos No relacionales

Una limitación con respecto a las bases de datos relacionales es la rigidez que conserva la relación entre las tablas. En respuesta a este problema, surgieron las bases de datos no relacionales o también conocidas como NoSQL que se traduce como “*Not Only SQL*”. Estas bases de datos no relacionales no siguen una estructura relacional haciendo mucho más fácil el almacenamiento de información (García, 2020). La popularidad de estas bases de datos radica en que son fáciles de desarrollar, implementar y llevarlo a gran escala (AWS, 2020). Este tipo de bases de datos se utilizan para la creación de modelos mucho más flexibles y que permitan adaptarse a aplicaciones más modernas. Las principales razones por las cuales usar una base de datos no relacional son las siguientes:

- **Flexibilidad:** Estas bases de datos no relacionales permite la creación de modelos más fáciles de desarrollar y perfectos para el almacenamiento de datos semiestructurados y no estructurados.
- **Alto rendimiento:** Las bases de datos no relacionales están optimizadas para el almacenamiento de datos específicos y con patrones de acceso que permiten tener un mejor rendimiento a comparación de las bases de datos relacionales.
- **Alta funcionalidad:** Estas bases de datos cuentan con API funcionales que están diseñadas para trabajar de manera adecuada con sus respectivos modelos.

Los tipos de bases de datos no relacionales más comunes son los siguientes:

- **Clave-valor:** Estos tipos de base de datos son de los más sencillos de crear. Consiste en una relación de una cada elemento posee una llave y un valor. Para acceder al valor de ese elemento, se hace un llamado a la clave de este. Los usos más comunes para este tipo de base de datos son para juegos e IoT dado que permite el almacenamiento de grandes cantidades de información (AWS, 2020).
- **Documentos:** Almacena los datos en documentos muy parecidos objetos tipo JSON. En este tipo de base de datos NoSQL se pueden almacenar datos con una arquitectura más compleja en donde un documento equivaldría un registro y una colección equivaldría a una tabla (García, 2020). Este tipo de base de datos funciona de manera adecuada con catálogos y en general con sistemas que se necesitan hacer una clasificación de contenido.
- **Orientada a columnas:** Este tipo de base de datos es muy parecido a una tabla relacional. El surgimiento de este tipo de base de datos NoSQL radica en que hay que hacer una recuperación de datos de manera conjunta tal cual como se haría en una base de datos relacional, no obstante, en esta base de datos orientada a columnas, es que estas se agrupan en familias para poder hacer la consulta respectiva (García, 2020).

2.2.9.2.1 *Firestore Database*

Firestore es una plataforma en la nube para el desarrollo de aplicaciones web y móviles. Fue adquirida por Google en el año 2014 en donde funcionaba principalmente como una base de datos en tiempo real, pero fue adquiriendo nuevas funcionalidades con el pasar de los años (López, 2020). Firestore posee varias funciones que permiten a los desarrolladores crear aplicaciones de una manera más ágil. Algunas de sus principales servicios que ofrece son los siguientes:

- **Realtime Database:** Probablemente, una de las características más importantes de Firestore sea su base de datos en tiempo real. Esta base de datos es NoSQL y los archivos se almacenan en formato JSON.
- **Almacenamiento en la nube:** En este sistema de almacenamiento, los usuarios pueden almacenar fotos, videojuegos, imágenes, entre otros. Firestore crea un fichero por cada usuario de tal modo que la información que ordenada y clasificada de forma correcta evitando la creación de este servicio por parte del desarrollador (López, 2020).
- **Crash reporting:** Poder gestionar los errores que tiene la aplicación y darle soporte a los mismos es vital para poder brindar una buena experiencia a los usuarios. Por tal motivo, Firestore ofrece esta opción a los desarrolladores de tal modo que los errores de

sus aplicaciones puedan ser atendidos y almacenados en un historial para poder hacer un análisis de estos (López, 2020).

- **Hosting:** Firebase ofrece un servidor para poder almacenar las aplicaciones de los desarrollares y cuenta con certificados de seguridad SSL y HTTP2 por cada dominio ofreciendo seguridad al navegar por la aplicación (López, 2020).

Adicionalmente, las principales ventajas de utilizar Firebase son las siguientes:

- Firebase es recomendable cuando se necesita almacenar y consultar data el tiempo real.
- Permite realizar monetización con las propias aplicaciones obteniendo rentabilidad sin recurrir en un costo adicional.
- Las aplicaciones realizadas pueden ir escalando, adoptando nuevas funcionalidades y conjunto con esto pueden ir variando los planes de pago.
- Firebase se concentra principalmente en el Backend para que los programadores puedan enfocar su atención en otros tipos de requerimientos de sus aplicaciones.

2.2.10 Hosting o Alojamiento Web

Un servicio de Hosting es un lugar en donde se almacenan las páginas web y hacen posible que estas últimas se encuentren siempre disponible para los usuarios todos los días de la semana y las 24 horas del día (Alonso, 2020).

Su función principal es que una determinada página web se encuentre disponible en internet con todos los protocolos adecuados de seguridad (Acibeiro , 2017). Las empresas suelen ubicar sus servidores (computadoras disponibles siempre y que son las que almacenan las páginas web) en lugares estratégicos para que funcionen de manera adecuada (Acibeiro , 2017).

En la Figura N°26, se puede observar cómo es que funciona un servicio de hosting. La primera figura del lado izquierdo es un servidor físico de la empresa. Este servidor puede alojar varios hostings y estos últimos a su vez puede almacenar una o varias páginas web las cuales se pueden visualizar en internet (Alonso, 2020).

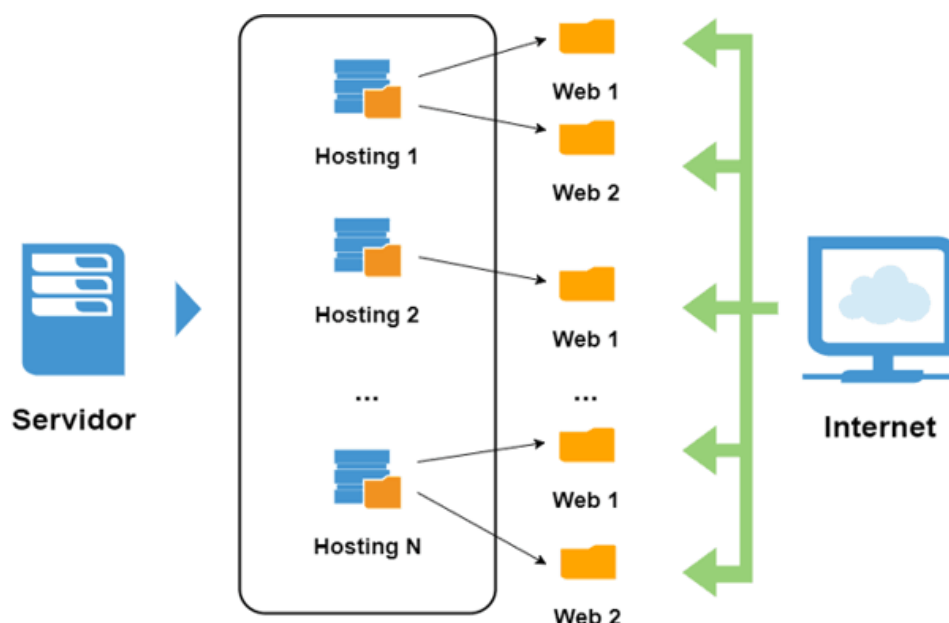


Figura N°26: Servicio de Hosting Web.

Fuente: Alonso, R. (2020). *¿Qué es un hosting o alojamiento web?*

Los tipos de hosting más utilizados se presentan a continuación:

- **Hosting Compartido:** Es uno de los más utilizado, dado que permite compartir el mismo hosting para utilizar varias páginas web. Dependiendo del servicio que se contrate, pueden cobrar o no por la cantidad de tráfico generado por las páginas web (Acibeiro , 2017).
- **Servidor dedicado:** Este es un tipo especial de servidor que se le asigna a un único usuario. Este usuario tiene acceso a todos los recursos que posee el servidor desde el ancho de banda hasta el total de almacenamiento.

2.2.10.1 Node.js

Es un entorno en tiempo real de JavaScript y cuya función principal es la ejecución de páginas o programas web que estén desarrollados en lenguaje de programación ya mencionado (Lucas, 2019). Node.js permite el correcto funcionamiento de las páginas web en un navegador dando la arquitectura Backend necesarios para que el código se ejecute de la manera adecuada.

Node.js a diferencia de otros servicios web, opera en un solo subproceso (Lucas, 2019). De este modo, ocupa mucho menos espacio en la RAM y permitiendo además soportar diversas conexiones al mismo tiempo.

Según Lucas (2019), las principales ventajas de usar Node.js son las siguientes:

- Cuenta incorporado JavaScript lo cual lo hace una herramienta fácil de entender y más por desarrolladores de java.
- Ayuda con la creación y almacenamiento de proyectos.
- Permite el manejo en paralelo de diversas peticiones de clientes.
- Su codificación no es muy compleja a comparación con otros motores de ejecución.
- Dado que un solo subproceso cuenta con más variedad de características implementadas de manera conjunta.

2.2.11 Deep Learning

Deep Learning es un tipo particular de Machine Learning que posibilita la creación de modelos con un gran poder y flexibilidad para aprender diversas actividades realizadas por los humanos. En Deep Learning, las redes neuronales, los cuales son algoritmos inspirados en el cerebro humano, aprenden a partir de una gran cantidad de datos (Bernard, 2018). Del mismo modo que los seres humanos aprenden, los algoritmos de Deep Learning realizan una tarea de manera repetitiva, ajustando sus valores para mejorar el resultado obtenido de la tarea que debe realizar (Bernard, 2018). La gran cantidad de data que existe actualmente en las redes ha posibilitado el crecimiento del Deep Learning, dado que es requisito fundamental contar con una gran cantidad de data para la creación de modelos (Bernard, 2018). Los modelos de Deep Learning pueden aprender incluso de data no estructurada, variada y que guarden relación o no entre sí. Entre más aprendan los algoritmos, mejor funcionaran dentro del entorno real (Bernard, 2018). En el siguiente apartado, se explicarán todos los aspectos teóricos y matemáticos detrás de las redes neuronales, los cuales son principal estructura para la creación de modelos de Deep Learning.

2.2.11.1 Redes Neuronales

Las redes neuronales se basan en los procesos que los seres humanos realizan reconocimiento de imágenes, detección de objetos, entre otros. El cerebro posee unas estructuras conocidas como neuronas que realizan cálculos para realizar las actividades que los humanos hacen de manera diaria (Haykin, 2009).

Una red neuronal es una máquina que ha sido creada con el objetivo de modelar los procesos internos que realiza el cerebro para realizar una determinada tarea (Haykin, 2009). El conocimiento es adquirido por la red neuronal a través de un proceso de aprendizaje y las interconexiones que existen entre las neuronas son representadas por pesos sinápticos que

almacenan en datos los conocimientos adquiridos ajustándose a sí misma mediante un algoritmo de aprendizaje hasta conseguir realizar la actividad deseada (Haykin, 2009).

2.2.11.1.1 El Perceptrón

El Perceptrón es una estructura que fue inventada por Frank Rosenblatt para representar una neurona artificial con dos salidas (Lillo, 2019) y es la pieza fundamental para el correcto funcionamiento de lo que es una red neuronal. En la Figura N°27, se puede apreciar la representación de un perceptrón.

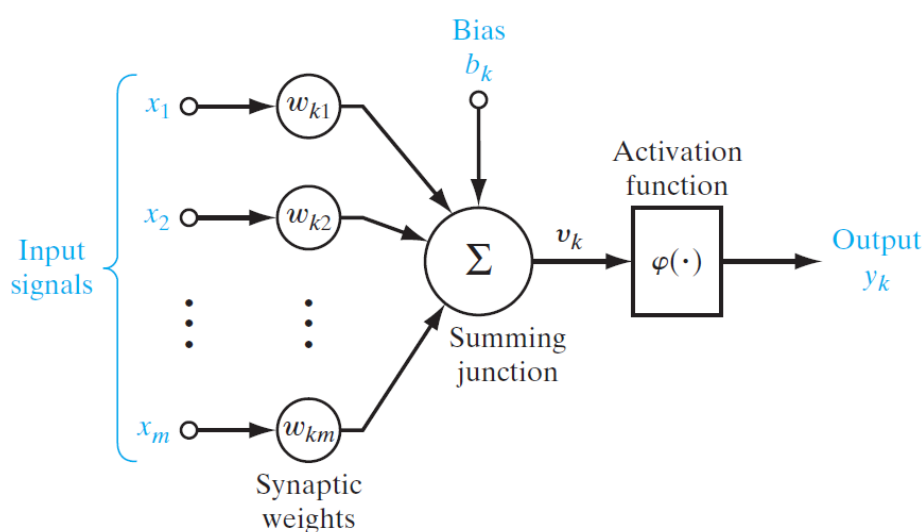


Figura N°27: Modelo de un perceptrón.

Fuente: Haykin, S. (2009). *Neural Networks and Learning Machines*. (p.40)

Como se puede apreciar en la Figura N°27, el perceptrón posee un *Output* o salida denotado por y_k que normalmente toma valores que se encuentran en un intervalo de $[0,1]$ o $[-1,1]$. Adicionalmente, en la Figura N°27, se observa un elemento llamado *Bias* o sesgo denotado por un b_k cuyo objetivo es el de aumentar o disminuir la entrada neta de la función de activación dependiendo si es positivo o negativo, respectivamente.

Generalmente, las salidas de un perceptrón suelen tomar valores de +1, en caso de que el output sea positivo o de -1 o 0 en caso de que sea negativo (Haykin, 2009).

El objetivo del perceptrón es clasificar correctamente un conjunto de estímulos o entradas x_n de tal modo que se puedan transformar en dos clases \mathcal{C}_1 y \mathcal{C}_2 . La regla de decisión para la clasificación es asignar el punto representado por las entradas x_1, x_2, \dots, x_m a la clase \mathcal{C}_1 si la salida del perceptrón es +1 y a la clase \mathcal{C}_2 si es -1.

En términos matemáticos, el modelo del perceptrón puede ser presentado por la ecuación:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

Y por la ecuación:

$$y_k = \varphi(u_k + b_k) \quad (2)$$

En donde:

- x_1, x_2, \dots, x_m son los *inputs* o señales de entrada.
- $w_{k1}, w_{k2}, \dots, w_{km}$ son los pesos de la neurona k .
- u_k es la salida de la sumatoria lineal de las entradas multiplicadas por sus pesos.
- b_k es el sesgo o *bias*.
- $\varphi(\cdot)$ es la función de activación.
- y_k es la señal de salida de la neurona.

El uso del bias b_k tiene un efecto de aumentar o decrementar el valor de la salida u_k tal como se mostró en la Figura N°27, de tal modo que:

$$v_k = \varphi(u_k + b_k) \quad (3)$$

Es preciso mencionar que el bias es un valor externo de la neurona k . Dependiendo del valor de este, el valor de v_k puede incrementar su valor o decrementar dependiendo además de la función de activación y el valor de la salida de la sumatoria de los valores de las entradas multiplicada por sus pesos.

De la misma Figura N°27, se puede apreciar que:

$$v_k = u_k + b_k \quad (4)$$

En donde:

- v_k es el resultado de la suma entre la sumatoria de las entradas multiplicadas por sus respectivos pesos más el valor del bias.

Haciendo una combinación entre la Ecuación (1) y la Ecuación (4) se tiene como resultado la siguiente ecuación:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (5)$$

Además:

$$y_k = \varphi(v_k) \quad (6)$$

En la Ecuación (5), se puede adherir una nueva entrada x_0 de tal modo que:

$$x_0 = +1 \quad (7)$$

Y su peso respectivo que sería equivalente a:

$$w_{k0} = b_k \quad (8)$$

Con las anteriores modificaciones a las ecuaciones, se puede plantear un nuevo modelo para una neurona k tal cual como se muestra en la Figura N°28.

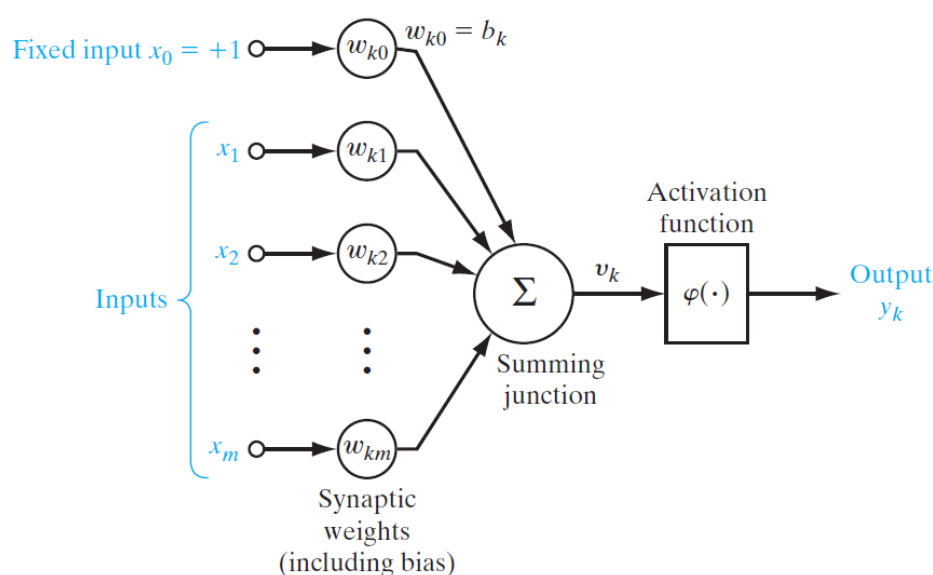


Figura N°28: Modelo del perceptrón incluyendo el bias.

Fuente: Haykin, S. (2009). *Neural Networks and Learning Machines*. (p.43)

De la Figura N°28, se puede reformular el modelo del perceptrón adhiriendo una nueva señal de entrada x_0 equivalente a 1 y un nuevo peso w_{k0} que equivale a b_k . De tal manera que a modo de resumen se podría definir a una red neuronal como un conjunto de señales de entrada multiplicadas por su respectivo peso adhiriendo a estas entradas una señal inicial x_0 con su respectivo peso de tal modo que v_k equivale a la sumatoria de las entradas multiplicadas por su respectivo peso. Finalmente, a este resultado, se le aplica la función de activación que da como resultado final la salida y_k .

2.2.11.1.2 Función de activación

De igual manera que las neuronas biológicas tienden a estar activas o inactivas, las neuronas artificiales tienen el mismo comportamiento (Ruiz, Basualdo, y Matich, 2001). Normalmente, tanto las neuronas biológicas como las artificiales tienden a tomar 2 valores, no obstante, las

neuronas artificiales pueden tomar cualquier valor numérico que se encuentre de un rango definido (Ruiz, Basualdo, y Matich, 2001).

La función de activación, denotada por $\varphi(v)$, calcula el estado final de la neurona k . Normalmente, el estado de la neurona toma valores entre el rango de $[0,1]$ o $[-1,1]$. Las funciones de activación más utilizadas son las siguientes:

1) *Función Sigmoide*: Este tipo de función de activación es una de las más comunes, la hace distinguible su peculiar y a la vez elegante forma de “S” (Haykin, 2009). Su ecuación se representa de la siguiente manera:

$$\varphi(v) = \frac{1}{1+\exp(-av)} \quad (9)$$

Además, su curva tiene un comportamiento peculiar dado que se mantiene lineal en los extremos, pero no lineal en el medio de la función (Haykin, 2009). La gráfica de la función sigmoide se presenta en la Figura N°29.

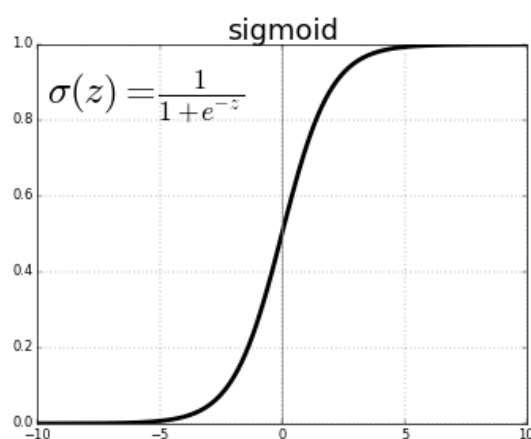


Figura N°29: Función Sigmoide.

Fuente: Alvarado, I. (2015). *Redes Neuronales*.

2) *Función RELU*: Este tipo de funciones de activación permiten el paso de valores numéricos positivos y todos los valores negativos que pueda tomar los iguala a cero. Su ecuación se representa de la siguiente manera:

$$f(x) = \max(0, x) = \{0 \text{ para } x < 0 ; x \text{ para } x \geq 0\} \quad (10)$$

La gráfica tiene un comportamiento lineal creciente para números positivos y lineal horizontalmente para los números negativos siendo constantes igual a cero de ser el resultado cualquier número negativo (Haykin, 2009). La gráfica de la función *RELU*, se presenta en la Figura N°30.

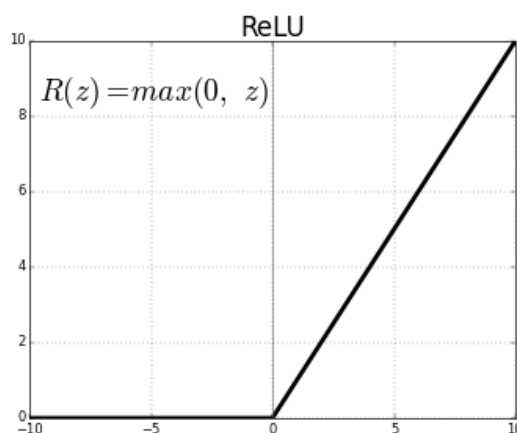


Figura N°30: Función RELU.

Fuente: Alvarado, I. (2015). *Redes Neuronales*.

2.2.11.1.3 Multilayer Perceptron (MLP)

Anteriormente, se ha mencionado sobre el perceptrón y que es la base para la construcción de redes neuronales profundas (Haykin, 2009). El Multilayer Perceptron es el conjunto de perceptrones agrupados de manera vertical en grupos conocido como capas. El MLP contiene 1 capa de entrada en donde ingresan los datos a la red. Seguidamente puede poseer una 1 o más capas que se encuentran entre la capa de entrada y la capa de salida conocidas como capas ocultas en donde se realizan los cálculos correspondientes y finalmente, una capa de salida en donde se obtienen los resultados por cada neurona que ,como se mencionó anteriormente, puede tomar valores de $[0,1]$ o $[-1,1]$. Las principales características del Multilayer Perceptron son los siguientes:

- El modelo de cada neurona posee una función de activación no lineal.
- La red contiene una o más capas que se encuentran ocultas tanto para la entrada como para nodos de salida.
- La red presenta un alto de conexión, cuyo alcance se determina por los pesos que presenta la red.

Es preciso mencionar que aquellas características traen consigo consecuencias que es preciso mencionar. En primer lugar, el hecho de que existan capas ocultas dentro de la red hace que sea más difícil tener un entendimiento teórico más profundo sobre esta (Haykin, 2009). Y,

en segundo lugar, el uso de múltiples capas oculta es mucho más difícil dar seguimiento al entrenamiento de la red (Haykin, 2009).

El método utilizado para el entrenamiento del Multilayer perceptrón es el algoritmo de *back-propagation*. Dicho entrenamiento incluye dos fases:

- 1) La fase *forward*, en donde los pesos de la red son fijos y la entradas se propagan a través de la red multiplicándose por sus pesos correspondientes capa por capa hasta llegar a la salida (Haykin, 2009). En esta fase, los únicos cambios que se dan son con respecto a las salidas de la red que pueden ser $[0,1]$ o $[-1,1]$.
- 2) La fase en *backward*, en donde se calcula la diferencia que existe entre la salida de la red y la salida esperada. El resultado de esa diferencia se propaga a través de la red hacia atrás capa por capa. En esta segunda fase, se realizan muchos más cambios dado que surgen sucesivos ajustes a los pesos que contiene la red mientras el error va propagándose hacia atrás (Haykin, 2009).

2.2.11.1.4 Algoritmo de *Back-Propagation*

El aprendizaje del Multilayer Perceptrón ha ido mejorando a lo largo del tiempo y parte de esas mejoras se debe al algoritmo de *Back-Propagation*. Para explicar el algoritmo, se tiene en la Figura N°31, a una neurona j alimentada por un conjunto de señales producidas por una capa de neuronas a su izquierda.

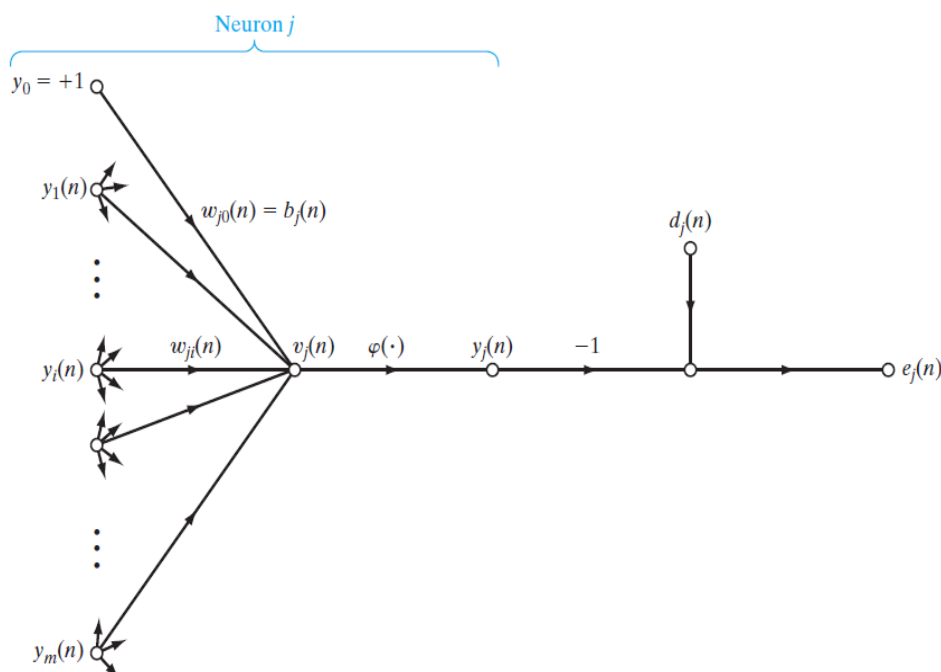


Figura N°31: Gráfico del flujo de salida para una neurona j .

Fuente: Haykin, S. (2009). *Neural Networks and Learning Machines*. (p.129)

El resultado debido a las señales de entrada es la variable $v_j(n)$ y su valor final se calcula mediante la siguiente ecuación:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n)y_i(n) \quad (11)$$

En donde:

- m es el número total de entradas sin incluir al sesgo aplicadas a la neurona j .
- El peso w_{j0} correspondiente a la entrada $y_0 = +1$ es igual al sesgo b_j aplicado a la neurona j .

Por lo tanto, $y_j(n)$ que aparece en la salida de la neurona j en la iteración n es:

$$y_j(n) = \varphi(v_j(n)) \quad (12)$$

El algoritmo de *Back-Propagation* aplica una corrección $\Delta w_{ji}(n)$ al peso sináptico $w_{ji}(n)$ que es proporcional a la derivada parcial $\partial \mathcal{E}(n) / \partial w_{ji}(n)$. Según la regla de la cadena, se puede expresar esta gradiente de la siguiente manera:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \quad (13)$$

La derivada parcial $\partial \mathcal{E}(n) / \partial w_{ji}(n)$ representa un factor de sensibilidad que determina la dirección de búsqueda en el espacio para el peso sináptico w_{ji} . De este modo, se tiene:

$$\frac{\partial \mathcal{E}(n)}{\partial e_j(n)} = e_j(n) \quad (14)$$

Además:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (15)$$

Ahora, derivando la Ecuación (12) con respecto a $v_j(n)$, se obtiene:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'_j(v_j(n)) \quad (16)$$

Posteriormente, derivando la Ecuación (11) con respecto a $w_{ji}(n)$ se tiene la siguiente ecuación:

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_j(n) \quad (17)$$

De igual manera, el uso de la Ecuación (14) hasta la Ecuación (17) en la Ecuación (13) queda de la siguiente manera:

$$\frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} = -e_j(n) \varphi'_j(v_j(n)) y_i(n) \quad (18)$$

La corrección aplicada $\Delta w_{ji}(n)$ aplicada a $w_{ji}(n)$ es definido por la regla delta, o:

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial w_{ji}(n)} \quad (19)$$

En donde:

- η es la tasa de aprendizaje del algoritmo de *Back-Propagation*.

El uso del signo menos de la Ecuación (19) tiene en cuenta el descenso de la gradiente en el espacio del peso (es decir, buscando una dirección para el cambio de peso que reduce el valor de $\mathcal{E}(n)$). En consecuencia, el uso de la Ecuación (18) en la Ecuación (19) queda representada de la siguiente manera:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad (20)$$

Donde la gradiente local $\delta_j(n)$ es definida por:

$$\begin{aligned} \delta_j(n) &= \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} \\ &= \frac{\partial \mathcal{E}(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= e_j(n) \varphi'_j(v_j(n)) \end{aligned} \quad (21)$$

De acuerdo con la Ecuación (21), la gradiente local $\delta_j(n)$ para una neurona j es igual al producto de la correspondiente señal de error $e_j(n)$ para esa neurona y la derivada $\varphi'_j(v_j(n))$ de la función de activación asociada.

De la Ecuación (20) a la Ecuación (21) se puede observar un factor el cual es importante para el cálculo del ajuste del peso $\Delta w_{ji}(n)$ y es la señal del error $e_j(n)$. En este escenario, se pueden identificar 2 distintos casos en función de la ubicación de la neurona j . En el primer caso, la neurona puede ser el *output* de la red. En este caso, se calcula el error mediante la diferencia entre el valor de la neurona y la respuesta deseada de tal modo que es más sencillo obtener el error. El otro caso que pueda suceder es que la neurona sea una neurona que se

encuentra dentro de una capa oculta. En estos casos particulares, el error se comparte por las neuronas que se encuentran en la misma capa oculta.

A continuación, se explican de una manera más detallada el proceso para calcular el error en función a la ubicación de una neurona:

1) **Caso 1 - Neurona como nodo de salida de la red:** Para calcular, el error en este primer caso, se puede usar la siguiente ecuación:

$$e_j(n) = d_j(n) - y_j(n) \quad (22)$$

Donde:

- $y_j(n)$ es la señal producida en la salida de la neurona j por un estímulo $X(n)$ aplicado en la capa de entrada.
- $d_j(n)$ es el i -ésimo del vector de respuesta deseada $d(n)$.

Habiendo determinado el valor de $e_j(n)$, se podría ya encontrar el valor de la gradiente local $\delta_j(n)$ utilizando la Ecuación (21).

- 2) **Caso 2 - Neurona como nodo oculto de la red:** Es este caso, no existe un valor deseado para la neurona y por lo tanto el valor del error se tiene que hallar de forma recursiva trabajando hacia atrás en términos de las señales de error de todas las neuronas a las cuales se encuentra esta neurona conectada (Haykin, 2009). Para dar una mejor explicación, en la Figura N°32, se puede observar a una neurona k conectada a una neurona j .

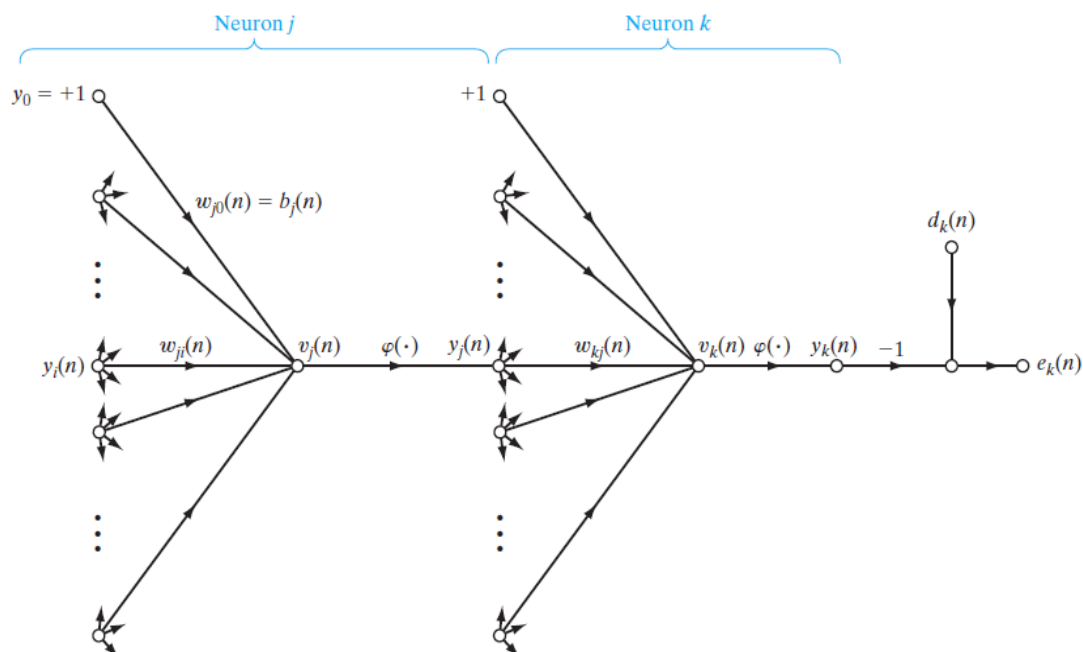


Figura N°32: Gráfico del flujo de salida para una neurona k conectada a una neurona j.
Fuente: Haykin, S. (2009). *Neural Networks and Learning Machines*. (p.132)

De acuerdo con la Figura N°32 y siguiendo la Ecuación (21), se puede redefinir la gradiente local $\delta_j(n)$ para la neurona oculta j de la siguiente manera:

$$\begin{aligned}\delta_j(n) &= \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \\ &= -\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} \varphi'_j(v_j(n))\end{aligned}\quad (23)$$

En donde en la segunda línea se reutilizó la Ecuación (16). Para calcular la derivada parcial $\partial \mathcal{E}(n) / \partial y_j(n)$, se debe de proceder de la siguiente manera:

$$\mathcal{E}(n) = \frac{1}{2} \sum_k c_k e_k^2(n)\quad (24)$$

Además, para evitar confusiones, se cambió la variable j por k dado que la neurona j se refiere a una que se encuentra dentro de una capa oculta. Derivando la Ecuación (24) con respecto a la señal $y_j(n)$, se obtiene que:

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = \sum_k e_k \frac{\partial e_k(n)}{\partial y_j(n)}\quad (25)$$

Seguidamente, se calcula la derivada parcial $\partial e_k(n) / \partial y_j(n)$ y si se reescribe la Ecuación (25) se obtiene lo siguiente:

$$\frac{\partial \mathcal{E}(n)}{\partial y_j(n)} = \sum_k e_k(n) \frac{\partial e_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial y_j(n)} \quad (26)$$

Además, de la Figura N°32, se puede observar que:

$$\begin{aligned} e_k(n) &= d_k(n) - y_k(n) \\ &= d_k(n) - \varphi_k(v_k(n)) \end{aligned} \quad (27)$$

Por lo tanto:

$$\frac{\partial e_k(n)}{\partial v_k(n)} = -\varphi'_k(v_k(n)) \quad (28)$$

También, de la Figura N°32, se puede observar que:

$$v_k(n) = \sum_{j=0}^m w_{kj}(n) y_j(n) \quad (29)$$

Donde:

- m es el número total de entradas (excluyendo el sesgo) aplicadas a la neurona k .

Nuevamente en este caso, el peso $w_{k0}(n)$ es igual al sesgo $b_k(n)$ aplicado a la neurona k y la entrada correspondiente se mantiene en 1. Derivando la Ecuación (29) con respecto a $y_j(n)$ se obtiene:

$$\frac{\partial v_k(n)}{\partial y_j(n)} = w_{kj}(n) \quad (30)$$

Usando la Ecuación (28) y Ecuación (30), se obtiene la derivada parcial deseada:

$$\begin{aligned} \frac{\partial \mathcal{E}(n)}{\partial y_j(n)} &= - \sum_k e_k(n) \varphi'_k(v_k(n)) w_{kj}(n) \\ &= - \sum_k \delta_k(n) w_{kj}(n) \end{aligned} \quad (31)$$

Es preciso mencionar que en la Ecuación (31) se ha utilizado la Ecuación (21) con el índice k sustituido por la j .

Finalmente, usando la Ecuación (31) en la Ecuación (23), se obtiene la formula final de gradiente para una neurona j que se encuentra en la capa oculta:

$$\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (32)$$

Ahora, resumiendo todas las relaciones vistas anteriormente, se tiene en primer lugar, la corrección $\Delta w_{ji(n)}$ aplicado al peso que conecta la neurona i a la neurona j está definida por la siguiente regla:

$$\Delta w_{ji(n)} = \eta \times \delta_j(n) \times y_i(n) \quad (33)$$

Donde:

- $\Delta w_{ji(n)}$ es la fórmula para la corrección del peso.
- η es la tasa de aprendizaje.
- $\delta_j(n)$ es la gradiente.
- $y_i(n)$ es la señal de entrada para la neurona j .

En segundo lugar, la gradiente $\delta_j(n)$ depende si la neurona es una neurona de salida o pertenece a la capa oculta:

- Si la neurona j es un nodo de salida, $\delta_j(n)$ es igual al producto de la derivada $\varphi'_j(v_j(n))$ y el error $e_j(n)$, ambos asociados a la neurona j .
- Si la neurona oculta es un nodo oculto, $\delta_j(n)$ es igual al producto de la derivada de $\varphi'_j(v_j(n))$ y la suma ponderada de δ_s para las neuronas en la siguiente capa oculta o de salida que esté conectada a la neurona j .

En el algoritmo de *Back-propagation* se distinguen 2 diferentes pases de computación. Teniendo en cuenta que se realizó la inicialización de los pesos de manera aleatoria, se tiene que, en el primer pase conocido como *forward*, los pesos se mantienen inalterados en toda la red y la señal de función de la red se calculan neurona por neurona (Haykin, 2009).

La señal de salida de la neurona se calcula como:

$$y_j(n) = \varphi(v_j(n)) \quad (34)$$

Donde:

- $v_j(n)$ es el valor del campo de la neurona j definido por:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n) \quad (35)$$

Donde:

- m es el número total de entradas (excluyendo el sesgo) aplicadas a la neurona j .
- $w_{ji}(n)$ es el peso que conecta la neurona i con la neurona j .

- n es la señal de entrada de la neurona j .

Si la neurona j está en primera capa oculta de la red entonces $m = m_0$ y el índice i se refiere al i -ésimo input de la red por lo cual:

$$y_j(n) = x_i(n) \quad (36)$$

Donde $x_i(n)$ es el i -ésimo elemento del vector de entrada. Si, por el contrario, la neurona j se encuentra en la capa de salida de la red entonces $m = m_L$ y el índice j se refiere al j -ésimo output de la red, de tal modo que:

$$y_j(n) = o_j(n) \quad (37)$$

Donde $o_j(n)$ es el j -ésimo elemento del vector de salida del perceptrón multicapa y se compara con la respuesta deseada $d_j(n)$ obteniendo de esta manera el error $e_j(n)$ para el j -ésimo neurona de salida.

En el segundo pase, conocido como *backward*, comienza desde la capa de salida pasando las señales del error hacia la izquierda propagándolo por toda la red capa por capa y calculando el gradiente por cada neurona (Haykin, 2009). Este proceso permite a los pesos de la red realizar los cambios necesarios de acuerdo con la Ecuación (33).

Una vez que se tienen calculados el gradiente por cada neurona que se presenta en la capa de salida, se utiliza la Ecuación (32) para realizar los cálculos hasta la penúltima capa y por lo tanto los cambios en todos los pesos de todas las conexiones que lo alimentan (Haykin, 2009). El cálculo recursivo continúa capa por capa propagando los cambios a todos los pesos de la red.

2.2.11.1.5 Ejemplo aplicativo

Dado que existe una serie de pasos y ecuaciones a utilizar, es preciso mencionar un ejemplo didáctico de tal manera que se pueda explicar de una manera más clara el proceso de aprendizaje de una red neuronal en donde intervienen los pases en *Forward* y *Backward*.

Para plantear el ejemplo, se tiene una red neuronal con dos entradas i_1 e i_2 , una capa oculta y 2 neuronas (o_1 y o_2) en la capa de salida. Además, la capa oculta y la capa de salida incluyen sus respectivos bias (b_1 y b_2). La estructura de la red neuronal para el ejemplo se presenta en la Figura N°33.

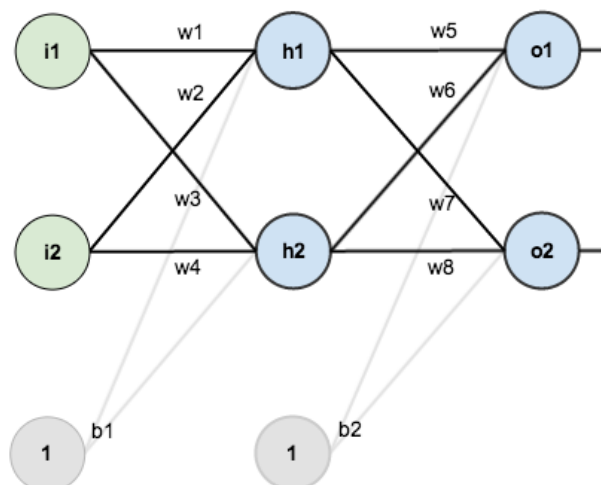


Figura N°33: Estructura de red para ejemplo aplicativo.

Fuente: Mazur, M. (2015). *A Step by Step Backpropagation Example*. Recuperado de: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

Seguidamente, se procede a inicializar con valores tanto a los inputs y outputs, *bias* y los pesos de la red. En el caso de este último, en la mayoría de los casos, los pesos de la red se generan de manera aleatoria. Los valores para cada elemento de la red se presentan en la Figura N°34.

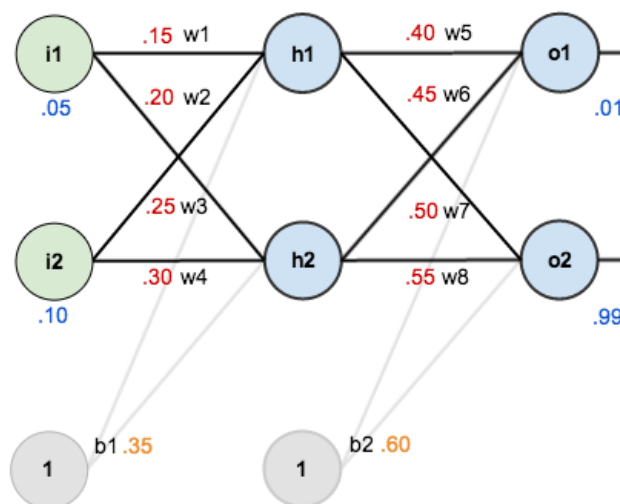


Figura N°34: Valores de los inputs, Outputs y pesos de la red.

Fuente: Mazur, M. (2015). *A Step by Step Backpropagation Example*. Recuperado de: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

El objetivo del algoritmo *Back-Propagation* es modificar y optimizar los pesos para que la red neuronal pueda aprender a tener como resultado en las neuronas de salida los valores reales que se desean (Mazur, 2015).

En este caso, las 2 entradas que corresponden a la red son respectivamente de 0.05 y 0.10 y lo que se desea es que las salidas sean de 0.01 y 0.99. Adicionalmente, la función de activación que se usará es la función logística y el valor del *bias* es de 1 tanto para la capa oculta como la capa de salida. A continuación, se describirán a detalle los cálculos a realizar por cada pase en el entrenamiento de la red neuronal.

- Pase en *Forward*

Dado que ya se tienen los valores de entrada, así como sus respectivos pesos y el valor del *bias*, lo primero que se debe de realizar es el resultado neto (net_{h_1}) para la primera neurona h_1 de la capa oculta. De tal modo, se debe de aplicar la Ecuación (11) en donde se realiza la sumatoria ponderada de cada entrada por su respectivo peso adicionando el *bias* de tal modo que se tendría el siguiente valor:

$$\begin{aligned} net_{h_1} &= w_1 * i_1 + w_2 * i_2 + b_1 * 1 \\ net_{h_1} &= 0.15 * 0.05 + 0.20 * 0.10 + 0.35 * 1 \\ net_{h_1} &= 0.3775 \end{aligned}$$

Obteniendo el valor neto de la neurona, se procede a utilizar la Ecuación (12) en donde se aplica la función de activación al resultado neto para calcular la salida (Out_{h_1}) de la neurona h_1 . La ecuación para una función logística es la siguiente:

$$\phi(x) = \frac{1}{1 + e^{-x}} \quad (38)$$

Aplicada la función de activación logística quedaría el resultado de la siguiente manera:

$$\begin{aligned} Out_{h_1} &= \frac{1}{1 + e^{-0.3775}} \\ Out_{h_1} &= 0.593269992 \end{aligned}$$

Ahora, se debe aplicar el mismo procedimiento en la segunda neurona h_2 de la capa oculta de la siguiente manera:

$$net_{h_2} = w_3 * i_1 + w_4 * i_2 + b_1 * 1$$

$$net_{h2} = 0.25 * 0.05 + 0.30 * 0.10 + 0.35 * 1$$

$$net_{h2} = 0.3925$$

Seguidamente, se vuelve a aplicar la función logística al resultado neto obtenido de tal modo que:

$$Out_{h2} = \frac{1}{1 + e^{-0.3925}}$$

$$Out_{h2} = 0.596884378$$

Habiendo calculado las salidas de las neuronas que se encuentran en la capa oculta, lo siguiente es saber el resultado final de las neuronas en la capa de salida. Para realizar lo anterior mencionado, se aplica el mismo procedimiento empezando por la primera neurona de la capa de salida O_1 . En primer lugar, se halla el resultado neto net_{o1} de la siguiente manera:

$$net_{o1} = w_5 * Out_{h1} + w_6 * Out_{h2} + b_2 * 1$$

$$net_{o1} = 0.40 * 0.593269992 + 0.45 * 0.596884378 + 0.60 * 1$$

$$net_{o1} = 1.105905967$$

Seguidamente, se aplica la función de activación al resultado anterior hallado:

$$Out_{o1} = \frac{1}{1 + e^{-1.105905967}}$$

$$Out_{o1} = 0.75136507$$

Ahora, se realiza el mismo procedimiento para la segunda salida de la red de la siguiente manera:

$$net_{o2} = w_7 * Out_{h1} + w_8 * Out_{h2} + b_2 * 1$$

$$net_{o2} = 0.50 * 0.593269992 + 0.55 * 0.596884378 + 0.60 * 1$$

$$net_{o2} = 1.224921404$$

Aplicando la función de activación al resultado anterior obtenido, se obtiene el siguiente resultado:

$$Out_{o2} = \frac{1}{1 + e^{-1.224921404}}$$

$$Out_{o2} = 0.772928465$$

Obtenido las salidas de la red neuronal, el siguiente paso es calcular el error por cada salida de la red para que finalmente se calcule el error total. La ecuación para determinar el error total de la red se representa de la siguiente manera:

$$E_{TOTAL} = \sum \frac{1}{2} (target - output)^2 \quad (39)$$

Donde:

- E_{TOTAL} es el error total de la red.
- Target es el valor original esperado.
- Output es la salida de la red neuronal.

Como se mencionó al inicio, se desea que las salidas de la primera y segunda neurona sean de 0.01 y 0.99 respectivamente. Aplicando la Ecuación (39) a la primera salida de la red que fue de 0.75136507, se tiene lo siguiente:

$$E_{O1} = \frac{1}{2} (target_{O1} - Out_{O1})^2$$

$$E_{O1} = \frac{1}{2} (0.01 - 0.75136507)^2$$

$$E_{O1} = 0.274811083$$

Aplicando el mismo proceso a la segunda salida de la red, se obtiene lo siguiente:

$$E_{O2} = \frac{1}{2} (target_{O2} - Out_{O2})^2$$

$$E_{O2} = \frac{1}{2} (0.99 - 0.772928465)^2$$

$$E_{O2} = 0.023560026$$

El error total de la red se obtiene sumando el error de la primera salida de la red y de la segunda salida de la siguiente manera:

$$E_{TOTAL} = E_{O1} + E_{O2}$$

$$E_{TOTAL} = 0.274811083 + 0.023560026$$

$$E_{TOTAL} = 0.298371109$$

- Pase en *Backward*

El objetivo principal de este segundo pase radica en que los pesos se actualicen de tal modo que la nueva salida de la red se asemeje mucho más a la original de tal modo que se minimice el error (Mazur, 2015).

Realizando los cálculos respectivos en la capa de salida:

Supóngase que lo que se desea es calcular en cuánto varía el peso con respecto al error total ($\frac{\partial E_{TOTAL}}{\partial w_5}$), de tal modo que se puede entender lo anterior mencionado de la siguiente manera:

$\frac{\partial E_{TOTAL}}{\partial w_5}$ es “la derivada parcial de E_{TOTAL} con respecto a w_5 ”. Pero también se puede leer como “la gradiente con respecto a w_5 ”.

Aplicando la regla de la cadena se tiene lo siguiente:

$$\frac{\partial E_{TOTAL}}{\partial w_5} = \frac{\partial E_{TOTAL}}{\partial out_{o1}} \times \frac{\partial out_{o1}}{\partial net_{o1}} \times \frac{\partial net_{o1}}{\partial w_5}$$

En la Figura N°35, se puede observar la aplicación de la regla en la propia arquitectura de red.

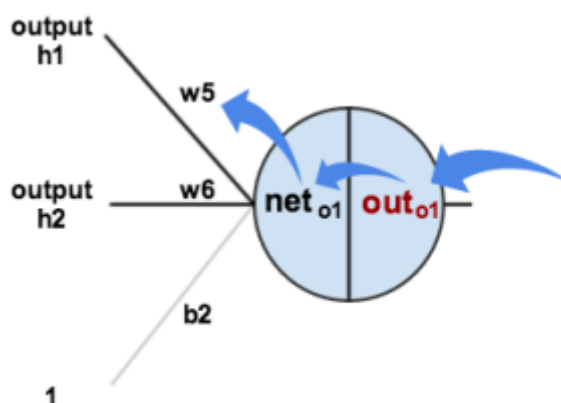


Figura N°35: Representación visual de regla de la cadena en la capa de salida.

Fuente: Mazur, M. (2015). *A Step by Step Backpropagation Example*. Recuperado de: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

Lo que se necesita ahora es calcular cada componente de la regla de la cadena. En primer lugar, lo que se necesita es determinar cuánto cambia el error total con respecto a las salidas. De tal modo que se puede obtener lo siguiente:

$$E_{TOTAL} = \frac{1}{2} (target_{01} - Out_{01})^2 + \frac{1}{2} (target_{02} - Out_{02})^2$$

$$\frac{\partial E_{TOTAL}}{\partial out_{01}} = 2 * \frac{1}{2} (target_{01} - Out_{01})^{2-1} * -1 + 0$$

$$\frac{\partial E_{TOTAL}}{\partial out_{01}} = -(target_{01} - Out_{01})$$

$$\frac{\partial E_{TOTAL}}{\partial out_{01}} = -(0.01 - 0.75136507)$$

$$\frac{\partial E_{TOTAL}}{\partial out_{01}} = 0.74136507$$

En segundo lugar, lo que se necesita es determinar en cuánto varía la salida out_{01} con respecto a net_{01} .

La derivada parcial de la función logística es la salida out_{01} multiplicada por uno menos la misma salida out_{01} . De tal modo que se obtiene lo siguiente:

$$\frac{\partial out_{01}}{\partial net_{01}} = out_{01}(1 - out_{01})$$

$$\frac{\partial out_{01}}{\partial net_{01}} = 0.75136507(1 - 0.75136507)$$

$$\frac{\partial out_{01}}{\partial net_{01}} = 0.186815602$$

Finalmente, en cuánto varía net_{01} con respecto a w_5 . Para esto último se tiene:

$$net_{01} = w_5 * Out_{h1} + w_6 * Out_{h2} + b_2 * 1$$

$$\frac{\partial net_{01}}{\partial w_5} = 1 * out_{h1} * w_5^{(1-1)} + 0 + 0$$

$$\frac{\partial net_{01}}{\partial w_5} = out_{h1}$$

$$\frac{\partial net_{01}}{\partial w_5} = 0.593269992$$

Una vez obtenidos todos los valores, se tiene finalmente lo siguiente:

$$\frac{\partial E_{TOTAL}}{\partial w_5} = \frac{\partial E_{TOTAL}}{\partial out_{01}} \times \frac{\partial out_{01}}{\partial net_{01}} \times \frac{\partial net_{01}}{\partial w_5}$$

$$\frac{\partial E_{TOTAL}}{\partial w_5} = 0.74136507 \times 0.186815602 \times 0.593269992$$

$$\frac{\partial E_{TOTAL}}{\partial w_5} = 0.082167041$$

Para disminuir el error, se resta el valor del peso actual multiplicada por una tasa de aprendizaje η de tal modo que, para este ejemplo, tomará el valor de 0.5. De este modo se tiene:

$$w_5^+ = w_5 - \eta * \frac{\partial E_{TOTAL}}{\partial w_5}$$

Donde:

- w_5^+ es el nuevo peso w_5 ajustado.

Reemplazando los valores se tendría que:

$$w_5^+ = 0.4 - 0.5 * 0.082167041$$

$$w_5^+ = 0.35891648$$

De esta manera es como se realiza el ajuste del peso para cada peso en la capa de salida hacia atrás. Repitiendo el exactamente el mismo proceso para los pesos w_6 , w_7 y w_8 , se tendría como resultados finales los siguientes pesos actualizados:

$$w_6^+ = 0.408666186$$

$$w_7^+ = 0.511301270$$

$$w_8^+ = 0.561370121$$

Realizando los cálculos respectivos en la capa oculta:

Ahora, continuando con la actualización de los pesos, se deben de calcular los nuevos pesos para w_1 , w_2 , w_3 y w_4 . Para realizar dicha tarea se necesitan lo siguiente:

$$\frac{\partial E_{TOTAL}}{\partial w_1} = \frac{\partial E_{TOTAL}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w_1}$$

En la Figura N°36, se puede observar el panorama general de lo que está sucediendo:

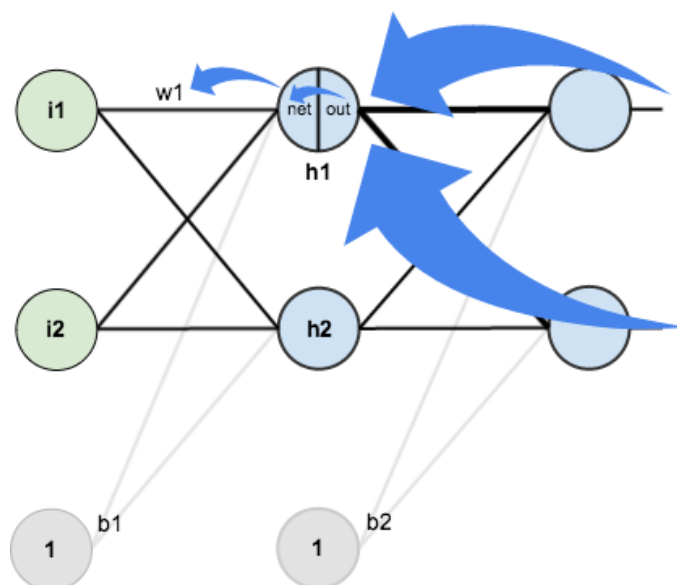


Figura N°36: Representación visual de regla de la cadena en la capa oculta.

Fuente: Mazur, M. (2015). *A Step by Step Backpropagation Example*. Recuperado de: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>

Se sabe que out_{h1} afecta tanto a out_{o1} y a out_{o2} por lo tanto $\frac{\partial E_{TOTAL}}{\partial out_{h1}}$ debe tener su efecto en ambas neuronas de salida. Entonces se tendría lo siguiente:

$$\frac{\partial E_{TOTAL}}{\partial out_{h1}} = \frac{\partial E_{O1}}{\partial out_{h1}} + \frac{\partial E_{O2}}{\partial out_{h1}}$$

Iniciando con el cálculo con $\frac{\partial E_{O1}}{\partial out_{h1}}$:

$$\frac{\partial E_{O1}}{\partial out_{h1}} = \frac{\partial E_{O1}}{\partial net_{O1}} * \frac{\partial net_{O1}}{\partial out_{h1}}$$

Se puede calcular $\frac{\partial E_{O1}}{\partial net_{O1}}$ en función a los cálculos que se han hecho anteriormente:

$$\begin{aligned} \frac{\partial E_{O1}}{\partial net_{O1}} &= \frac{\partial E_{O1}}{\partial out_{O1}} * \frac{\partial out_{O1}}{\partial net_{O1}} \\ &= 0.74136507 * 0.186815602 \\ &= 0.138498562 \end{aligned}$$

Y $\frac{\partial net_{O1}}{\partial out_{h1}}$ es igual a w_5 dado que:

$$net_{o1} = w_5 * Out_{h1} + w_6 * Out_{h2} + b_2 * 1$$

$$\frac{\partial net_{O1}}{\partial out_{h1}} = w_5 = 0.40$$

Y reemplazando los valores se tendría lo siguiente:

$$\begin{aligned}\frac{\partial E_{O1}}{\partial out_{h1}} &= \frac{\partial E_{O1}}{\partial net_{O1}} * \frac{\partial net_{O1}}{\partial out_{h1}} \\ &= 0.138498562 * 0.40 \\ &= 0.055399425\end{aligned}$$

Siguiendo el mismo proceso con $\frac{\partial E_{O2}}{\partial out_{h1}}$ se tiene lo siguiente:

$$\frac{\partial E_{O2}}{\partial out_{h1}} = -0.019049119$$

Por lo tanto:

$$\begin{aligned}\frac{\partial E_{TOTAL}}{\partial out_{h1}} &= \frac{\partial E_{O1}}{\partial out_{h1}} + \frac{\partial E_{O2}}{\partial out_{h1}} \\ &= 0.055399425 + -0.019049119 \\ &= 0.036350306\end{aligned}$$

Ahora que se tiene $\frac{\partial E_{TOTAL}}{\partial out_{h1}}$, se necesita obtener $\frac{\partial out_{h1}}{\partial net_{h1}}$ y luego $\frac{\partial net_{h1}}{\partial w}$ por cada peso:

$$\begin{aligned}out_{h1} &= \frac{1}{1 + e^{-net_{h1}}} \\ \frac{\partial out_{h1}}{\partial net_{h1}} &= \partial out_{h1} (1 - \partial out_{h1}) \\ &= 0.59326999(1 - 0.59326999) \\ &= 0.241300709\end{aligned}$$

Seguidamente, se calcula la derivada parcial de la entrada neta total de h_1 con respecto a w_1 de la misma manera que se hizo con la neurona de salida:

$$\begin{aligned}\partial net_{h1} &= w_1 * i_1 + w_3 * i_2 + b_1 * 1 \\ \frac{\partial \partial net_{h1}}{\partial w_1} &= i_1 = 0.05\end{aligned}$$

Reuniendo todos los valores obtenidos se tendría:

$$\begin{aligned}\frac{\partial E_{TOTAL}}{\partial w_1} &= \frac{\partial E_{TOTAL}}{\partial out_{h1}} \times \frac{\partial out_{h1}}{\partial net_{h1}} \times \frac{\partial net_{h1}}{\partial w_1} \\ \frac{\partial E_{TOTAL}}{\partial w_1} &= 0.036350306 \times 0.241300709 \times 0.05 \\ &= 0.000438568\end{aligned}$$

De esta manera, se puede llegar a reajustar el valor del peso w_1 :

$$\begin{aligned}w_1^+ &= w_1 - \eta * \frac{\partial E_{TOTAL}}{\partial w_1} \\ w_1^+ &= 0.15 - 0.5 * 0.000438568 \\ &= 0.149780716\end{aligned}$$

De este modo es como se realiza el ajuste del peso para cada peso en la capa oculta hacia atrás. Repitiendo el exactamente el mismo proceso para los pesos w_2 , w_3 y w_4 , se tendría como resultados finales los siguientes pesos actualizados:

$$\begin{aligned}w_2^+ &= 0.19956143 \\ w_3^+ &= 0.24975114 \\ w_4^+ &= 0.29950229\end{aligned}$$

Finalmente, se han actualizado todos los pesos. Cuando originalmente se tenían las entradas 0.05 y 0.1, el error en la red era de 0.298371109.

Después de la retropropagación, el nuevo error total se ha reducido a 0.291027924. Este cambio no parece tan significativo, pero si se remite el exactamente el mismo proceso, por ejemplo, unas 100 000 veces, el error decrecería aún más de tal modo que los resultados de la salida de la red se asemejen mucho más a los resultados deseados.

Este ejemplo desarrollado se ha hecho con el objetivo de explicar de una manera más comprensible cómo es que una red neuronal aprende a través de las fases de *Forward* y *Backward*.

2.2.12 Machine Learning

Machine Learning es una rama de la Inteligencia Artificial surgido en los años 80 cuyo principal objetivo es poder detectar patrones a través de un conjunto de datos y en función a eso poder realizar predicciones (Recuero, 2017).

Esta rama de la IA consiste en realizar un modelo que, entrenado a través de un conjunto de datos, permita detectar tendencias o patrones de nuevos conjuntos de datos y a su vez estos resultados obtenidos permitan ayudar a las empresas a ser más eficientes (Recuero, 2017).

La llegada de los modelos de Machine Learning permitieron a los modelos aprender por cuenta propia e ir mejorando sus algoritmos con el paso del tiempo de tal modo que los problemas que la Inteligencia Artificial podía resolver eran ahora problemas más complejos y con un mayor impacto en la organización de tal modo que diversas empresas empezaron a utilizar modelos de Machine Learning para la toma de decisiones dado que contaban con gran cantidad de data de las cual los modelos podían aprender (Potentia Analytics Inc, 2020).

Según Recuero (2017), los principales usos del Machine Learning son los siguientes:

- Detección de anomalías y fraudes informáticos.
- Reconocimiento de imágenes.
- Conducción autónoma.
- Evaluación de exámenes de salud y detección de enfermedades.
- Análisis de mercado.
- Clasificación de contenido y sugerencias de búsqueda.

Actualmente, Machine Learning se divide en tres áreas tal cual como se presenta en la Figura N°37, estas áreas son el aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo (Potentia Analytics Inc, 2020). No obstante, en esta sección se profundizará sobre el aprendizaje por refuerzo que es el más común aplicado a videojuegos.

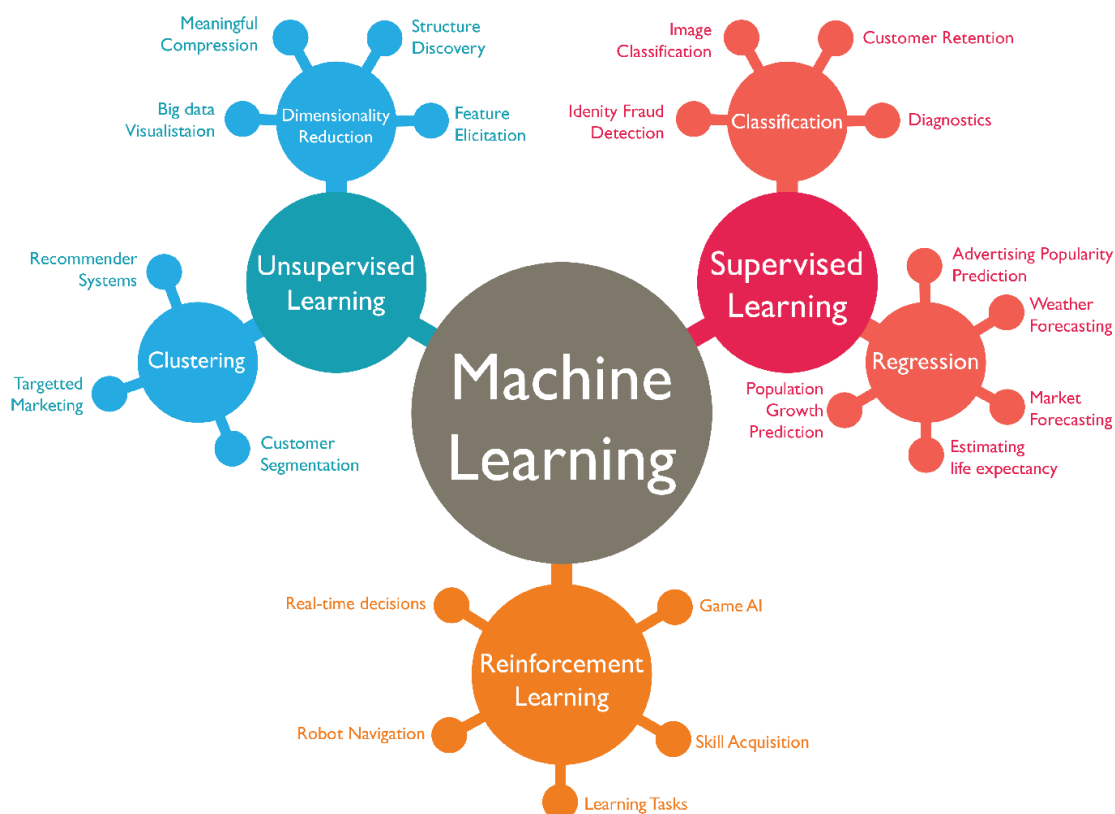


Figura N°37: Tipos de Machine Learning.

Fuente: Shankar, S. (2020). *Types of Machine Learning Algorithms*. Recuperado de: <https://www.7wdata.be/visualization/types-of-machine-learning-algorithms-2/>.

2.2.12.1 Aprendizaje por refuerzo

El aprendizaje por refuerzo se basa en cómo aprenden los seres humanos de tal modo que se recopilan datos y se ejecutan acciones. Estas acciones a su vez son premiadas si es que la acción se ha realizado correctamente o se penaliza si no es la acción esperada de tal modo que el algoritmo entienda cuál es la acción correcta (Potentia Analytics Inc, 2020).

En este tipo a aprendizaje, los algoritmos aprender de diversas situaciones y se mejoran a sí mismos hasta conseguir que el modelo entienda la acción que deben realizar (Potentia Analytics Inc, 2020).

Basado en el concepto psicológico de condicionamiento, el aprendizaje por refuerzo funciona poniendo a prueba el algoritmo en un ambiente similar al real a modo de entrenamiento y programando un sistema de recompensas que otorgue un premio o refuerzo por cada vez que ejecute la acción de manera adecuada y una penalidad por cada que lo ejecute mal (Potentia Analytics Inc, 2020).

Este tipo de Machine Learning, a diferencia de los 2 anteriores, aprende de sus errores de tal modo que su objetivo es maximizar la recompensa que obtiene por cada acción realizada de tal modo que la inteligencia deduzca qué acción es la que debe de realizar (Shankar, 2020).

En la Figura N°38, se puede observar el flujo de entrenamiento aplicando aprendizaje por refuerzo. Como se puede observar, el flujo contiene un elemento llamado *Agent* o agente, este elemento es el que se desea que tenga un comportamiento inteligente. El otro elemento es conocido como *environment* o ambiente de entrenamiento en donde es el lugar en donde el agente va a entrenar. Este ambiente envía señales al agente el cual interpreta y devuelve una acción a realizar dentro del *environment*. Como consecuencia a esta acción, se genera un nuevo estado y la acción es premiada o castigada en función a lo que se desea que la Inteligencia Artificial deba realizar.

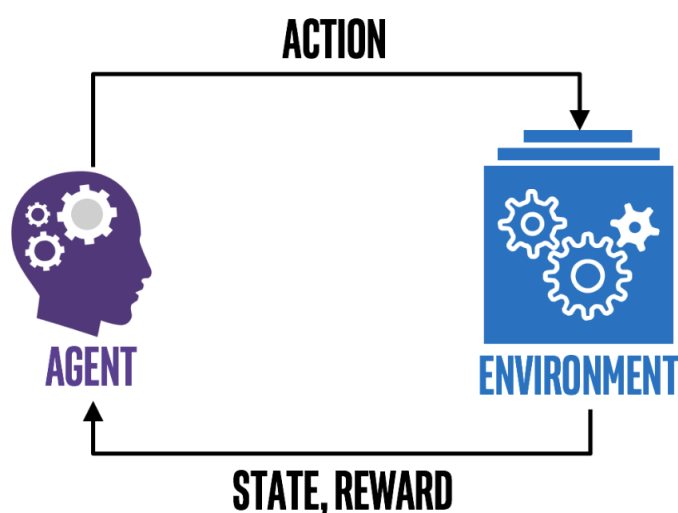


Figura N°38: Flujo de entrenamiento en el Aprendizaje por refuerzo.

Fuente: Shankar, S. (2020). *Types of Machine Learning Algorithms*. Recuperado de: <https://www.7wdata.be/visualization/types-of-machine-learning-algorithms-2/>.

Este tipo de aprendizaje es uno de los más utilizados dentro del mundo de los videojuegos dado que los usuarios esperan enfrentarse a un oponente (agente entrenado) que les ofrezca un reto y una vez superado, pasar al siguiente nivel. Tal cual como se presenta en la Figura N°39, la misma lógica de aprendizaje descrita anteriormente, se aplican a videojuegos tradicionales de tal modo que los agentes entrenados aprendan a saltar, caminar, esquivar obstáculos, atacar, entre otras.

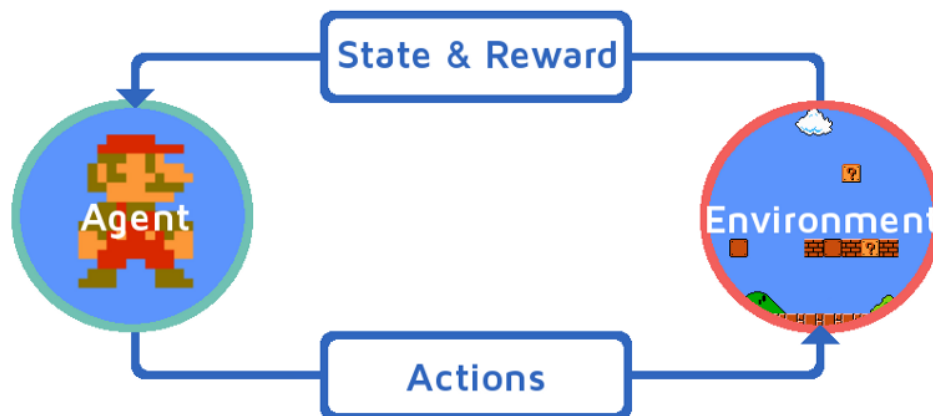


Figura N°39: Aprendizaje por refuerzo aplicado a videojuegos.
 Fuente: Shankar, S. (2020). *Types of Machine Learning Algorithms*. Recuperado de: <https://www.7wdata.be/visualization/types-of-machine-learning-algorithms-2/>.

2.2.12.1.1 Ejemplo aplicativo

Para ejemplificar el aprendizaje por refuerzo se tiene el siguiente ejemplo aplicativo: Se requiere que un gato aprenda a realizar un truco nuevo. El flujo del entrenamiento se presenta en la Figura N°40.

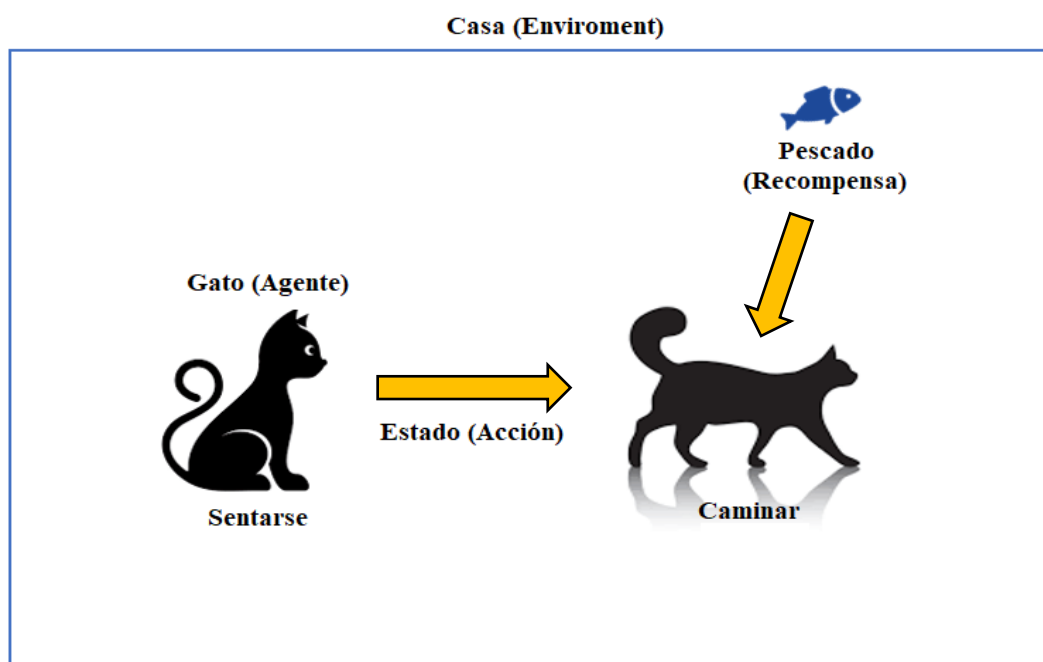


Figura N°40: Ejemplo aplicativo del aprendizaje por refuerzo.
 Fuente: Guru. (2021). *Reinforcement Learning: What is, Algorithms and Applications*.

Algunas de las consideraciones del caso se presentan a continuación:

- Como el gato no entiende ningún idioma hablado por los humanos, no se le puede explicar la acción que debe realizar.
- Se pretende representar la situación entrenando al gato (*agente*) en su hogar (*Environment*). Cada que el gato realice la acción de caminar, se le dará un pescado (*Reward*) tal cual como se muestra en la Figura N°40.
- El agente realizará acciones de forma aleatoria hasta realizar aquella acción que le permita obtener la recompensa deseada.

Como se aprecia en la Figura N°40, el agente se expone ante un ambiente de entrenamiento que en este caso es el hogar del gato. El gato reacciona ante la interacción entre él y el ambiente de entrenamiento pasando de un estado a otro (de estar sentado a caminar). Luego del cambio de estado producto de la acción realizada por el agente, se le otorga una recompensa (pescado) si la acción que ha realizado es la correcta. De esta forma, el gato “aprende” a ejecutar una acción determinada.

2.2.12.1.2 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) o conocido también como aprendizaje por refuerzo profundo combina las técnicas de Deep Learning y Reinforcement Learning para crear agentes mucho más avanzados que puedan aprender, a través de redes profundas, a realizar cualquier tipo de tarea (Shao et al., 2019). DRL ha logrado un gran avance en la Inteligencia Artificial en los videojuegos como StarCraft, Dota2 y ViZDoom (Shao et al., 2019).

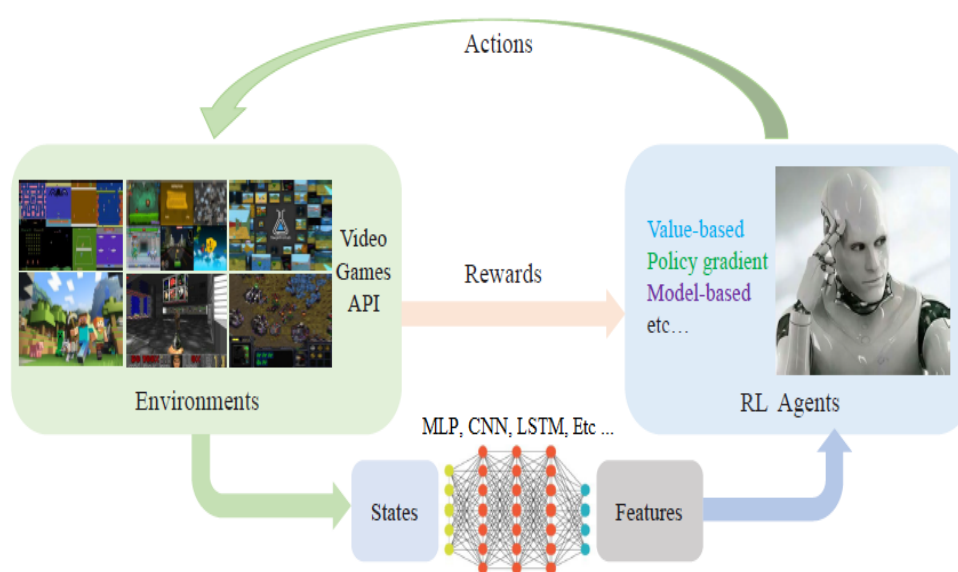


Figura N°41: Flujo de trabajo de DRL aplicado a videojuegos.

Fuente: Shao, K., Tang, Z., Zhu, Y., Li, N., y Zhao, D. (2019). *A Survey of Deep Reinforcement Learning in Video Games*. (p.2)

Como se observa en la Figura N°41, el *environment* envía señales a la red neuronal que puede ser una MLP, CNN o LSTM. El tipo de red que se escoja depende del tipo de videojuego que se desee realizar. A la red profunda ingresan los estados que el *environment* envía y devuelve como salidas las acciones que el agente debe de realizar. Una vez que el agente realiza las acciones, se evalúa si la acción que ha realizado es correcta o no. Si la acción realizada es la que se tenía planeada, entonces se le da una recompensa o *reward* de lo contrario se le da un refuerzo negativo. Para explicar los conceptos matemáticos del aprendizaje por refuerzo primero se debe de entender el Proceso de Decisión de Markov que se encuentra representado en la Figura N°42.

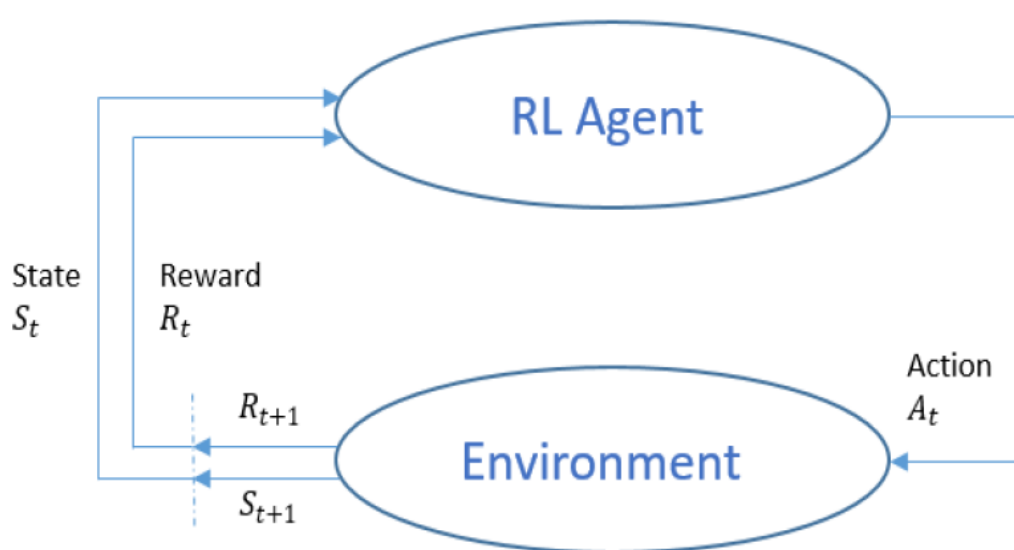


Figura N°42: Proceso de decisión de Markov.

Fuente: Han, M., Zhao, J., Zhang, X., Shen, J., y Li, Y. (2020). *The reinforcement learning method for occupant behavior in building control: a review*. (p.5)

La Figura N°42 representa el Proceso de decisión de Markov en donde S_t representa el estado inicial que presenta el *environment* en un tiempo t . Posteriormente, el agente realiza una determinada acción A_t y que a su vez esa acción intenta maximizar la recompensa o *Reward* representado por un R_t . Como resultado de esa acción, se genera un nuevo estado S_{t+1} así como una recompensa R_{t+1} y el flujo se repite nuevamente.

Además, al momento de que el agente realice una nueva acción, determina una recompensa esperada a obtener por la ejecución de dicha acción. La ecuación para representar dicha recompensa que se espera recibir queda modelada de la siguiente manera:

$$r(s, a) = \mathbb{E}[R_t | S_{t-1} = s, A_{t-1} = a] \quad (40)$$

De la Ecuación (40), se puede observar que la recompensa esperada está en función del estado y la acción anterior.

Adicionalmente, existe otro concepto conocido como *Policy* o política que es de vital importancia para el entendimiento de DRL. Una *Policy* denotada por π es una distribución sobre las acciones dados un conjunto de estados y que puede ser considerada como una función de comportamiento (Shao et al., 2019). Una política define el accionar de un agente indicándole cómo debe actuar cuando se encuentra en un nuevo estado en función a los errores cometidos anteriormente. Una política tiene como principal objetivo evaluar el rendimiento futuro esperado cuando se realice una acción a desde el tiempo t de tal modo que la sumatoria de las recompensas durante un episodio sea la mayor posible. La ecuación que representa la sumatoria de todas las recompensas durante un episodio determinado está representada de la siguiente manera:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_{t+T_{m\acute{a}x}} = \sum_{i=0}^{T_{m\acute{a}x}-1} R_{t+i+1} \quad (41)$$

Donde:

- $T_{m\acute{a}x}$ denota el último paso o acción de un episodio.

Como se había mencionado, una *Policy* o política es una función que depende de las acciones y los estados. La política selecciona una acción $a \in A$ que puede realizar para cada estado $s \in S$. La función de un agente es seguir esa política π en un tiempo t de tal modo que específicamente una política está representada por $\pi(a_t|s_t)$. Durante todo el proceso de entrenamiento, una política se va actualizando de tal modo que el agente, en función a la experiencia (obtención de los premios y castigos recibidos anteriormente) pueda escoger la acción más adecuada dentro de la política.

Otro concepto importante en *Deep Reinforcement Learning* es la Función de Estado-Valor. Esta función está representada por $V_{\pi}(s)$. Donde π es la política o *Policy* y s es el estado. La Función de Estado-Valor muestra qué tan bueno es estar en el estado actual donde se encuentra el agente, esta función está representada de la siguiente manera:

$$V_{\pi}(s) = E \left[\sum_{i=0}^{T_{m\acute{a}x}} \gamma^i R_{t+i+1} | s_t = s \right] \quad (42)$$

Donde:

- γ es el factor de descuento que va en un intervalo de $[0,1]$. Siendo cero cuando la recompensa es futura o mejor dicho cuando no es la recompensa actual y es 1 cuando se trabaja con la actual recompensa.

Del mismo modo que existe una Función de Estado-Valor, existe la Función Acción-Valor que estima qué tan buena es tomar una acción a en un estado s bajo una política π . Esta función esta representada por $Q_\pi(a, s)$. Así, esta función estima la recompensa esperada a partir de un estado s tomando una acción a y siguiendo una política π . Esta función es representada por la siguiente ecuación:

$$Q_\pi(a, s) = E \left[\sum_{i=0}^{T_{\max}} \gamma^i R_{t+i+1} | \mathbf{s}_t = s, \mathbf{a}_t = a \right] \quad (43)$$

En este apartado, se ha profundizado en términos concretos de *Deep Reinforcement Learning* como estados, acciones, políticas, entre otros, así como las ecuaciones matemáticas detrás de este aprendizaje dado que son conceptos necesarios antes de pasar a la siguiente subsección en donde se detallará un método particular de esta técnica conocido como *Proximal Policy Optimization*.

2.2.12.1.2.1 Proximal Policy Optimization (PPO)

Este algoritmo surge en respuesta de que los algoritmos tradicionales de DRL que crean nuevas políticas de un agente eran en ocasiones demasiado extensas por la cantidad de acciones que la nueva política tenía (Schulman et al., 2017). Este algoritmo tiene la principal función de optimizar la cantidad de acciones de la *Policy* siguiente del agente de tal modo que se obtenga una nueva política mucho más eficiente en función a las políticas anteriores y que con estas nuevas políticas optimizadas se pueda obtener una recompensa acumulada máxima (Schulman et al., 2017).

El algoritmo de *Proximal Policy Optimization* utiliza un método conocido como *Actor-Critic* el cual es un método que combina la Función de Estado-Valor y una política. La parte del *Actor* se refiere a una política optimizada y *Critic* se refiere a la Función de Estado-Valor que está aprendiendo (Adamsson, 2018).

El método conocido como *Actor-Critic* utiliza la Ecuación (42) y la política $\pi(a_t | s_t)$ para resolver los problemas de aprendizaje por refuerzo. Un agente que se encuentra en un estado s está tratando de escoger la acción a más adecuada de tal modo que esa acción permita maximizar su recompensa. El método *Actor-Critic* está representado por la siguiente ecuación:

$$A_\pi(\mathbf{s}_t, \mathbf{a}_t) = Q_\pi(\mathbf{s}_t, \mathbf{a}_t) - V_\pi(\mathbf{s}_t) \quad (44)$$

Si el valor que obtiene la Ecuación (44) es mayor a cero quiere decir que el desempeño del agente fue mejor de lo esperado en promedio. La ecuación del método *Actor-Critic* no solo permite determinar no solo qué tan buena es la acción tomada por el agente, sino qué tal bien es esta con respecto al valor que se espera obtener (Adamsson, 2018).

Mnih et al. (2016) citado en Adamsson (2018) popularizó en su trabajo de investigación la implementación del método *Actor-Critic* reformulando la Ecuación (44) de la siguiente manera:

$$A_{\pi}(\mathbf{s}_t, \mathbf{a}_t) = \sum_{i=0}^{K-1} \gamma^i R_{t+i} + \gamma^k V_{\pi}(\mathbf{s}_t, k) - V_{\pi}(\mathbf{s}_t) \quad (45)$$

Donde:

- K es conocido como horizonte de tiempo o *Time Horizont*.
- R es la recompensa.
- γ es el factor de descuento.

El horizonte del tiempo o *Time Horizont* define la longitud de una trayectoria. Del mismo modo, el objetivo de una política es maximizar la recompensa del agente sobre las trayectorias. El horizonte del tiempo debe ser lo suficientemente largo como para que el agente reciba una recompensa significativa dentro de este.

En el método *Actor-Critic* se deben de minimizar dos funciones de pérdida. La primera es conocida como *Value Loss* cuya ecuación está representada de la siguiente manera:

$$L_{Value} = \left(\sum_{i=0}^{K-1} \gamma^i R_{t+i} + \gamma^k V_{\pi}(\mathbf{s}_t + k) - V_{\pi}(\mathbf{s}_t) \right)^2 \quad (46)$$

El segundo es conocido como *Policy Loss*, cuya ecuación es la siguiente:

$$L_{Policy} = -\log(\pi(\mathbf{a}_t | \mathbf{s}_t)) A_{\pi}(\mathbf{s}_t, \mathbf{a}_t) \quad (47)$$

Donde:

- $\pi(\mathbf{a}_t | \mathbf{s}_t)$ es la probabilidad de seleccionar una acción \mathbf{a}_t en un estado \mathbf{s}_t .

Un término que mide la entropía se puede agregar se puede agregar al *Policy Loss* para evitar que el agente se quede atascado en un óptimo local. La entropía es una medida que mide la dispersión de las probabilidades de las acciones (Adamsson, 2018). Este término se puede agregar con el objetivo de aumentar la exploración. Así, la Ecuación (47), incluyendo el término de la entropía quedaría de la siguiente manera:

$$L_{Policy} = -\log(\pi(a_t|s_t))A_{\pi}(s_t, a_t) - \beta H(\pi) \quad (48)$$

Donde:

- β es la magnitud de regularización.
- $H(\pi)$ es el valor de la entropía.

$H(\pi)$ está representada mediante la siguiente ecuación:

$$H(\pi(a_t|s_t)) = -\sum_{i=1}^M P(a_t|s_t)\log P(a_i|s_t) \quad (49)$$

Donde:

- M es el número de acciones
- $\pi(a_t|s_t) = [P(a_1|s_t), \dots, P(a_M|s_t)]$

El algoritmo de *Proximal Policy Optimization* se basa en los métodos de gradiente de políticas (*Policy Gradients*, en inglés) y se utiliza para el aprendizaje de una política parametrizada. Además, el algoritmo de *Proximal Policy Optimization* puede ser utilizado para el aprendizaje de una política, pero no es necesaria para la selección de acciones. El vector de políticas es denotado por θ . La probabilidad de tomar acción a partiendo de un estado s dado y un vector de políticas θ en un tiempo t es formulado como $\pi(a|s, \theta) = P\{a_t = a|s_t = s, \theta_t = \theta\}$.

Los métodos de gradiente de políticas son utilizados para minimizar una función de pérdida como la Ecuación (46) y Ecuación (48). Por lo tanto, el ascenso de gradiente se utiliza para moverse en las direcciones y así aumentar la función objetivo. Esto último se logra ajustando los parámetros de la política en la dirección del gradiente. El más común estimador de gradiente se presenta en la siguiente ecuación:

$$\hat{g} = \hat{\mathbb{E}}_t [\Delta_{\theta} \log \pi_{\theta}(a_t|s_t) \hat{A}_t] \quad (50)$$

Donde:

- π_{θ} es la política.
- Δ_{θ} es la gradiente
- \hat{A}_t es un estimador de la función en un tiempo t .
- $\hat{\mathbb{E}}_t [\dots]$ indica el promedio empírico sobre un conjunto finito de muestras en un algoritmo que alterna entre muestreo y optimización.

Si la función es positiva, es decir, si el agente recibió una gran recompensa del entorno, la gradiente debe de ajustarse para favorecer ese comportamiento específico. Por otro lado, si la función es negativa, entonces la gradiente debe de ajustarse para evitar ese comportamiento específico. La política se actualiza calculando la función objetivo y ajustando en la dirección del gradiente.

El algoritmo de *Proximal Policy Optimization* utiliza una función objetivo sustituto recortada (*Clipped surrogate Objective Function*, en inglés) para evitar grandes políticas al momento de la actualización de estas. Llevan el nombre de “sustituto recortada” (*Clipped surrogate*, en inglés) dado que desalienta grandes cambios entre la política anterior y la nueva política “recortando” la función objetivo. La función objetivo sustituto recortada se define de la siguiente manera:

$$L^{CLIP}(\theta) = \mathbb{E}_t [\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (51)$$

Donde:

- ϵ es un hiperparámetro usualmente entre 0.1 y 0.2.
- \hat{A}_t es un estimador de la función en un tiempo t .
- El hiperparámetro $r_t(\theta)$ es el ratio entre la antigua y la nueva política.

$r_t(\theta)$ está definida como:

$$r_t(\theta) = \frac{\pi_\theta}{\pi_{\theta \text{ anterior}}} \quad (52)$$

Si $r_t(\theta)\hat{A}_t$ se vuelve demasiado extensa, entonces se recorta. Esto previene que $r_t(\theta)$ salga del intervalo de $[1 - \epsilon, 1 + \epsilon]$. El método de gradiente de políticas actualiza a un máximo de $1 + \epsilon$ si la función objetivo es positiva y $1 - \epsilon$ si la función objetivo es negativa. Realizar esto último previene una nueva política que sean más largas que las anteriores.

De este modo surgió un nuevo algoritmo para la optimización de políticas conocido como *Proximal Policy Optimization* o PPO (por sus siglas en inglés) con el objetivo de que las nuevas políticas o conjunto de acciones que surjan en función a la política anterior no sean excesivamente grandes de tal modo que se pueda disminuir el costo computacional y que a su vez el tiempo que se tome el agente en realizar esa nueva política sea el menor.

El algoritmo de PPO implementa el método *Actor-Critic*. La función de *Policy Loss* usado en el algoritmo de PPO es una función objetivo sustituto recortada combinado con la función de *Value Loss*.

De este modo, la ecuación del algoritmo de *Proximal Policy Optimization* queda representada de la siguiente manera:

$$L^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t [L^{CLIP}(\theta) - C_1 L_t^{VF}(\theta) + \beta H(\pi)] \quad (53)$$

Donde:

- θ es el parámetro de la política o *Policy*.
- C_1 y β son coeficientes que describen la magnitud de la función de *Value Loss* $L_t^{VF}(\theta)$ y la entropía $H(\pi)$ respectivamente.

En el algoritmo de *Proximal Policy Optimization* la política inicial interactúa con el entorno o *environment* para generar una secuencia de episodios (Quang y Hoon, 2020). Luego se recopilan las acciones iniciales y se empiezan a crear experiencias con el objetivo de establecer una nueva política $L^{CLIP}(\theta)$ y esta vez esta nueva política contendrá acciones optimizadas de tal modo que se obtenga una recompensa adecuada a estas nuevas acciones realizadas (Quang y Hoon, 2020).

Para la implementación del algoritmo de PPO en un lenguaje máquina se empieza con la definición de la cantidad de iteraciones que se desea realizar. Seguidamente se establecen la cantidad de actores o agentes que van a participar dentro del *environment*.

Cuando la red neuronal profunda recibe las señales transmitidas por el *environment*, las convierte en acciones a través de sus salidas y el conjunto de estas origina la primera *Policy* o política. Esta nueva política se ejecuta en el ambiente de entrenamiento de tal modo que esta última se transforma en una política antigua ($\theta_{\pi_{old}}$) que ha sido ejecutada en un tiempo T. Luego se calcula la pérdida en los NT pasos de tiempo y este error se minimiza con un minibatch para K épocas.

En función a las iteraciones realizadas se obtiene una nueva política en función a las anteriores realizadas en un determinado lapso. Esta nueva política reemplaza a la anterior de tal modo que la cantidad de acciones a realizar para la ejecución de esta nueva política es la mínima tal cual como se mostró en la Ecuación (51).

A modo de resumen, *Deep Reinforcement Learning* combina las técnicas de Deep Learning y Reinforcement Learning o aprendizaje por refuerzo. El ciclo de aprendizaje de un agente se establece a través del criterio de decisión de Markov. Una vez establecido el ambiente de entrenamiento, se establece cuáles serán las entradas a la red. La red neuronal profunda devuelve una salida que se transforman en acciones que el agente deberá realizar. En función

de las acciones realizadas por el agente se dará una recompensa o premio de tal modo que se vaya obteniendo una recompensa promedio y se le penalizará si es que la acción realizada por el agente es la incorrecta. Este conjunto de acciones se les denomina *Policy* o política. Dado que se ha establecido la cantidad de iteraciones que el modelo de DRL va a realizar, el algoritmo de *Proximal Policy Optimization* interviene transformando las acciones pasadas del agente en experiencias de tal modo que el agente establece una nueva política que contenga una menor cantidad de acciones de tal manera que se pueda maximizar la recompensa obtenida. La función principal del algoritmo de PPO es optimizar las acciones que deba realizar el agente en la siguiente iteración a fin de ahorrar esfuerzo computacional y obteniendo la mayor cantidad de recompensa promedio posible.

2.2.13 Técnica estadística para el análisis de resultados

En esta sección se explicarán los diferentes conceptos y las principales técnicas estadísticas que se pueden aplicar a los resultados obtenidos, así como el estadístico a utilizar para la elaboración del presente proyecto.

2.2.13.1 Formulación de las hipótesis

Para contrastar una hipótesis siempre se formularán 2 alternativas conocidas como Hipótesis Nula (H_0) e Hipótesis Alterna (H_1). A la llamada Hipótesis Nula será a la cual se le conceda en beneficio de la duda. El objetivo será determinar si se tiene las suficientes pruebas para aceptar o rechazar esta hipótesis nula. Esto último se conocerá mediante las propias características que posean nuestros datos de tal manera que se muestren las evidencias necesarias para rechazar el H_0 . En el caso de que se acepte H_0 será por el hecho de que no se tuvieron las pruebas suficientes para en contra de esta. Por otro lado, si se rechaza el H_0 , será por el hecho de que se encontraron pruebas suficientes dentro de nuestra data para estar completamente seguros de rechazar la Hipótesis Nula (Botella, Alacreu, y Martínez, 2012).

2.2.13.2 P- valor

El llamado P-valor o valor “P” es la probabilidad de aceptar la hipótesis alterna como cierta cuando en realidad la verdadera podría ser la hipótesis nula (Rubio y Berlanga, 2012). Entre más alto sea el p-valor, será mayor la probabilidad de error al escoger la hipótesis alterna (Botella et al., 2012).

2.2.13.3 Nivel de significancia

El nivel de significancia representado por el símbolo “ α ” es el valor límite o umbral para la probabilidad del error o p-valor de tal manera que se puede afirmar que la probabilidad de

equivocarse es mínima o no (Botella et al., 2012). En los estudios estadísticos, el nivel de significancia suele tomar los valores de 0.05, 0.01 o 0.1. El tomar un valor más cercano a 1 implica asumir un mayor riesgo de poder equivocarse (Botella et al., 2012).

2.2.13.4 Decisión de aceptación o rechazo de la hipótesis

Para aceptar o rechazar la hipótesis nula, se debe conocer tanto los valores del P-valor como del nivel de significancia. Si el p-valor es mayor al nivel de significancia, se debe aceptar la hipótesis nula ya que la probabilidad de error sería mayor al riesgo que se desea asumir. Por otro lado, si el p-valor es menor al nivel de significancia, entonces se rechaza la hipótesis nula ya que la probabilidad de error de escoger la hipótesis alterna es menor al riesgo que se está dispuesto a asumir. Lo anterior mencionado, se puede resumir en la Ecuación (54) y Ecuación (55).

$$P - \text{valor} < \alpha \text{ (Se rechaza } H_0) \quad (54)$$

$$P - \text{valor} > \alpha \text{ (Se acepta } H_0) \quad (55)$$

2.2.13.5 Prueba T para dos muestras independientes

La Prueba T para muestras independientes se utiliza cuando se tienen dos muestras independientes y consiste en comparar las medias de una variable para ambos grupos en un mismo intervalo de tiempo con el objetivo de ver un cambio significativo entre las medias de ambas muestras (Rubio y Berlanga, 2012). En este tipo de prueba se suele someter a uno de los grupos a la utilización de un método nuevo propuesto por el investigador y el otro grupo suele permanecer utilizando un método tradicional. Dentro de la propia fuente de datos debe de contener una columna con los datos de la propia variable de interés y una columna en donde se especifique a qué grupo pertenecen esos datos (Rubio y Berlanga, 2012).

2.2.13.6 Metodología para la obtención del estadístico T de Student

El estadístico T-student es un estadístico que se utiliza para examinar dos muestras pequeñas e independientes que posean una distribución normal e igualdad de varianzas. La prueba T de Student sirve para “comparar dos muestras de tamaño ≤ 30 ” (Sánchez, 2015, p. 59). Para utilizar la una prueba T de Student primero se deben de formular las hipótesis H_0 (hipótesis nula) y H_1 (hipótesis alterna). La hipótesis nula tiene que mencionar que no existe una diferencia significativa entre la media de las dos muestras y si existiera diferencia a simple vista, esta se debe al azar. Por otro lado, la hipótesis alterna debe afirmar que existe una diferencia significativa entre la media de las dos muestras (Sánchez, 2015). Al encontrar el p-valor del estadístico T-Student, este se debe de comparar con el nivel de significancia

establecido por el investigador de tal manera que la decisión de aceptación o rechazo de la hipótesis nula viene a ser la misma planteada en la Ecuación (54) y Ecuación (55). Según Sánchez (2015), la metodología para determinar el valor del estadístico T-Student es el siguiente:

- **Paso 1:** Verificar que cada muestra siga una distribución normal (Prueba de normalidad).

Todos los valores de la variable dependiente deben de seguir una distribución normal. La prueba de normalidad permite verificar que el conjunto de datos con el cual se está trabajando sigue una distribución normal o gaussiana (Rubio y Berlanga, 2012). Para la aplicación de esta prueba, se deben de redactar las siguientes hipótesis:

H_0 = Los datos provienen de una distribución normal

H_1 = Los datos no provienen de una distribución normal

Posterior a eso, se debe de establecer el valor del α con el cual se planea trabajar. La prueba de normalidad tiene dos tipos: Prueba de Shapiro Wilk y Prueba de Colmogorov Smirnoff (Rubio y Berlanga, 2012). Para la elección de la prueba de normalidad adecuada se debe de tener presente el tamaño de la muestra. Si la muestra es inferior a 30, se debe usar la prueba de Shapiro Wilk. Por otro lado, si la prueba es superior a 30, se debe de usar la prueba de Colmogorov Smirnoff (Rubio y Berlanga, 2012). Para que los datos provengan de una distribución normal el P-valor encontrado debe ser mayor al nivel de significancia propuesto (Rubio y Berlanga, 2012). El criterio de aceptación o rechazo, son los mismos planteados en la Ecuación (54) y Ecuación (55).

- **Paso 2:** Por cada muestra, encontrar su respectivo tamaño, media y varianzas.

Por cada con la cual se trabaje, se debe hallar su respectivo tamaño, media y varianza de tal manera que se cumpla la terminología presente en la Ecuación (56):

n_1 = *tamaño de la muestra 1*

m_1 = *media de la muestra 1*

v_1 = *varianza de la muestra 1*

n_2 = *tamaño de la muestra 2*

m_2 = *media de la muestra 2*

v_2 = *varianza de la muestra 2* (56)

- **Paso 3:** Verificar que las varianzas sean homogéneas (Prueba de Levene).

Para la realización de la prueba de Levene, primero se deben de definir las siguientes hipótesis:

H0: Las varianzas de los datos son iguales en ambos grupos.

H1: Existe una diferencia significativa en las varianzas de ambos grupos.

Posteriormente, se debe de definir el nivel de significancia. La prueba de Levene debe de arrojar una significación mayor de 0.05 para que se cumpla el requisito de homogeneidad de varianzas. Por último, para la aceptación o rechazo de la hipótesis, se debe de utilizar el mismo criterio definido en Ecuación (54) y Ecuación (55).

- **Paso 4:** Establecer las diferencias entre las medias y calcular la varianza común de las dos muestras.

Se deben de restar las diferencias que existen entre las medias de ambas muestras y obtener la varianza común tal como se muestra en la Ecuación (57) y Ecuación (58).

$$D_m = m_1 - m_2 \quad (57)$$

Donde: D_m : Diferencia de las medias de ambas muestras

$$V_c = ((n_1 - 1) \cdot v_1 + (n_2 - 1) \cdot v_2) / (n_1 + n_2 - 2) \quad (58)$$

Donde: V_c : Varianza común de ambas muestras

- **Paso 5:** Calcular el error estándar de la diferencia de medias (ESM).

El error estándar de la diferencia de las medias se define como se muestra en la Ecuación (59):

$$ESM = \sqrt{V_c \cdot \frac{n_1 + n_2}{n_1 \cdot n_2}} \quad (59)$$

Donde: ESM : Error estándar de la diferencia de medias

- **Paso 6:** Calcular el valor del estadístico T-student.

El valor del estadístico T-Student se define como el cociente entre la diferencia de medias entre el error estándar de la diferencia de medias. Esto se puede apreciar en la Ecuación (60):

$$t = \frac{D_m}{ESM} \quad (60)$$

Donde: t : Valor de estadístico T-Student

- **Paso 7:** Encontrar el respectivo p-valor en una tabla de T-Student

El último paso consiste en buscar en las tablas de T- Student para encontrar el valor del p-valor. Para esto último, es necesario conocer los grados de libertad. Los grados de libertad se calculan como se muestra en la Ecuación (61).

$$GL = n_1 + n_2 - 2 \quad (61)$$

Donde: GL : Grados de Libertad

2.3 Marco Conceptual o Contexto de Investigación

En esta última sección, se detallarán los principales conceptos utilizados en el presente trabajo de investigación y una breve descripción del sector al cual va dirigido.

2.3.1 Marco Conceptual

A continuación, se presentan los principales conceptos a utilizar en el presente trabajo de investigación.

1. **Videojuego:** Es “un juego usando una computadora y un visor de video. Puede ser un computador, un teléfono móvil o una consola de juegos” (Eguia et al., 2012, p.5).
2. **Metodología en cascada:** Es una metodología usada en el desarrollo de software que consta de 5 fases que se siguen de manera secuencial (Pressman, 2010).
3. **Aprendizaje basado en videojuegos:** Nueva tendencia que consiste en usar un videojuego con el fin de poder enseñar de una mejor manera a las personas a través de la mejora del proceso de aprendizaje (Trends E. , 2016).
4. **Comprensión Lectora:** Es el resultado de la interacción entre el texto y el lector en donde este último, a través de los conocimientos que posee trata de dar un significado a lo leído para obtener una interpretación adecuada del contenido (Gordillo y Del Pilar (2009), p. 97).
5. **Lectura Comprensiva:** Es una actividad primordial cuyo objetivo principal es llegar a profundizar lo leído utilizando estrategias en medio o al final de la lectura (Javier, 2013).

6. **Realidad Virtual:** Es un tipo de tecnología que permite crear otro mundo pero este es virtual y a través de un ordenador u otro dispositivo se puede acceder a el, estando consientes de que esa realidad no existe (Rivera, 2018).
7. **Base de datos:** Repositorio en donde se pueden almacenar grandes cantidades de información de manera ordenada y que posteriormente puede ser consultada a través de uno o muchos usuarios (Pérez D. , 2007).
8. **Hosting:** Repositorio donde se almacenan las páginas web y hacen posible que estas últimas se encuentren siempre disponible para los usuarios todos los días de la semana y las 24 horas del día (Alonso, 2020).
9. **Deep Reinforcement Learning:** Subcampo del aprendizaje por refuerzo que combina las técnicas de Deep Learning y Reinforcement Learning para crear agentes mucho más avanzados que puedan entender a través de redes profundas a realizar cualquier tipo de tarea (Shao et al., 2019).
10. **Agente inteligente:** Cualquier modelo en 3D que tiene un comportamiento inteligente y que ha sido entrenado para la realización de tarea determinada pudiéndola ejecutar de manera adecuada (Pierre , 2020).
11. **Prueba T de Student:** Es una prueba estadística que sirve para “comparar dos muestras de tamaño ≤ 30 ” (Sánchez, 2015, p. 59).
12. **Muestras independientes:** Son un tipo de muestra en donde el o los grupos seleccionados no poseen características en común (Rubio y Berlanga, 2012).

2.3.2 Contexto de Investigación

El sector al cual va dirigido el presente trabajo es al sector educación. Este sector actualmente se encuentra en proceso de crecimiento sostenido y desarrollo inclusivo (Guadalupe, León, Rodríguez , y Vargas, 2017). El Ministerio de Educación ha promovido nuevos cambios en este sector incrementando la inversión en el mismo, mejorando la capacitación de los docentes, mejorando las infraestructuras educativas y mejorando los procesos de enseñanza y especialmente en mejorar los logros del aprendizaje de los estudiantes (Guadalupe et al., 2017).

No obstante, una de las principales quejas que tienen los docentes es que no cuentan con material educativo de calidad para una correcta enseñanza de los temas. Además, a esto se suma un desconocimiento de estrategias de aprendizaje que posibilitan poder entender mejor los temas por parte de los alumnos (Guadalupe et al., 2017).

Por otro lado, los indicadores de la Unesco y por la OCDE sugieren que el Perú se encuentra muy por debajo de la inversión que los países pertenecientes a la OCDE invierten en educación. Por otro lado, se muestra que nivel de gasto por estudiante es inferior al de otros países. Se ha observado que el incremento del nivel de gasto por estudiante no asegura exactamente una mejora en la calidad educativa y si esta mejora, el incremento no es el esperado. Esto quiere decir que no se encuentra una relación lineal entre el gasto por estudiante y el rendimiento académico (Guadalupe et al., 2017).

El Perú a pesar de haber empezado un incremento de inversión en el sector educativo. Las mejoras no son notables. Esta consecuencia puede deberse a una mala inversión por parte del Ministerio de Educación (Guadalupe et al., 2017). Los colegios se preocupan más por la infraestructura que poseen que por la calidad de enseñanza que puedan ofrecer al alumnado. Es cierto que la infraestructura suele tener un papel importante, pero si este no va de la mano con la calidad educativa que aparenta poseer, los estudiantes no podrán obtener un mejor rendimiento académico (Guadalupe et al., 2017).

Por otro lado, es importante contar con docentes adecuadamente capacitados para un correcto desenvolvimiento dentro de las clases y que estos puedan contar con materiales de buena calidad para que puedan brindar sus clases de una mejor manera y posibiliten una mejor enseñanza para el alumnado (Guadalupe et al., 2017).

Con la aparición del nuevo coronavirus en el mundo, el Perú empezó a implementar medidas con el objetivo de disminuir el número de contagiados. A partir del 15 marzo del 2020, se dispuso el cierre de los colegios en el Perú dado que es un foco de contagio debido a la aglomeración de alumnos. Esta medida inicialmente fue impuesta por 15 días, no obstante se prolongó hasta el 30 de julio. Luego de esta última fecha, se dispuso que tanto colegios, institutos y universidades deberían de tomar esta modalidad de educación a distancia hasta que finalice el estado de emergencia sanitaria.

Si bien es cierto que los colegios, principalmente los privados, estaban ante una situación de incertidumbre debido a las clases que se estaban perdiendo poco a poco, principalmente entre los meses de abril y junio, fueron adaptándose a una modalidad de educación a distancia utilizando todas las herramientas tecnológicas existentes para poder mantener la interacción con sus alumnos.

Adicionalmente, como un plan de acción, el MINEDU aprobó la compra de kits de limpieza para poder distribuirse entre colegios públicos y privados para mantener los ambientes educativos limpios.

Asimismo, MINEDU propuso medidas que los profesores deben de implementar en sus salones de clases virtuales para contribuir con la disminución de casos de Covid-19 y poder cuidar la salud de sus alumnos. Algunas de estas medidas son las siguientes:

- Promover siempre el lavado correcto de las manos por 20 segundos a sus alumnos.
- Explicar a los alumnos que no se deben de tocar la nariz, ojos o boca mientras se tenga las manos sucias.
- Informar a los alumnos que cuando se estornuda o se tosa, no es correcto cubrirse con las manos sino con el codo.
- Promover el saludo a distancia a sus alumnos con los miembros de su familia.
- Explicar la importancia de mantener los ambientes ventilados.

CAPITULO III: METODOLOGÍA DE LA INVESTIGACIÓN

En este capítulo se explicará el diseño, tipo y enfoque del presente trabajo de investigación. Del mismo modo, se detallará la muestra utilizada y la operacionalización de las variables del proyecto. También se expondrán los criterios utilizados para seleccionar la metodología para el desarrollo del videojuego, el tipo de realidad virtual, así como la base de datos a utilizar, la técnica de Inteligencia Artificial para la creación de los enemigos y la técnica estadística para el análisis de los resultados. Por otro lado, se explicarán las etapas del proyecto, así como la descripción de cada fase de la metodología aplicada. Además, se hará una explicación detallada del prototipo desarrollado. También se explicará la implementación del tipo de realidad virtual utilizada, de la base de datos para el videojuego, de la técnica de Inteligencia Artificial para implementar un agente inteligente que persiga al personaje en primera persona controlado por el estudiante, del servicio de hosting que fue utilizado y las técnicas de aprendizaje que el videojuego contiene para que los estudiantes puedan ejercitar su nivel de comprensión lectora. Seguidamente, se presentará la metodología para la medición de los resultados con respecto al videojuego construido, con respecto a los experimentos desarrollados para seleccionar al mejor agente inteligente y también se detallará la medición realizada con respecto al entrenamiento de los estudiantes tanto para el análisis cuantitativo y cualitativo. Finalmente, el presente capítulo concluirá con la presentación del cronograma de actividades para la realización del presente proyecto de investigación.

3.1 Diseño de Investigación

En este apartado, se explicará el diseño, tipo de investigación, enfoque del proyecto, la población y muestra utilizada y finalmente, la operacionalización de las variables con las cuales se trabajó.

3.1.1 Diseño

El diseño de la investigación es de tipo cuasi experimental debido a que los grupos de estudiantes con el cual se trabajó no fueron establecidos al azar, sino que estos ya habían sido definidos desde un comienzo por el propio colegio (Moreno, Vahos, y Mazo, 2019). La idea es aprovechar característica propia de los colegios de dividir a sus alumnos de cada grado en grupos diferentes creando secciones para poder trabajar con estos grupos divididos.

La idea de utilizar secciones pertenecientes a un mismo grado fue poder comparar 2 o más grupos de estudiantes que posean características similares como la edad, nivel socioeconómico, entre otros.

Uno de los principales beneficios de comparar a las secciones de un mismo grado de un colegio es que tradicionalmente los alumnos son clasificados en secciones en función a las notas que poseen dentro de sus cursos manteniendo a alumnos con una nota aceptable en la primera sección, los que tienen una nota regular los colocan en una siguiente sección y así sucesivamente en función a la cantidad total de alumnos que posea una sección (Layona, et al., 2018).

Por otro lado, la presente investigación es transversal debido a que se evaluó a 12 grupos independientes de alumnos pertenecientes a 2 salones diferentes por cada uno de los 6 grados que posee un nivel primario en un mismo intervalo de tiempo (Layona, et al., 2018).

3.1.2 Tipo

La propuesta es de tipo tecnológico dado que tal cual como se desarrolló Huerta (2017), Moreno, Vahos, y Mazo (2019) y Castro (2015), se utilizaron las tecnologías de realidad virtual no inmersiva, base de datos y servicio de hosting para poder crear un producto tecnologico que sea funcional y adaptable para los participantes del proyectos. Adicionalmente, se realizó la implementación de la técnica de Inteligencia Artificial conocida como *Deep Reinforcement Learning*, tal cual como se aplicó en el trabajos de investigación de Expósito (2018), Ropilo (2019) y Adamsson (2018) con el objetivo de poder implementar un agente inteligente que sea el enemigo de los usuarios y pueda quitarles vida consiguiendo un mayor grado de realismo dentro del videojuego.

3.1.3 Enfoque

Se optó por usar un enfoque cuantitativo-cualitativo o conocido también como un enfoque mixto (Geronazzo et al., 2019). En primer lugar, se utilizó un enfoque cuantitativo dado que se recopilaron las notas de los pretest y postest de cada uno de los grupos participantes por cada grado y a través de una técnica estadística se verificó que el grupo que forme parte del experimento haya mejorado su nivel de comprensión lectora en un mayor nivel que el grupo que sigue una metodología tradicional (Didehbani et al., 2016). Este proceso se repitió para cada uno de los 6 grados pertenecientes a una educación primaria.

Con respecto al análisis cualitativo, se realizó una encuesta de satisfacción a todos los grupos que formen parte del experimento para determinar si el videojuego que se desarrolló fue de su agrado. Además, los participantes dieron su opinión sobre qué aspectos del videojuego poder mejorar para que este pueda volverse un videojuego más aceptable por parte de los participantes.

3.1.4 Población y muestra

Para seleccionar una muestra para el presente trabajo de investigación, se visitaron alrededor de 6 diferentes colegios particulares de Lima en noviembre del 2019 para proponerles la realización de este proyecto en sus salones de estudio con sus alumnos. El colegio que aceptó la realización del videojuego para utilizarse dentro de las aulas de clase fue la I.E.P Colegio “Mater Christi” ubicado en el distrito de Pueblo Libre. Este colegio forma a estudiantes de educación primaria de 1ro a 6to grado que poseen dos secciones por cada grado que son la sección “A” y sección “B”.

El organigrama de este colegio se muestra en el Anexo 1. Además, según las entrevistas con los profesores de lectura, la metodología tradicional que aplicaban antes de la pandemia del Covid-19 era que el docente antes de iniciar una lectura, hacía preguntas relacionadas al tema que abordaba el texto a leer. El material que se suele utilizar para la enseñanza de la comprensión lectora son los libros Santillana donde hay diversas lecturas para poder ejercitar el nivel de comprensión lectora, así como actividades para resolver. Un ejemplo del material educativo utilizado se muestra en el Anexo 2.

Los alumnos dan su opinión con respecto al tema abordado y posteriormente el docente inicia la lectura y continuaba un alumno al azar que salía al frente del salón y leía la lectura a sus compañeros tal cual como se muestra en el Anexo 3. Finalmente, se resolvían unas actividades referentes a la lectura, tal cual como se muestra en el Anexo 4, para poder reforzar los conocimientos que el texto quería transmitir.

No obstante, con la aparición del nuevo coronavirus, los colegios tuvieron que modificar sus metodologías de enseñanza. El colegio “Mater Christi” cambió su forma de enseñar de manera presencial a una virtual. Las plataformas que se utilizan para las clases virtuales son Zoom y WhatsApp para poder contactarse con los alumnos por llamada y preguntarles si tuvieron alguna duda sobre las tareas encomendadas para desarrollarlas en casa.

Las clases de comprensión lectora consta de 2 clases a la semana de 45 minutos cada una para todos los grados. En la primera parte de la clase se enseña a los alumnos la teoría necesaria con respecto a la comprensión lectora. En la segunda parte de la clase, se suele practicar con los propios materiales que usa la institución.

Antes de presentar el total de participantes, es preciso resaltar que con la aparición del Covid-19, la cantidad de alumnos totales disminuyó casi en 50% dado que antes las poblaciones totales según los profesores eran de 200 a 220 alumnos. Si bien es cierto que las clases se

vuelven más personalizables para el aprendizaje con una menor cantidad de alumnos, la deserción de alumnos ha traído consecuencias económicas tanto en este colegio en particular como en la mayoría de los colegios de lima.

La muestra utilizada fue de 112 alumnos siendo esta de tipo censal (Cheng y Tsaib, 2019) con un rango de edad de los estudiantes de 6 a 11 años pertenecientes al nivel socioeconómico A y B y divididos en 2 salones por cada grado. En este trabajo de investigación se dividió a cada grado en grupo de control y experimental. Dado que, según los profesores, los alumnos de la sección “A” suelen tener mejores notas en general que los alumnos de la sección “B” por lo que se optó para que los grupos de control sean cada sección “A” de cada grado y cada grupo experimental sea el pertenecientes a la sección “B”. Todos los grupos de control siguieron la metodología tradicional de clases mientras que los grupos experimentales utilizaron la nueva metodología en donde se combinó la metodología tradicional de clases y el uso del videojuego. Los grupos participantes se muestran en la Tabla 9.

Tabla 9: Población y muestra del proyecto de investigación

Grado	Sección A		Sección B	
	Grupo de control	Grupo experimental	Total	Rango de edad
1°	11	9	20	6-7 años
2°	8	10	18	7-8 años
3°	9	11	20	8-9 años
4°	7	9	16	9-10 años
5°	9	10	19	10-11 años
6°	10	9	19	11-12 años
Total	54	58	112	

Fuente: Elaboración Propia.

3.1.4.1 Justificación de división en grupos de control y experimental

La división de la muestra en grupos de control y experimental parte de la necesidad de establecer un grupo con el cual comparar los resultados obtenidos mediante el uso del videojuego. Aquí radica la importancia del grupo de control. Un grupo de control es aquel en donde la variable independiente no afecta en el aprendizaje de los involucrados (GreeLane, 2020). Es decir, es un grupo en donde el videojuego desarrollado no se utiliza de tal modo que siguen simplemente las clases virtuales tal cual como el colegio lo dicta. La utilización de un grupo de control en donde no intervenga la variable independiente es importante dado que los

resultados obtenidos en este grupo se comparan con los resultados obtenidos del grupo experimental de tal modo que se pueda llegar a una conclusión que el uso del videojuego dentro de las clases de lectura es mejor que la metodología tradicional (GreeLane, 2020).

Los estudios experimentales en donde se utiliza un grupo de control sirven para evaluar el grado de eficacia de una intervención (Zurita et al., 2018). Estos tipos de estudios sirven tradicionalmente en la realización de ensayos clínicos de modo que estableciendo un grupo experimental y un grupo de control se puede determinar si un determinado tratamiento es eficaz a no realizarlo.

Para el presente trabajo de investigación, como ya se ha mencionado, lo que se busca es provechar los grupos ya divididos en los colegios por cada grado en secciones dado que el criterio para que el colegio ponga a un alumno en una sección u otra está en función a las notas que obtuvo. El flujo de trabajo para el aprendizaje de ambos grupos, se observa en la Figura N°43. Antes de la intervención en ambos grupos, se tomó una evaluación previa conocido como un pretest. Este examen es con el objetivo es para comparar las notas de ambos grupos y verificar si existe o no existe una diferencia significativa antes de la intervención. Posteriormente, se ejecutó el aprendizaje. Este para el grupo experimental se dió a través del videojuego en todos los grados aplicando a su vez el protocolo de aprendizaje de tal modo que haya un tiempo distribuido entre la parte teórica dictada y la parte práctica. El grupo de control fue el grupo que siguió la metodología tradicional de enseñanza con los materiales de estudio y métodos que los profesores suelen utilizar. Al final del periodo de entrenamiento de los alumnos, se realizó una nueva evaluación conocido como posttest. Esta última evaluación se tomó con el objetivo de determinar si existe una mejora significativa del grupo experimental con respectivo al grupo de control. Mediante los resultados obtenidos se pudo comprobar la hipótesis general planteada y se verificó si la intervención con el videojuego realmente es eficaz para el aprendizaje de la comprensión lectora de los alumnos.

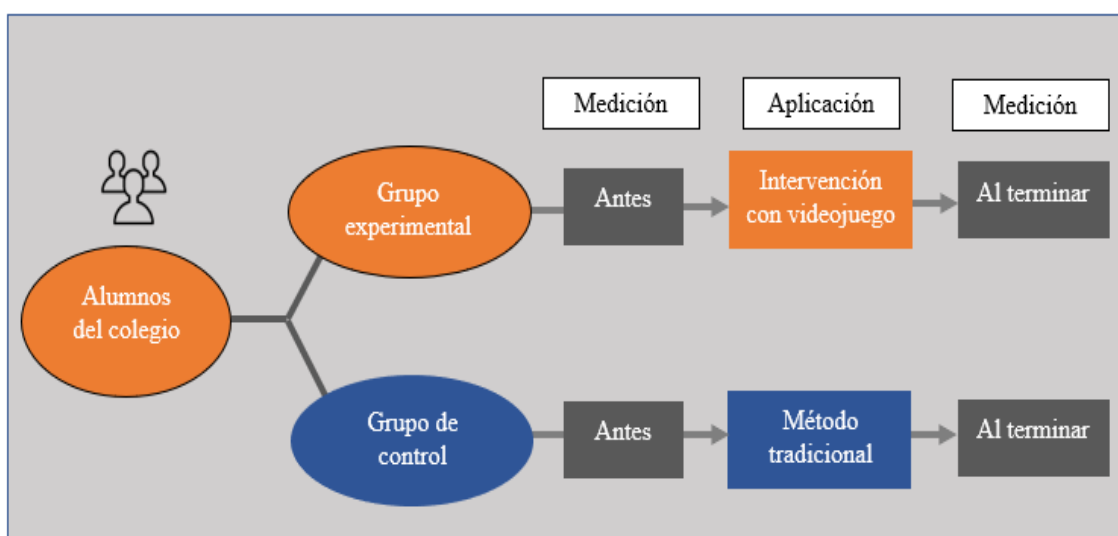


Figura N°43: Flujo de trabajo para los grupos de control y experimental.

Fuente: Zurita, J., Villasís, M., Miranda, G., y Márquez, H. (2018). Estudios experimentales: diseños de investigación para la evaluación de intervenciones en la clínica.

3.1.5 Operacionalización de las variables

Para la elaboración de los indicadores tanto para la variable dependiente e independiente, se utilizaron las propuestas de los trabajos desarrollados en Huerta (2017), Cheng y Tsaib (2019) y en Moreno, Vahos, y Mazo (2019). El cuadro de operacionalización para la variable dependiente y para la independiente se presentan en la Tabla 10. En dicha tabla se visualiza el nombre de la variable a analizar, el tipo de variable, la dimensión de la variable, los indicadores asociados a cada variable, las fórmulas para detectar el porcentaje de nivel literal, inferencial y crítico así como el porcentaje de aceptación del videojuego, las escalas de medición y los instrumentos utilizados para medir cada variable.

Tabla 10: Operacionalización de la variable dependiente e independiente

Nombre	Tipo	Dimensión	Indicador	Formula	Item	Escala de medición	Instrumento
Nivel de comprensión lectora de estudiantes	Variable dependiente	Nivel literal	Reconoce personajes del texto	$\frac{N^{\circ} \text{ de respuestas correctas}}{\text{Total de preguntas}} \times 100\%$	1,6	Nominal	Pretest y Postest
			Ordena secuencia de hechos en el texto		2,7		
		Nivel inferencial	Analiza hechos presentes en el texto	$\frac{N^{\circ} \text{ de respuestas correctas}}{\text{Total de preguntas}} \times 100\%$	3,8		
			Interpreta información expuesta en el texto o reacciones de los personajes		4,9		
		Nivel crítico	Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	$\frac{N^{\circ} \text{ de respuestas correctas}}{\text{Total de preguntas}} \times 100\%$	5,10	Nominal	
Uso de videojuego desarrollado	Variable independiente	Satisfacción de uso de videojuego	Experiencia de uso de videojuego	$\frac{N^{\circ} \text{ de respuestas por opción}}{\text{Total de participantes}} \times 100\%$	1	Escala de Likert (Buena, Regular, Mala)	Encuesta de videojuego de Moreno, Vahos, y Mazo (2019)
				-	2,3,4	Preguntas abiertas	

Fuente: Elaboración Propia.

3.2 Instrumentos de medida

Se realizaron 3 evaluaciones para la medición de los resultados: 2 exámenes conocidos como pretest y posttest y una encuesta de satisfacción. El pretest fue tomado como un examen inicial antes de que se empiece el entrenamiento en ambos grupos por cada grado, una vez finalizado el entrenamiento se tomó un posttest con el objetivo de medir si hubo un cambio o no en el nivel de comprensión lectora por cada grado. Finalmente, se tomó una encuesta de satisfacción con el objetivo de determinar si el videojuego fue o no del agrado de los niños y que ellos puedan explicar qué aspectos de aquel se pueden mejorar para hacerlo más atractivo. El flujo general que se ha explicado anteriormente para la realización de las evaluaciones se presenta en la Figura N°44. En las secciones 3.2.1 y 3.2.2, se explicarán a detalle la estructura de estas evaluaciones utilizadas.

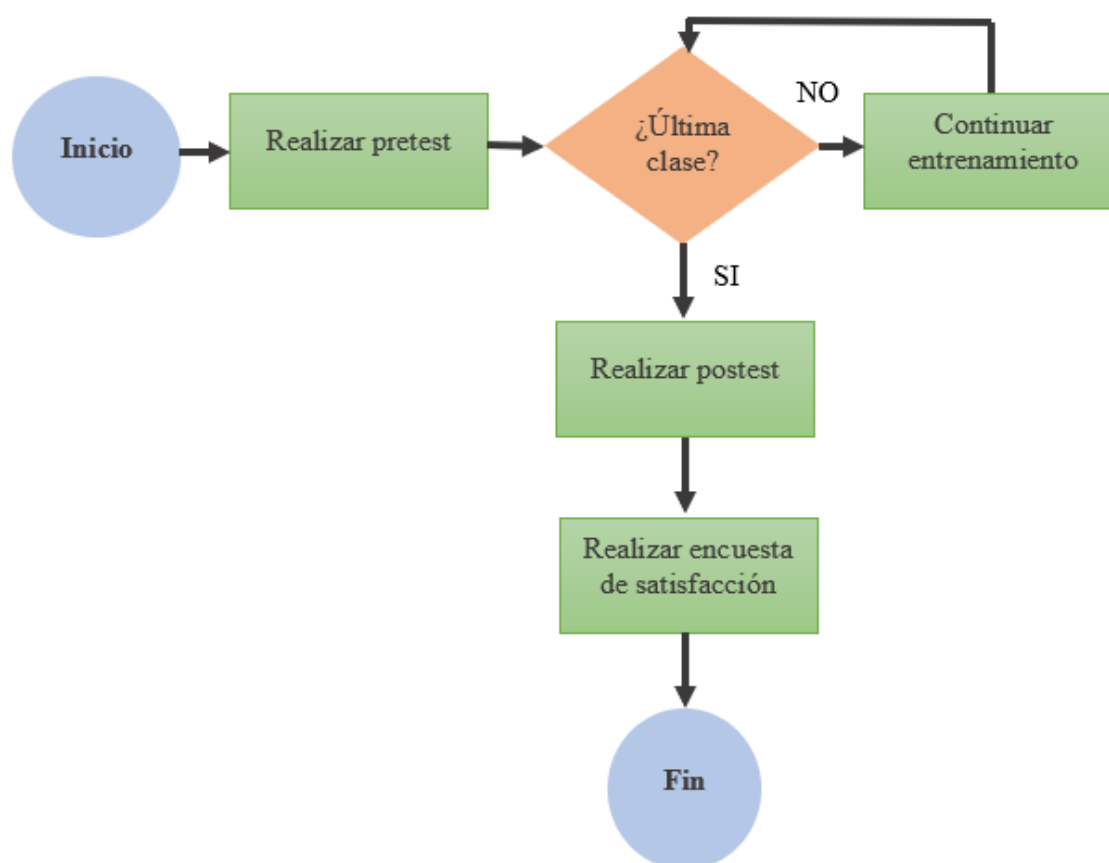


Figura N°44: Secuencia de actividades para implementar los instrumentos de medición.
Fuente: Elaboración Propia.

3.2.1 Pretest y posttest

Para medir los resultados y saber si existe un cambio significativo en las notas de comprensión lectora de los estudiantes de cada grado pertenecientes al grupo de control y grupo experimental, se tomaron dos test, uno antes de la intervención y uno después. Los test fueron

los mismos que se suelen tomar tradicionalmente en las clases. Estas 2 pruebas fueron elaboradas por el propio profesor y fueron tomadas tanto al grupo de control como al grupo experimental de cada grado. Los exámenes tuvieron lecturas con preguntas que midan los 3 niveles de la comprensión lectora y fueron tomados durante un mismo intervalo de tiempo para ambos grupos de cada grado. Los dos test tuvieron las siguientes características generales:

- Cada test tuvo 2 lecturas que fueron escogidas por el propio profesor y que son lecturas con el mismo formato que se les enseña en las clases.
- Cada lectura tuvo 5 preguntas. En total tanto el pretest como el postest se tuvo un total de 10 preguntas y la máxima nota fue de 20. Esto es debido a que se realizó el mismo proceso de evaluación de la propia clase de comprensión lectora.
- Con respecto a una lectura, esta tuvo 5 preguntas, como se había mencionado en el apartado anterior, de las cuales: 2 preguntas fueron para medir el nivel literal, 2 preguntas para medir el nivel inferencial y 1 final para medir el nivel crítico del alumno. En total, por cada test, se tuvieron 4 preguntas de nivel literal, 4 preguntas de nivel inferencial y 2 preguntas de nivel crítico.

En la Tabla 11, se presenta un resumen de la estructura del pretest y postest que se tomaron a los estudiantes por cada grado. Como se ha mencionado, los 2 test fueron elaborados y revisados por el propio docente de comprensión lectora ya que la metodología tradicional de evaluación debe de mantenerse igual para los salones de todos los grados.

Tabla 11: Estructura general de pretest y postest

Tipo de evaluación	N° Lecturas	Total de preguntas	Total de preguntas de nivel literal	Total de preguntas de nivel inferencial	Total de preguntas de nivel crítico
Pretest	2	10	4	4	2
Postest	2	10	4	4	2

Fuente: Elaboración Propia.

3.2.2 Encuesta de satisfacción

Con respecto a la variable independiente para la determinación si el videojuego fue o no del agrado de los estudiantes, se reutilizó la encuesta de satisfacción de videojuego elaborada y validada por Moreno, Vahos, y Mazo (2019). Esta encuesta fue tomada solamente a los grupos experimentales ya que solamente ellos jugaron el videojuego desarrollado. La encuesta de

satisfacción constó de 4 preguntas. La primera pregunta fue cerrada con una escala de Likert de 1 a 3. Las otras tres preguntas restantes fueron con el objetivo de saber el nivel de satisfacción que tuvieron los niños con el videojuego (Moreno, Vahos, y Mazo, 2019). La encuesta de satisfacción que tomada a los estudiantes, se presenta en el Anexo 5. Para validar la encuesta de videojuego, se aplicó un instrumento de validación en donde un experto en el área de la psicología brindó su aprobación para poder utilizar dicha encuesta. El instrumento de validación aplicado se presenta en el Anexo 6.

3.3 Técnica de recolección de datos

En este apartado, se explicará la técnica que sirvió para recolectar la información necesaria para el desarrollo del videojuego.

3.3.1 Entrevista

Se utilizó principalmente una entrevista con el objetivo de hacer un levantamiento de los requerimientos del profesor. En esta entrevista, no solamente se escucharon los requerimientos del profesor de comprensión lectora, sino que también se expuso los conocimientos que se posee en el desarrollo de videojuegos con el objetivo de poder desarrollar un videojuego que se encuentre de las capacidades que uno tiene como desarrollador ya que solamente una persona fue la encargada de desarrollar el videojuego. De esta forma, se buscó llegar a un acuerdo para poder crear el videojuego de la mejor manera posible, sabiendo que se cuenta con el conocimiento y recursos necesarios para no tener inconvenientes en el proceso de desarrollo.

3.4 Criterio de selección de la metodología para el desarrollo del videojuego

En este apartado se explicará el proceso que se siguió para la selección de la metodología adecuada para el desarrollo del videojuego. El método utilizado es el propuesto en Méndez (2006). La selección de la metodología se compone de 3 pasos: el primero es establecer un conjunto de metodologías para el desarrollo del producto que se desea crear (en este caso es un producto de software), el segundo paso consiste en la selección de las metodologías que se suelen aplicar especialmente para el producto en particular que se desea crear. En ese segundo paso, intervendrán las metodologías que se han utilizado en los trabajos de investigación explicados en el capítulo anterior. Con esto se consiguió reducir la población de metodologías a un tamaño que se pueda comparar. Finalmente, se utilizó la matriz propuesta en el trabajo de Méndez (2006) y que se ha descrito en el capítulo anterior para realizar la comparación de metodologías. Los 3 pasos para la selección de la metodología se detallaran en las siguientes sub secciones.

3.4.1 Población de metodologías de desarrollo de software

Para establecer una población de metodologías, en Méndez (2006) se desarrollo una encuesta para que sean llenados por equipos de desarrollo de software de tal modo que se pueda identificar qué metodología son las más utilizadas para desarrollar productos de software. También se les consultó a estos equipos de desarrollo cuáles son los principales criterios que suelen seguir para poder seleccionar una metodología para desarrollar un producto de software. Estos criterios son los que se explicaron anteriormente en el Capítulo II cuando se describió la matriz de evaluación de metodologías. Los resultados demostraron que las metodologías que más se suelen utilizar para el desarrollo de software son 6 principalmente: Desarrollo adaptativo de software (DAS), Metodología en cascada, Modelo incremental, Modelo en espiral, Programación extrema y la Metodología SCRUM.

3.4.2 Metodologías utilizadas en papers de videojuegos

Una vez establecida la población de metodologías, lo que se sugiere en Méndez (2006) es seleccionar las metodologías que más se ajusten al producto de software que se va a realizar. Para realizar este paso, se siguieron las metodologías que más se suelen utilizar en los trabajos de investigación planteados en el Capítulo II específicamente en la Tabla 5 que se realizó en el capítulo ya mencionado. En dicha tabla se realizó un resumen de las metodologías más utilizadas en el desarrollo de videojuegos. Las metodologías tradicionales de software que más se han utilizado en los trabajos de investigación descritos anteriormente son principalmente la metodología en Cascada y SCRUM. Como se mencionó en el capítulo anterior en la Tabla 5 también existen metodologías que se conocen como mixtas en donde no solamente se construye el videojuego sino también, dado que se utilizada realida virtual semi inmersiva y requiere la utilización de componentes que los niños utilizarán como cascos de realidad virtual, suelen involucrarse también profesores, psicólogos y hasta doctores de tal modo que tanto los desarrolladores y especialistas de la educación se agrupan de tal modo que el videojuego y los cascos de realidad virtual puedan ser utilizados de la manera más apropiada posible. Como en este proyecto, solo contó con un desarrollador y el tipo de realidad virtual fue la no inmersiva (la más básica), no se utilizaron estas metodologías mixtas por lo que solamente se optó por usar las 2 metodologías más utilizadas en el desarrollo de videojuegos tradicionales.

3.4.3 Resultados de matriz de evaluación de metodologías

Una vez que ya se tienen en las metodologías que se van a comparar, se procedió a aplicar la matriz descrita en el capítulo anterior. Los resultados de la matriz se presentan en la Figura N°45.

<i>Factor clave de éxito</i>	<i>Ponderación</i>	Metodología en Cascada		Metodología SCRUM	
		<i>Puntuación</i>	<i>Resultado Ponderado</i>	<i>Puntuación</i>	<i>Resultado Ponderado</i>
La metodología debe ajustarse a los objetivos	0.080	1	0.08	1	0.08
La metodología debe cubrir el ciclo entero del desarrollo de software	0.086	2	0.172	2	0.172
La metodología debe integrar las distintas fases del ciclo de desarrollo	0.086	3	0.258	4	0.344
La metodología debe incluir la realización de validaciones	0.080	2	0.16	1	0.08
La metodología debe soportar la determinación de la exactitud del sistema a través del ciclo de desarrollo	0.075	2	0.15	1	0.075
La metodología debe ser la base de una comunicación efectiva	0.078	2	0.156	2	0.156
La metodología debe funcionar en un entorno dinámico orientado al usuario	0.064	3	0.192	3	0.192
La metodología debe especificar claramente los responsables de resultados	0.078	3	0.234	3	0.234
La metodología debe emplearse en un entorno amplio de proyectos de software	0.083	1	0.083	2	0.166
La metodología se debe de poder enseñar	0.075	4	0.3	2	0.15
La metodología debe estar soportada por herramientas CASE	0.070	1	0.07	1	0.07
La metodología debe soportar la eventual evolución del sistema	0.075	1	0.075	2	0.15
La metodología debe contener actividades conducentes a mejorar el proceso de desarrollo de software	0.072	1	0.072	1	0.072
TOTAL			2.002		1.941

Figura N°45: Resultados de la matriz de evaluación de metodologías.
Fuente: Elaboración propia.

Las puntuaciones que se les colocaron a cada metodología del cuadro de la matriz corresponden a las puntuaciones que se sugiere en Méndez (2006) para la metodología en cascada y para la metodología SCRUM y a través de los objetivos propios que persigue el presente trabajo así como los recursos con los que se cuenta de tal modo se han colocado las puntuaciones de la forma más objetiva posible. Los resultados demuestran que la metodología que mejor se adapta a este proyecto en particular es la metodología en cascada y es debido a que solamente una persona es la encargada de desarrollar y gestionar el proyecto. Además, el cliente no tuvo un grado de involucramiento elevado, sino que este ayudó brindando sus requerimientos y verificando que lo que se haya construido es igual a lo que se solicitó. Una de las ventajas de poder utilizar la metodología tradicional en cascada es que se pudo obtener una mayor documentación de todo el proceso del videojuego. Esto sirve para poder detectar los errores de una manera más rápida.

3.5 Criterio de selección del tipo de realidad virtual a utilizar

Para seleccionar el tipo de realidad virtual para realizar el videojuego, se hizo un resumen de cada tipo de realidad virtual utilizada en cada uno de los trabajos de investigación descritos en el Capítulo II en donde se desarrolló un videojuego para el aprendizaje de un determinado tema o mejora de una habilidad. Este resumen se presenta en la Tabla 12.

Tabla 12: Resumen de tipo de realidad virtual aplicada en los trabajos de investigación

Autor	Tipo de realidad virtual
Didehbani et al. (2016)	No inmersiva
Cheng y Tsaib (2019)	Semi-inmersiva
Layona, Yulianto, y Tunardi (2018)	No inmersiva
Moreno, Vahos, y Mazo (2019)	No inmersiva
Castro (2015)	No inmersiva

Fuente: Elaboración Propia.

Como se puede apreciar, los tipos de realidad virtual que más resaltan de la Tabla 12 es la realidad virtual no inmersiva y semi-inmersiva. De la definición de realidad virtual realizado en el capítulo anterior, la realidad semi-inmersiva se aplica cuando se utilizan cascos de realidad virtual dado que tradicionalmente tienen dos lentes. No obstante, dado que este trabajo fue desarrollado bajo una situación de emergencia sanitaria y los estudiantes están realizando sus actividades desde sus casas, implementar este tipo de realidad virtual en primer lugar es muy costoso dado que se tendrían que adquirir cascos de realidad virtual para cada niño que lo utilizaría y en segundo lugar no sería práctico dado que se tendría que dejar el casco de realidad virtual en cada casa de los participantes. De este modo, lo que se ha decidido fue implementar el tipo de realidad virtual no inmersiva que es la más común en donde el videojuego se visualiza en una sola pantalla y se controla a través de un mouse, teclado o ambos. De este modo, se buscó que todos los estudiantes pudieran tener acceso al videojuego por igual y que puedan controlarlo desde sus propias casas con las computadoras que ya poseen.

3.6 Criterio de selección de base de datos a utilizar

Para seleccionar la base de datos a utilizar dentro del videojuego, se utilizó el trabajo realizado en Ohyver et al., (2019). En este trabajo se compararon las bases de datos de MySQL Database y Firebase Database con el objetivo de determinar cuál de las dos bases de datos tienen un mejor rendimiento al momento de ser implementadas y la comparación específicamente en con respecto al tiempo de respuesta (Ohyver et al., 2019).

La técnica utilizada para analizar el tiempo de respuesta de ambas bases de datos fue el test de *Wilcoxon Signed-Rank*. Este test es una prueba no paramétrica que se utiliza para comparar pares de observaciones que provienen de dos poblaciones (Ohyver et al., 2019). La hipótesis nula es que la diferencia de ambas medias es cero. Por otro lado, la hipótesis alternativa es que la diferencia de ambas medias no es cero.

La data que se utilizó fue de información sobre la nutrición de niños en donde se tenían los campos del peso del niño, la edad, la puntuación de su nutrición y su id. La idea fue crear dos bases de datos independientes, pero con la misma información. Con respecto a la base de datos de MySQL se creó una tabla tradicional con filas y columnas y en el caso de Firebase Database se registró la información a través de ficheros en formato json.

Para cada base de datos creada, se realizaron las 4 diferentes operaciones conocidas como CRUD de Create, Read, Update, Delete. En la Tabla 13, se puede observar la descripción de cada operación y su aplicación tanto en MySQL como en Firebase.

Tabla 13: Operaciones aplicadas a una base de datos

CREATE	READ	UPDATE	DELETE
Se mide el tiempo que se demora cada base de datos en registrar un nuevo dato.	Permite medir cuánto tiempo demora una base de datos en devolver un determinado valor.	Mide el tiempo que se demora la base de datos en corregir un dato en particular.	Permite medir el tiempo que se demora cada base de datos en eliminar un dato.
Uso en MySQL: INSERT	Uso en MySQL: SELECT	Uso en MySQL: SELECT + UPDATE	Uso en MySQL: DELETE
Uso en Firebase: <i>SetValue()</i> .	Uso en Firebase: <i>ValueEventListener()</i> .	Uso en Firebase: <i>SetValue()</i> .	Uso en Firebase: <i>removevalue()</i> .

Fuente: Elaboración Propia.

Para la realización de los experimentos, se trabajó con cada una de las operaciones en donde se demostró que en cada operación Firebase presentó mayor superioridad que MySQL. Mediante los resultados obtenidos en Ohyver et al., (2019), se decidió que, para este proyecto de videojuego, se implemente Firebase Database dado que, como se mencionó al inicio, lo que se desea es que los alumnos puedan registrarse y loguearse para acceder al videojuego. Lo que se necesita es una base de datos rápida para poder almacenar la información dado que, si un alumno se registra dentro del videojuego, pero si no se llega a almacenar rápidamente dentro de la base de datos esto puede ocasionar que el alumno no pueda ingresar a tiempo en el videojuego originando una mala experiencia de usuario por lo que Firebase Database con su base de datos en tiempo real puede ayudar a minimizar este error dentro del videojuego desarrollado.

3.7 Criterio de selección de técnica de Inteligencia Artificial

Para poder darle al videojuego un mayor grado de realismo, se desarrolló a los enemigos de tal forma que puedan tener un comportamiento inteligente por lo que se deseó introducir Inteligencia Artificial al videojuego. Para realizar dicho objetivo, se hizo un resumen de los trabajos de Inteligencia Artificial desarrollados en el Capítulo II. En la Tabla 14, se presenta un resumen de los trabajos realizados anteriormente de tal modo que se pueda tener una noción de qué técnica de Inteligencia Artificial utilizar para crear a los enemigos del videojuego.

Tabla 14: Resumen de técnica de Inteligencia Artificial aplicada en los trabajos de investigación

Autor	Motor gráfico	Técnica de IA	Librería de IA	Algoritmo de optimización
Expósito (2018)	Unity	DRL	ML-Agents	PPO
Ropilo (2019)	Unity	DRL	ML-Agents	PPO
Adamsson (2018)	Unity	DRL	ML-Agents	PPO

Nota. IA = Inteligencia Artificial; DRL = Deep Reinforcement Learning; PPO = Proximal Policy Optimization.

Fuente: Elaboración Propia.

Como se observa en la Tabla 14, la plataforma de videojuegos que se suele utilizar para crear a una Inteligencia Artificial es Unity, la técnica que se aplica es *Deep Reinforcement Learning* o conocido también como Aprendizaje por refuerzo profundo, la librería de Machine Learning que se utiliza es ML-Agents y el algoritmo utilizado para optimizar el entrenamiento es el algoritmo de *Proximal Policy Optimization*.

3.7.1 Selección del tipo de red neuronal

Para seleccionar la red adecuada se han utilizado dos criterios. En primer lugar, el tipo de arquitectura de red utilizado en los trabajos de investigación desarrollados en el Capítulo II y en segundo lugar el tipo de red que se suelen utilizar tradicionalmente en los juegos de mercado. Para presentar el primer criterio, se presenta en la Tabla 15 un resumen de los trabajos de investigación desarrollados en donde se implementa una Inteligencia Artificial dentro del videojuego.

Tabla 15: Resumen de tipo de arquitectura de red aplicada en los trabajos de investigación

Autor	Tipo de red neuronal
Expósito (2018)	MLP
Ropilo (2019)	MLP
Adamsson (2018)	MLP

Nota. MLP = Multilayer Perceptrón.

Fuente: Elaboración Propia.

Como se puede apreciar en la Tabla 15, el tipo de red neuronal que se suele aplicar para el entrenamiento de un agente inteligente es el Multilayer Perceptrón. Este tipo de red neuronal, así como los procesos que involucra, se han explicado ya en el Capítulo II.

Para el segundo criterio, se ha elaborado la Tabla 16 basado en la investigación realizada en Risi y Togelius (2015) en donde se realiza un estado del arte sobre el avance de la Inteligencia Artificial aplicada a videojuegos. En la Tabla 16 se observa a los distintos videojuegos más populares del mercado como Pac-Man, GO y Super Mario Bros. El tipo de arquitectura de red utilizado por cada videojuego se presenta en la última columna de la tabla. Se puede apreciar que se utiliza una MLP y un perceptrón en la mayoría de videojuegos populares. Dado que en los dos criterios descritos anteriormente, se visualiza que se utiliza una MLP para el entrenamiento de la Inteligencia Artificial se ha decidido aplicar esta arquitectura de red para este videojuego en particular de tal modo que se pueda entrenar a una Inteligencia Artificial que persiga al jugador.

Tabla 16: Arquitecturas de red utilizados en videojuegos tradicionales de mercado.

Tipo	Nombre de Videojuego	Arquitectura de Red
Videojuegos de evaluación de estados o acciones	GO (7X7)	MLP
	Ms. Pac-Man	MLP
	Chess	MLP
Videojuegos de acción directa	Car Racing	MLP
	Battle Domain	MLP
	Keepaway Soccer	MLP
Videojuegos de selección de estrategias	EvoCommander	MLP
Videojuegos de aventura	Super Mario Bros	MLP, Perceptrón

Nota. MLP = Multilayer Perceptrón

Fuente: Risi, S., y Togelius, J. (2015). Neuroevolution in Games: State of Art and Open Challenges.

3.8 Criterio de selección de técnica estadística para el análisis de resultados

Para seleccionar la técnica estadística a utilizar para el análisis de la variable comprensión lectora, se realizó un resumen de las técnicas estadísticas utilizadas en los antecedentes de investigación el cual se presenta en la Tabla 17.

Tabla 17: Resumen de técnica estadística utilizada en los trabajos de investigación

Autor	Evaluación	Análisis estadístico
Didehbani et al. (2016)	Pretest y Postest	Prueba T de student
Cheng y Tsaib (2019)	Pretest y Postest	Prueba T de student
Layona, Yulianto, y Tunardi (2018)	Pretest y Postest	Prueba T de student
Moreno, Vahos, y Mazo (2019)	Pretest y Postest	Prueba T de student
Castro (2015)	Pretest y Postest	Prueba T de student

Fuente: Elaboración Propia.

Como se puede apreciar en la Tabla 17, la prueba que se suele utilizar para evaluar los resultados cuantitativos es la prueba T de student aplicados a los pretest y postest. Ahora, existen 2 tipos de prueba T y la elección del tipo de prueba dependerá de la cantidad de grupos con los que se va a trabajar y/o comparar (Soto, 2018). La prueba T para muestras dependientes es utilizando un solo grupo y revisando sus resultados antes y después. Por otro lado, existe la prueba T para muestras independientes que se aplica a muestras en donde a dos grupos se evalúan en un mismo intervalo de tiempo (Soto, 2018). Para la elección del tipo de prueba T a utilizar, se utilizó la matriz para la elección de pruebas estadísticas utilizada en Soto (2018) la cual se presenta en la Figura N°46.

		PRUEBAS NO PARAMÉTRICAS			PRUEBAS PARAMÉTRICAS
Variable aleatoria		NOMINAL DICOTÓMICA	NOMINAL POLITÓMICA	ORDINAL	NUMÉRICA
Variable Fija					
Estudio Transversal Muestras independientes	Un grupo	X^2 Bondad de ajuste Binomial	X^2 Bondad de ajuste	X^2 Bondad de ajuste	T de Student para una muestra
	Dos grupos	X^2 de Homogeneidad, Corrección de Yates, Test Exacto de Fisher	X^2 de Homogeneidad	U Mann-Withney	T de Student para muestras independientes
	Mas de dos grupos	X^2 de Homogeneidad	X^2 de Homogeneidad	H Kruskai-Wallis	ANOVA con un factor intersujetos
Estudio Longitudinal Muestras relacionadas	Dos medidas	Mc Nemar	Q de Cochran	Wilcoxon	T de Student para muestras relacionadas
	Mas de dos medidas	Q de Cochran	Q de Cochran	FriedMan	ANOVA para medidas repetidas

Figura N°46: Selección de prueba estadística a utilizar.Fuente: Soto, J. (2018). *Estadística: Test de hipótesis*.

Tal como se mencionó al inicio del capítulo, el presente estudio es transversal dado que se evaluaron a 2 grupos pertenecientes a un mismo grado para cada uno de los 6 grados de educación primaria en un mismo intervalo de tiempo. Realizando el cruce respectivo de la Figura N°46 seleccionando de forma horizontal la fila correspondiente de estudio transversal con la última columna de variable numérica dado que la variable a evaluar de comprensión lectora es numérica se obtiene como prueba estadística la Prueba T para muestras independientes.

Dado que ya se tienen definidos los principales aspectos del proyecto a realizar como los objetivos, hipótesis y el diseño de la investigación así como las tecnologías y la técnica estadística para el análisis de la información, en el Anexo 7 se presenta la matriz de consistencia del presente trabajo de investigación.

3.9 Etapas del proyecto de investigación

El presente proyecto de investigación consistió en 4 fases siguiendo el trabajo de investigación desarrollado en Cheng y Tsaib (2019) presentado en la sección de antecedentes del capítulo anterior. Las 4 fases que tiene el trabajo de investigación son: el desarrollo propio del videojuego en función a los requerimientos del docente, la realización del protocolo de aprendizaje de tal modo se distribuya el tiempo de enseñanza de forma tradicional a través de los materiales y teorías que se suelen abordar normalmente y el uso del videojuego. La tercera etapa consistió en el entrenamiento propio de los estudiantes siguiendo el protocolo de aprendizaje establecido y finalmente la presentación de los resultados obtenidos luego del entrenamiento. La descripción general de lo que se realizó en cada etapa se presentará en la cuatro siguientes sub secciones.

3.9.1 Desarrollo del videojuego

Para el desarrollo del videojuego se utilizó la metodología en cascada como se definió anteriormente. El motor de base de datos que se utilizó para el desarrollo del videojuego es Unity con la versión 2019.4.0f1. Luego de instalar el software principal para el desarrollo del videojuego, se procedió a recopilar los requerimientos del cliente. La idea fue entender cuál es la necesidad del profesor con respecto al problema de la comprensión lectora para poder desarrollar un videojuego acorde a sus requerimientos.

Dado que actualmente el Perú se encuentra en una situación de estado de emergencia y no es posible tener un control total de los alumnos que están participando dentro del experimento,

se desarrolló una escena de Login y registro para que los alumnos registren sus datos e ingresen al videojuego a través de la base de datos de Firebase.

El objetivo de poder desarrollar la base de datos fue tener cierto control de los alumnos que están participando y poder comparar la lista oficial de los alumnos con los registrados dentro del videojuego con el objetivo de verificar que los alumnos realmente se encuentran jugando el videojuego.

Además, para el desarrollo del videojuego se tuvo contemplado introducir una técnica de Inteligencia Artificial con el objetivo de hacer del videojuego mucho más realista y desafiante. La técnica en especial que se utilizó se conoce como *Deep Reinforcement Learning* en donde se combinan las técnicas de redes neuronales y aprendizaje por refuerzo. La idea fue entrenar a un agente en un entorno de aprendizaje que simule el videojuego real de tal modo que aquel aprenda a seguir a un objetivo.

El videojuego que se desarrolló fue puesto en el servidor de Firebase dado que esta herramienta ofrece un servicio de hosting. La idea es poder aprovechar las funcionalidades que ofrece Firebase para el desarrollo del videojuego. Una vez que el videojuego se encuentre en los servidores de Firebase, se generó un link de tal modo que al ponerlo en un navegador pudo reproducirse el videojuego. El principal sustento por el cual el videojuego fue jugado de esta manera es que los estudiantes puedan jugar desde sus propias casas sin tener que salir para reunirlos en un salón o sala de cómputo del colegio. La idea fue poder adaptarse a esta nueva modalidad de enseñanza de educación a distancia utilizando las herramientas tecnológicas que se encuentren al alcance.

En el desarrollo del videojuego se tuvo contemplado la utilización de la realidad virtual no inmersiva el cual es el tipo más común que existe en los videojuegos y según la teoría para la aplicación de este tipo de realidad virtual es necesario que el videojuego se controle a través de periféricos. En este caso los periféricos utilizados fueron el propio teclado de las computadoras, así como el mouse. Se utilizaron estos periféricos dado que son los más comunes que poseen todos los computadores. Esto es más beneficioso que utilizar otros componentes como cascos de realidad virtual o equipos más avanzados dado que sería complicado buscar a cada niño para entregarle los equipos, así como que se requiere un manual para poder utilizarlos. Unity provee librerías propias ya configuradas en donde se agregan personajes en primera persona y se puede configurar para que se puedan mover con botones específicos del teclado.

Una vez finalizado el videojuego, se les envió el link a los profesores para que ellos mismos puedan dar su aprobación con respecto al videojuego. Luego de que los profesores revisaron el producto y verificaron que no existen errores se dio por terminado esta primera fase del proyecto. En la Figura N°47, se presenta el flujo de actividades para la aprobación del videojuego.

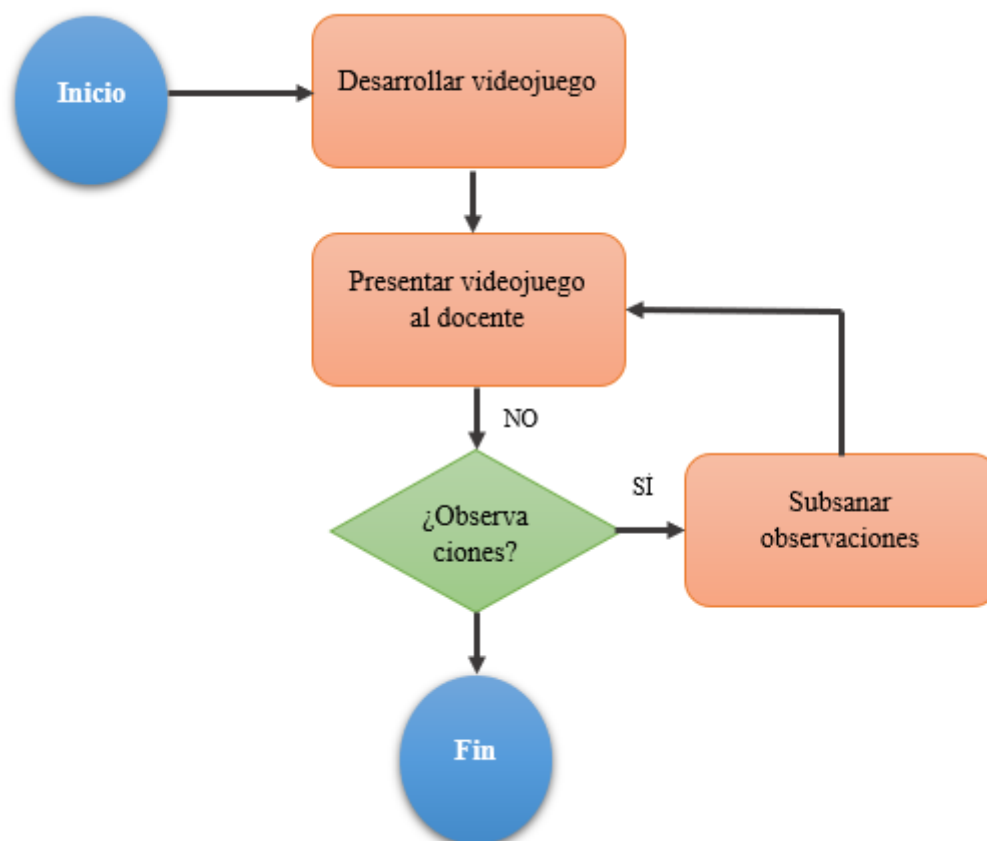


Figura N°47: Diagrama de flujo para el desarrollo de la primera etapa.
Fuente: Elaboración Propia.

En esta sección se ha presentado de forma general la primera etapa que consiste en el desarrollo propio del videojuego y de las principales acciones que se realizaron. No obstante, esta etapa será explicada con un mayor detalle en las secciones 3.10, 3.11, 3.12, 3.13, 3.14, 3.15 y 3.16 del presente capítulo. Además, se detallará sobre la implementación de las tecnologías que fueron aplicadas en el videojuego como realidad virtual no inmersiva, la técnica de *Deep Reinforcement Learning*, la base de datos en tiempo real de Firebase y el servicio de hosting. Finalmente, se explicará a detalle las técnicas de aprendizaje que se utilizaron dentro del videojuego para que los estudiantes puedan ejercitar su nivel de comprensión lectora de tal forma que se pueda tener una mejora significativa.

3.9.2 Protocolo de aprendizaje

Luego de haberse aprobado producto de software, se procedió a elaborar un procedimiento pedagógico y tecnológico en donde se detallaron las semanas a jugar el videojuego, así como el tiempo destinado al uso propio del videojuego y al tiempo utilizado para las clases tradicionales de comprensión lectora. Al procedimiento desarrollado se le conoce formalmente como protocolo de aprendizaje y el objetivo de desarrollarlo fue para que los alumnos puedan combinar la metodología tradicional de clases y el uso propio del videojuego de tal modo que se pueda visualizar la aplicación del aprendizaje basado en videojuegos.

Para realizar el protocolo de aprendizaje se tuvo en cuenta el tiempo con el cual disponen los profesores para que los niños puedan entrenar con el videojuego. Según los trabajos de investigación desarrollados en el anterior capítulo, protocolo de aprendizaje se suele desarrollar con un periodo no menor a 2 semanas. En este proyecto se buscó que el entrenamiento en lo posible pueda ser mayor a estas dos semanas. No obstante, el tiempo que se establezca depende mucho de la decisión de los propios profesores.

Una vez definidas las semanas del entrenamiento, se tomó el tiempo total que existe para una clase completa del curso de comprensión lectora y se estableció el tiempo a utilizar para las clases tradicionales y el tiempo para utilizar el videojuego. Es preciso mencionar que cada grado tiene horarios diferentes, así como días diferentes de enseñanza del curso por lo que, en este caso, también se tuvo contemplado la visualización de qué días son los que jugarán para cada grado.

En el protocolo de aprendizaje también se visualiza en qué fechas se desarrolló las evaluaciones del pretest y postest para cada uno de los grados de tal modo que se tuvo un horario completo antes de pasar a la siguiente etapa. La idea fue que se pueda aplicar en lo posible todo lo que estipula el protocolo de aprendizaje de tal modo que se puedan obtener los mejores resultados posibles dentro del trabajo de investigación.

Un resumen de las principales actividades que se desarrollaron en esta etapa se pueden visualizar en la Figura N°48.

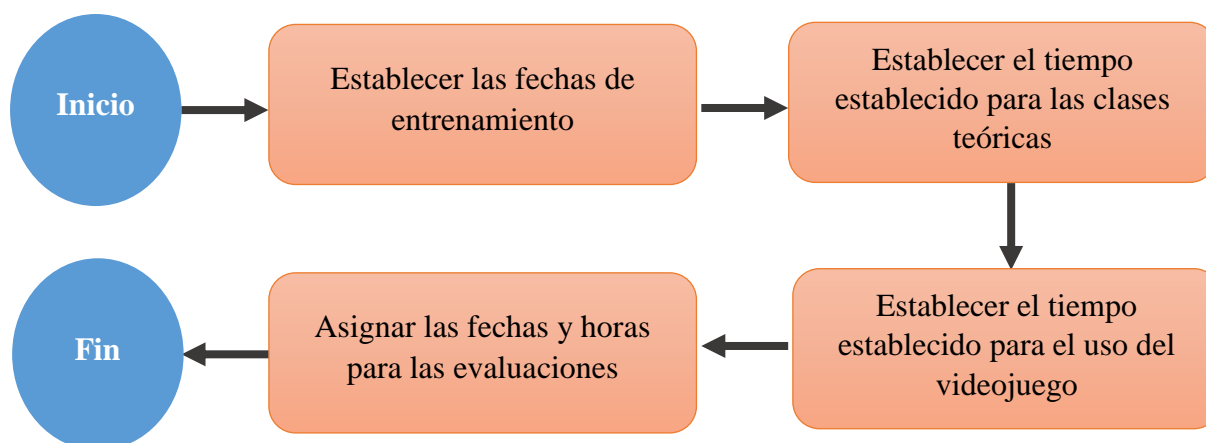


Figura N°48: Secuencia de actividades para el desarrollo del protocolo de aprendizaje.
Fuente: Elaboración Propia.

En esta sección, se ha dado una visión general de la segunda fase del proyecto de investigación. No obstante, esta fase se explicará más a detalle en la sección 3.17 del presente capítulo en donde se explicará la aplicación del aprendizaje basado en videojuego a través del desarrollo y utilización del protocolo de aprendizaje.

3.9.3 Entrenamiento de los estudiantes

Una vez construido el videojuego y establecido el protocolo de aprendizaje, el siguiente paso fue que los niños entrenen a través del videojuego respetando el protocolo de aprendizaje desarrollado. A diferencia del aprendizaje tradicional de enseñanza en donde el videojuego se utiliza en las salas de cómputo del colegio, esta vez los niños jugaron dentro de las clases virtuales.

Como se había mencionado anteriormente, la base de datos en tiempo real sirve para saber por lo menos que los alumnos hayan ingresado al videojuego dado que como una educación a distancia por motivos del Covid-19, no se tiene un control total de lo que los alumnos realizan en las clases virtuales.

Además, para verificar la asistencia de los alumnos en el periodo de entrenamiento, los profesores pasaron asistencia a los alumnos de manera tradicional y se verificó que todos los alumnos se encuentren asistiendo a las semanas establecidas para el periodo de entrenamiento.

De este modo, se hizo una supervisión diaria de la asistencia de los alumnos y se registraron las faltas por si es que hubiera. Al final se presentó el total de faltas que hubo por grado. La idea es que todos los niños asistan al periodo de entrenamiento tanto del grupo de control como

del grupo experimental de tal modo que se puedan obtener los mejores resultados posibles. El resumen del flujo de trabajo seguido en esta etapa se muestra en la Figura N°49.

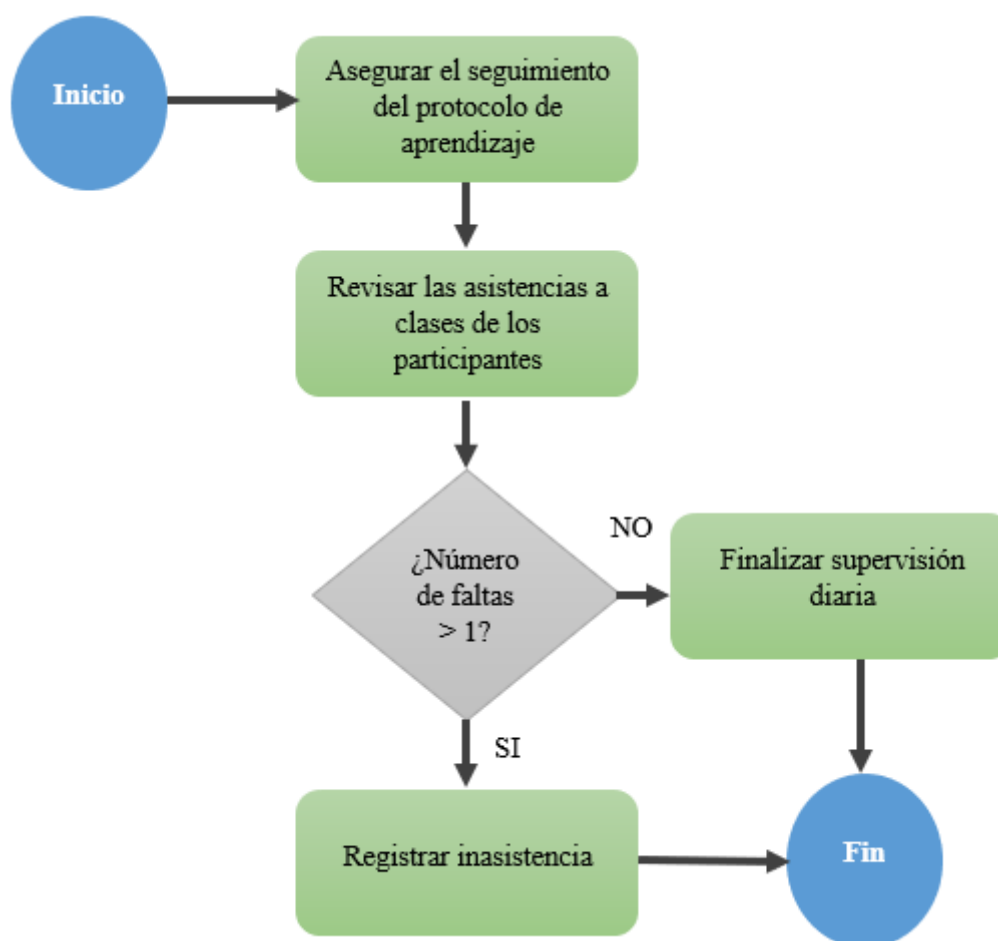


Figura N°49: Secuencia de actividades para el desarrollo del periodo de entrenamiento.
Fuente: Elaboración Propia.

3.9.4 Presentación de resultados

En esta última etapa se presentaron los resultados obtenidos con respecto al videojuego que incluye los resultados obtenidos en la metodología por cada etapa que se desarrolló y el resultado de las tecnologías aplicadas como la realidad virtual no inmersiva, la base de datos en tiempo real y el servicio de hosting.

Además, se presentaron los resultados obtenidos luego del entrenamiento de los agentes inteligentes dado que se reutilizaron los modelos de DRL planteados en el Capítulo II, así como se realizaron 4 propuestas de modelos de DRL en función a los valores propuestos por los autores con el objetivo de determinar qué modelo es el más adecuado. La idea fue poder seleccionar al mejor modelo con el objetivo de utilizar ese agente mejor entrenado dentro de los enemigos del videojuego.

Por último, se presentaron los resultados con respecto al entrenamiento de los alumnos a través del uso del videojuego combinado con sus clases tradicionales. En esta última parte del proyecto se recibió como input las notas del pretest y postest por cada grado. Además, se recibió los resultados de la encuesta de satisfacción para ver qué tanto les agradó a los estudiantes la introducción del videojuego en sus clases tradicionales de lectura. Debido a que se utilizó un análisis cuantitativo-cualitativo para hacer el análisis de los datos brindados, se procedió a dividir el análisis de los resultados en 3 partes para cada grado:

1. **Análisis cuantitativo descriptivo:** Este tipo de análisis permitió observar el promedio y desviación estándar de notas inicial de ambos grupos tanto del pretest como del postest. De igual modo, se presentaron los resultados de los indicadores de la variable de comprensión lectora para cada grado de primaria.
2. **Análisis cuantitativo inferencial:** En este análisis se sometió a una prueba de hipótesis tanto en el pretest como en el postest con el objetivo de saber si existe una diferencia significativa en el promedio de notas finales de los alumnos de ambos grupos (Didehbani et al., 2016). Es preciso mencionar que en el postest es donde se validó la hipótesis general (H.G) planteada al inicio del presente trabajo.
3. **Análisis cualitativo:** El primer y segundo análisis fue para medir la variable cuantitativa y usando el pretest y postest. En este último análisis, se explicaron las respuestas que los alumnos escribieron en la encuesta de satisfacción con respecto al uso del videojuego (Geronazzo et al., 2019). Se pudo observar qué tan agradable les pareció el videojuego para que ellos mejoren su nivel de comprensión lectora en función a sus respuestas que sirvieron para obtener un Feedback general con respecto al contenido que posee el videojuego.

En la Figura N°50, se presenta un resumen de las actividades desarrolladas para poder realizar el análisis respectivo a las notas de los alumnos.

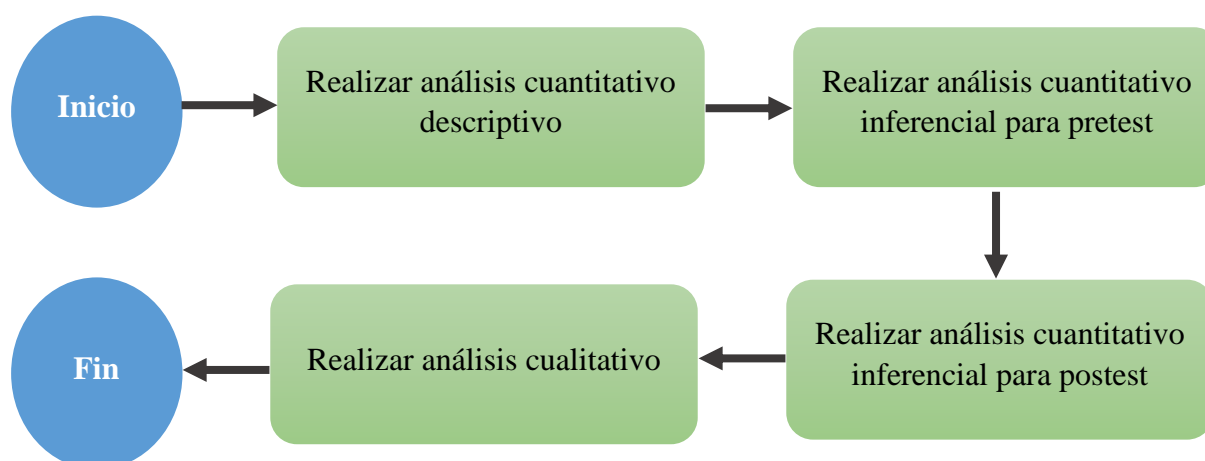


Figura N°50: Secuencia de actividades para el análisis de las evaluaciones a los estudiantes.
Fuente: Elaboración Propia.

En esta sección se ha descrito de manera general esta última etapa del presente proyecto de investigación. No obstante, esta etapa se explicará en mayor detalle en la sección 3.18 que corresponde a la medición de los resultados.

3.10 Metodología de implementación de la solución

Dado que anteriormente se realizaron los pasos necesarios para seleccionar la metodología correcta y según la matriz de evaluación metodológica resultó que el método en cascada era la metodología que más se adecuaba al presente trabajo de investigación, en esta sección se explicará a detalle cómo fue la aplicación de esta metodología.

3.10.1 Metodología en cascada

Para utilizar la metodología en cascada en este proyecto, se utilizó la misma secuencia de actividades para cada una de las 5 etapas planteada en el trabajo de investigación de Layona, Yulianto, y Tunardi (2018) desarrollado en la sección de antecedentes de investigación del capítulo anterior. En la Figura N°51 se puede visualizar las etapas que comprende la metodología en cascada, así como las principales actividades que se realiza en cada etapa. En las próximas sub secciones se detallarán las actividades realizadas para cada etapa de la metodología en cascada aplicada al prototipo.

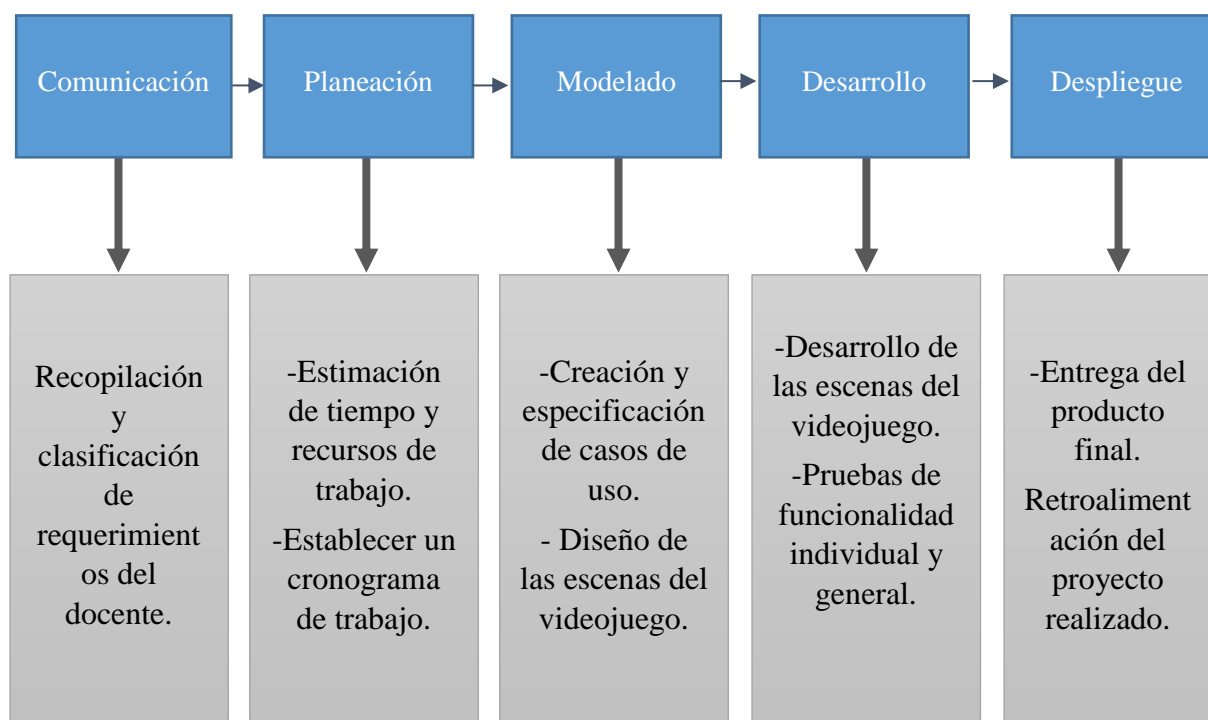


Figura N°51: Metodología en cascada aplicado al presente proyecto de investigación.

Fuente: Layona, R., Yulianto, B., y Tunardi, Y. (2018). *Web based Augmented Reality for Body Anatomy Learning*. *Computer Science*, 135,457-464.

3.10.1.1 Comunicación

La primera fase de la aplicación de la metodología en cascada consistió principalmente en la recopilación y clasificación de los requerimientos del cliente. Las actividades mencionadas serán descritas en las siguientes sub secciones.

3.10.1.1.1 Actividad 1: Recopilación de requerimientos del docente

Se inició el desarrollo del prototipo con la presentación ante el director en primer lugar y luego ante el docente de comprensión lectora. Se brindó una visión general de los objetivos que se buscaban mediante el videojuego desarrollado. El principal de estos fue la mejora de los estudiantes en su nivel de comprensión lectora. Una vez establecido lo que se desea trabajar y obtenida la confirmación de los profesores, se procedió a recopilar los requerimientos para el desarrollo del videojuego.

Además, en la reunión se pidió al docente los materiales con los cuales se suele trabajar de manera tradicional de tal modo que estos puedan estar presentes dentro del videojuego. La idea fue que los niños puedan visualizar estos mismos materiales de estudio a través del uso del videojuego y a su vez puedan responder las preguntas de las lecturas a través del prototipo.

Lo que se tuvo pensado como una lógica para poder implementar los textos al igual que las preguntas y que estos formen parte de la jugabilidad del videojuego es que los niños al perder en un determinado nivel del videojuego sean redirigidos a una lectura o varias lecturas del videojuego en función al grado que corresponda el alumno de tal modo que una vez que el estudiante se encuentre dentro de las lecturas, este pueda responderlas y si acierta todas pueda regresar al nivel en donde se encontró de lo contrario, tenga que iniciar el videojuego desde el inicio. Por este motivo, fue importante la obtención del material educativo por parte del docente, de preferencia el material se pidió en formato pdf para poder introducirse dentro del videojuego y que se pueda observar de una manera nítida por los estudiantes.

A su vez, se detalló al docente como parte de la explicación del proyecto, que los alumnos deberían de completar una encuesta de satisfacción con el objetivo de determinar si el videojuego que se desarrolló fue del agrado de los estudiantes. Esto fue básicamente para recopilar las opiniones de los estudiantes de cada grado con el objetivo de a futuro poder mejorar el videojuego desarrollado.

También se explicó al docente que al inicio y final el periodo de entrenamiento, el mismo debe de evaluar a sus alumnos mediante un examen tradicional de comprensión lectora. Estos dos exámenes que se desarrollaron para cada grado fueron revisados y calificados por el propio docente. Lo único que se recibió para el desarrollo del proyecto es un resumen de las notas obtenidas por los alumnos de tal modo que, a través del método estadístico definido anteriormente, se pudo verificar si existe o no un cambio significativo en el nivel de comprensión lectora de los estudiantes.

3.10.1.1.2 Actividad 2: Clasificación de requerimientos

Luego de la recolección de la información necesaria por parte del docente, los requerimientos fueron clasificados en funcionales y no funcionales. Para poder realizar la clasificación de los requerimientos, se analizaron aquellos que tengan un impacto directo con la funcionalidad del sistema a desarrollar para el caso de los funcionales. En el caso de los requerimientos no funcionales, se tuvieron aquellos que su descripción no tenga un impacto directo en el prototipo. En la Tabla 18 se presenta un resumen de las actividades, tareas y entregables a presentar en la etapa de comunicación.

Tabla 18: Resumen de actividades a realizar en la etapa de comunicación

Actividades	Descripción	Tareas	Entregables
Recopilación de requerimientos del docente	Obtención de las necesidades del docente para poder desarrollar el prototipo.	Presentación ante docente, explicación general del proyecto, beneficios esperados del proyecto y recopilación de los requerimientos.	-
Clasificación de requerimientos del docente	Clasificación de requerimientos en funcionales (impacto directo en la funcionalidad del videojuego) y no funcionales.	Análisis y división de los requerimientos obtenidos por parte del docente en funcionales y no funcionales	Requerimientos clasificados en funcionales y no funcionales.

Fuente: Elaboración Propia

3.10.1.2 Planeación

En la etapa de planeación se realizaron 2 actividades principales que consistieron en la estimación de los recursos necesarios para el desarrollo del videojuego y la elaboración de un cronograma de trabajo para desarrollar el prototipo.

3.10.1.2.1 Actividad 1: Establecimiento de recursos de trabajo

Obtenidos los requerimientos del docente, se procedió a estimar el tiempo necesario para el construir el prototipo teniendo en cuenta además los conocimientos que se posee en desarrollo de videojuegos.

Además, dentro del desarrollo del producto de software a crear, se tuvo en claro contar con los recursos necesarios para el desarrollo del videojuego. El objetivo principal fue verificar que se cuenta con una conexión a internet estable al igual que el equipo de cómputo necesario para llevar a cabo el desarrollo del videojuego.

Dado que el principal recurso a utilizar fue una laptop para el desarrollo del videojuego, se realizó una descripción de las especificaciones técnicas del equipo de trabajo, así como la definición de los participantes estableciendo el rol que cumple cada colaborador.

3.10.1.2.2 Actividad 2: Desarrollo de cronograma de trabajo

La siguiente actividad de la fase de planeación fue el desarrollo de un cronograma que permita visualizar las actividades a realizar para el desarrollo del videojuego en función a los requerimientos establecidos por parte del cliente.

En el cronograma desarrollado se visualizan el nombre de cada etapa de la metodología en cascada, así como las actividades que implica el desarrollo de cada etapa. También se aprecia las fechas asignadas para la elaboración en general de cada etapa, así como las fechas asignadas para la elaboración de cada actividad que corresponde a cada etapa de la metodología en cascada. La idea fue poder llevar un control de cada actividad que se iba desarrollando y verificar cada día que se encontraba desarrollando la actividad que corresponde. Finalmente, en la última columna del cronograma de trabajo se visualizaba al responsable encargado de realizar cada actividad registrada. En la Tabla 19 se pueden visualizar las actividades, tareas y entregables a desarrollar.

Tabla 19: Resumen de actividades a realizar en la etapa de planeación

Actividades	Descripción	Tareas	Entregables
Establecimiento de recursos de trabajo	Realización de especificaciones técnicas de equipo de trabajo y descripción de los involucrados en el proyecto.	Recopilar de características de equipo de trabajo, recopilar datos y roles de los participantes.	Especificaciones técnicas del equipo a utilizar y descripción de los participantes del proyecto.
Desarrollo de cronograma de trabajo	Realización de cronograma de trabajo para el desarrollo del videojuego	Estimar fechas de entrega para cada fase de la metodología en cascada, definir responsable para cada actividad.	Cronograma de trabajo desarrollado

Fuente: Elaboración Propia

3.10.1.3 Modelado

En la etapa de modelado se desarrollaron 5 principales actividades que corresponden al desarrollo de los casos de uso, bosquejos iniciales del prototipo a desarrollar, el desarrollo de la arquitectura de las escenas, la secuencia de Gameplay del videojuego y la descarga correspondiente de los modelos en 3D a utilizar.

3.10.1.3.1 Actividad 1: Desarrollo de casos de uso

Los casos de uso del sistema fueron desarrollados con el objetivo de modelar el comportamiento del prototipo cuando interactúe con el estudiante. Para el desarrollo de los casos de uso, se tuvieron en cuenta los requerimientos funcionales detectados. Se establecieron las dependencias que existían entre casos de uso como la acción de registrarse antes de poder ingresar al videojuego. Una vez desarrollado el caso de uso general del sistema, se desarrolló la especificación de cada caso de uso en donde se plasmaron el flujo tradicional de funcionamiento y el flujo alternativo por una acción realizada por el usuario. También se especificaron la pre condición que debía cumplir cada caso de uso antes poder ejecutarse una acción determinada y la post condición sobre cómo debería de terminar la actividad realizada por el caso de uso.

3.10.1.3.2 Actividad 2: Desarrollo de bosquejos del prototipo

Como segunda actividad, procedió a establecer los primeros bosquejos del videojuego en función a las especificaciones que debía de poseer el videojuego. Para la creación del diseño del videojuego, se tuvo en cuenta los diseños de los videojuegos realizados en Castro (2015) y en Didehbani et al. (2016) dado que se solicitaron funcionalidades similares a los videojuegos de los autores ya mencionados como el login del videojuego o niveles dentro del prototipo. Se realizaron diseños iniciales con respecto al Login del videojuego, del registro de los estudiantes, del menú principal, así como de cómo debería ser cada nivel del videojuego. Se tuvo en cuenta en dónde deberían de aparecer los campos en donde el estudiante debía de ingresar información para que sea lo más amigable posible para los estudiantes. Con respecto a los niveles del videojuego, se probaron los diversos lugares en la pantalla en donde se podía visualizar el contador de vidas del estudiante, así como el score del alumno y el Minimap. También se tuvo en cuenta la posición de las lecturas que iba a contener el videojuego, así como la escena de las preguntas de opción múltiple de cada lectura debido a que estas funcionalidades debían de poder visualizarse claramente para los usuarios.

3.10.1.3.3 Actividad 3: Desarrollo de la arquitectura de las escenas

Para el desarrollo de la arquitectura de las escenas, se definieron las posiciones finales de cada botón, texto de entrada y modelos en 3D en general antes del desarrollo de las escenas. Al desarrollar la arquitectura de las escenas, se tuvo una visión más específica del aspecto de cada escena del videojuego de modo que la creación del videojuego sea más exacta. En el caso de la escena del login, se establecieron dos InputField en la parte central para que los estudiantes ingresen sus nombres y sus apellidos. En la parte inferior de la escena del login se agregó el botón para que puedan ingresar el videojuego. Con respecto a la escena de registro, se agregaron los campos para el ingreso del nombre, apellido y grado del estudiante. En el caso de este último atributo, se agregaron opciones para que el estudiante puede seleccionar el grado correspondiente. Con respecto a la ubicación de las estadísticas del videojuego como el score acumulado por el estudiante o la cantidad de enemigos destruidos se ubicaron en la parte superior de cada escena. El Minimap se decidió colocar en la parte inferior izquierda del videojuego y las oraciones en la parte inferior central para que el estudiante pueda leer las recomendaciones que el prototipo va mostrando al estudiante. En relación con las lecturas que posee el videojuego, se definió que sería una escena independiente para cada lectura de modo que esta ocupara en mayor proporción la pantalla del computador y a su lado derecho de cada imagen se colocó un botón para cada pregunta para que pueda ser redirigido el estudiante a las preguntas de opción múltiple para cada lectura.

3.10.1.3.4 Actividad 4: Desarrollo de secuencia de Gameplay

Teniendo desarrollado los diseños y arquitectura de las principales escenas del prototipo, se procedió a realizar la secuencia de Gameplay del videojuego que es la secuencia que el estudiante debe de seguir para poder jugar el videojuego. El Gameplay del videojuego se realizó en forma de diagrama de flujo dado que el sistema muestra un determinado contenido en función a las acciones que el estudiante va realizando. Las condiciones establecidas para la secuencia de Gameplay estuvieron en función a las vidas que posee el estudiante y el nivel en donde se encuentre. La secuencia de Gameplay establecida muestra las acciones que estudiante debe de realizar para llegar hasta los créditos finales y pasar el juego, pero también muestra el flujo con acciones incorrectas que dirigen al usuario a un Game Over inmediato.

3.10.1.3.5 Actividad 5: Descargas de modelos en 3D

Los personajes utilizados son los enemigos en 3D que vienen a ser los zombies del videojuego. Los enemigos fueron descargados de la página de <https://www.mixamo.com/>. Para descargar los zombies primero se seleccionaron los modelos y se descargaron los *Idles*

correspondientes a cada uno seleccionando la opción “For Unity” para que los modelos puedan ser leídos por el motor del videojuego. Por cada modelo en 3D descargado, también se buscaron descargar animaciones de *walking*, *death* y *attack* de tal modo que se pueda tener una gran variedad de visualizaciones de movimientos dentro del videojuego. Se descargaron alrededor de 18 diferentes modelos de zombies para que puedan estar dentro del videojuego. También se duplicaron los modelos para poder tener más de un mismo modelo dentro cada nivel.

Para introducir terrenos al videojuego se descargaron diferentes modelos en 3D de terrenos de forma gratuita de la página de <https://www.turbosquid.com/>. Cada modelo descargado fue puesto dentro de una carpeta general de terrenos y arrastradas dentro del proyecto general del videojuego. El modelo de cada terreno fue arrastrado dentro de cada nivel del videojuego y asignándole solidez al modelo en 3D de tal modo que se pueda caminar encima de este.

Finalmente, los modelos de las armas y municiones fueron descargados desde la página de <https://www.turbosquid.com/> y fueron agrupados en una sola carpeta que fue puesta dentro del proyecto de Unity. En total son diez diferentes modelos que fueron utilizados y que se pueden intercambiar con los numerales del teclado. En la Tabla 20, se pueden observar las actividades realizadas en la etapa de modelado, así como las tareas pertenecientes a cada etapa y los entregables para cada actividad.

Tabla 20: Resumen de actividades a realizar en la etapa de modelado

Actividades	Descripción	Tareas	Entregables
Desarrollo de casos de uso	Desarrollo de diagrama de casos de uso general del sistema y especificación de cada caso de uso.	Adaptación de requerimientos funcionales recopilados en el comportamiento del videojuego, establecimiento de precondition y postcondición para cada caso de uso.	Diagrama general de casos de uso y especificación de casos de uso.
Desarrollo de bosquejos del prototipo	Desarrollo de diseños iniciales de las escenas del videojuego	Realización de diseños de escenas principales del sistema	Bosquejos de las escenas del videojuego
Desarrollo de la arquitectura de las escenas	Especificación de la ubicación de los componentes que formaran parte de cada escena del videojuego	Análisis y especificación de los componentes que formarán parte de cada escena (botones, textos, entre otros)	Arquitectura de las escenas del videojuego
Desarrollo de secuencia de Gameplay	Desarrollo del flujo para la jugabilidad del videojuego	Desarrollo de secuencia lógica principal y secuencia de Game Over.	Secuencia de Gameplay completa del prototipo
Descarga de modelos en 3D	Descarga de modelos en 3D para la utilización dentro de las escenas del videojuego.	Descarga de modelos en 3D y Descarga de animaciones	Modelos en personajes enemigos y terrenos en 3D descargados

Fuente: Elaboración propia

3.10.1.4 Desarrollo

En la fase de desarrollo de la metodología en cascada aplicada al prototipo se realizaron 5 principales actividades que consistieron en el desarrollo de las escenas del videojuego, la implementación de las tecnologías seleccionadas (realidad virtual no inmersiva, base de datos de Firebase y servicio de hosting de Firebase), la implementación de Deep Reinforcement Learning como técnica de Inteligencia Artificial, la implementación de las técnicas de aprendizaje dentro del videojuego y las pruebas de funcionalidad de cada escena y general con todas las escenas integradas.

3.10.1.4.1 Actividad 1: Desarrollo de las escenas del videojuego

Para el desarrollo de las escenas del videojuego se utilizó el motor de videojuegos Unity con la versión 2019.4.0f1 y el editor de código Visual Basic con la versión 14.0.23107.178. En primer lugar, se creó un nuevo proyecto para crear el prototipo. Se construyó cada escena en función a los diseños y arquitectura desarrollados en la etapa de modelado. Los modelos en 3D descargados fueron agregados en función a la escena que corresponda. Del mismo modo, las lecturas utilizadas en cada grado fueron colocadas dentro del videojuego. Para esto, las lecturas estuvieron en formato pdf con el objetivo de recortar solamente la imagen donde aparece la lectura y colocarlo dentro del videojuego convirtiendo la imagen del texto en un *sprite*. En esta actividad de desarrollaron las escenas del login y registro de los usuarios. Con respecto a los niveles del videojuego, cada uno posee un terreno diferente para darle una mayor variedad para el usuario. Se desarrollaron escenas adicionales como la escena de créditos finales y la colocación de imágenes para prevenir el Covid-19 en niños de primaria.

3.10.1.4.2 Actividad 2: Implementación de tecnologías

Para mejorar la funcionalidad del videojuego y para que este pueda ser jugado desde los hogares de los estudiantes, se implementaron 3 tecnologías que son la realidad virtual no inmersiva, base de datos y servicio de hosting.

El tipo de realidad virtual utilizada fue la no inmersiva. Este tipo de realidad virtual se caracteriza por la manipulación del videojuego a través de periféricos como un mouse y un teclado que son los más comunes de encontrar dentro de un equipo de computadora. La idea de implementar este tipo de realidad fue aprovechar los recursos que poseen los propios estudiantes dentro de sus casas para que puedan jugar el prototipo desarrollado desde sus hogares. La implementación de esta tecnología se explicará a detalle en la sección 3.12 del presente capítulo.

Como segunda tecnología se implementó una base de datos dentro del videojuego dado que se deseaba tener un control del registro de los alumnos. En esta ocasión, se utilizó la base de datos en tiempo real de Firebase como se había definido anteriormente. La idea fue contar con una escena especialmente diseñada para el registro de los alumnos de tal modo que ingresen sus nombres, apellidos y el grado al cual pertenecen. Una vez que los niños hayan registrado sus datos, con sus nombres y apellidos puedan acceder al videojuego. La base de datos almacena en ficheros la información de cada alumno y posteriormente esa información fue descargada con el objetivo de determinar si todos los alumnos jugaron con el videojuego. El contenido del videojuego se encuentra clasificado en función al grado del estudiante de tal modo que si en el registro el estudiante escoge por ejemplo el sexto grado, cuando juegue el videojuego le aparecerán lecturas correspondientes a ese grado en particular. La creación de una base de datos en tiempo real utilizando Firebase y la conexión respectiva con Unity serán explicados a detalle en la sección 3.13 de presente capítulo.

Por último, dado que el videojuego lo debían de jugar los estudiantes desde sus propias casas, se decidió utilizar un servicio de hosting para el alojamiento del videojuego. Dado que se estaba utilizando Firebase para la implementación de la base de datos, se decidió utilizar de igual manera su servicio de hosting que posee de manera gratuita. Luego de utilizar el servicio de hosting, se creó un link el cual fue puesto en los navegadores para que el videojuego se reproduzca en un computador. La explicación de la implementación del servicio de hosting de Firebase será presentada en la sección 3.15 del presente capítulo.

3.10.1.4.3 Actividad 3: Implementación de técnica de Inteligencia Artificial

Para darle al videojuego un mayor grado de realismo, se decidió utilizar una técnica conocida como *Deep Reinforcement Learning (DRL)* que combina la teoría de Deep learning y Reinforcement Learning. Para implementar la técnica de DRL, se utilizó la librería de Machine Learning de Unity conocida como ML-Agents. Esta librería permite la configuración de un componente Brain que es en el fondo una red neuronal. En este caso se utilizó una MLP para el entrenamiento del agente inteligente. El agente fue creado con el objetivo de seguir al personaje controlado por los alumnos y al estar a una determinada distancia, el agente reproducía un script de ataque para quitarle vida a los alumnos. La lógica principal para el aprendizaje del agente para que pueda aprender a perseguir un objetivo fue capturar la distancia inicial y final entre el agente y el objetivo de tal modo que si la distancia final del agente producto de la salida de la red neuronal es menor a la distancia inicial quiere decir que la Inteligencia Artificial se ha acercado al objetivo por lo que se le da un refuerzo de + 1, de lo

contrario se le dio un refuerzo de -1. La explicación detallada de la implementación de la técnica de *Deep Reinforcement Learning* será presentada en la sección 3.14 del presente capítulo.

3.10.1.4.4 Actividad 4: Implementación de técnicas de aprendizaje

Para ejercitar el nivel de comprensión lectora de los estudiantes, se utilizaron 3 maneras diferentes para que los estudiantes puedan leer dentro del texto. La primera técnica utilizada consistió es la aparición de textos dentro de la lectura. La idea fue ir narrando una historia dentro del videojuego de tal modo que es estudiante pueda ir entrando en contexto dentro del videojuego y entienda cuál era la misión que debía seguir. También, se utilizaron oraciones dentro de cada nivel con el objetivo de poder complementar el texto presentado al inicio. La segunda forma de aprendizaje fue a través de las lecturas propias del colegio. Como se mencionó al inicio la idea fue que las lecturas se encuentren dentro del videojuego y que los estudiantes puedan responder las preguntas que se les plantea con respecto a estas. Las preguntas fueron las mismas que acompañan a las lecturas solo que estas se visualizaban en escenas propias del videojuego. Finalmente, como última técnica se utilizó una sección especial en donde se visualizaban diferentes estrategias para mejorar el nivel de comprensión lectora que proponen los autores, pero redactados con un lenguaje sencillo para que los alumnos puedan entender la estrategia sin problemas. Las explicaciones a detalle para la implementación de estas 3 técnicas de lectura para el aprendizaje de la comprensión lectora serán descritas en la sección 3.16 del presente capítulo.

3.10.1.4.5 Actividad 5: Pruebas de funcionalidad individual y general

La fase de desarrollo culminó realizando las pruebas de funcionalidad de las escenas desarrolladas tanto desde el propio motor de Unity, así como después de que se encontraron dentro del servidor de Firebase con el objetivo de determinar si el comportamiento se mantenía al exportar el videojuego. La idea de realizar las pruebas de funcionalidad individual fue probar el videojuego escena por escena dado que cada escena tiene sus propios scripts desarrollados y era importante verificar que cada una funcionara de manera adecuada para que no haya una mala experiencia de usuario. Verificado que las escenas funcionan de manera adecuada individualmente, la última parte fue la verificación general del funcionamiento de las escenas a partir del registro de los alumnos hasta llegar al final del videojuego de tal modo de poder comprobar que el videojuego funciona correctamente antes de entregar el producto final al docente. En la Tabla 21, se puede visualizar el resumen de actividades desarrolladas en la etapa de desarrollo así como las tareas y entregables que le corresponden a cada actividad.

Tabla 21: Resumen de actividades a realizar en la etapa de desarrollo

Actividades	Descripción	Tareas	Entregables
Desarrollo de las escenas del videojuego	Desarrollo de las escenas diseñadas en la etapa de modelado.	Implementar los modelos de 3D descargados, archivos de audio e imágenes. Introducir botones, textos y diseños a cada escena del videojuego.	Escenas desarrolladas del videojuego.
Implementación de tecnologías	Implementación de la realidad virtual no inmersiva, base de datos de Firebase y servicio de hosting.	Implementar en el videojuego a un personaje en primera persona (<i>FPS Controller</i>), realizar la conexión entre la base de datos con Unity, exportar el videojuego y exportar el proyecto final para subir el prototipo al servidor de Firebase.	Personaje en primera persona, base de datos implementada y link del videojuego.
Implementación de técnica de Inteligencia Artificial	Implementación de la técnica de Deep Reinforcement Learning.	Creación del agente inteligente, creación del ambiente de entrenamiento, desarrollo de los modelos de DRL a entrenar e implementación del mejor agente dentro del videojuego.	Mejor agente inteligente implementado en el prototipo.
Implementación de técnicas de aprendizaje	Implementación de las técnicas de lectura que posibilitan ejercitar la habilidad de comprensión lectora.	Introducir oraciones dentro del videojuego, adaptar los textos tradicionales para implementarlos dentro del prototipo, recopilar las estrategias de lecturas para crear una sección propia dentro del menú principal.	Visualización de lecturas, oraciones dentro del videojuego y sección de estrategias de lectura.
Pruebas de funcionalidad individual y general	Realizar las pruebas del prototipo desarrollado.	Prueba de control de personaje principal, prueba de visualización de contenidos de escenas, prueba de base de datos.	Prototipo terminado y operativo

Fuente: Elaboración Propia

3.10.1.5 Despliegue

En la fase final de la etapa de la metodología en cascada, se realizaron dos principales actividades que consistieron en la entrega del producto desarrollado y realizar una retroalimentación del proyecto realizado.

3.10.1.5.1 Actividad 1: Entrega del producto desarrollado

Una vez terminado de desarrollar el videojuego, este fue mostrado al docente de comprensión lectora con el objetivo de saber si cumple o no con los requerimientos iniciales y para que verifique que está completamente funcional. Se hizo una explicación de cada escena que poseía el videojuego, así como la forma en la cual los estudiantes a través de las lecturas que posee el sistema, las oraciones que van apareciendo en cada nivel y la sección de estrategias de lectura, los estudiantes pueden ejercitar su habilidad de comprensión lectora para obtener la aprobación del docente.

3.10.1.5.2 Actividad 2: Retroalimentación del proyecto

Como última actividad, se realizó una retroalimentación de todo el proyecto desarrollado de tal modo que se pudiera obtener un resumen de las principales actividades que se desarrollaron en cada fase con el objetivo de tener un Feedback del videojuego desarrollado y detectar posibles mejoras en el proceso de la aplicación de la metodología en cascada aplicada al videojuego. En la Tabla 22 se pueden visualizar el resumen de las actividades y tareas de la última etapa de la metodología en cascada.

Tabla 22: Resumen de actividades a realizar en la etapa de despliegue

Actividades	Descripción	Tareas	Entregables
Entrega del producto desarrollado	Entrega del videojuego desarrollado al docente	Exhibición de las escenas del videojuego, explicación de técnicas de aprendizaje para los estudiantes.	Aprobación del docente para la utilización del videojuego
Retroalimentación del proyecto	Retroalimentación del desarrollo del videojuego	Realización de Feedback de las principales actividades desarrolladas	-

Fuente: Elaboración Propia

Finalmente, en la Figura N°52 se presenta un resumen de la lógica para la implementación de la solución resumiendo las principales actividades para el desarrollo del videojuego.

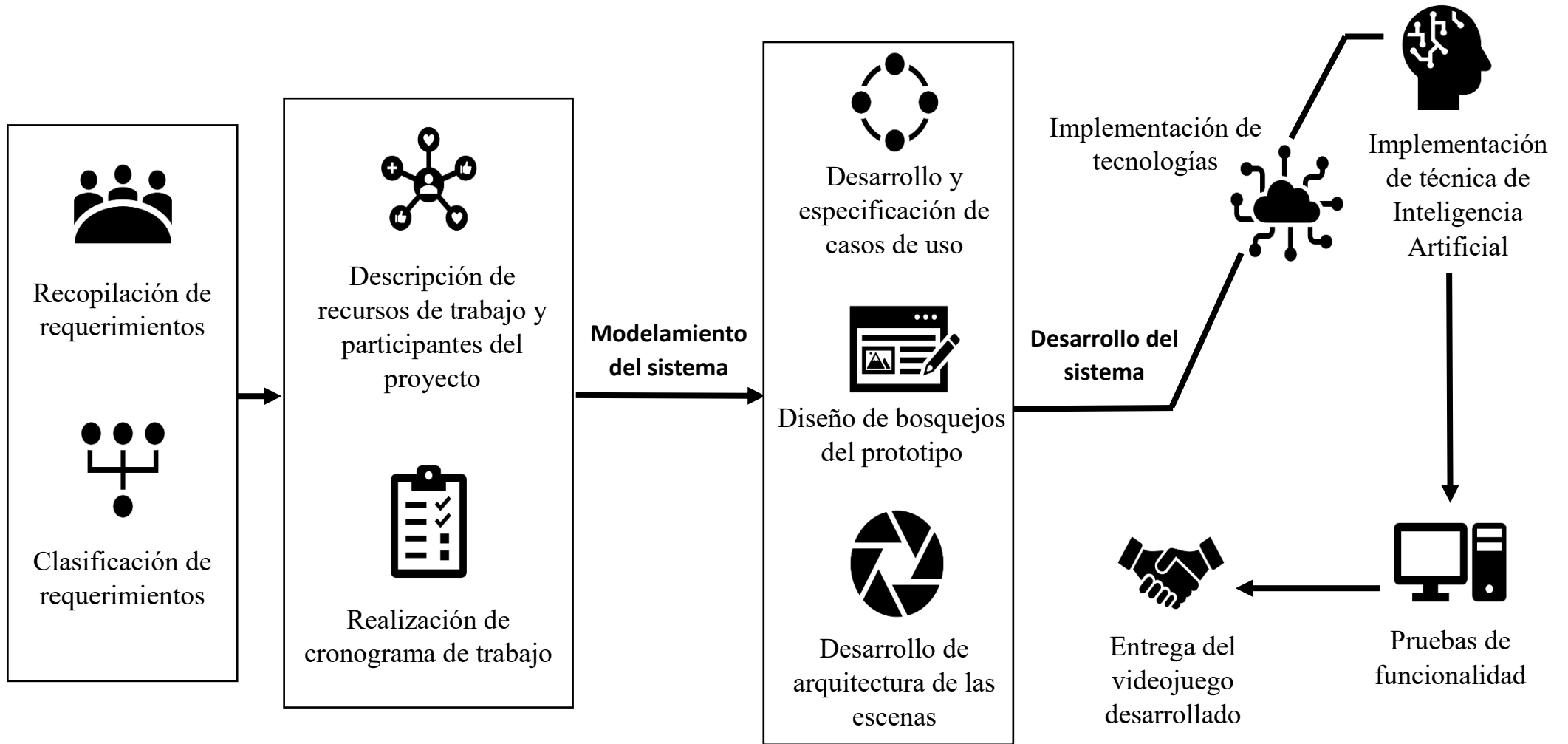


Figura N°52: Resumen gráfico de la metodología para la implementación de la solución.
Fuente: Elaboración Propia.

3.11 Descripción general del prototipo

El videojuego que se desarrolló se llama “Invasión Zombie” cuya historia principal fue la de un virus que ha aparecido en el mundo convirtiendo a las personas en zombies apoderándose de 12 ciudades que fueron los niveles creados para el videojuego y estas ciudades a su vez son diferentes modelos de terrenos en 3D por donde los usuarios debieron andar. El objetivo de los usuarios que vienen a ser los estudiantes de primaria participantes del experimento fue la de destruir a 25 zombies para que lo lleven al siguiente nivel. Cada nivel tiene alrededor de 35 zombies cuyo objetivo principal será la de perseguir y atacar al personaje en primera persona que controle el estudiante. Los usuarios contaban con 5 vidas que son restadas cada vez que los zombies ataquen.

En la Figura N°53, se puede observar los principales componentes del videojuego. De la Figura N°53, se aprecia que el videojuego cuenta con una base de datos en tiempo real. Los usuarios a través del Login ingresaban al videojuego, pero no sin antes haberse registrado. Cuando los alumnos se registran en el videojuego, se envían sus datos de nombres, apellidos y el grado a la base de datos de Firebase creándose un nuevo registro. Una vez que los estudiantes se encuentran registrados, son redirigidos al Login de tal modo que puedan ingresar al videojuego. Los estudiantes digitan los nombres y apellidos que habían registrado anteriormente, la base de datos verifica que esos datos existen dentro de los ficheros creados y extrae el grado al cual corresponde ese usuario de tal modo que sea dirigido al menú correspondiente. Cada estudiante de 1ro a 6to grado es enviado al mismo videojuego, pero con diferentes menús y a su vez cada menú contiene las lecturas que corresponden a cada grado dado que lo que se desea es clasificar el contenido del videojuego para que a cada estudiante pueda presentarse las lecturas que les corresponde.

Firestore, dentro de sus funcionalidades, ofrece un servicio de hosting gratuito el cual fue utilizado para colocar el videojuego con todos los archivos que implica luego de la exportación. En total se utilizaron dos servicios de Firestore, en primer lugar, su base de datos en tiempo real y en segundo lugar el servicio de hosting gratuito.

Además, de la Figura N°53 se puede apreciar que el videojuego cuenta con un modelo de *Deep Reinforcement Learning (DRL)*. Este modelo fue entrenado con el objetivo de que un Zombie pueda ejecutar la acción de seguimiento a los usuarios. Se entrenaron 7 modelos de DRL, 3 de ellos descritos en la presentación de antecedentes del capítulo anterior y los 4 restantes fueron propuestos en función a los mismos valores de los 3 modelos que propusieron

los autores descritos en el anterior capítulo. Dado que la idea principal del entrenamiento de los agentes es que estos acumulen la mayor cantidad de recompensa posible, el mejor modelo seleccionado para que se encuentre dentro del videojuego fue aquel que haya conseguido la mayor cantidad de recompensa en la menor cantidad de tiempo (Adamsson, 2018). La idea de utilizar esta técnica de Inteligencia Artificial fue hacer del videojuego mucho más profesional de tal modo que sean los estudiantes los que compitan con una Inteligencia Artificial y hacer de este videojuego mucho más atractivo y desafiante.

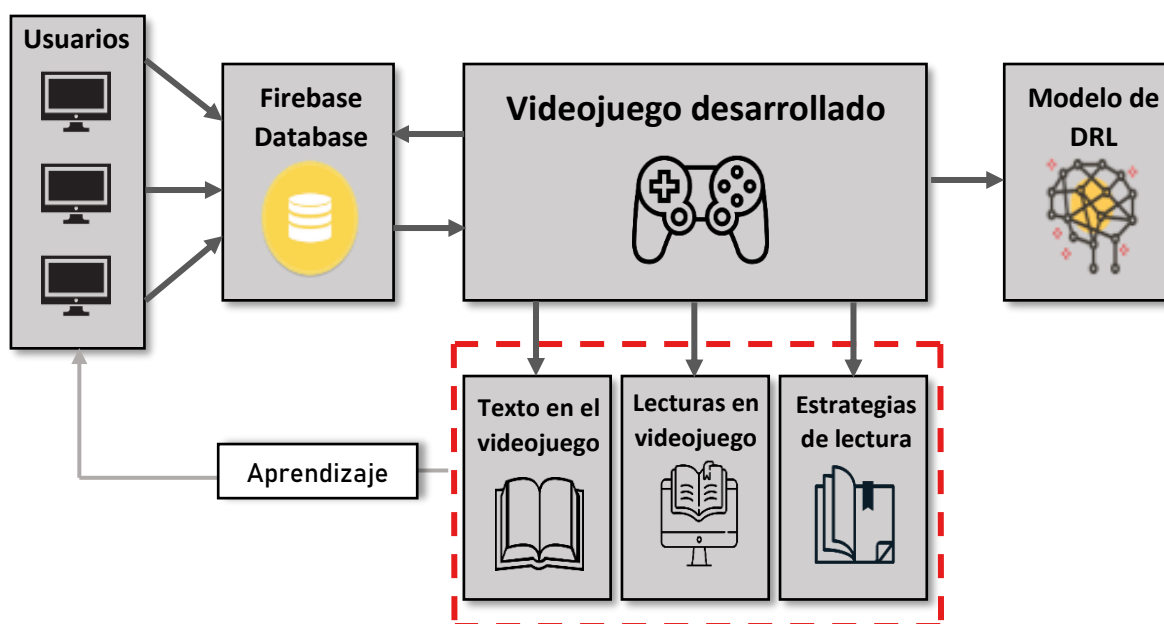


Figura N°53: Descripción general del prototipo.

Fuente: Elaboración Propia.

Para que los estudiantes puedan atacar a los zombies, se les proporcionaron armas en 3D que podrían ser intercambiadas con los numerales del teclado. Dado que se está aplicando la realidad virtual no inmersiva, el niño controlaba a su personaje en primera persona a través del teclado específicamente con las teclas de las flechas de izquierda, derecha, arriba y abajo pero también podría utilizar las teclas A, S para el desplazamiento en horizontal y W, S para el desplazamiento en vertical. De igual modo, para que los estudiantes puedan dispararles a los enemigos, estos utilizaron la tecla de enter, pero también presionando el botón izquierdo del mouse. A su vez los estudiantes podrían recargar sus balas recogiendo las municiones que se encuentran dispersas por todo el terreno para que puedan seguir disparando.

Para el aprendizaje de la comprensión lectora se utilizaron 3 técnicas que estuvieron presentes en todo momento dentro del videojuego. En primer lugar, fue la aparición de textos dentro del videojuego. Cuando los estudiantes iniciaban el videojuego, se les presentaba una

historia ubicándolos en un contexto determinado de tal modo que mediante la lectura de este texto los estudiantes puedan ir desde ya ejercitando su capacidad de comprensión de textos desde el inicio del videojuego. Acabada la presentación de la lectura, se presentaba el primer nivel que corresponde a un terreno determinado. Para complementar esta primera técnica, en la parte inferior del videojuego se presentaban oraciones que complementaban la historia narrada al inicio, así como los objetivos que se deben de cumplir.

La segunda técnica se aplicaba cuando los estudiantes perdían sus 5 vidas, cuando esto sucede aparece un texto especificando que han perdido el nivel a manos de los zombies y que, para recuperar todas sus vidas, deben leer una lectura que poseerá 3 preguntas. Cada escena de las preguntas contenía un contador que va del 1 al 3. El contador se incrementa cada vez que el estudiante respondía adecuadamente una pregunta. Si responde correctamente todas las preguntas del texto leído, era enviado de regreso al nivel en donde se quedó con la totalidad de sus vidas de tal modo que pueda continuar con el videojuego de lo contrario eran enviados al menú de inicio en donde debían empezar desde el primer nivel.

La tercera técnica de comprensión lectora consistió en la creación de un botón que enviaba a los estudiantes a escenas que muestren algunas estrategias para mejorar el nivel de comprensión lectora. Mediante esta sección de estrategias, los niños podrán aplicarlas dentro de las lecturas que contenía el videojuego con el objetivo de ayudarlos a que puedan leer mucho mejor. A través de las 3 técnicas descritas anteriormente es como los estudiantes aprenden a leer mucho mejor y por ende puedan contribuir con la mejora de su nivel de comprensión lectora.

Dado que actualmente, el Perú se encuentra en una situación difícil por la pandemia generada a causa del Covid-19, la idea es contribuir de cierta manera a través del uso del videojuego al cuidado que los niños deben de tener en casa para que prevengan el contagio del Covid-19. Para esto se ha decidido que luego de los créditos aparezcan imágenes con recomendaciones que los niños deben de tener en sus casas para que puedan cuidarse de esta enfermedad altamente contagiosa. Las imágenes que se pusieron dentro del videojuego son las recomendaciones realizadas por BBC News (2020). Estas imágenes fueron colocadas en una escena nueva convirtiéndolas en *sprites* para que se puedan visualizar en el videojuego. Un ejemplo de las recomendaciones que se utilizaron se muestra en la Figura N°54.



Figura N°54: Recomendaciones para niños para prevenir el Covid-19.

Fuente: BBC News. (2020). *Coronavirus: 6 ilustraciones para enseñar a los niños a protegerse (y para que se entretengan coloreando)*. Recuperado de: <https://www.bbc.com/mundo/noticias-52092558>

3.12 Implementación de realidad virtual no inmersiva

Para implementar la realidad virtual no inmersiva, en primer lugar se seleccionó la opción de WebGL al momento de crear el proyecto en Unity. Al seleccionar esta opción para desarrollar el videojuego, cuando el videojuego se exportó, se creó un archivo Index.html que es el archivo que contiene el diseño principal de la página web, se creó también una carpeta llamada “TemplateData” que almacena todas las figuras e íconos que posee el archivo index.html y finalmente se creó un archivo llamado “Build” que es el principal archivo ya que almacena todos los modelos en 3D utilizados así como los scripts necesarios para que el videojuego funcione. Estos 3 archivos fueron colocados dentro de un servidor para que el prototipo desarrollado pueda visualizarse en la pantalla computadora de cualquier estudiante.

Para que el estudiante controle a su personaje, se importó la librería llamada “Standard Assets” de Unity. Esta librería permite utilizar un objeto ya configurado llamado “FPSController” que al arrastrarlo a la escena permite que el usuario controle a un personaje en primera persona a través del teclado y mouse. Por ejemplo, para el desplazamiento del personaje en primera persona en el eje x, se utiliza las teclas A y D, así como las flechas horizontales de izquierda y derecha. Como se aprecia en la Figura N°55, el Input Manager ya se encuentra configurado por defecto para que el personaje en primera persona se desplace con determinadas teclas que son las teclas más usuales dentro de los videojuegos tradicionales por lo que no se realizaron configuraciones adicionales. Mediante el personaje en primera persona,

el usuario puede desplazarse en el eje x con las teclas A, D y las flechas horizontales, desplazarse en el eje z con las teclas W, S y las flechas verticales y finalmente se puede saltar con la tecla de espacio.

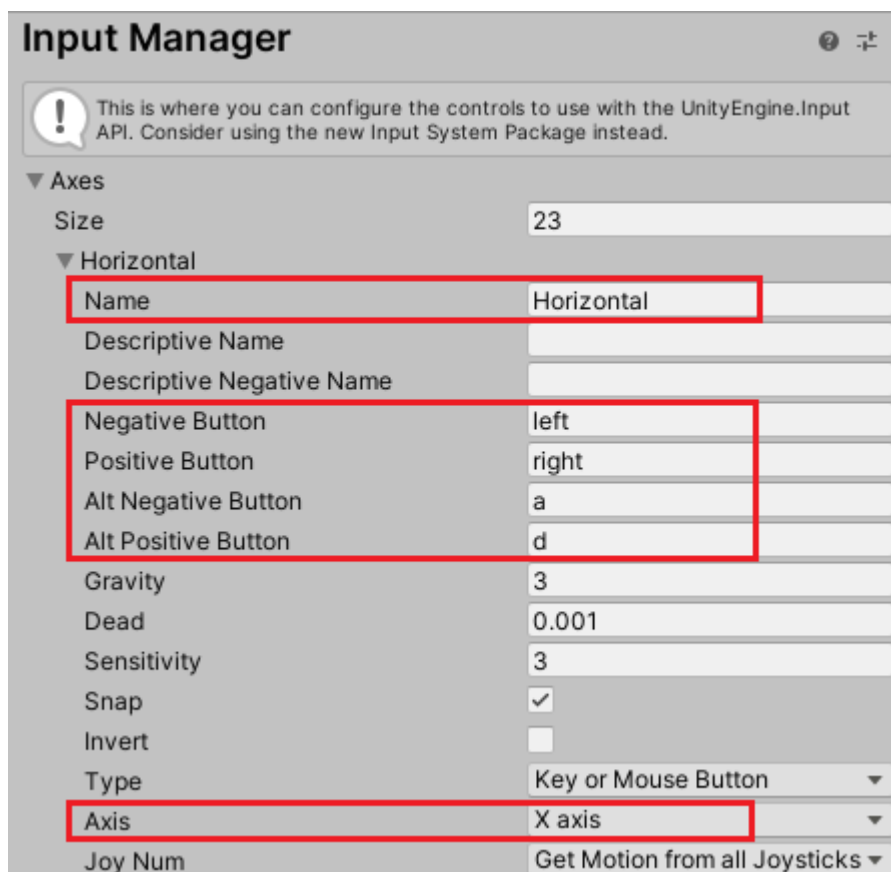


Figura N°55: Configuración del Input Manager para configuración del teclado.

Fuente: Elaboración propia.

3.13 Implementación de la base de datos en tiempo real de Firebase

La base de datos utilizada fue la de Firebase la cual es una base de datos no relacional debido a su velocidad para el almacenamiento y consulta de información. En esta sección, se explicará la conexión con Unity para el envío y consulta de información.

3.13.1 Conexión con Unity

Para crear un nuevo proyecto en Firebase, se debió contar primero con una cuenta Gmail. Con solo registrarse con la cuenta Gmail en Firebase, se pudieron acceder a todas las funcionalidades que posee este último dado que esta tecnología es de Google. Una vez creado el proyecto de Firebase, se procedió a ir a la sección de “Realtime Database” y se creó una nueva base de datos. Firebase pide el nombre de la base de datos a crear y cuando esta se encuentre creada proporcionó un link que sirvió para la conexión con Unity. Esta base de datos

se creó totalmente vacía y además en las reglas de la base de datos se configuraron las reglas de “read” y “write” de tal modo que se encuentren en estado True dado que por defecto ambas aparecen como False. Mediante este cambio realizado, se puede leer el contenido de la base de datos y a su vez realizar inserciones de datos.

Por el lado de Unity, se descargó e importó la librería “Rest Client For Unity” con la versión 2.6.1. Esta librería permite recibir y enviar información desde Unity a través de la función GET y POST respectivamente. Para importar esta librería dentro del código se escribió “Using Proyecto26”. Además, se creó un nuevo script llamado “Usuario”. En este es script se creó una clase pública del usuario y en donde se definió 3 variables: las dos primeras variables son públicas de tipo texto en donde se almacenan los nombres y apellidos del usuario y la tercera variable es pública del tipo Integer que contiene un valor que va del 1 al 6 refiriéndose a los grados de primero a sexto de primaria. De igual modo se creó un constructor de esta clase de tal modo que estas tres variables que han creado inicialmente de nombre, apellido y grado hagan referencia a un nuevo script llamado “Database” que posee dos variables públicas del tipo InputField que corresponde a los campos para almacenar en la escena del registro de alumnos el nombre y apellido de los estudiantes.

Para realizar la función de GET para consultar información de la base de datos de Firebase, el script llamado “Database” contiene además dos variables privadas de tipo string llamadas db_Name y db_Apellido. Estas dos variables son exclusivas para el Login de tal modo que el texto que ingresen los estudiantes sean iguales al valor de estas dos variables que contienen el valor consultado de la base de datos de Firebase. El botón que contiene el Login de “Ingresar” contiene una función creada dentro del propio script de “Database” que se llama RecibirdeBD(). Dentro de esta función se utilizó el comando RestClient.Get<Usuario> colocando entre paréntesis luego del comando el link generado de Firebase y de igual modo se escribe el nombre del fichero a donde consultar. Este nombre es el mismo que el nombre de los estudiantes, y este fichero contiene los atributos de nombre, apellidos y grado. Una vez que se consulta a la base de datos y se localiza el fichero con el nombre escrito por el estudiante en la escena de Login, los nombres y apellidos se almacenan en las variables declaradas de db_Name y db_Apellido de tal modo que si el valor que los alumnos ingresan textualmente dentro del Login es igual al valor almacenado dentro de aquellas variables, entonces accede a una condición en donde se captura el valor del atributo del grado del alumno y lo redirecciona al menú correspondiente. Para realizar esto último se creó una función llamada RecibirYVerificarData() en donde se reciben los datos de la base de datos llamando a la función

RecibirdeBD() explicada anteriormente y se compara si el texto escrito por el estudiante en el Login son iguales al nombre del fichero creado, de ser ese el caso, se extrae el valor del atributo grado del fichero encontrado y con la función Switch-Case se evalúa si el valor del grado es 1, 2, 3, 4, 5 o 6. En función a cada número del grado, el estudiante es redireccionado al menú correspondiente con la función de Unity SceneManager.LoadScene(). En el caso que el estudiante ingrese un campo incorrecto dentro del Login o deje un campo vacío, esos textos ingresados se consultan a la base de datos y al no encontrar el fichero correspondiente, aparece un mensaje en el Login que dice “Los campos ingresados son incorrectos”. De este modo el usuario puede saber si tiene o no una cuenta dentro del videojuego, de ser esto último, debe redirigirse a la escena de registro y completar los datos solicitantes y los datos se envían a Firebase a través del método POST.

Para realizar la función POST y enviar los datos que el estudiante ingrese en la escena de registro de usuarios, en primer lugar, se verificó que el InputField del nombre y apellidos sean diferentes de vacíos de tal modo que se envíe la información completa, de lo contrario aparece un mensaje que dice “Campos incompletos”. Una vez que los campos del nombre y apellido se encuentren completos, lo siguiente es que el estudiante seleccione el grado al cual pertenece y se envíe un número del 1 al 6 a la base de datos para lo cual se agregó un objeto tipo ToggleGroup de tal modo que el estudiante al marcar una casilla, se almacene un número correspondiente. Para realizar esto, se creó una función llamada ResultadoGrado() y se estableció un bucle For con la función Group.transform.childCount y la función GetComponent<Toggle>().isOn. Estas dos funciones permiten hacer referencia a cada Toggle dentro del ToggleGroup de izquierda a derecha. Y con el comando .isOn permite verificar si un Toggle respectivo es seleccionado. Dado que se cuentan con 6 diferentes botones tipo Toggle correspondiente a cada grado ubicados en dos filas cada una con 3 Toggles, el primer Toggle de la izquierda que corresponde a la primera fila contiene el valor de 1 si es seleccionado por el estudiante, el que se encuentra a su derecha tiene el valor de 2 que corresponde al segundo grado, el que se encuentra a la derecha de este último tiene el valor de 3 que corresponde al tercer grado y así sucesivamente hasta llegar al grado número 6 que corresponde al sexto grado. Cuando el estudiante escriba su nombre, apellido y Toggle correspondiente en función al grado al cual pertenezca, el alumno envía la información a través del botón de “Guardar” que contiene la escena de registro de alumnos. Este botón activa una función llamada EnviarPorPost(). Esta función crea un objeto de tipo Usuario (la clase creada al inicio de esta explicación) y dentro de este objeto creado se almacena la información del

alumno y es enviada en formato tipo JSON a la base de datos creándose así un nuevo registro en la base de datos que posteriormente será consultado por el alumno cuando se dirija al Login e ingrese al videojuego. En la Figura N°56, se puede ver un resumen del flujo de trabajo para enviar y consultar información de la base de datos de Firebase.

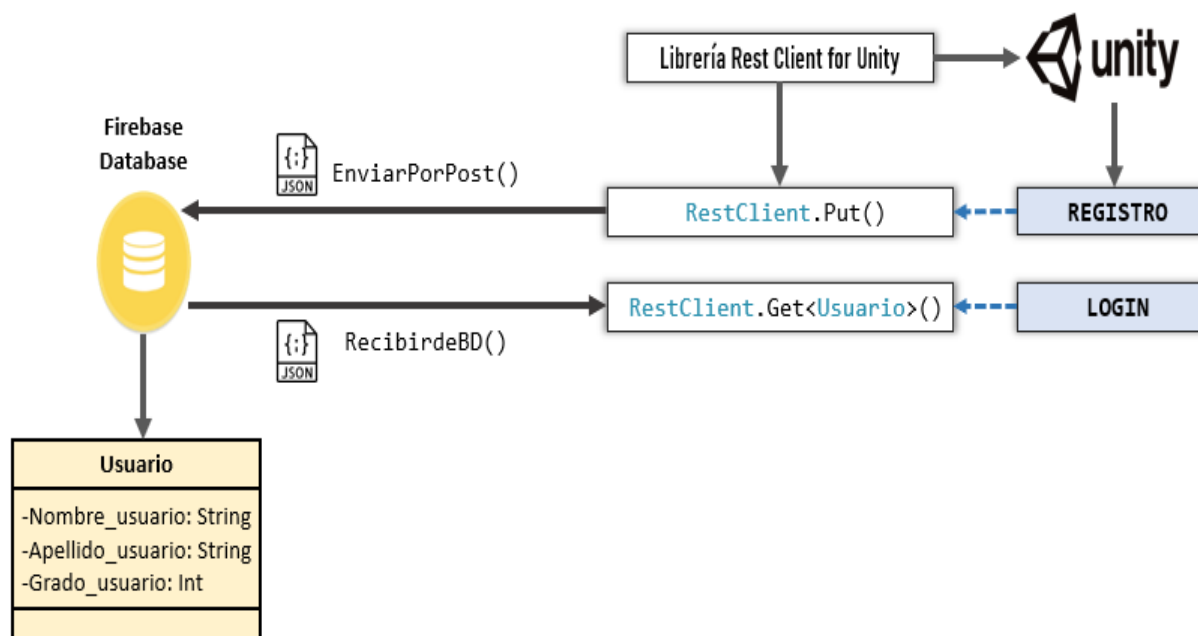


Figura N°56: Flujo de la conexión entre la base de datos de Firebase y Unity.

Fuente: Elaboración propia.

3.14 Implementación de Deep Reinforcement Learning (DRL)

Se implementó la técnica de DRL o aprendizaje por refuerzo profundo con el propósito de entrenar a un agente inteligente para que aprenda a desplazarse y seguir a un objeto en 3D de tal modo que al momento de introducirlo en el videojuego luego del entrenamiento, este pueda seguir al jugador reemplazando el objeto en 3D que sirvió para que el agente entrene por el personaje en primera persona controlado por el estudiante. Se aplicó el flujo de entrenamiento planteado por Shao et al.(2019) que se planteó en el capítulo anterior. Este flujo para el entrenamiento de un agente inteligente adaptado a este proyecto de investigación se presenta en la Figura N°57.

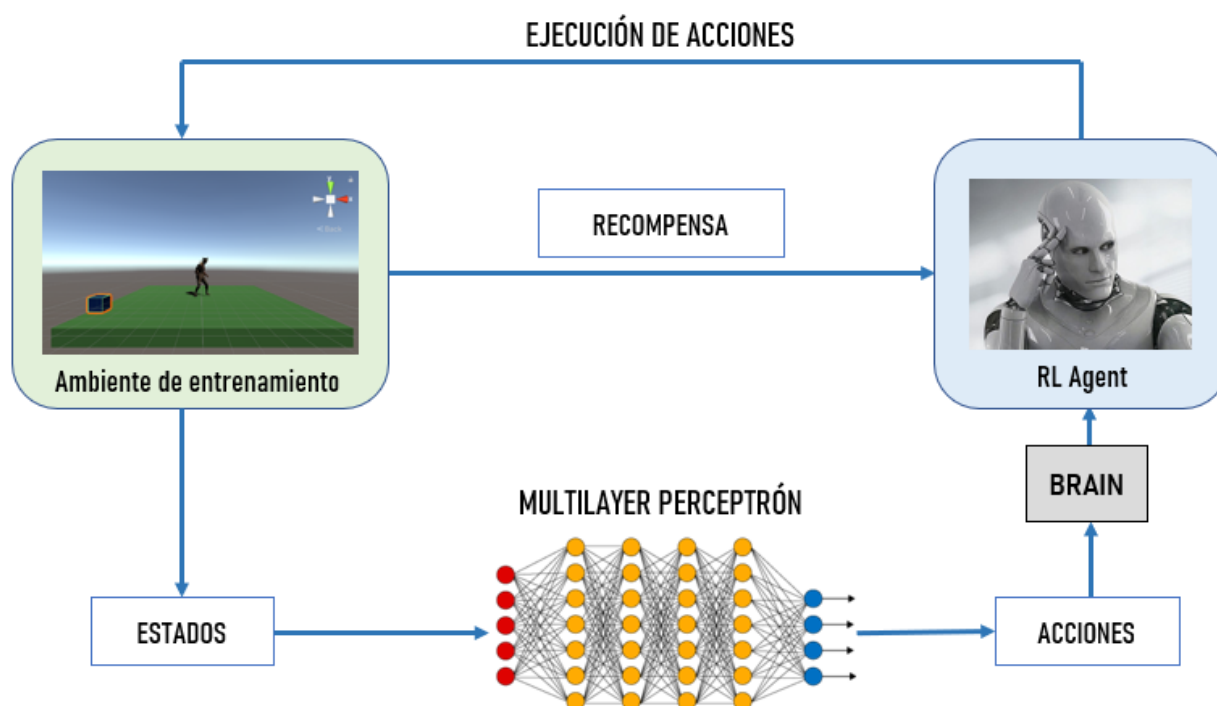


Figura N°57: Flujo de trabajo para la aplicación de DRL en el prototipo.

Fuente: Elaboración propia.

Para poder implementar esta técnica, se descargó Anaconda Distribution y la librería ML-Agents desde Unity y a través de la página oficial de GitHub de Unity utilizando el link <https://github.com/Unity-Technologies/ML-Agents>. Como se observa en la Figura N°57, se necesita en primer lugar un agente el cual será entrenado. Este agente fueron los modelos en 3D de los zombies descargados. Además, este agente entrenó en un ambiente de entrenamiento que simula el terreno del videojuego. El ambiente de entrenamiento envía señales o estados a la red neuronal que en este caso es un Multilayer Perceptrón y las salidas de la red son transformadas en acciones que debe ejecutar el agente a través del componente Brain. Al ejecutar el agente las acciones, el ambiente devuelve al agente una recompensa o refuerzo que puede ser positivo o negativo y además se generan nuevos estados debido a las acciones del agente. El objetivo del agente es acumular la mayor cantidad de recompensa en la menor cantidad de tiempo. Entre mayor sea la recompensa del agente, mejor entrenado estará para ejecutar la acción encomendada. El agente en función con la interacción con el *environment* estará aprendiendo a través de la experiencia a perseguir un objeto en 3D que será el objetivo.

3.14.1 Creación del agente inteligente

Una vez instalado la librería de ML-Agents, se creó una nueva escena para desarrollar el entrenamiento del agente. Para crear el agente inteligente, se creó un script llamado “AgentXZ”

dado que el zombie se desplazará sobre el terreno creado solamente en el eje x y el eje z. En este nuevo script se importó las librerías de ML-Agents escribiendo “Using Unity.MLAgents;” y además “Unity.MLAgents.Sensors;” para enviar las entradas a la red neuronal. De igual modo, por defecto, los scripts se crearon con la clase “MonoBehaviour”, esta fue reemplazada y se utilizó la clase “Agent”. En esta misma clase, se creó una variable pública de tipo GameObject llamada “Objetivo”. Dentro de esta variable se almacenó el objeto en 3D que el agente seguirá y una variable de tipo float llamada “Speed” que es la velocidad con la cual el agente se desplaza.

Dentro de la clase OnEpisodeBegin() en donde se establecen las acciones a realizar al inicio de un episodio se realizaron dos acciones. En primer lugar, cada vez que un episodio comience, el modelo en 3D del zombie se pondrá al centro del ambiente de entrenamiento para que realice sus movimientos. Para esto último, se utilizó la función `transform.localPosition = new Vector3(0, 1.4f, 0)`. Esta función permite que cada vez que un episodio comience la posición del zombie con respecto al eje x sea 0, con respecto al eje y sea 1.4 y con respecto al eje z sea de 0. La segunda acción fue con respecto al modelo en 3D del “Objetivo”. Este objeto al comenzar un nuevo episodio aparece de manera aleatoria en algún punto del terreno para que el zombie lo siga. Para que la posición del objetivo se inicialice de manera aleatoria se utilizó la línea de código `Objetivo.transform.localPosition = new Vector3 (Random.Range(3f,7f) * (Random.value <= 0.5 ? 1 : -1), 0, Random.Range(3f,7f) * (Random.value <= 0.5 ? 1 : -1))` de tal modo que la posición del Objetivo varíe con respecto al eje x y con respecto al eje z. La anterior línea de código permite que la posición del objetivo tome valores con respecto al eje x y z que vayan desde el -3 hasta el 7 de tal modo que pueda empezar un nuevo episodio en una posición nueva.

Dentro de la clase CollectObservations(VectorSensor sensor) es donde se envían las señales a la red neuronal. La red neuronal tiene una capa de entrada de 4 neuronas y se envían 4 valores. En primer lugar, se envía a través de la función `sensor.AddObservation()` la posición inicial del agente con respecto al eje x a través de la línea de código `transform.localPosition.x`. La segunda entrada de la red es la posición inicial del agente con respecto al eje z usando la línea de código `transform.localPosition.z`. El tercer valor que se envía a la red es la posición inicial del Objetivo con respecto al eje x que es a través del código `Objetivo.transform.localPosition.x` y el cuarto valor a la red es la posición inicial del Objetivo con respecto al eje z que es a través del código `Objetivo.transform.localPosition.z`.

Con respecto a la clase `OnActionReived(float [] vectorAction)`, en esta se realizaron las acciones recibidas por la red neuronal. Se creó un vector tridimensional de ceros llamado `controlSignal` con la función `Vector3.zero`. La red neuronal tiene dos salidas que son almacenadas en las variables `vectorAction[0]` y `vectorAction[1]` respectivamente. La variable `controlSignal.x` es igual al `vectorAction[0]` quiere decir que la posición del vector de ceros con respecto al eje x es igual a la salida de la primera neurona de la capa de salida. Del mismo modo, la variable `controlSignal.z` es igual al `vectorAction[1]`. Esto último quiere decir que el valor del vector de ceros creado con respecto al eje z es igual a la segunda salida de la red neuronal. Las salidas de la cada neurona de la capa de salida de la red neuronal son 0,1 y 2 y esas salidas se igualan a las posiciones con respecto al eje x y z del vector de ceros creado como se mencionó anteriormente. Si cada neurona de la red tiene como resultado 2, entonces el valor para su eje respectivo en el vector de ceros es de 1. En el caso de que la neurona tenga como resultado 0 o 1, el valor del eje respectivo es el negativo de la salida de red por lo que en general el desplazamiento con respecto al eje x del agente sería 1 (desplazamiento una unidad a la derecha), 0 (no se desplaza) y -1 (desplazamiento una unidad a la izquierda). De igual modo, el desplazamiento con respecto al eje z del agente es de 1 (desplazamiento una unidad hacia arriba), 0 (no se desplaza) y -1 (desplazamiento una unidad hacia abajo). Con esta lógica, el agente se puede desplazar de diversas maneras realizando diferentes combinaciones de movimientos o acciones sobre el terreno.

Para asignarle la recompensa o refuerzo al agente se capturó la distancia inicial entre el agente y el objetivo con la función `Vector3.Distance()`. Al final el movimiento del agente se volvió a calcular la nueva distancia entre el agente y el objetivo de tal modo que, si la distancia final entre el agente y el objetivo es menor a la distancia inicial calculada anteriormente, se le da al agente una recompensa de +1, de lo contrario se le da una penalidad de -1. Con la asignación de la recompensa, se termina un episodio y el ciclo se repite.

En esta sección, se ha explicado a detalle la creación del agente inteligente para el desplazamiento de este y que pueda aprender a seguir a un objetivo. Se ha hablado de las salidas de la red neuronal, así como el otorgamiento de los refuerzos positivos y negativos porque forman parte de la creación propia del agente. No obstante, la explicación de la arquitectura de la red neuronal, así como el detalle de sus salidas de la red y la aplicación del aprendizaje por refuerzo se profundizará en la sección 3.14.3 del presente capítulo.

3.14.2 Creación del ambiente de entrenamiento

En Adamsson (2018) y Ropilo (2019) se afirma que un ambiente de entrenamiento debe ser una escena independiente al videojuego pero que se debe de asemejar a las características de este para que el agente pueda aprender a ejecutar la acción que se desea. Habiendo revisado los trabajos de investigación de los autores en donde entrenan a sus propios agentes y teniendo en cuenta que los terrenos en donde se iba a realizar el movimiento el agente es un plano horizontal en el fondo, para crear un ambiente de entrenamiento se creó una nueva escena y se agregó un nuevo cubo en 3D al que se le ha llamado “Suelo”. Este cubo tiene un largo de 16 de largo con respecto al eje x y otros 16 de largo con respecto al eje z y contiene un ancho de una unidad. De este modo, el cubo se convierte en un plano que hará la representación del terreno que posee el videojuego por donde transitará el agente. De igual modo, para darle solidez a este suelo, se agregó un objeto a este de tipo Box Collider. Ya teniendo representado el terreno del videojuego, se agregó un objeto en 3D que es el objetivo el cual el agente debe intentar de perseguir de tal modo que simule al personaje en primera persona que el estudiante controla. Este personaje es un simple cubo en 3D que es colocado encima del terreno creado. Finalmente, se agregó el modelo del zombie en 3D que fue colocado al centro del suelo para asignarle el script de “AgentXZ” desarrollado en la sección anterior. De este modo, el entorno de entrenamiento donde el agente persiga a un objeto quedó representado por un suelo en 3D y un cubo en 3D que es el objeto que el zombie debe de perseguir simulando de esta manera el terreno original que posee el videojuego y el personaje en primera persona que el estudiante controla respectivamente.

3.14.3 Arquitectura y componentes del Multilayer Perceptrón

Como se había detallado en la sección 3.14.1 en donde se explica la creación del agente inteligente, se utilizó una red neuronal Multilayer Perceptrón con 4 neuronas en la capa de entrada. La profundidad y número de neuronas por cada capa oculta varía en función al modelo utilizado. La capa de salida de la red posee dos neuronas. Los resultados de la primera neurona de la capa de salida corresponden al desplazamiento del agente en el eje horizontal y la segunda neurona corresponde al desplazamiento con respecto al eje vertical. La arquitectura general de la red neuronal utilizada se presenta en la Figura N°58.

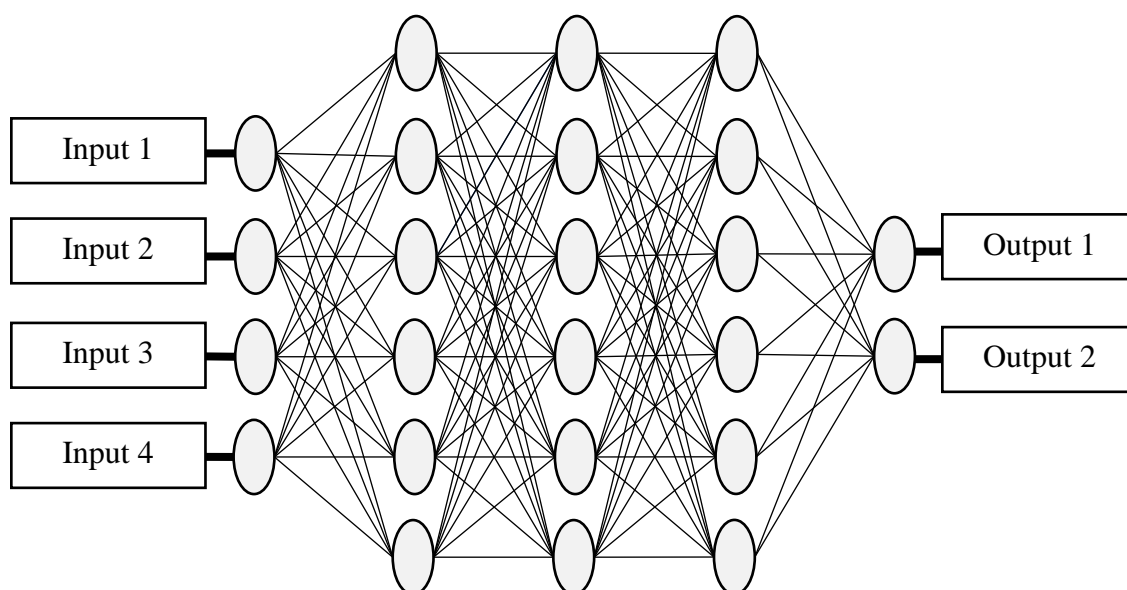


Figura N°58: Arquitectura general de la red neuronal a utilizar

Fuente: Elaboración propia.

La descripción de las entradas y salidas del Multilayer Perceptrón a utilizar se presentan en la Tabla 23.

Tabla 23: Descripción de componentes de la red neuronal

Tipo	Descripción	Rango de valores
Input 1	Posición del agente con respecto al eje x	[-7,7]
Input 2	Posición del agente con respecto al eje z	[-7,7]
Input 3	Posición del objetivo con respecto al eje x	[-7,7]
Input 4	Posición del objetivo con respecto al eje z	[-7,7]
Output 1	Desplazamiento del agente en el eje x	[0,2]
Output 2	Desplazamiento del agente en el eje z	[0,2]

Fuente: Elaboración Propia

Como se observa de la Tabla 23, las entradas de la red corresponden a la posición con respecto al eje x y con respecto al eje z del agente y las otras dos entradas corresponden a la posición del cubo objetivo con respecto al eje x y al eje z estas mismas entradas son utilizadas en Adamsson (2018) para el desplazamiento del agente.

En relación con los componentes de la red, la función de activación que utiliza la librería de ML-Agents por defecto es la tangente hiperbólica cuya ecuación es la siguiente:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (62)$$

Para cada valor de x , este tomará un valor de -1 y 1 . Esta función posee una media de cero y es muy similar a la función de activación sigmoide solo que en este caso la tangente hiperbólica posee un lado negativo (Rubiales, 2020). El *bias* establecido por defecto por la librería de ML-Agents de 0.5 . Las salidas que corresponden a la red, sirvieron para ser sumadas a la posición actual del agente con respecto al eje x y para el eje z de tal modo que el agente consiga desplazarse (Adamsson, 2018).

Las entradas a la red fueron normalizadas a través de un hiperparámetro llamado “*Normalize*” que forma parte de los hiperparámetros de un modelo de DRL. Este hiperparámetro es un booleano que de ser igual a *True* se aplica la siguiente ecuación para la normalización de los datos de entrada:

$$z = \frac{x - \mu}{\sigma} \quad (63)$$

Donde:

- El valor de X representa el valor a transformar.
- El valor de μ representa la media de las entradas de la red.
- El valor de σ representa la desviación estándar de los datos ingresados.

Posteriormente, para el entrenamiento de la red neuronal se aplicó la Ecuación (11) en donde se realiza la suma producto de las entradas de la red multiplicada por sus respectivos pesos para cada neurona en la capa oculta. De igual modo, se aplicó la Ecuación (12) para el cálculo del valor de una neurona en la capa de salida. Realizado el pase en *Forward*, se tuvo que calcular el error por cada neurona que se encuentra en la capa de salida que corresponde a la Ecuación (22) para finalmente realizar la sumatoria y obtener el error total. Para aplicar el pase en *Backward*, se aplicó la regla de la cadena que corresponde a la Ecuación (13), Ecuación (14), Ecuación (15), Ecuación (16), Ecuación (17), Ecuación (18), Ecuación (19), Ecuación (20), Ecuación (21), Ecuación (23), Ecuación (24), Ecuación (25), Ecuación (26), Ecuación (27), Ecuación (28), Ecuación (29), Ecuación (30), Ecuación (31), Ecuación (32) y termina con la Ecuación (33).

3.14.3.1 Configuración del Multilayer Perceptrón desde Unity

Para configurar la red neuronal se utiliza un componente de Unity llamado “*Behavior Parameters*” que se encuentra disponible al instalar la librería ML-Agents desde Unity y que es el componente por el cual se configura el nombre del modelo a probar, las entradas de la red y el valor de las salidas (Ropilo, 2019). En el campo “*Space Type*” se especifica si las salidas de la red serán de tipo discreta o continua. En este caso se utilizaron salidas del tipo discretas que parten desde el cero hacia adelante. En el campo “*Branches Size*” se especifica el número de salidas de la red neuronal y en los sub campos de este último se especifica hasta qué valor toma la salida de la red a partir de cero. Por ejemplo, si en los sub campos del campo “*Branches Size*” se coloca 3, la primera neurona de la capa de salida toma los valores de 0,1 y 2 dado que el número 3 representa la cantidad total de salidas por cada neurona. En la Figura N°59, se presenta la configuración de la red neuronal a través del componente “*Behavior Parameters*”.

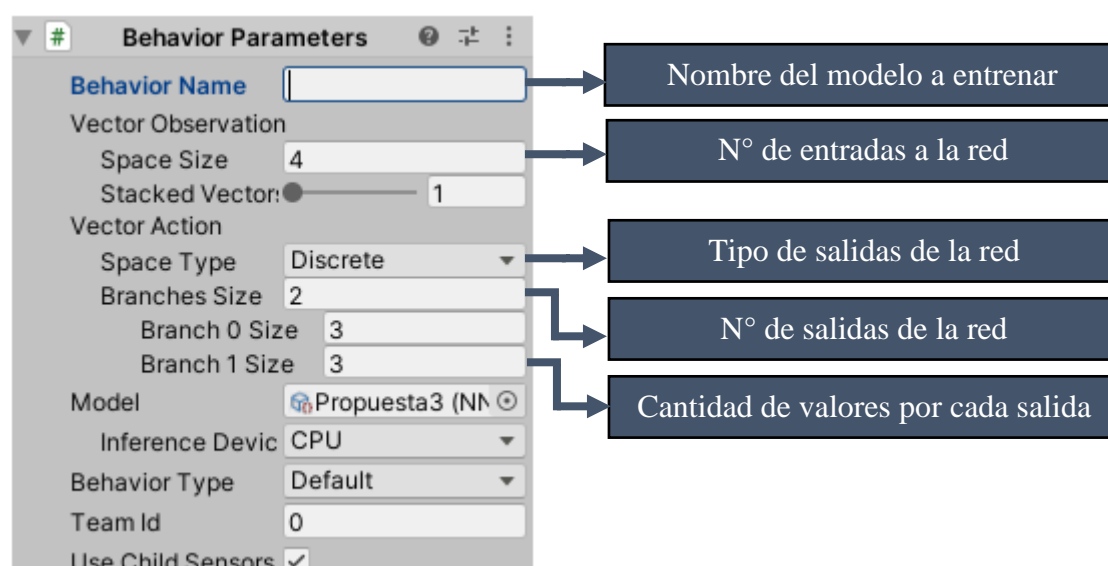


Figura N°59: Configuración de la red neuronal a través de Unity.
Fuente: Elaboración propia.

3.14.3.2 Diseño y selección de mejor arquitectura de red

Para diseñar las arquitecturas de red a utilizar, se hizo una revisión de literatura de trabajos en donde se hace la implementación de la técnica de Deep Reinforcement Learning. Los autores, para la elaboración de los modelos de DRL, suelen realizar consultas a expertos en el campo en donde explican el objetivo principal del agente y la cantidad de tareas que este debe de realizar (Adamsson, 2018).

Para el desarrollo de las arquitecturas de red a utilizar en este proyecto, se visualizaron los valores del Multilayer Perceptron planteados en Expósito (2018), Ropilo (2019) y Adamsson (2018) con el objetivo de tener un intervalo para los valores para establecer el número de capas ocultas a utilizar para cada arquitectura y el número de neuronas por cada capa oculta. Se observó que el número de capas ocultas utilizadas fueron de 128, 512 y 1024 respectivamente. Por otro lado, en la revisión de literatura se visualizó que el número de neuronas por cada capa oculta que sugieren los expertos en el campo es principalmente entre 2 y 3 neuronas. En función a los valores detectados anteriormente, se decidió entrenar 7 modelos diferentes de DRL, cada uno con 7 diferentes arquitecturas de Multilayer Perceptrón. La estructura de los modelos a utilizar se presentan en la Tabla 24.

Tabla 24: Arquitecturas de Multilayer Perceptrón para cada modelo de DRL

Modelo	Nº entradas	Nº Salidas	Nº Capas ocultas	Nº de neuronas por capa oculta
Adamsson (2018)	4	2	128	2
Ropilo (2019)	4	2	512	2
Expósito (2018)	4	2	1024	3
Propuesta 1	4	2	128	2
Propuesta 2	4	2	128	2
Propuesta 3	4	2	512	3
Propuesta 4	4	2	1024	3

Fuente: Elaboración Propia.

Con respecto a la selección de la mejor arquitectura de red neuronal se aplicó una técnica heurística en función a los valores de las métricas de *Cumulative Reward*, *Policy Loss*, *Value Loss* y *Entropy*. Adamsson (2018) y Ropilo (2019) sostienen que la mejor arquitectura de una red neuronal aplicada a *Deep Reinforcement Learning* son aquellas que al final del entrenamiento posee el menor valor en *Value Loss* y el mayor *Cumulative Reward*. La anterior afirmación es debido a que el indicador de *Value Loss* muestra el error entre la recompensa que el agente espera recibir y la recompensa real que recibe producto de las salidas de la red neuronal (Adamsson, 2018). Entre menor sea el valor del *Value Loss*, la red neuronal estará mejor capacitada para predecir los movimientos que debe de realizar el agente para cumplir con la acción asignada (Ropilo, 2019). Por otro lado, el *Cumulative Reward* representa la recompensa que va recibiendo la red neuronal producto de sus salidas. Si las salidas de la red

producen una acción adecuada sobre el agente, se obtiene un *Reward* o recompensa positiva, de lo contrario se obtiene una recompensa negativa.

En Expósito (2018), adicionalmente a las dos métricas descritas anteriormente, se utiliza el *Policy Loss* y *Entropy* para seleccionar la mejor arquitectura de red. Esto es debido a que la primera métrica en mención se relaciona con el cambio de decisión de toma de acciones del agente. Entre menor sea el valor de aquel indicador, el agente habrá comprendido de una mejor manera la acción que debe realizar producto de las salidas de la red. Por otro lado, la métrica de *Entropy* representa la aleatoriedad de las acciones del agente (Expósito, 2018). Entre menos aleatorias sean las salidas de la red neuronal, el agente podrá ejecutar las acciones de desplazamiento con una mayor probabilidad de obtener recompensas positivas.

Teniendo en consideración lo propuesto por expertos en el campo de *Deep Reinforcement Learning*, en este proyecto de investigación se utilizaron como principales métricas para seleccionar la mejor arquitectura de red y por lo tanto el modelo de DRL el *Value Loss*, *Policy Loss*, *Cumulative Reward* y *Entropy*. Dado que se utilizaron 7 modelos de DRL para entrenar a 7 modelos diferentes, también se utilizó el tiempo de entrenamiento de cada modelo para determinar el agente más eficiente que haya acumulado el mayor *Cumulative Reward* en la menor cantidad de tiempo.

Para el entrenamiento de cada arquitectura de red en DRL, se recibe como entradas las posiciones del agente y del cubo objetivo del ambiente de entrenamiento en el eje x y en el eje z para los desplazamientos horizontales y verticales. Se generan luego una sucesión de transformaciones dentro de las capas ocultas como se muestra en la Figura N°60.

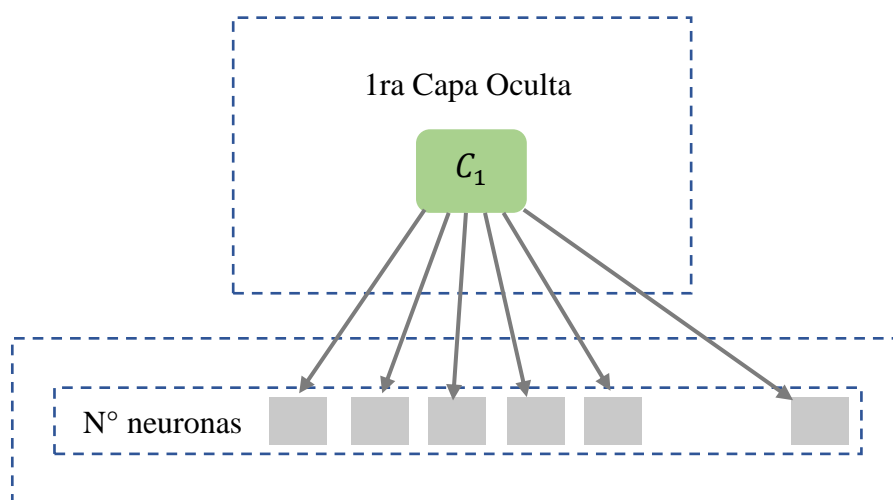


Figura N°60: Funcionamiento de capas oculta en red Multilayer Perceptrón
Fuente: Elaboración propia.

Las salidas de la red funcionan como las unidades a desplazarse del agente inteligente. Producto del Multilayer Perceptrón, el agente se aproxima o se aleja del cubo objetivo. Si la distancia final entre el agente inteligente y el cubo objetivo es menor a la distancia inicial, se le otorga al agente una recompensa positiva, de lo contrario se le otorga una recompensa negativa. La recompensa real obtenida al final de una iteración se compara con la recompensa esperada cuya diferencia es el error de la red neuronal (*Value Loss*). El error se propaga de derecha a izquierda ajustando los pesos de la red. En una arquitectura de red adecuada, el *Cumulative Reward* tiende a incrementarse a lo largo de todo el entrenamiento (Expósito, 2018). Además, la red neuronal al iniciar con el entrenamiento presenta una aleatoriedad de acciones elevada (*Entropy*), en un entrenamiento exitoso, esta métrica debe disminuir a lo largo del proceso de aprendizaje. La disminución de la entropía sugiere que la red neuronal cada vez se vuelve más precisa al momento de decidir las salidas que corresponden a un determinado conjunto de estados lo que sugiere que la arquitectura de red utilizada es la óptima. El resumen de la lógica de selección de mejor arquitectura se presenta en la Figura N°61.

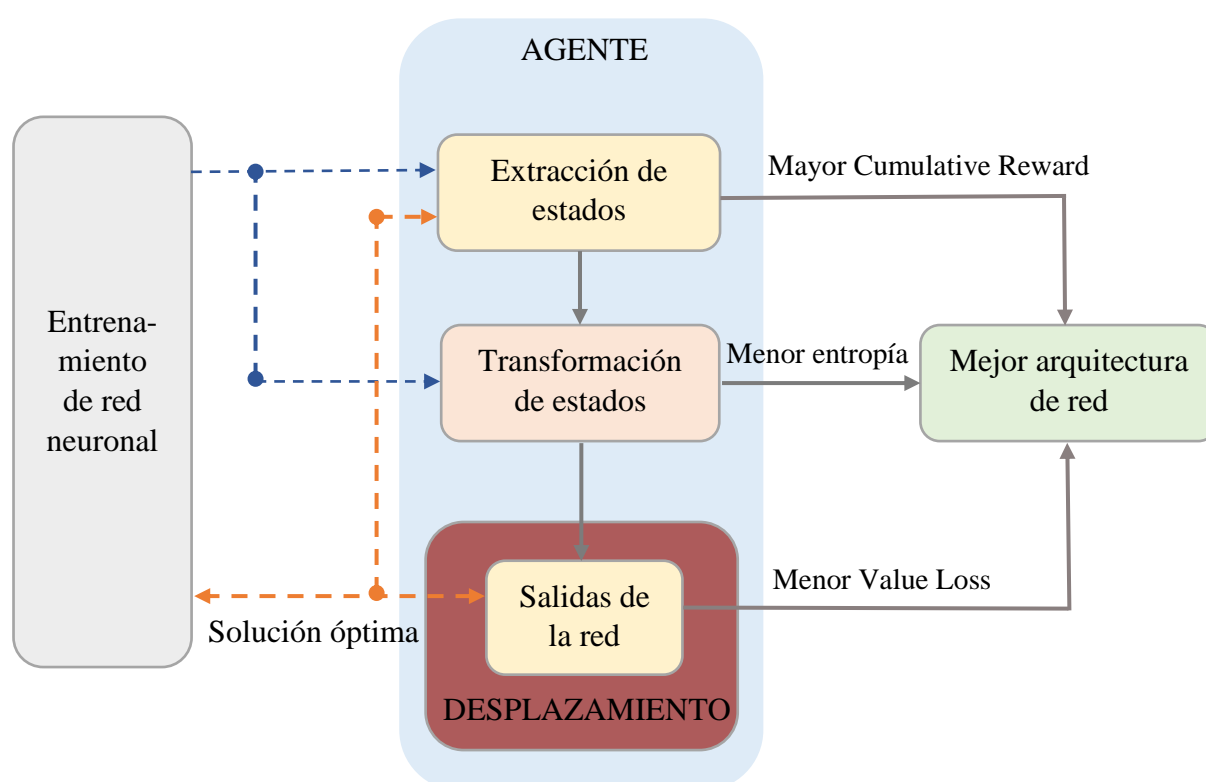


Figura N°61: Lógica de entrenamiento y selección de la mejor arquitectura de red.
Fuente: Elaboración propia.

Adicionalmente a las métricas utilizadas para seleccionar la mejor arquitectura de red, los resultados de los entrenamientos a través del Tensorboard muestran métricas complementarias

para entender el aprendizaje de cada agente a lo largo de todo el entrenamiento como *Extrinsic Value Estimate*, *Episode Length*, *IsTraining*, entre otros. La explicación detallada de los hiperparámetros de cada modelo a utilizar se explicarán en la sección 3.14.4. Además, las métricas completas que proporciona Tensorboard para entender el aprendizaje del agente, se explicarán en la sección 3.18.2.1.

3.14.3.3 Interpretación de las salidas de la red sobre el agente

Como se había mencionado anteriormente, la capa de salida de la red tiene dos neuronas. La primera neurona corresponde al desplazamiento con respecto al eje x del agente y la segunda neurona corresponde a la salida con respecto al eje z. Ambas neuronas tienen como salidas 3 valores discretos que son 0,1 y 2. En el caso que salga 2, el desplazamiento es cambiado a 1. En el caso que salgan 0 o 1, el valor de desplazamiento es el negativo de estos. De tal modo que el negativo de 0 seguirá siguiendo cero y el negativo de uno será -1. Con estos cambios el agente logra desplazarse una unidad a la derecha, una unidad a la izquierda y el cero se refiere a que no habrá movimiento. En la Tabla 25, se puede observar un resumen de lo anterior mencionado.

Tabla 25: Interpretación de las salidas de la red para el desplazamiento del agente

Capa de salida	Valores de Salida de la red	Transformación de valor de salida	Descripción de movimiento del agente
1ra neurona	0	0	Sin desplazamiento horizontal
	1	-1	Movimiento de 1 unidad a la izquierda
	2	1	Movimiento de 1 unidad a la derecha
2da neurona	0	0	Sin desplazamiento vertical
	1	-1	Movimiento de 1 unidad hacia abajo
	2	1	Movimiento de 1 unidad hacia arriba

Fuente: Elaboración Propia.

3.14.3.4 Aplicación del aprendizaje por refuerzo

En este apartado se explicará cómo es que se le dio una recompensa o refuerzo positivo o negativo al agente de tal modo que, mediante la interacción con el ambiente de entrenamiento, este agente pueda entender que la tarea que debe de desarrollar es perseguir a un objeto 3D que en este caso es un simple cubo llamado objetivo.

3.14.3.4.1 Otorgamiento de premios y castigos al agente

En el trabajo desarrollado en Adamsson (2018), se utilizó como criterio para otorgar los refuerzos positivos y negativos las distancias iniciales y finales de cada episodio entre el agente y el objetivo tal cual como se mencionó en la sección 3.14.1. Para otorgarle el premio y castigo al agente se almacenó en una variable tipo float la distancia inicial que existe entre el agente y el objetivo y después del desplazamiento del agente producto de las salidas de la red neuronal, se almacenó en otra variable tipo float la distancia final que existe entre el agente y el objetivo. Si la distancia final es menor a la distancia inicial quiere decir que el agente se ha acercado al cubo objetivo por lo que se le dio una recompensa o refuerzo positivo de +1. Por el contrario, si la distancia final es mayor a la distancia inicial entonces quiere decir que el agente se ha alejado del cubo objetivo por lo que se le dio una penalidad de -1. Para otorgarle los refuerzos positivos y negativos a través del código se utilizó la función `SetReward()` en donde dentro del paréntesis se colocó el valor que se le dará al agente. Los autores sugieren que los valores de las recompensas y castigos deben de ser pequeñas dado que darle un refuerzo positivo o negativo muy elevado podría entorpecer el entrenamiento del agente por lo que solo se le dio un refuerzo positivo de +1 al agente si es que ejecuta la acción adecuada y un refuerzo negativo de -1 si lo hace incorrectamente.

3.14.4 Desarrollo de experimentos

Para entrenar a un agente inteligente, fue necesario establecer un modelo que posea determinados hiperparámetros para el entrenamiento del agente. En el presente proyecto de investigación se utilizaron 7 modelos de DRL para la creación de 7 diferentes agentes. Una vez entrenados los agentes con los 7 modelos, se determinó qué agente es el más adecuado en función a los resultados obtenidos del entrenamiento. La presentación de los resultados de los modelos se explicarán con más detalle en la sección 3.18 del presente capítulo. De los 7 modelos a utilizar, 3 modelos fueron los que propusieron en los trabajos de investigación de Adamsson (2018), Ropilo (2019) y Expósito (2018). Cada uno de estos 3 autores establecieron un modelo de DRL en función a la consulta a expertos en el campo por lo que la idea fue reutilizar los mismos modelos para determinar cual es el mejor. Los 4 modelos siguientes

fueron propuestas que se basaron en la combinación de los 3 modelos de los autores descritos anteriormente, de tal modo que se obtengan 4 nuevos modelos. Es preciso resaltar que en este campo de DRL aplicado a videojuegos no existe un proceso formal para realizar un tuneo de hiperparámetros y seleccionar los mejores valores sino que se hace a través de consultas a expertos en el campo de DRL y en función a los valores que estos expertos ofrescan, se creen los modelos de DRL adecuados para el entrenamiento de los agentes.

3.14.4.1 Modelos de DRL utilizados y aplicación del algoritmo de PPO

Los modelos de DRL constan de 16 hiperparámetros, como se había dicho anteriormente, se utilizaron los modelos de Adamsson (2018), Ropilo (2019) y Expósito (2018) y en función a los valores de los modelos se crearon 4 nuevos modelos que fueron propuestas. De los 16 hiperparámetros, 7 de estos fueron constantes dado que no hubo variación en los 3 modelos de los autores. Dado que estos 7 hiperparámetros tienen valores constantes, en los 4 modelos propuestos también los tuvieron. En la Tabla 26, se presentan estos 7 hiperparámetros constantes para cada uno de los 7 modelos a entrenar.

Tabla 26: Hiperparámetros con valores constantes para los 7 modelos de DRL

Modelo	Trainer	Lambda	Normalize	Num_Epoch	Time_Horizon	Summary_Freq	Strength
Adamsson (2018)	PPO	0.95	True	3	64	10000	1.0
Ropilo (2019)	PPO	0.95	True	3	64	10000	1.0
Expósito (2018)	PPO	0.95	True	3	64	10000	1.0
Propuesta 1	PPO	0.95	True	3	64	10000	1.0
Propuesta 2	PPO	0.95	True	3	64	10000	1.0
Propuesta 3	PPO	0.95	True	3	64	10000	1.0
Propuesta 4	PPO	0.95	True	3	64	10000	1.0

Fuente: Elaboración Propia.

Como se puede observar en la Tabla 26, el algoritmo de *Proximal Policy Optimization* (PPO) se aplica dentro del entrenamiento cuando al hiperparámetro llamado “*Trainer*” se escribe como valor de entrada “PPO” en los modelos de DRL. Al aplicar este algoritmo para optimizar las nuevas políticas o conjunto de acciones del agente, se está aplicando la Ecuación (53) que es la ecuación del algoritmo de *Proximal Policy Optimization* presentada en el capítulo anterior. De igual modo, se aprecia que todos los modelos de DRL, se aplicó la normalización a los datos de entrada a la red a través del hiperparámetro “*Normalize*” que utiliza la Ecuación (63).

Una vez establecidos los 7 hiperparámetros constantes, lo siguiente fue definir los 9 hiperparámetros que son variables. Dado que los modelos propuestos por Adamsson (2018), Ropilo (2019) y Expósito (2018) han dado diferentes valores para estos hiperparámetros la idea fue obtener todos los valores para cada hiperparámetro y para los 4 modelos propuestos estos valores se ordenaron de menor a mayor. Por ejemplo, para el hiperparámetro *Batch_Size*, los autores Adamsson (2018), Ropilo (2019) y Expósito (2018) proponen los valores de 1024, 128 y 32 respectivamente. Por lo que ordenando estos valores de menor a mayor para los cuatro modelos propuestos será 32,32,128 y 1024. Como se aprecia, se repetirá el menor valor del hiperparámetro hasta llegar a cubrir los 4 modelos. Un segundo ejemplo es con el hiperparámetro *Epsilon*, los autores Adamsson (2018), Ropilo (2019) y Expósito (2018) proponen los valores de 0.2,0.2 y 0.1 respectivamente, por lo que ordenando estos valores de menor a mayor para los cuatro modelos propuestos quedaría 0.1, 0.1, 0.2, 0.2 para la propuesta 1 hasta la propuesta 4 respectivamente. Este mismo criterio se aplicó para el resto de hiperparámetros. Los 9 hiperparámetros restantes para los 7 modelos aplicando el criterio descrito anteriormente, se presenta en la Tabla 27. Es preciso mencionar que el criterio de ordenar de menor a mayor puede variar por ejemplo ordenando de mayor a menor o con valores intercalados, etc. Esto dependerá del propio investigador dado que, como se dijo anteriormente, en este campo no existe un proceso formal para crear los modelos de DRL sino que los autores de los diversos trabajos de investigación se basaron la opinión de expertos en el campo. La idea principal para crear estos 4 nuevos modelos que son propuestas fue reutilizar esos mismos valores que en su momento recomendaron los expertos en DRL a los cuales se consultaron en los trabajos de Adamsson (2018), Ropilo (2019) y Expósito (2018).

Tabla 27: Hiperparámetros con valores variables para los 7 modelos de DRL

Modelo	Batch_Size	Beta	Buffer_Size	Epsilon	Hidden_Units	Learning_rate	Num_Layers	Gamma	Max_steps
Adamsson (2018)	1024	5.0e-3	10240	0.2	128	3.0e-4	2	0.99	27.0e4
Ropilo (2019)	128	1.0e-1	2048	0.2	512	3.0e-4	2	0.99	5.0e5
Expósito (2018)	32	1.0e-2	2048	0.1	1024	5.0e-4	3	0.7	1.0e6
Propuesta 1	32	5.0e-3	2048	0.1	128	3.0e-4	2	0.7	2.0e5
Propuesta 2	128	5.0e-3	2048	0.1	128	3.0e-4	2	0.7	2.0e5
Propuesta 3	128	1.0e-2	10240	0.2	512	5.0e-4	3	0.99	25.0e4
Propuesta 4	1024	1.0e-1	10240	0.2	1024	5.0e-4	3	0.99	25.0e4

Fuente: Elaboración Propia.

De este modo se tuvieron los 7 modelos de DRL a entrenar para conseguir al mejor agente inteligente. Además, se menciona que los nuevos valores para el hiperparámetro `Max_steps` pertenecientes a las 4 propuestas fueron recopilados de los trabajos de Liessner et al. (2018) y González (2018) dado que estos mismos autores sugieren valores para este hiperparámetro que vayan entre $2.0e5$ y $25.0e4$ para un aprendizaje adecuado. Con esto se consiguió que los 4 nuevos modelos entrenen con la menor cantidad de pasos acumulados posibles pero que a su vez puedan en ese periodo acumular la mayor cantidad de recompensa. Para visualizar los hiperparámetros de los 7 modelos de DRL de forma individual se puede consultar al enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN> que redirecciona al repositorio de GitHub del proyecto.

Es preciso mencionar que hay un hiperparámetro adicional que se llama `Visual_Encoder_Type`. Este hiperparámetro se puede configurar de tal modo que se pueda implementar un Multilayer perceptrón si se usa `Visual_Encoder_Type = simple`. No obstante, este hiperparámetro no se suele introducir en los modelos dado que la librería ML-Agents asume que se va a utilizar el valor de “simple” para el hiperparámetro `Visual_Encoder_Type` de tal modo que lo que se necesita es configurar simplemente el número de capas ocultas de la red, así como el número de neuronas por cada capa oculta y el número de las épocas a través del hiperparámetro `Hidden_Units`, `Num_layers` y `Num_epoch` respectivamente.

3.14.4.2 Explicación de Hiperparámetros y rangos sugeridos por Unity

En este apartado se explicará los hiperparámetros que presentan los modelos de DRL presentados anteriormente. La definición de cada hiperparámetro, así como el rango que sugiere Unity son los siguientes:

- **Trainer:** Permite seleccionar el tipo de algoritmo de DRL a utilizar para el entrenamiento. La librería ML-Agents permite implementar el algoritmo de *Proximal Policy Optimization* (PPO) y *Soft Actor Critic* (SAC) (González, 2018). Por defecto, suele aplicarse PPO dado que es el más común para el entrenamiento de agentes inteligentes.
- **Batch_Size:** Es el número de experiencias utilizada para una iteración para la actualización del descenso de gradiente que trata de minimizar el error de la predicción (González, 2018). El rango sugerido por Unity para este hiperparámetro es de 512 hasta 5120 para valores continuos y de 32 hasta 512 para valores discretos.

- **Beta:** Permite regularizar la entropía lo que posibilita que el agente explore las diversas acciones que puede realizar durante el entrenamiento (González, 2018). Conforme avanza el entrenamiento, la entropía debería de disminuir. El rango sugerido por Unity para este hiperparámetro es de 0.0001 y 0.01.
- **Buffer_Size:** Indica cuántas experiencias (observaciones, acciones y recompensas) deben de ser recolectadas antes de actualizar el modelo. Entre mayor sea el tamaño del Buffer_Size, más estable serán las actualizaciones del modelo (González, 2018). El rango típico que sugiere Unity para este hiperparámetro son valores comprendidos entre 2048 y 409 600.
- **Epsilon:** Representa el umbral de diferencia aceptable entre las antiguas y las nuevas políticas durante una actualización del descenso de gradiente (González, 2018). El rango que sugiere Unity para este hiperparámetro es entre 0.1 (para actualizaciones más estables) y 0.3 (para un entrenamiento más ágil) (González, 2018). Mediante la modificación de este hiperparámetro, es como se aplica el algoritmo de PPO, dado que la idea es mantener este hiperparámetro dentro de un intervalo tal cual como se mostró en la Ecuación (51) de tal modo que las nuevas políticas no superen en cantidad de acciones a las anteriores.
- **Hidden_Units:** Corresponde al número de capas ocultas que tiene la red neuronal (González, 2018). El rango que sugiere Unity para este hiperparámetro es entre 32 y 512.
- **Lambda:** Reduce la varianza de las estimaciones de la gradiente de política de tal modo que se administre de una mejor manera los datos de entrada. Si se utiliza un valor muy bajo, el agente confiará más en la estimación del valor actual (habría un mayor sesgo) y un valor muy alto ocasiona que el agente confíe más en el valor real de las recompensas recibidas (González, 2018). El rango sugerido por Unity es entre 0.9 y 0.95.
- **Learning_rate:** Representa la fuerza de cada descenso de gradiente (González, 2018). El valor de este hiperparámetro debería de reducirse si el entrenamiento es estable y la recompensa no aumenta de forma constante (González, 2018). El rango sugerido por Unity es entre 0.000001 y 0.0003.
- **Max_steps:** Indica cuantos pasos debe de realizar el entrenamiento del modelo. El valor debe de incrementarse si es que un modelo ya ha sido pre entrenado y se desea continuar

con el entrenamiento (González, 2018). El rango de pasos sugerido por Unity para este hiperparámetro es entre 500 000 y 10 000 000 (para problemas más complejos).

- **Normalize:** Es un valor booleano que indica si se aplica la normalización al conjunto de datos de entrada a la red (González, 2018). Este hiperparámetro aplica la Ecuación (63) vista con anterioridad.
- **Num_epoch:** Es el número de épocas para la red neuronal durante el descenso de gradiente (González, 2018). El rango típico sugerido por Unity es entre 3 y 10.
- **Num_layers:** Corresponde al número de neuronas por cada capa oculta de la red neuronal (González, 2018). El incremento de este hiperparámetro se suele aplicar para problemas de aprendizaje más complejos (González, 2018). El rango sugerido por Unity es entre 1 y 3.
- **Time_horizon:** Representa el número de pasos que se deben de recopilar antes de agregar esos pasos al búfer (memoria temporal de almacenamiento de datos) (González, 2018). El rango típico sugerido por Unity es entre 32 y 2048.
- **Summary_freq:** Es el número total de pasos que ejecutará el modelo por cada episodio del entrenamiento (González, 2018). Por ejemplo, si escribe 20 000 como valor para este hiperparámetro, se verá en la impresión de la consola que el modelo incrementará el número de pasos de 20 000 en 20 000 para todos los episodios hasta llegar al valor establecido en el Max_steps.
- **Strength:** Corresponde a la fuerza con la que se permite que las acciones recolectadas influyan en la política (González, 2018). El rango sugerido por Unity es entre 0.1 y 0.5.
- **Gamma:** Indica si el accionar del agente debe ser en función a las recompensas futuras (si el valor es grande) o si las acciones del agente dependen de las recompensas inmediatas (recompensas inmediatas) (González, 2018). El rango sugerido por Unity es entre 0.8 y 0.995.

3.14.4.3 Archivo Trainer_config.yaml y ejecución de modelo de DRL

Al momento de descargar la librería ML-Agents desde la página de GitHub de Unity, dentro de esa carpeta se encontrará un archivo llamado Trainer_config.yaml. Dentro de este archivo se colocaron todos los modelos de DRL que iban a entrenar. Por defecto, dentro de este archivo, se encuentran ya algunos ejemplos de modelos que vienen para las escenas de ejemplo que vienen junto con la librería como se muestra en la Figura N°62. Este archivo puede abrirse desde cualquier editor de texto y es sensible a las tabulaciones.

```
default:
  trainer: ppo
  batch_size: 1024
  beta: 5.0e-3
  buffer_size: 10240
  epsilon: 0.2
  hidden_units: 128
  lambda: 0.95
  learning_rate: 3.0e-4
  learning_rate_schedule: linear
  max_steps: 5.0e5
  normalize: false
  num_epoch: 3
  num_layers: 2
  time_horizon: 64
  summary_freq: 10000
  reward_signals:
    extrinsic:
      strength: 1.0
      gamma: 0.99

FoodCollector:
  normalize: false
  beta: 5.0e-3
  batch_size: 1024
  buffer_size: 10240
  max_steps: 2.0e6
```

Figura N°62: Estructura interna del Archivo Trainer_config.yaml.

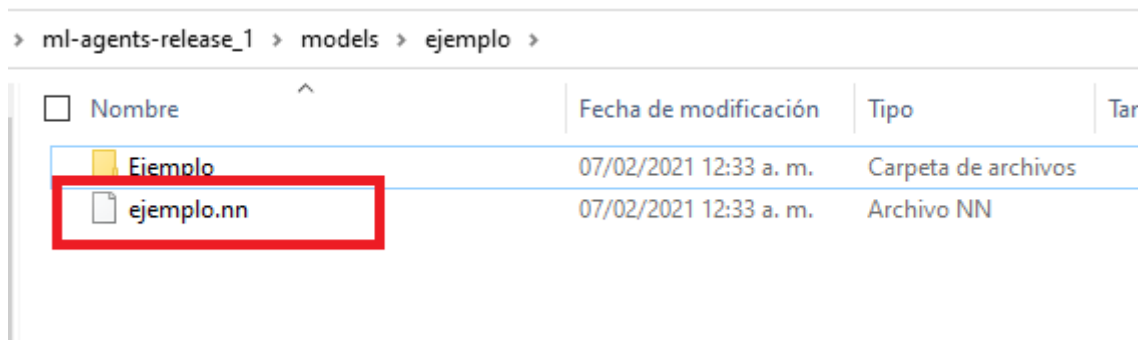
Fuente: Elaboración propia.

Para entrenar los 7 modelos de DRL, se escribieron los modelos dentro de este archivo y se les asignó un nombre. Cada modelo registrado tuvo el mismo nombre de los autores que los proponen y el nombre de los 4 modelos que serán las propuestas fueron llamados “Propuesta1”, “Propuesta2” y así sucesivamente. Una vez registrados los modelos en este archivo, dentro de Unity en el componente “*Behavior Parameters*” en el campo de “*Behavior Name*” se escribió el nombre del modelo a entrenar que se encuentra escrito dentro del archivo Trainer_config.yaml.

Teniendo el ambiente de entrenamiento ya creado, se abrió el Anaconda Prompt, se cambió la ruta original por la ruta en donde se encuentra la librería ML-Agents descargada de Unity y se activó el *environment* creado. Una vez activado el *environment* y teniendo Unity abierto en la escena donde se encuentra el ambiente de entrenamiento y además habiendo escrito el nombre del modelo a entrenar en el componente de “*Behavior Parameters*”, el último paso corresponde a la ejecución del modelo de DRL a través de la consola de Anaconda.

Una vez finalizado el entrenamiento, se crea automáticamente una carpeta con el nombre del id del entrenamiento y dentro de esa carpeta se encuentra un archivo csv con el nombre del id. Dentro de este último archivo se encuentran todas las impresiones que se visualizaron en la

consola durante el entrenamiento. A su vez, en la carpeta se muestra el modelo entrenado en un archivo con extensión “.nn” tal cual como se muestra en la Figura N°63.



The screenshot shows a file explorer window with the following structure:

- Path: ml-agents-release_1 > models > ejemplo >
- Table with columns: Nombre, Fecha de modificación, Tipo, Tar
- Row 1: Ejemplo (Carpeta de archivos), 07/02/2021 12:33 a. m.
- Row 2: ejemplo.nn (Archivo NN), 07/02/2021 12:33 a. m.

Nombre	Fecha de modificación	Tipo	Tar
Ejemplo	07/02/2021 12:33 a. m.	Carpeta de archivos	
ejemplo.nn	07/02/2021 12:33 a. m.	Archivo NN	

Figura N°63: Modelo de DRL entrenado.

Fuente: Elaboración propia

Este archivo es arrastrado dentro de Unity y posteriormente arrastrado dentro del campo “Model” que se encuentra dentro del componente “Behavior Parameters tal cual como se muestra en la Figura N°64. Cuando esa acción es realizada y se reproduce la escena, el agente se desplaza automáticamente dado que ya cuenta con un modelo de DRL ya entrenado para seguir a un objeto en 3D. El mejor modelo fue utilizado para todos los agentes (Modelos en 3D de los zombies) y fueron arrastrados a los terrenos originales del videojuego. De igual modo, se reemplazó el objeto del cubo en 3D y se colocó como nuevo componente a seguir al personaje en primera persona del jugador. Mediante esta implementación, el zombie persigue al usuario y sumado con un script de ataque que reproduce una animación del zombie atacando y restándole una vida al usuario se logra el movimiento completo del enemigo.

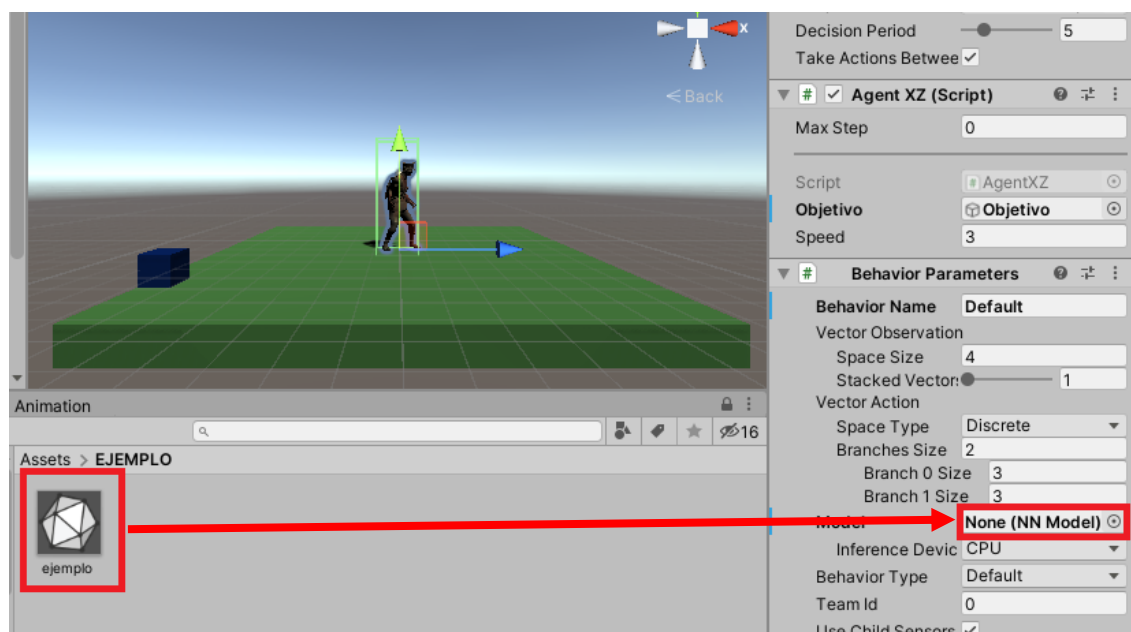


Figura N°64: Utilización del modelo entrenado dentro del Agente.
Fuente: Elaboración propia

3.15 Implementación del servicio de Hosting

Una vez finalizado el desarrollo del videojuego, se realizó la exportación respectiva. Se creó una carpeta para guardar los 3 archivos que fueron creados. Los 3 archivos que se crearon al momento de la exportación son el Index.html, la carpeta “TemplateData” que almacena todos los íconos e imágenes que el archivo Index.html posee y la carpeta “Build” que contiene todos los Assets en general que se han utilizado como los modelos de 3D de los personajes, los terrenos, los sonidos, las armas, entre otros. Para utilizar el servicio de hosting de Firebase primero se descargó Node.js que es el framework de backend para JavaScript. Node.js se descargó desde la página oficial <https://nodejs.org/> en la sección de descargas. Teniendo exportado el videojuego y habiendo instalado Node.js, lo siguiente fue abrir Node.js command prompt que es la consola de Node.js y se ejecutó el comando “Npm install -g firebase-tools”. Lo siguiente fue abrir el símbolo del sistema y cambiar la ruta hacia la carpeta específica creada en donde se encuentran los archivos exportados por Unity del videojuego creado. Una vez dentro de la carpeta a través del símbolo del sistema, se escribió y ejecutó el comando “Firebase init” y para luego introducir el comando “Firebase deploy”. Al finalizar el servicio, Firebase otorgó un link final que fue puesto en el navegador de tal modo de poder visualizar el videojuego desarrollado a través de la computadora. Ese link fue compartido a los docentes del curso de lectura para que estos puedan compartirlo con sus estudiantes.

3.16 Implementación de técnicas para el aprendizaje de la comprensión lectora

Para que los estudiantes puedan aprender a través del uso del videojuego se utilizaron 3 formas diferentes para que los niños puedan ejercitar su nivel de comprensión lectora. Estas tres técnicas que se utilizaron dentro del videojuego se basan en los videojuegos desarrollados en la sección de antecedentes de investigación del capítulo anterior.

3.16.1 Narración de historia y oraciones dentro del videojuego

La primera técnica utilizada para que los estudiantes puedan leer dentro del prototipo desarrollado fue la aparición de texto dentro del videojuego. Esta técnica se basa en la aparición de oraciones dentro de los videojuegos desarrollados en Huerta (2017) y en Moreno, Vahos, y Mazo (2019) presentados en el capítulo anterior. Cuando se inicia el videojuego haciendo click en el botón de “Iniciar partida” del menú, el videojuego narra una historia para que el estudiante pueda ir leyendo y con esto aquel pueda ir sumergiéndose dentro del videojuego.

Para narrar una historia inicial, se agregó una nueva escena al videojuego y se le colocó un fondo oscuro. Además se agregó un objeto tipo Texto en la parte inferior de la escena de tal modo que se pueda escribir con un color de letra blanco para que resalte del fondo. Seguidamente, se creó una animación simple para que el objeto de tipo Texto creado se desplace desde la parte inferior hasta la parte superior de la pantalla de tal modo que el estudiante pueda ir leyendo la historia que el texto va narrando. Adicionalmente, se agregó una música de fondo que acompañe a la escena. Esto último se consiguió arrastrando una canción a la escena y se reproduce automáticamente al iniciar la escena.

Del mismo modo, se agregaron oraciones dentro de los propios niveles del videojuego de tal modo que pueda complementar la historia narrada al inicio así como también oraciones que orienten al estudiante para que pueda superar el nivel. Para que aparezcan oraciones dentro del texto, se creó una variable pública de tipo GameObject. Se utilizó la función `GetComponent<Text>().text` con el objetivo de hacer referencia al texto de la variable creada. Las oraciones aparecen por un tiempo de 3 segundos y luego desaparecen de la escena para que el estudiante siga jugando. Para lograr este efecto, se utilizó la función `yield return new WaitForSeconds(3)` y debajo de esta función se volvió a utilizar la función `GetComponent<Text>().text` pero esta vez se igualó a unas comillas dobles sin escribir una oración de tal modo que la oración en la escena desaparece.

3.16.2 Adaptación de textos educativos dentro del videojuego

La segunda técnica a utilizar dentro del videojuego fue la adaptación de los textos tradicionales de lectura para que puedan estar presentes en las escenas que posee el videojuego. Para utilizar las lecturas que se usan de manera tradicional, el docente compartió su material en formato pdf de las lecturas que se suelen utilizar en las clases de lectura para cada grado. Cada pdf fue revisado y se recortaron solamente las lecturas que poseen los documentos de tal modo de tener diferentes imágenes de lecturas para cada grado. Todas las lecturas fueron guardadas dentro de una carpeta que fue arrastrada hacia el proyecto de Unity. Cada lectura fue convertida dentro del videojuego a un formato *sprite* para que pueda visualizarse en la escena. Posteriormente, se agregó una nueva escena y se agregó un objeto de tipo Image a la escena. Se arrastró cada imagen al objeto de tipo Image creado específicamente al campo Source Image. Al realizar esta acción, aparecerá la imagen dentro de la propia escena. Para que la aparición de la lectura sea más dinámica, se crearon 4 diferentes animaciones de tal forma que desplacen las imágenes desde la izquierda, derecha, arriba o abajo hacia el centro de la escena. Además, cada escena donde se muestre la lectura, se agregó un botón que redirija a las preguntas de la lectura en cuestión. Si el estudiante responde adecuadamente las 3 preguntas de la lectura, regresa al nivel en donde perdió sus 5 vidas, de lo contrario, empezará nuevamente desde el primer nivel.

3.16.2.1 Configuración del contenido para cada grado

El contenido de las lecturas por cada grado fue clasificado de tal modo que, a cada estudiante de un determinado grado, le aparezcan lecturas que corresponden a su nivel de lectura. Para realizar la clasificación de contenido de estas lecturas, al momento que el estudiante da click en el botón de “Ingresar” que posee la escena del Login, se verifica que el nombre y el apellido ingresado por el alumno se encuentren dentro de la base de datos buscando el fichero correspondiente, de encontrarse la información del alumno, se obtiene el valor del atributo grado que se encuentra dentro del registro y dentro de la función del script de la base de datos creado en Unity se creó una condicional con un *Switch Case* en donde se verificaba el valor del atributo del grado del alumno. De tener este el valor de 1 lo redirige al estudiante al Menú 1, de tener este el valor de 2, lo redirige al estudiante al Menú 2 y así sucesivamente. En total se crearon 6 diferentes menús para cada grado. Además, cada escena de Menú posee un objeto vacío llamado “Identificador”. Este objeto vacío se le colocó una etiqueta en función al número del menú. Por ejemplo, si la escena es el Menú 1 que le corresponde al primer grado, la etiqueta para el objeto identificador será de 1. Cada menú posee su propio identificador. A

su vez, este identificador que fue creado, se le incluyó la función DontDestroy(). Cuando a un objeto de Unity posee esta función, cuando cambia de escena, el objeto se mantiene siempre, no se destruye por lo que este identificador aparece dentro de todas las escenas del videojuego. Cuando el estudiante pierde sus 5 vidas encontrándose en un determinado nivel, este posee ya el identificador asociado y se activa un script para verificar si la escena posee un identificador y el valor de este. En función al valor que posea, se carga la escena donde se encuentra la lectura correspondiente al grado que le corresponde ya definido en el código y a su vez cada botón de las preguntas de la lectura redirecciona al estudiante a la escena correspondiente a las preguntas de la lectura que el estudiante acaba de leer. Esta misma lógica se aplica cuando el estudiante pierde, dado que se verifica si la escena de Game Over posee un identificador y en función al valor de este es redirigido al Menú correspondiente.

3.16.3 Sección de estrategias de lectura

Como última técnica para que los estudiantes puedan mejorar su nivel de comprensión lectora, se incluyó una sección propia de estrategias tal como realizó Huerta (2017) en su propio videojuego presentado en el capítulo anterior en la sección de antecedentes. Para ir a la sección de estrategias, el menú posee un botón de estrategias que lleva al estudiante a una escena en donde se muestre una estrategia para que el estudiante pueda ponerlas en práctica y como consecuencia pueda leer mucho mejor. Las estrategias presentadas en el videojuego serán las propuestas en Roa (2020) presentadas en la sección 2.2.7 del capítulo anterior pero adaptadas a un lenguaje sencillo de tal modo que cualquier estudiante lo pueda entender.

Para crear la sección de estrategias se creó una nueva escena y se agregó un nuevo objeto Canvas. Dentro de ese objeto, se agregó un nuevo objeto tipo Texto que llevó como título “Estrategia 1” y que se encuentra ubicado en la parte superior, el número de la estrategia va variando conforme se vayan agregando nuevas escenas. Debajo del título, se agregó un nuevo objeto tipo Texto en donde se escribe la estrategia de tal modo que se encuentre al centro de la escena. Debajo de la estrategia, se agregaron dos botones. Uno se encuentra ubicado en la parte izquierda de la escena y uno en la parte derecha. El primer botón es para que el estudiante se dirija a la siguiente estrategia y el segundo botón es para que el estudiante regrese al menú principal.

3.17 Aplicación del aprendizaje basado en videojuegos

El aprendizaje basado en videojuegos se logra mediante la utilización de un videojuego con el objetivo de mejorar alguna habilidad en un grupo de estudiantes o el aprendizaje de un determinado tema (Trends E., 2016). El videojuego desarrollado en este trabajo de investigación sirvió como un complemento tecnológico para el docente para que pueda mantener a sus estudiantes más enfocados dentro de las clases de lectura, pero a su vez este videojuego contiene, como se ha visto anteriormente, 3 diferentes técnicas de lectura para que los estudiantes puedan ejercitar su habilidad de comprensión de textos.

La aplicación del aprendizaje basado en videojuegos se logró mediante la utilización del videojuego dentro de las clases de lectura de cada grado de la I.E.P Colegio “Mater Christi”. Para lograr este objetivo fue necesario distribuir el tiempo que los alumnos utilizarán para sus clases tradicionales de lectura de manera virtual y el tiempo del uso del videojuego de tal modo que se tenga una nueva metodología de enseñanza para poder mejorar el nivel de lectura de los estudiantes. A esta distribución de tiempo en donde se combinan las clases tradicionales y el uso del videojuego tiene como nombre protocolo de entrenamiento o de aprendizaje (Cheng y Tsaib, 2019).

3.17.1 Desarrollo del protocolo de aprendizaje

Según los trabajos de investigación de Didehbani et al. (2016) y Cheng y Tsaib (2019) desarrollados en el capítulo anterior, los protocolos de aprendizaje suelen desarrollarse entre dos y tres semanas. Esto es debido a que en este tipo de trabajos de investigación se dependen de diversos factores como la autorización del director (a), de los docentes así como de los padres de familia. Otro factor importante para la utilización del videojuego dentro de las salas de cómputo de un colegio es particular es la tecnología que poseen dado que si las computadoras con la que cuenta el colegio son antiguas, el videojuego no podrá jugarse de la mejor manera.

Dada previa coordinación con el colegio, este videojuego se utilizó por 5 semanas para el entrenamiento de los estudiantes dado que en esta oportunidad no se depende de las computadoras del colegio, sino que cada estudiante ha jugado desde sus hogares con la computadora que posean para acceder a sus clases virtuales.

Los pasos para la realización del protocolo de aprendizaje ya han sido planteados en la sección 3.9.2 en donde se establecieron las fechas para iniciar el entrenamiento. Estas fechas tanto para la realización del pretest, como el entrenamiento y el postest no fueron las mismas

para todos los grados dado que no todos llevan el curso de comprensión lectora el mismo día por lo que estas fechas han variado.

De igual modo, el tiempo que establecen tanto Cheng y Tsaib (2019) como Didehbani et al. (2016) en sus videojuegos son de 15 minutos para un protocolo de entrenamiento de 2 semanas. Dado que en esta ocasión, el periodo de entrenamiento ha sido casi el triple de los que los anteriores autores han desarrollado, el uso del videojuego fue por 10 minutos para cada grado y las clases tradicionales fueron de 35 minutos dado que el total de tiempo para las clases tradicionales son de 45 minutos.

3.18 Metodología para la medición de los resultados

En esta sección, se detallará la metodología seguida para la medición y presentación de los resultados con respecto a la metodología aplicada, experimentos y entrenamiento de los estudiantes. Con respecto a los 7 modelos desarrollados, se brindaron los resultados generales con respecto a estos y se seleccionó el mejor de estos modelos para hacer una descripción detallada del mismo dado que en los trabajos de investigación solamente se suele describir un único modelo. Posteriormente, en esta sección se explicarán los cálculos necesarios para poder realizar el análisis estadístico para los resultados del pretest y posttest y los resultados cualitativos de la encuesta de satisfacción. Finalmente, se presentará el cronograma de actividades para la realización de las actividades del presente proyecto de investigación.

3.18.1 Medición de resultados - Metodología aplicada al videojuego

Dado que la metodología aplicada fue la metodología en cascada para el desarrollo del videojuego, se presentaron los resultados obtenidos por cada una de sus 5 fases.

Con respecto a la fase de comunicación, se presentó el total de requerimientos recopilados por parte del docente y con respecto a la clasificación de estos, se presentó cuantos fueron funcionales y cuantos no funcionales.

Acerca de la fase de planeación, se reportó el total de actividades planificadas por cada etapa, así como el tiempo total que se tomó para la realización de cada etapa y por las 5 etapas juntas.

En relación con la fase de modelado, se mostró el total de casos de uso desarrollados para el videojuego así como la especificación de cada uno de ellos. De igual modo, se presentó el total de bosquejos realizados y el total de arquitecturas diseñadas para el videojuego. También, se presentó el total de modelos en 3D que fueron descargados para los enemigos, terrenos y las

armas disponibles para los estudiantes. De igual modo, se presentó el total de links utilizados para la música y sonidos del videojuego. Finalmente, se reportó el total de lecturas para que se encuentren dentro del videojuego y la cantidad para cada grado, así como la cantidad de preguntas de opción múltiple para cada lectura.

Con respecto a la fase de desarrollo, se presentaron el total de escenas y el total de scripts que fueron construidos para el videojuego. Además, se presentó el total de niveles que posee el videojuego y el total de modelos en 3D implementados. También, se presentaron el total de tecnologías aplicadas para el desarrollo del prototipo. En relación con las técnicas de lectura, se mostraron el total de lecturas y oraciones implementadas dentro del videojuego, así como el total de estrategias de lectura introducidas. De igual modo, se presentaron el total de audios dentro del videojuego, así como el total de menús que se crearon. Finalmente, se presentaron el total de modelos de DRL que fueron entrenados para implementar al mejor agente dentro del videojuego.

Finalmente, en relación con el despliegue, se detalló si hubo alguna observación por parte de los profesores o si se aprobó el videojuego para su utilización dentro de las aulas virtuales sin observaciones.

3.18.2 Medición de resultados – Experimentos de DRL

Una vez entrenados los 7 modelos de DRL, se presentaron los gráficos del comportamiento del entrenamiento del agente por cada uno de los modelos desde el Tensorboard. Para presentar los resultados de los Modelos de DRL del presente proyecto de investigación, se realizaron dos presentaciones. En primer lugar, se presentaron los resultados generales de los 7 modelos por cada gráfica proveniente del Tensorboard. En este análisis general, se describió el comportamiento de los 7 modelos del entrenamiento a través de las métricas de *Cumulative Reward*, *Episode Length*, *IsTraining*, *PolicyLoss*, *ValueLoss*, *Entropy*, *Extrinsic Reward*, *Extrinsic Value Estimate* y *Learning rate*.

Al finalizar este análisis, se presentó un cuadro resumen de las métricas obtenidas con el objetivo de verificar cuál es el mejor modelo. Las métricas principales para seleccionar el mejor modelo fueron el *PolicyLoss*, *ValueLoss*, *Entropy*, *Cumulative Reward* tal cual como se detalló en la sección 3.14.3.2.

Una vez seleccionado el mejor modelo, la segunda presentación de resultados consistió en describir a detalle este modelo seleccionado dado que los autores en sus trabajos de investigación solo entrenan y presentan 1 solo modelo de DRL. Estas dos presentaciones de los resultados (generales de los modelos de DRL e individual del mejor modelo) serán explicados a detalle en las sub secciones 3.18.2.1 y 3.18.2.2 respectivamente.

3.18.2.1 Resultados generales de los modelos de DRL desarrollados

Los resultados provenientes del Tensorboard sobre los modelos de DRL se presentan en 9 métricas que serán descritas a continuación:

- *Cumulative Reward*: Muestra la recompensa acumulada de todos los agentes involucrados para cada entrenamiento. En un entrenamiento exitoso, este valor debe de aumentar para cada episodio dado que el objetivo del agente es incrementar su recompensa (Adamsson, 2018). La fórmula para el cálculo de la recompensa acumulada se presentó anteriormente en la Ecuación (41) del capítulo anterior. Un ejemplo de esta gráfica se presenta en la Figura N°65 - (a).
- *Episode Length*: Presenta el número promedio de acciones que un agente suele realizar durante un episodio del entrenamiento (González, 2018). Un ejemplo de esta gráfica se presenta en la Figura N°65 - (b).
- *IsTraining*: Revela si el entrenamiento ha sido continuo o si hubo una interrupción del entrenamiento para luego continuarlo. Cuando el entrenamiento no se interrumpe, el gráfico mostrará un 1, si es que el gráfico se interrumpe en un determinado punto del entrenamiento cambia a 0. Un ejemplo de esta gráfica se presenta en la Figura N°65 - (c).
- *PolicyLoss*: Exhibe los cambios en el proceso de decisión de un agente (González, 2018). Una Policy o política es un conjunto de acciones que el agente debe de ejecutar para tener un comportamiento determinado ante situaciones diferentes durante el entrenamiento. Para un entrenamiento exitoso, el Policy Loss, debería de ir a la baja y su gráfica debe de ser casi constante (Adamsson, 2018). La fórmula para el cálculo del Policy Loss se presentó anteriormente en la Ecuación (47) del capítulo anterior. Un ejemplo de esta gráfica se presenta en la Figura N°65 - (d).
- *ValueLoss*: Representa el error del valor de la recompensa obtenida de la salida de la red neuronal (González, 2018). Esta gráfica inicia tendiendo hacia el alta pero luego debe de disminuir. Lo anterior mencionado es debido a que un determinado agente, cuando ejecuta una política espera recibir una recompensa, no obstante, recibe otra

recompensa que es la real. Esa diferencia suele ser alta al inicio de un entrenamiento dado que el agente está empezando a interactuar con el entorno. No obstante, el algún momento del entrenamiento, el agente empezará a acercarse más a la recompensa real obtenida por lo que la diferencia debe de ser mínima por lo que el valor debería ir a la baja (González, 2018). La fórmula para el cálculo del Value Loss se presentó anteriormente en la Ecuación (46) del capítulo anterior. Un ejemplo de esta gráfica se presenta en la Figura N°65 - (e).

- *Entropy*: Representa la aleatoriedad de las decisiones del agente. Cuanto mayor sea la entropía, es más difícil para el agente tomar una decisión para ejecutar una acción determinada por lo que en un entrenamiento exitoso la entropía debería de disminuir lentamente porque la toma de decisiones se vuelve más precisa al avanzar el entrenamiento del agente. La fórmula para el cálculo de la entropía se presentó anteriormente en la Ecuación (49) del capítulo anterior. Un ejemplo de esta gráfica se presenta en la Figura N°65 - (f).
- *Extrinsic Reward*: Esta gráfica es la misma que la gráfica de *Cumulative Reward* dado que las recompensas que se le asignan al agente son otorgadas a través del ambiente de entrenamiento (Adamsson, 2018). No obstante, a las recompensas generadas del ambiente de entrenamiento también se le pueden sumar un componente llamado *Curiosity* de tal modo de poder asignarle recompensas adicionales al agente por encontrar estados que puedan sorprenderlo y que generen curiosidad. Esto permite al agente generar nuevas estrategias para poder aprender a realizar una determinada actividad. De sumar este elemento el nombre de la gráfica cambia a *Intrinsic Reward* (González, 2018). En este proyecto en particular, solo se utilizaron las recompensas generadas del propio ambiente de entrenamiento. Un ejemplo de la gráfica *Extrinsic Reward* se presenta en la Figura N°65 - (g).
- *Extrinsic Value Estimate*: Revela la recompensa acumula promedio que el agente podría recibir en la próxima iteración. Aumenta en medida que aumenta la recompensa acumulada. En un entrenamiento exitoso, esta gráfica debería de ir en aumento de forma constante (González, 2018). Un ejemplo de esta gráfica se presenta en la Figura N°65 - (h).
- *Learning rate*: Muestra la tendencia de la tasa de aprendizaje y para un aprendizaje exitoso esta tasa debería de disminuir conforme avanza el entrenamiento (González, 2018). Al inicio del entrenamiento, el cerebro trata de realizar acciones de manera

aleatoria tratando de determinar las mejores acciones para aumentar la recompensa. Conforme avanza el entrenamiento, el cerebro intenta escoger la mejor elección haciendo que la tasa de aprendizaje disminuya (González, 2018). En este tipo de entrenamiento, la tasa de aprendizaje (η) no se mantiene fija como se presentó en el capítulo anterior en la Ecuación (19) sino que va disminuyendo conforme avanzan las iteraciones. De este modo, esta tasa de aprendizaje variable se representa de la forma η_k , donde k representa el número de iteración del modelo en entrenamiento. De este modo, el decrecimiento de la de la tasa de aprendizaje se representa mediante la siguiente ecuación:

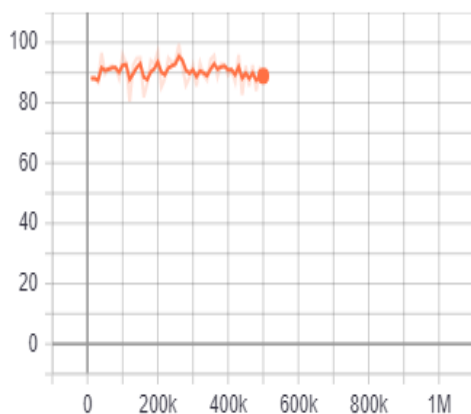
$$\eta_k = (1 - \alpha_k)\eta_0 + \alpha_k\eta_t \quad (64)$$

Donde:

- $\alpha_k = \frac{k}{t}$. Luego de la iteración t , el valor de α_k se mantiene de forma constante.
- η_0 es el valor inicial establecido para el hiperparámetro de la tasa de aprendizaje.
- η_t es el valor de la tasa de aprendizaje en el tiempo t .

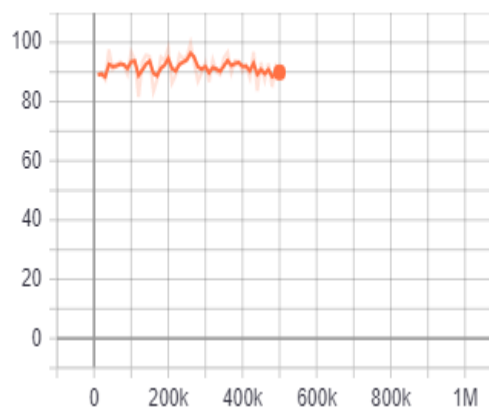
Un ejemplo de esta gráfica se presenta en la Figura N°65 - (i).

Cumulative Reward
tag: Environment/Cumulative Reward



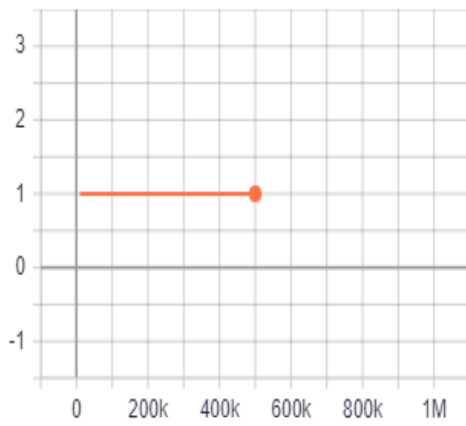
(a)

Episode Length
tag: Environment/Episode Length



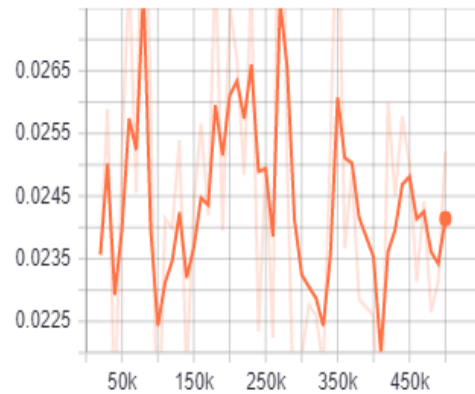
(b)

Is Training



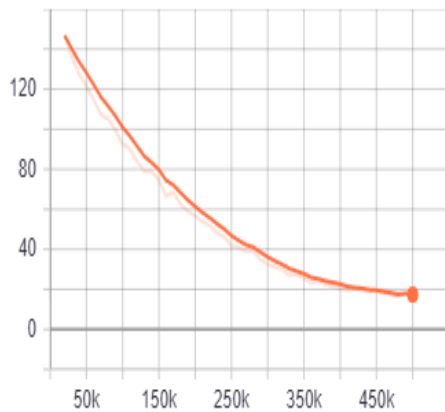
(c)

Policy Loss
tag: Losses/Policy Loss



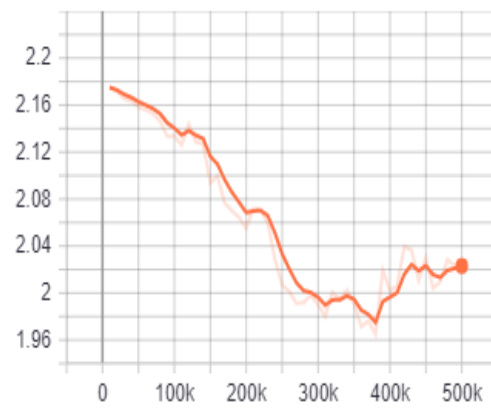
(d)

Value Loss
tag: Losses/Value Loss



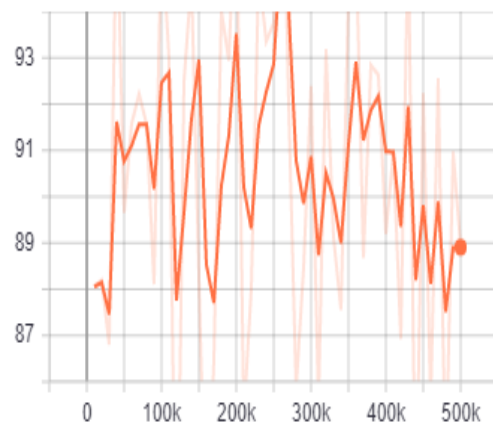
(e)

Entropy
tag: Policy/Entropy



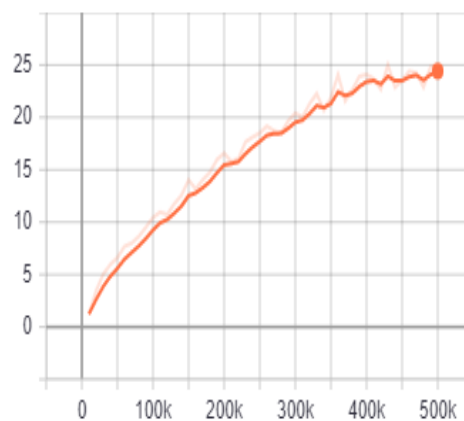
(f)

Extrinsic Reward
tag: Policy/Extrinsic Reward

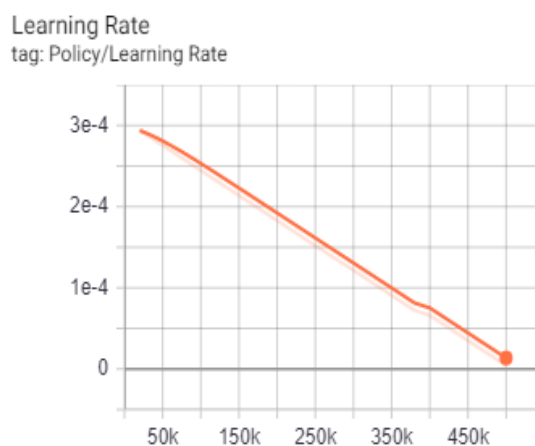


(g)

Extrinsic Value Estimate
tag: Policy/Extrinsic Value Estimate



(h)



(i)

Figura N°65: Ejemplo de los resultados del Tensorboard para los experimentos de DRL.
Fuente: Elaboración propia.

Como se había mencionado a inicios de la presente sección, la idea fue poder realizar una descripción general de los 7 modelos entrenados de tal modo que se pueda seleccionar al mejor modelo en función al total de recompensa acumulada. Para realizar la selección de este modelo, se realizó una tabla resumen de los 9 modelos del Tensorboard en donde se visualizaron los resultados por cada métrica y se agregó una columna adicional que es el tiempo que se ha demorado cada modelo en entrenar al agente de tal modo que se pudo tomar una decisión para seleccionar el mejor de los 7 modelos de DRL.

3.18.2.2 Resultados del mejor modelo de DRL

Una vez seleccionado el mejor modelo, se procedió a describir los resultados de este en función a los resultados obtenidos por la consola en cada episodio del modelo. Además, se analizaron los resultados otorgados por el Tensorboard de manera individual para este modelo y con respecto a los resultados de la red neuronal, se presentaron los valores de entrada y salidas de la red, así como la matriz de pesos. Estos resultados se describen en las próximas sub secciones.

3.18.2.2.1 Resultados del entrenamiento del agente por episodio

En primer lugar, se presentó los resultados impresos en la consola de Anaconda Prompt. Estos resultados del entrenamiento se presentaron en una tabla en donde figuraba el *Cumulative Reward*, *Episode Length*, *Is Training*, *Policy Loss*, *Value Loss*, *Entropy*, *Extrinsic Reward*, *Extrinsic Value Estimate*, *Learning Rate* y el número de pasos acumulados por cada episodio

(*Step*). Además, se realizó una descripción de cada resultado mencionado anteriormente con respecto a la evolución que tuvo durante el entrenamiento.

3.18.2.2.2 Tabla de entradas y salidas de la red

En segundo lugar, se presentó una tabla con las entradas a la red normalizadas aplicando la Ecuación (63) y sus respectivas salidas. Dado que son redes profundas, solo se hizo una visualización de las primeras salidas de la red en donde se visualizaron 4 columnas que representan los datos de entrada y 2 columnas adicionales que representan las salidas con respecto a la primera y segunda neurona de la capa de salida.

3.18.2.2.3 Matriz de pesos

Como parte de la presentación de los resultados del mejor modelo, se presentaron los resultados con respecto a los pesos reajustados de la red. Dado que la mayoría de los modelos tienen más de 100 capas ocultas, solamente se detalló las primeras 6 capas ocultas. Se presentó la tabla de los pesos iniciales para las primeras 6 capas ocultas de la red antes del entrenamiento. Además, se reportó los pesos iniciales para las neuronas de la capa de salida antes del entrenamiento. Posteriormente, se exhibió la tabla de pesos actualizados para las primeras 6 capas ocultas y para la capa de salida de la red neuronal. Es preciso mencionar que para la actualización de los pesos de la red se calculó el error por cada neurona de la capa de salida que corresponde a la Ecuación (22) para finalmente realizar la sumatoria y obtener el error total. Para aplicar el pase en *Backward* y actualizar los pesos de la red, se aplicó la regla de la cadena que corresponde a la Ecuación (13), Ecuación (14), Ecuación (15), Ecuación (16), Ecuación (17), Ecuación (18), Ecuación (19), Ecuación (20), Ecuación (21), Ecuación (23), Ecuación (24), Ecuación (25), Ecuación (26), Ecuación (27), Ecuación (28), Ecuación (29), Ecuación (30), Ecuación (31), Ecuación (32) y termina con la Ecuación (33).

3.18.3 Medición de resultados - Entrenamiento de los estudiantes

Como se mencionó al inicio del capítulo, el enfoque de este proyecto de investigación es cuantitativo-cualitativo. Por el lado cuantitativo, se realizó una prueba T para muestras independientes por cada grado con el objetivo de determinar si existen una mejora significativa de tal modo que se pueda validar o rechazar la hipótesis general planteada en este proyecto de investigación. Por el lado cualitativo, los estudiantes realizaron una encuesta de satisfacción de tal modo que los resultados de las preguntas puedan presentarse gráficamente con el objetivo de determinar si el videojuego fue del agrado de los estudiantes y qué mejoras sugieren que debe haber dentro del videojuego para que este sea más disfrutable. La ampliación de la

medición de los resultados cuantitativos y cualitativos, se presentan en las sub secciones 3.18.3.1 y 3.18.3.2.

3.18.3.1 Análisis cuantitativo de las evaluaciones por cada grado

El análisis cuantitativo del videojuego se dividió en dos tipos de análisis: cuantitativo descriptivo y cuantitativo inferencial. Con respecto al análisis cuantitativo descriptivo se realizó un resumen estadístico de los resultados tanto para el pretest y posttest para el grupo de control y el grupo experimental para cada grado. En este resumen, se presentó el promedio de notas obtenidas por cada evaluación, la desviación estándar y la mediana de las notas. Finalmente, se presentó los resultados de los indicadores en el pretest para el grupo de control y experimental, así como los resultados de in indicadores en el posttest para el grupo de control y experimental. En la Figura N°66, se presenta un resumen de las actividades descritas anteriormente para la presentación de los resultados con respecto al análisis cuantitativo descriptivo.

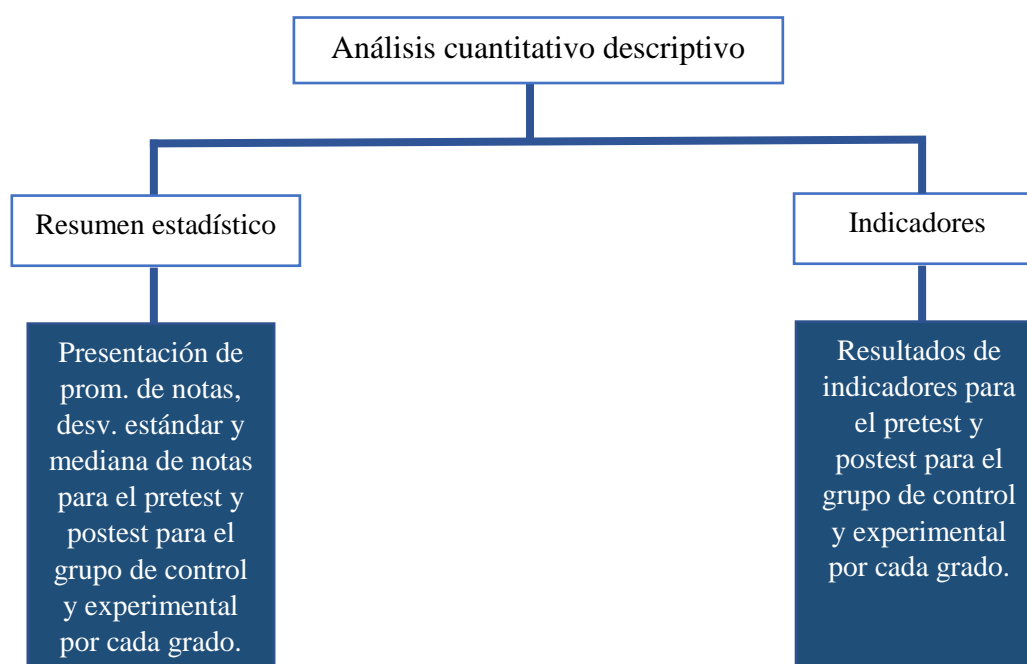


Figura N°66: Resumen de las actividades para el análisis cuantitativo descriptivo.

Fuente: Elaboración propia.

Con respecto al análisis cuantitativo inferencial, dado que se definió la aplicación de la prueba T para muestras independientes para las evaluaciones por cada grado, se aplicaron los pasos establecidos en la sección 2.2.13 del capítulo anterior. Esta prueba se aplicó tanto para el pretest como el posttest para cada grado con el objetivo de determinar si la utilización del videojuego dentro del salón de clases ha ocasionado una mejora significativa en el nivel de comprensión lectoras de los estudiantes de primaria. La secuencia de pasos a seguir para validar

o rechazar la hipótesis general establecida al inicio del documento, se presenta en la Figura N°67. Para aplicar esta técnica estadística, se aplicaron la Ecuación (54), Ecuación (55), Ecuación (56), Ecuación (57), Ecuación (58), Ecuación (59), Ecuación (60) y Ecuación (61) descritas en el capítulo anterior.

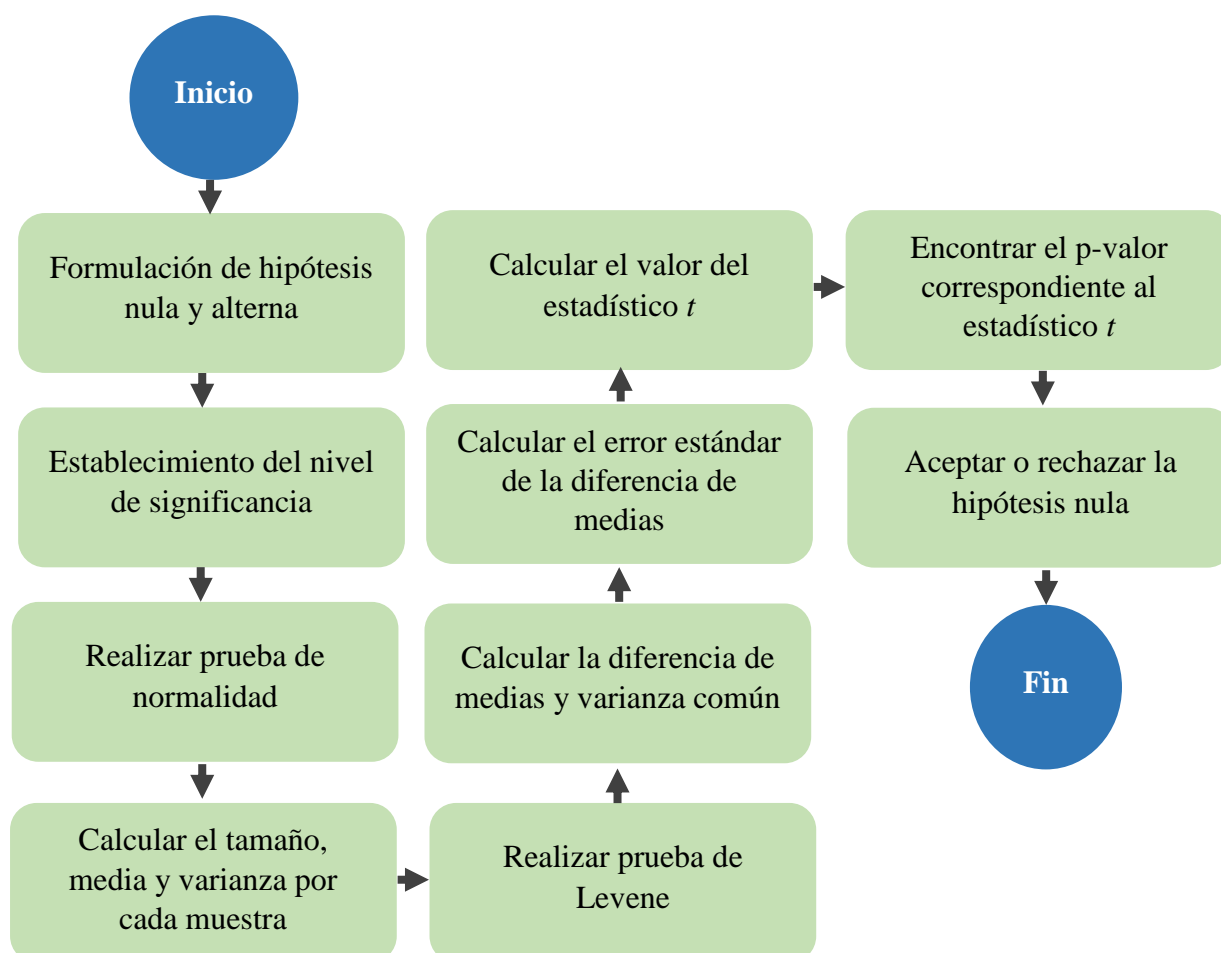


Figura N°67: Secuencia de actividades para aplicar la prueba T para muestras independientes. Fuente: Elaboración Propia.

3.18.3.1.1 Resumen de resultados generales de las evaluaciones

Una vez realizadas las pruebas T para muestras independientes para el pretest y postest por grado, se realizó un cuadro resumen en donde se observan los resultados generales obtenidos por cada grado de tal modo de observar el rechazo o aceptación de la hipótesis general planteada.

3.18.3.2 Análisis cualitativo de la encuesta de satisfacción

Con respecto al análisis cualitativo, se presentaron los resultados de la encuesta de satisfacción realizada a los grupos experimentales de cada grado. Dado que la primera pregunta es de opción múltiple, se presentó a través de un diagrama de barras la cantidad de usuarios

que les había parecido el videojuego bueno, regular y malo. Con respecto a las 3 preguntas siguientes que son para redactar. Se escogieron solamente la opinión de tres alumnos en representación del grupo experimental para cada grado con el objetivo de poder redactarlas y dar una visión general de lo que cada grado sugiere incluir dentro del videojuego.

3.19 Cronograma de actividades

Para poder realizar las actividades del proyecto de forma ordenada y a su vez para dar seguimiento a aquellas, fue necesaria la elaboración de un cronograma de trabajo. El cronograma general realizado para el desarrollo del presente proyecto se presenta en la Figura N°68.

Actividades	Fecha Inicio	Fecha Final	4-Nov-19	9-Nov-19	11-Nov-19	16-Nov-19	18-Nov-19	20-Abr-20	04-May-20	15-Set-20	24-Set-20	28-Set-20	29-Set-20	05-Oct-20	06-Nov-20	05-Ene-21	13-Mar-21	15-Mar-21	22-Mar-21
Recolección de requerimientos del docente	04-11-19	09-11-19	■	■															
Verificación de tiempo y recursos necesarios	11-11-19	16-11-19			■	■													
Desarrollo del modelado del comportamiento del videojuego y bosquejos iniciales de las escenas	18-11-19	20-04-20					■	■											
Desarrollo del videojuego en función a las especificaciones y diseños realizados	04-05-20	15-09-20							■	■									
Desarrollo de las pruebas de funcionalidad individual y general	15-09-20	24-09-20								■	■								
Aprobación del videojuego por parte de docentes	28-09-20	29-09-20										■	■						
Entrenamiento de los estudiantes de primaria	05-10-20	06-11-20												■	■				
Presentación, análisis e interpretación de resultados obtenidos	05-01-21	13-03-21														■	■		
Presentación de conclusiones y recomendaciones del proyecto	15-03-21	22-03-21																■	■

Figura N°68: Cronograma general para el desarrollo del proyecto de investigación.

Fuente: Elaboración propia.

CAPITULO IV: DESARROLLO DE LA SOLUCIÓN

En este capítulo se explicará en primer lugar, las alternativas existentes para desarrollar el prototipo y se detallará la propuesta escogida para el desarrollo del videojuego. De igual manera se desarrollarán las 3 fases iniciales del presente proyecto de investigación que constan del desarrollo propio del prototipo a través de la aplicación de la metodología en cascada, el desarrollo del protocolo de aprendizaje y el entrenamiento de los estudiantes.

4.1 Determinación y evaluación de las alternativas de solución

Para que los estudiantes puedan mejorar su nivel de comprensión lectora, se ha propuesto como solución el desarrollo de un videojuego de tal modo de poder aprovechar las bondades de este para poder ejercitar la habilidad de lectura de los estudiantes y para que a su vez sirva como una herramienta tecnológica adicional para el docente. Para realizar este objetivo, se recopilaron y analizaron diversos trabajos de investigación de videojuegos aplicados en un contexto educacional con el objetivo de recopilar las metodologías más resaltantes para el desarrollo del videojuego, los diferentes tipos de realidad virtual utilizados, las diferentes técnicas de aprendizaje utilizados, así como las técnicas estadísticas para realizar la evaluación de los resultados obtenidos.

Los tipos de realidad virtual más resaltantes dentro de los trabajos de investigación desarrollados fueron la realidad virtual no inmersiva y semi inmersiva. El tipo de realidad virtual no inmersiva se caracteriza por la utilización de una pantalla a través del control de mouse y teclado o pantalla táctiles. Por otro lado, la realidad virtual semi inmersiva se caracteriza por la utilización de dos lentes que normalmente estos se encuentran dentro de los cascos de realidad virtual. A través de la utilización de estos dispositivos, los usuarios pueden utilizar los movimientos de su cabeza para poder controlar o visualizar diversas escenas del videojuego. Se descartó el uso de la realidad virtual semi inmersiva dado que como actualmente se está viviendo en el mundo las consecuencias de la aparición del Covid-19, conseguir los componentes para implementar este tipo de realidad resultaría más costoso, así como la entrega de estos cascos a las casas de cada estudiante sería más complicado. Por lo tanto, se optó por el tipo de realidad virtual no inmersiva dado que es la más común de aplicar dentro de los videojuegos más tradicionales y resulta más fácil de implementar por parte de Unity. Además, los estudiantes podrán aprovechar la tecnología que poseen en sus hogares (computadoras y teclados) para que puedan controlar el videojuego.

Por otro lado, se estudiaron las diferentes técnicas utilizadas en los trabajos de investigación para poder ejercitar el nivel de comprensión lectora de los usuarios. Las técnicas más resaltantes fueron la aparición de textos dentro del propio videojuego, así como la aparición de lecturas con preguntas de retroalimentación con el objetivo de que los estudiantes puedan leer dentro del propio videojuego y responder preguntas de opción múltiple que corresponden a la lectura leída. De igual modo, diversos trabajos de investigación introducen secciones de estrategias de lectura dentro del propio videojuego con el objetivo de que los estudiantes puedan tener en cuenta estas y puedan desenvolverse de una mejor manera dentro de las lecturas que plantea el videojuego.

De igual modo, para poder realizar el análisis de los resultados luego del entrenamiento de los estudiantes, los trabajos de investigación apuntan a la utilización del estadístico T de student realizando una pretest y posttest. Es preciso resaltar que esta técnica estadística se divide en 2: prueba T para muestras dependientes y prueba T para muestras independientes. El primero evalúa a un solo grupo antes y después de la intervención aplicando un pretest y posttest a través de la evaluación de una variable numérica que en este caso es la variable de comprensión lectora dado que corresponden a las notas obtenidas de los estudiantes. El segundo corresponde a la evaluación de dos grupos (control y experimental) con el objetivo de comparar los resultados de ambos y determinar si existe una mejora significativa. El grupo de control sigue la metodología tradicional de enseñanza y el grupo experimental utiliza la nueva metodología propuesta a través del uso del videojuego de modo que se pueda verificar que la nueva metodología propuesta es mejor a la metodología tradicional.

Dado que el videojuego fue desarrollado en un contexto donde los estudiantes aprenden a través de una educación a distancia y no hay manera de verificar si realmente han ingresado al videojuego, se decidió implementar una base de datos. Para seleccionar la base de datos adecuada para el videojuego, se realizó una comparación entre la base de datos MySQL Database en representación de las bases de datos relacionales y Firebase Database en representación de las bases de datos no relacionales. En función al trabajo desarrollado en Ohyver et al., (2019). El trabajo en mención realizó un análisis estadístico para verificar la existencia de una diferencia significativa entre el tiempo que se demora registrar, consultar, eliminar y actualizar datos. La base de datos que mostró superioridad con respecto a las 4 operaciones fue la base de datos de Firebase. La idea de implementar una base de datos es poder registrar los datos de los estudiantes con el objetivo de verificar que verdaderamente han entrado al videojuego.

Adicionalmente, se realizó una recopilación de videojuegos desarrollados aplicando Inteligencia Artificial con el objetivo de hacer este prototipo mucho más realista creando a un agente enemigo de tal modo que pueda enfrentarse a los estudiantes. La técnica utilizada en los diversos trabajos fue *Deep Reinforcement Learning* dado que según los propios autores es la técnica más común que se suelen aplicar en los videojuegos. Esta técnica combina 2 técnicas: Deep Learning que corresponde a la teoría de redes neuronales y Reinforcement Learning que corresponde a la aplicación de premios y castigos que son otorgados al agente cada vez que ejecute la acción deseada y cada vez que el agente se equivoca de tal modo que el objetivo del agente es acumular la mayor cantidad de recompensa y así logre aprender la tarea encomendada.

4.1.1 Propuesta de solución

La propuesta del presente proyecto de investigación para que los estudiantes puedan mejorar su habilidad de comprensión lectora consistió en el desarrollo de un videojuego que contiene una base de datos en tiempo real de Firebase para el registro y Login de los estudiantes de tal modo de poder verificar su ingreso al videojuego. Por otro lado, este videojuego contiene 3 técnicas de aprendizaje diferentes para que los estudiantes puedan ejercitar su nivel de comprensión lectora. La primera técnica consiste en la aparición de textos dentro del propio videojuego a través de la narración de una historia al iniciar el juego y la aparición de oraciones dentro del propio videojuego. La segunda técnica de aprendizaje consiste en la aparición de lecturas dentro del videojuego que contienen 3 preguntas diferentes. Estas lecturas aparecen cada que el estudiante pierda sus 5 vidas y este debe responder adecuadamente las 3 preguntas para que pueda regresar al nivel donde se encontró de lo contrario, inicia el videojuego desde cero. La última técnica de lectura consiste en la aparición de estrategias de lectura en una sección aparte del videojuego para que los estudiantes lo puedan tener en cuenta al momento de leer los textos que se les presente. El control del videojuego es a través de la utilización del teclado y mouse de las computadoras de los estudiantes aplicando de esta manera realidad virtual no inmersiva. Para hacer el videojuego más agradable y desafiante para los estudiantes, se implementó la técnica de *Deep Reinforcement Learning* (DRL) con el objetivo de desarrollar un agente inteligente de tal modo que pueda perseguir a los estudiantes. Al momento de que el agente se acerque al personaje en primera persona controlado por el estudiante y haya una distancia de 1, se activa un script de ataque de tal modo que reste vida al estudiante y al perder sus 5 vidas sea redirigido a las lecturas. Para implementar esta técnica, se desarrolló un ambiente de entrenamiento en donde los agentes que en este caso fueron los modelos en 3D de

los zombies entrenaron persiguiendo un cubo en 3D. Para el aprendizaje de los agentes, se entrenaron 7 modelos de DRL en donde 3 de estos fueron propuestas de autores que desarrollaron sus propios agentes y 4 fueron propuestas que se formaron en función a los valores de los 3 modelos iniciales dado que, para realizar aquellos modelos, los autores consultaron a expertos en el campo de DRL. Una vez entrenados los 7 modelos, se escogió al mejor agente en función a la recompensa obtenida al final del entrenamiento y al tiempo que se demoró el agente para obtener esa recompensa. Seleccionado al mejor modelo, se procedió a utilizar ese modelo entrenado dentro del agente y finalmente fue arrastrado al videojuego de tal modo que pueda perseguir al personaje en primera persona controlado por el estudiante. Este videojuego desarrollado contiene 12 niveles con diferentes terrenos por los cuales el estudiante deberá pasar y destruir a los 25 zombies para ir al siguiente nivel. A su vez, el estudiante para que ataque a los zombies tiene la opción de escoger 10 diferentes armas con diferentes modelos, nombres y sonidos para que el estudiante pueda disparar. Finalizado el videojuego, este fue exportado y se utilizó el servicio de hosting de Firebase para que pueda jugarse en cualquier computadora a través de un link que puede ser colocado en cualquier navegador.

4.2 Desarrollo del videojuego

En esta sección del presente capítulo se presentará el desarrollo del videojuego aplicando la metodología en cascada como se había definido en el capítulo anterior. En cada sub sección se procederá a explicar el desarrollo de cada etapa de la metodología en cascada aplicado al presente proyecto de investigación.

4.2.1 Fase de comunicación

Para el desarrollo de esta primera fase de la metodología, lo que se hizo en primer lugar en el mes de noviembre del 2019 fue dialogar con la directora de la I.E.P. Colegio “Mater Christi” que a la vez es psicóloga y coach. Se explicó el proyecto que se deseaba desarrollar el cual era un videojuego para mejorar el nivel de comprensión lectora en los estudiantes de primaria de la institución. Posteriormente, se estableció una cita para poder conversar con la profesora María Pascua Chotti la cual es la profesora de comprensión lectora de la institución. En la reunión con la profesora, se conversó acerca del funcionamiento que tendría el videojuego para poder recabar los requerimientos y que estos sean claros. No solamente se escuchó todas las necesidades de la profesora, sino que también se propuso que el videojuego tenga algunas funcionalidades que probablemente no se tenía pensado por parte del docente. De igual modo, se procedió a explicar lo relacionado con respecto a las evaluaciones durante el periodo de

entrenamiento dado que el profesor mismo tenía que realizar las propias evaluaciones para los estudiantes del primer al sexto grado. A su vez, se solicitaron los textos utilizados para desarrollar las clases de comprensión lectora, de preferencia en formato pdf. Además, se explicaron algunas funcionalidades que ya se tenían pensada con respecto al videojuego que se iba a desarrollar como la sección de estrategias de lectura para los estudiantes y la aparición de textos dentro del propio videojuego dado que diversos estudios implementaron aquellos métodos y obtuvieron resultados positivos por parte de los involucrados. De igual modo, se recopilaron las secciones con las cuales se iba a trabajar, se definió la sección “B” para cada grado dado que normalmente, los estudiantes de estas secciones poseen menores notas que los estudiantes de la sección “A” para cada grado. También, se definió que el videojuego sea en 3D dado que poseen gráficos mucho más realistas que pueden ser más agradables para los estudiantes. En la Figura N°69, se presenta la entrevista que se realizó con el docente del curso de comprensión lectora para recabar los requerimientos para el desarrollo del videojuego.



Figura N°69: Recolección de requerimientos del docente.

Fuente: Elaboración propia.

Al finalizar la recopilación de los requerimientos, se procedió a clasificarlos en funcionales y no funcionales con el objetivo de tener una mayor claridad acerca de las principales funcionalidades del videojuego a desarrollar. Los requerimientos funcionales recopilados fueron los siguientes:

- El videojuego debe de poseer un menú de inicio.
- El videojuego debe de tener personajes que sean del agrado de los estudiantes.
- El videojuego debe de poseer las lecturas enseñadas en las clases.
- Las lecturas del videojuego deben mostrarse de forma nítida.
- Cuando el estudiante lea una lectura, no debe haber música de fondo o animaciones.
- Las lecturas deben de poseer preguntas de retroalimentación.

- El estudiante debe de responder tres preguntas por cada pregunta.
- El videojuego debe poseer una sección de estrategias de lectura para los estudiantes.
- La historia del videojuego debe ser simple para que los estudiantes puedan entenderla.
- El prototipo debe tener niveles de forma que el niño pueda jugar en diversos escenarios.
- El videojuego debe tener una sección para explicar las instrucciones.
- El videojuego debe presentar textos (historia y oraciones) dentro de los propios niveles.
- La música y sonidos del videojuego deben ser del agrado de los estudiantes.

Del mismo modo, los requerimientos no funcionales recopilados fueron los siguientes:

- El videojuego debe de ser entendible por los estudiantes.
- El videojuego debe tener un inicio y un final para el estudiante.
- El tiempo de respuesta del videojuego no debe ser muy largo.
- La instalación del videojuego debe de ser segura.

4.2.2 Fase de planeación

Luego de haber recopilado los requerimientos del docente, se procedió a estimar el tiempo necesario para la elaboración del videojuego, así como el aseguramiento de contar con los recursos necesarios para crear el videojuego. Se estimó que la duración del desarrollo de todo el videojuego hasta entregar el producto final al docente iba a tener una duración aproximada de 1 año. Las etapas más críticas de la metodología en cascada son el modelado y la de desarrollo por lo que se estableció una duración de aproximada de 5 meses para cada una. La última fase que corresponde al despliegue del producto en donde se realiza la entrega final del producto iba a tener una duración aproximada de una semana.

Con respecto al aseguramiento de los recursos para llevar a cabo el videojuego, se contó con una laptop con las especificaciones técnicas presentadas en la Tabla 28.

Tabla 28: Especificaciones técnicas del equipo a utilizar para el desarrollo del prototipo

Campo	Descripción
Modelo de equipo	Lenovo™ YOGA 520
Procesador	Intel® Core™ i3-7100U
CPU	2.40 GHz
RAM	4.00 GB
Tipo de Sistema	64 bits, procesador x64
Sistema Operativo	Windows 10

Fuente: Elaboración Propia.

De igual modo, se contó con una velocidad de internet de 15 Mbps. Solamente se contó con un desarrollador para construir todo el videojuego, 1 experto en desarrollo de videojuegos y con 2 docentes del colegio que revisaron el producto final. Los participantes del proyecto se presentan en la Tabla 29.

Tabla 29: Participantes del proyecto de investigación

Participante	Rol establecido
Alvaro Machuca Breña	Desarrollador del videojuego.
Dr. Deyby Huamanchahua	Experto en desarrollo de videojuegos y consultor del proyecto de investigación.
Prof. María Pascua Chotti	Profesora del curso de comprensión lectora para el 1er, 2do y 3er grado de educación primaria y a la vez cliente principal que solicitó el videojuego.
Prof. Camila Cavero Limachi	Profesora del curso de comprensión lectora para el 4to, 5to y 6to grado de educación primaria.

Fuente: Elaboración Propia

Como última actividad de esta segunda fase de la metodología en cascada, se desarrolló el cronograma de trabajo para poder desarrollar el videojuego en donde se detallan las actividades a realizar por cada etapa. El cronograma desarrollado se puede visualizar consultando el siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

4.2.3 Fase de modelado

En esta tercera fase de la metodología en cascada, se procedió a modelar el comportamiento del sistema, así como la creación de los bosquejos y arquitectura del videojuego. A continuación, se presentan las actividades desarrolladas.

4.2.3.1 Creación y especificación de casos de uso

En la Figura N°70, se presenta el diagrama de casos de uso creado para el desarrollo del videojuego.

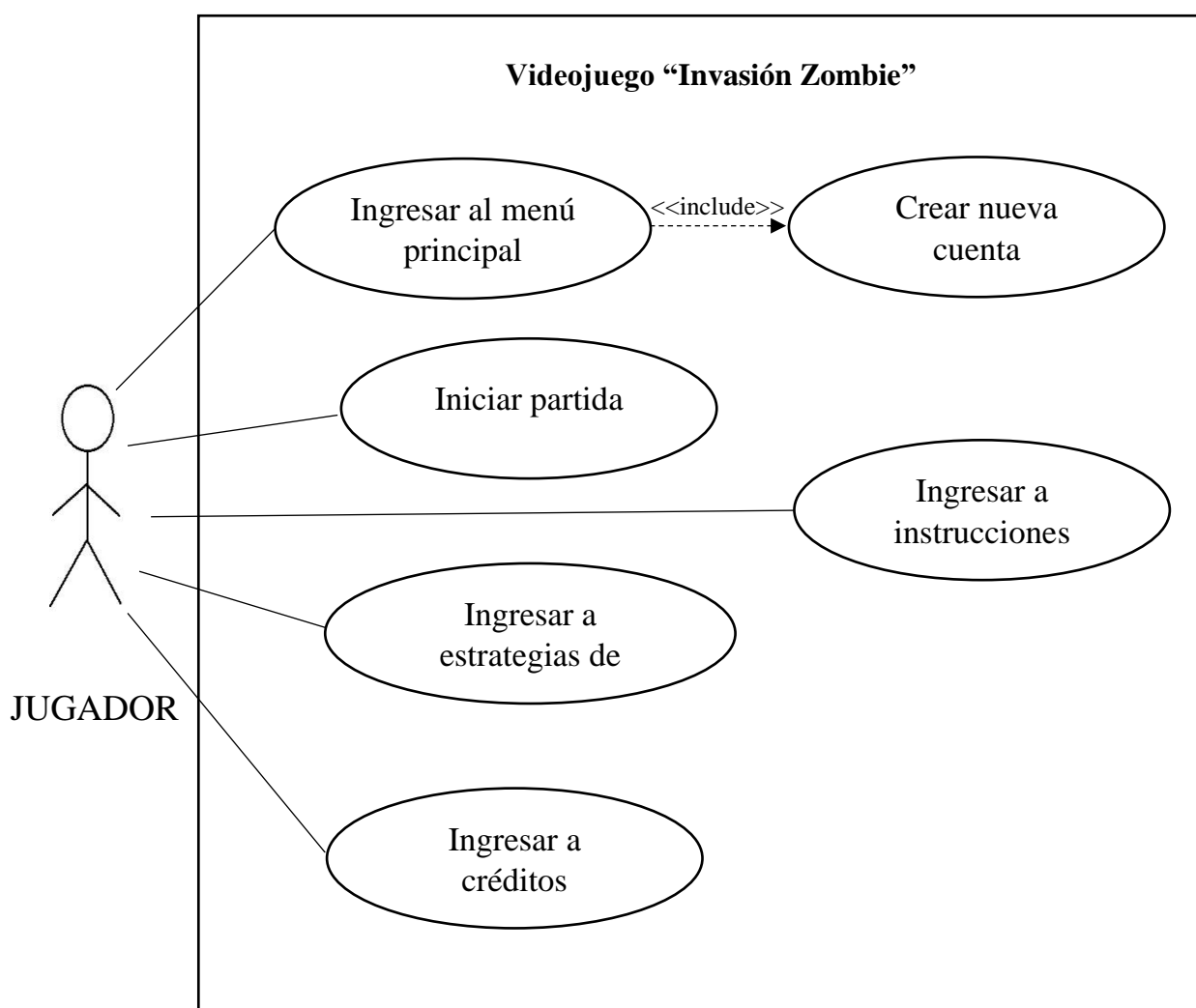


Figura N°70: Diagrama de casos de uso del videojuego.

Fuente: Elaboración propia.

Como se puede observar en la Figura N°70, se desarrollaron 6 diferentes casos de uso para modelar el comportamiento general del videojuego en función a las especificaciones brindadas por el docente del curso de lectura.

En la Tabla 30, se presenta una descripción general de cada caso de uso presentado en el diagrama desarrollado en el punto anterior.

Tabla 30: Descripción general de los casos de uso del videojuego

Código	Caso de Uso	Descripción general
CUN01	Ingresar al menú principal	El estudiante ingresa al videojuego luego de loguearse y visualiza el menú general correspondiente a su grado.
CUN02	Crear nueva cuenta	El estudiante se registra dentro del videojuego ingresando su nombre, apellido y grado correspondiente.
CUN03	Iniciar partida	El estudiante inicia la partida a través del botón “Iniciar Partida” dentro del menú y el videojuego redirecciona al alumno al primer nivel del videojuego.
CUN04	Ingresar a instrucciones	El estudiante ingresa a las instrucciones del videojuego a través del botón “Instrucciones” ubicado dentro del menú.
CUN05	Ingresar a estrategias de lectura	El estudiante visualiza las estrategias de lectura del videojuego haciendo click en el botón de “Estrategias” ubicado dentro del menú.
CUN06	Ingresar a créditos	El estudiante es redirigido a los créditos del videojuego haciendo click en el botón de “Créditos” del menú principal.

Fuente: Elaboración Propia.

Además, en la Figura N°71, Figura N°72, Figura N°73, Figura N°74, Figura N°75 y Figura N°76 se presenta la especificación de cada caso de uso del diagrama de caso de uso presentado.

Código: CUN01	Nombre: Ingresar al menú principal	
Descripción: En este caso de uso se describe el comportamiento del sistema cuando el estudiante ingresa al videojuego.		
Actor: Jugador		
Pre-condición: Haberse creado una cuenta para el videojuego.		
Flujo básico	Flujo Alternativo	
<ol style="list-style-type: none"> 1. El jugador ingresa sus nombres y apellidos en la escena de Login. 2. El jugador da click en el botón de “Ingresar”. 3. El sistema redirecciona al jugador al menú principal del videojuego. 4. El sistema muestra los 4 botones de “Iniciar partida”, “Instrucciones”, “Estrategias” y “Créditos”. 	<ol style="list-style-type: none"> 2.1 El sistema muestra un mensaje que dice “El usuario no existe o los datos son incorrectos” en el caso que el estudiante haya ingresado incorrectamente sus datos o no se haya registrado dentro del videojuego. 	
A. Post-condición: El niño se encuentra dentro del menú principal.		

Figura N°71: Especificación de caso de uso CUN01.

Fuente: Elaboración propia.

Código: CUN02	Nombre: Crear nueva cuenta	
Descripción: En este caso de uso se describe el comportamiento del sistema cuando el estudiante se encuentra en la escena de Registro del videojuego.		
Actor: Jugador		
Pre-condición: El alumno se encuentra en la escena del Login del videojuego.		
Flujo básico	Flujo Alternativo	
<ol style="list-style-type: none"> 1. El jugador hace click en el botón de “Registrarse” ubicado en la parte inferior de la escena del Login. 2. El sistema redirecciona al estudiante a la escena del Registro del videojuego. 3. El estudiante registra sus nombres, apellidos y grados al cual pertenece. 4. El estudiante hace click en el botón “Guardar”. 5. El sistema muestra el siguiente mensaje: “¡¡Felicidades! ¡¡Ya estás registrado!!”. 6. El estudiante hace click en el botón de “Ir al Login”. 	<ol style="list-style-type: none"> 4.1 El sistema muestra un mensaje que dice “¡Datos incompletos!” en la parte superior de la escena del registro en el caso que el estudiante haya omitido escribir ya sea su nombre o apellido. 	
A. Post-condición: El estudiante se encuentra nuevamente en el Login.		

Figura N°72: Especificación de caso de uso CUN02.

Fuente: Elaboración propia.

Código: CUN03	Nombre: Iniciar Partida
Descripción: En este caso de uso se describe el comportamiento del sistema cuando el estudiante hace click en el botón de “Iniciar Partida”.	
Actor: Jugador	
Pre-condición: El estudiante se encuentra dentro del menú principal.	
Flujo básico 1. El jugador hace click en el botón de “Iniciar Partida” del menú principal. 2. El sistema muestra la escena de narración de la historia principal del videojuego. 3. Al final la escena de narración de la historia, el videojuego envía al estudiante al primer nivel del videojuego.	Flujo Alternativo —
A. Post-condición: El estudiante se encuentra en el nivel 1.	

Figura N°73: Especificación de caso de uso CUN03.

Fuente: Elaboración propia.

Código: CUN04	Nombre: Ingresar a instrucciones	
Descripción: En este caso de uso se describe el comportamiento del sistema cuando el estudiante hace click en el botón de “Instrucciones”.		
Actor: Jugador		
Pre-condición: El estudiante se encuentra dentro del menú principal.		
Flujo básico	Flujo Alternativo	
<ol style="list-style-type: none"> 1. El jugador hace click en el botón de “Instrucciones” del menú principal. 2. El sistema muestra una escena con una imagen del botón que se debe de apretar dentro del videojuego y la acción que se ejecutará al presionar ese botón. Esta misma escena muestra el botón de “Ir a la siguiente instrucción” y “Regresar al menú”. 3. El estudiante hace click en el botón de “Ir a la siguiente instrucción”. 4. El sistema muestra la instrucción N°02 para jugar el videojuego. 	<ol style="list-style-type: none"> 3.1 El estudiante hace click en el botón de “Regresar al menú”. 3.2 El sistema redirecciona al estudiante al menú principal del videojuego. 	
A. Post-condición: El estudiante se encuentra en la sección de instrucciones.		

Figura N°74: Especificación de caso de uso CUN04.

Fuente: Elaboración propia.

Código: CUN05	Nombre: Ingresar a estrategias de lectura	
Descripción: En este caso de uso se describe el comportamiento del sistema cuando el estudiante hace click en el botón de “Estrategias”.		
Actor: Jugador		
Pre-condición: El estudiante se encuentra dentro del menú principal.		
Flujo básico	Flujo Alternativo	
<ol style="list-style-type: none"> 1. El jugador hace click en el botón de “Estrategias” del menú principal. 2. El sistema muestra una escena con la primera estrategia de lectura en la parte central y el botones de “Ir a la siguiente estrategia” y “Regresar al menú”. 3. El estudiante hace click en el botón de “Ir a la siguiente estrategia”. 4. El sistema muestra la estrategia N°02 para mejorar la habilidad de comprensión lectora. 	<ol style="list-style-type: none"> 3.1 El estudiante hace click en el botón de “Regresar al menú”. 3.2 El sistema redirecciona al estudiante al menú principal del videojuego. 	
A. Post-condición: El estudiante se encuentra en la sección de estrategias de lectura.		

Figura N°75: Especificación de caso de uso CUN05.

Fuente: Elaboración propia.

Código: CUN06	Nombre: Ingresar a créditos	
Descripción: En este caso de uso se describe el comportamiento del sistema cuando el estudiante hace click en el botón de “Créditos”.		
Actor: Jugador		
Pre-condición: El estudiante se encuentra dentro del menú principal.		
Flujo básico	Flujo Alternativo	
<ol style="list-style-type: none"> 1. El jugador hace click en el botón de “Créditos” del menú principal. 2. El sistema muestra al alumno los créditos finales del videojuego mostrando los involucrados en el proyecto. 3. Finalizando los créditos, el sistema muestra una foto del trabajo realizado con un mensaje en la parte superior que dice: “Presione Enter”. 4. El estudiante da click en enter. 5. El sistema muestra la siguiente foto del trabajo realizado. 6. Finalizando las fotos del proyecto desarrollado, el sistema muestra imágenes con el objetivo de prevenir el Covid-19 en niños mostrando en la última imagen el botón de “Regresar al menú”. 	—	
A. Post-condición: El estudiante se encuentra en la sección de créditos.		

Figura N°76: Especificación de caso de uso CUN06.
Fuente: Elaboración propia.

4.2.3.2 Creación de bosquejos y arquitectura de las escenas

Se realizaron diferentes bosquejos iniciales de las principales escenas que el videojuego iba a contener con el objetivo de establecer la apariencia de cada escena. En la Figura N°77, Figura N°78, Figura N°79 y Figura N°80 un ejemplo de los bosquejos desarrollados.

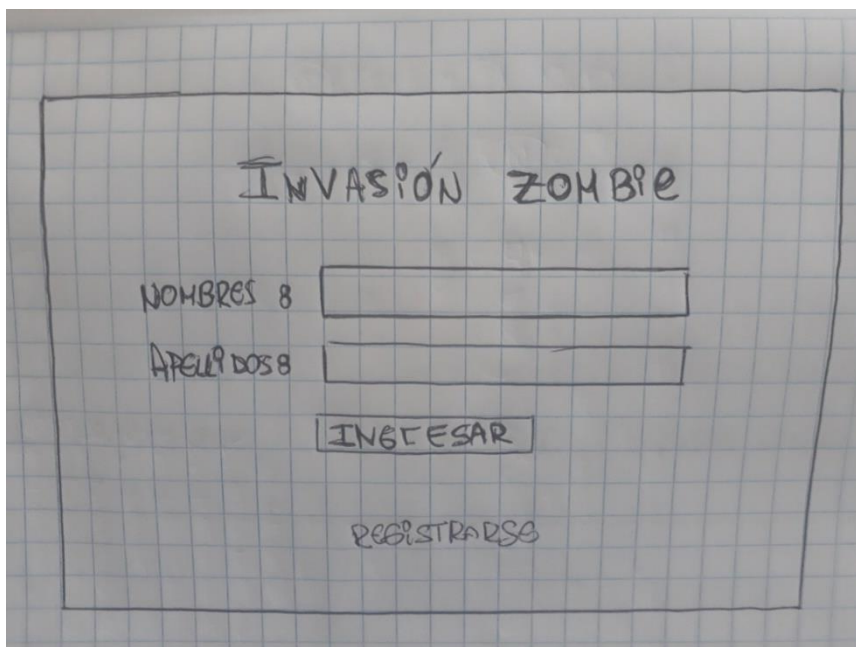


Figura N°77: Bosquejo inicial del Login del videojuego.
Fuente: Elaboración propia.

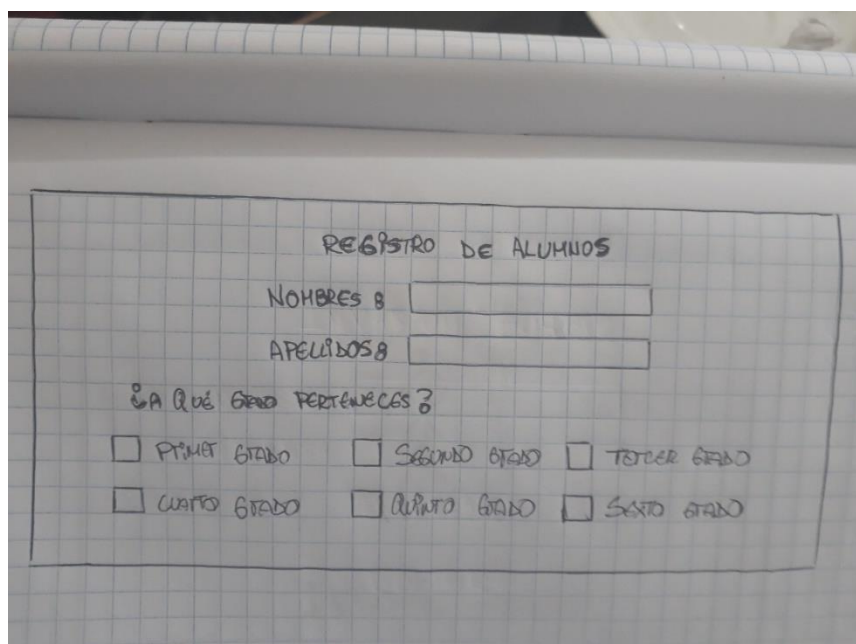


Figura N°78: Bosquejo inicial del registro de los alumnos.
Fuente: Elaboración propia.

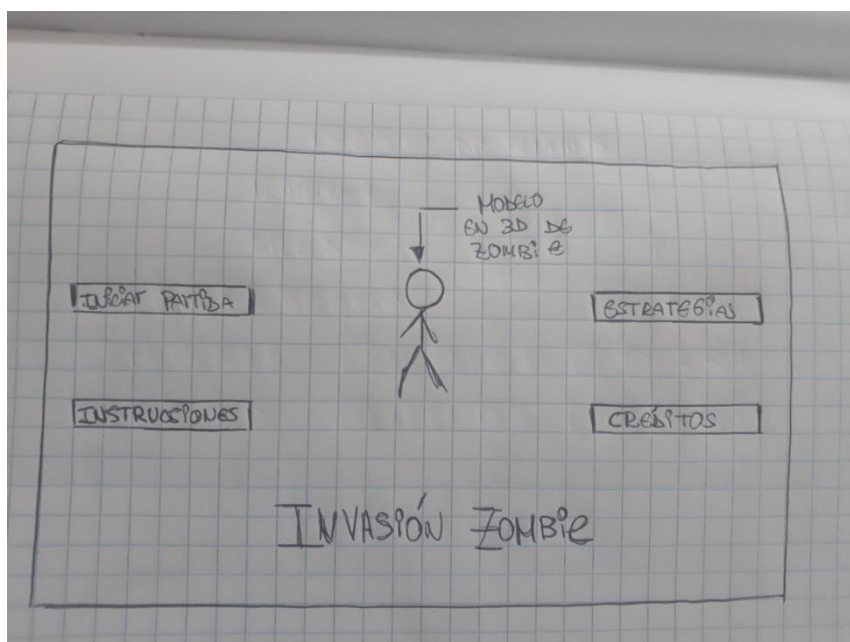


Figura N°79: Bosquejo inicial del menú principal del videojuego.

Fuente: Elaboración propia.

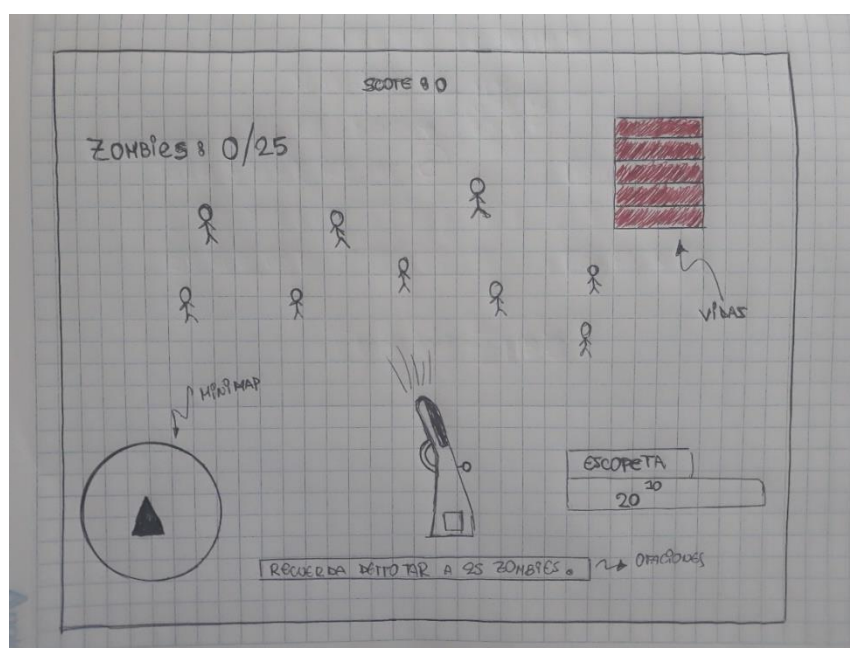


Figura N°80: Bosquejo inicial de los niveles del videojuego.

Fuente: Elaboración propia.

Definidos los bosquejos del videojuego se desarrolló la arquitectura de las escenas con el objetivo de definir los lugares de cada componente que iban a estar dentro de cada una de aquellas.

Con respecto al Login del videojuego, este posee dos InputField para que el estudiante ingrese sus nombres y apellidos, así como tres objetos tipo Texto y dos botones para que el estudiante haga click e ingrese al videojuego o para que se redirija a la escena de registro de los usuarios. La arquitectura del Login del videojuego, se presenta en la Figura N°81.

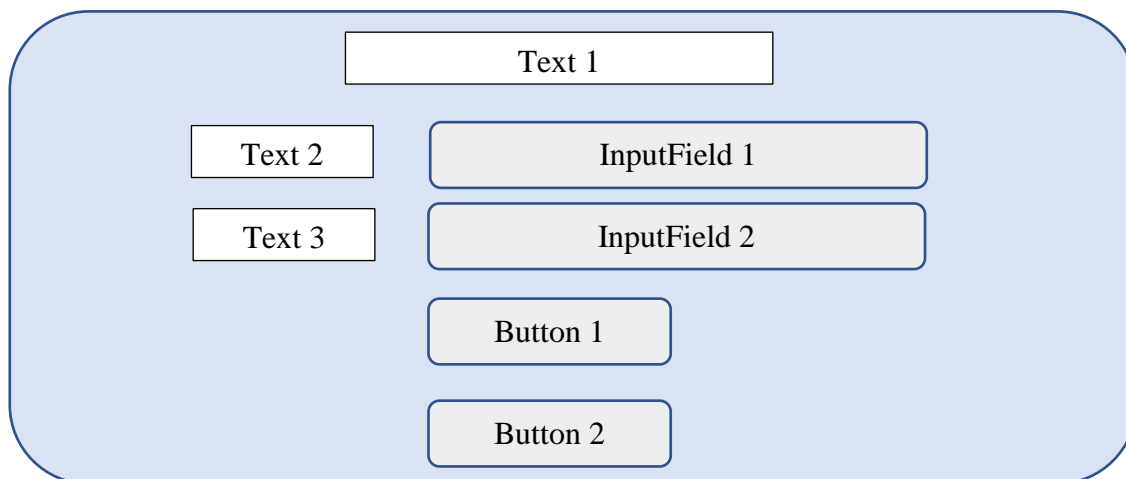


Figura N°81: Arquitectura del Login del videojuego.
Fuente: Elaboración propia.

Con respecto a la arquitectura de la escena del registro, este posee 2 InputField en la parte central para que el estudiante escriba sus nombre y apellidos. De igual modo, posee un ToggleGroup con 6 Toggles con el objetivo de que el estudiante seleccione el grado al cual pertenece. Finalmente, posee un botón de guardar para enviar la información a la base de datos. En la Figura N°82, se puede observar la arquitectura de esta escena.

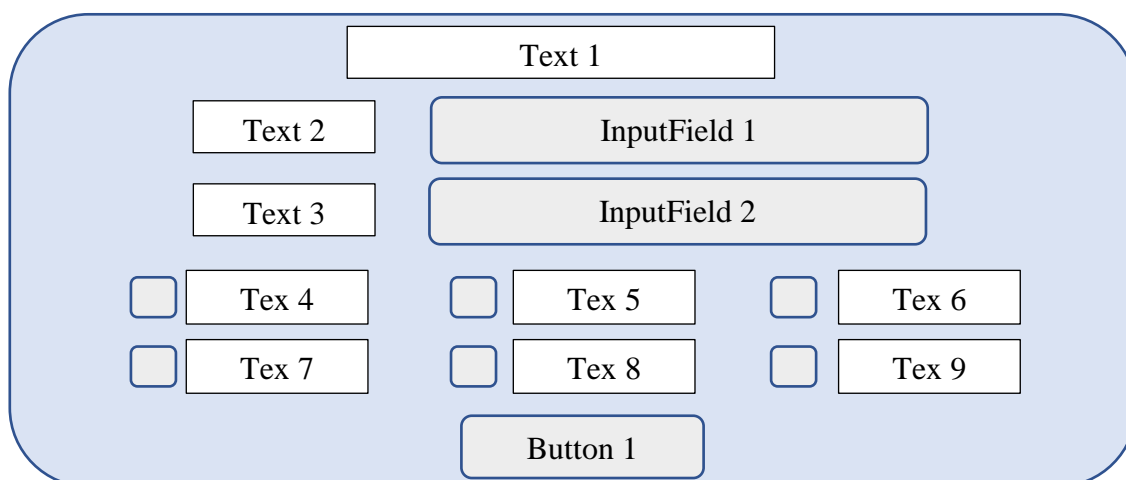


Figura N°82: Arquitectura de escena de Registro de alumnos.
Fuente: Elaboración propia.

La escena del menú principal posee 4 botones, dos ubicados en la parte izquierda y los otros 2 restantes a la derecha. Al centro de cada menú aparecerá un modelo en 3D de un zombie diferente para cada grado con el objetivo de diferenciar al menú en función al grado en el cual se encuentre el estudiante y en la parte inferior de la escena aparece el nombre del videojuego. En la Figura N°83, se presenta la arquitectura del menú del videojuego.

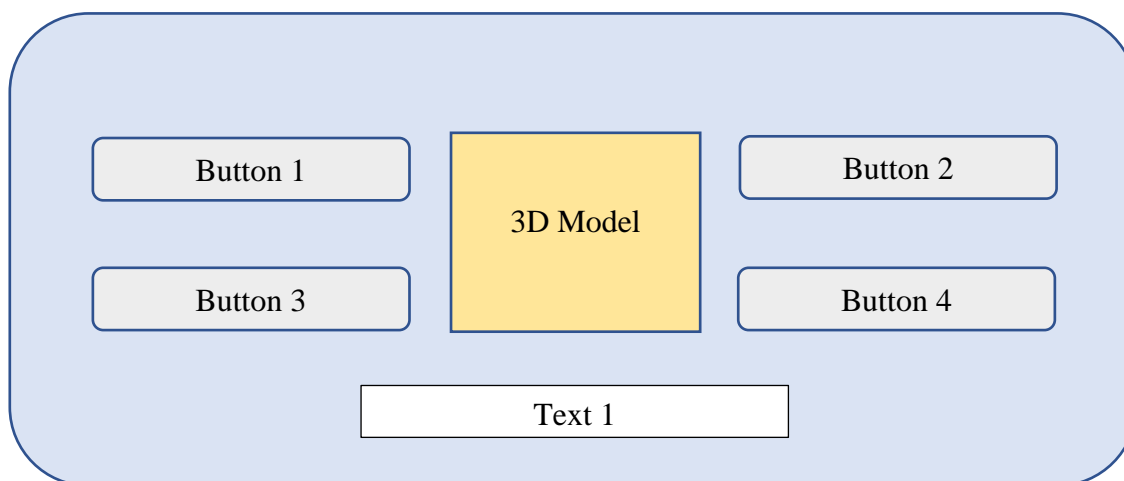


Figura N°83: Arquitectura del menú general del videojuego
Fuente: Elaboración propia.

Las 12 escenas de los niveles conservan el mismo formato. En la parte inferior izquierda se encuentra el Minimap y en la parte superior del mismo lado, se presenta el score del estudiante con respecto a la cantidad de zombies que ha destruido. De igual modo, en la parte central del videojuego se visualizarán los zombies enemigos que persiguen al personaje en primera persona controlado por el estudiante. En la parte inferior central de la escena, se visualizará el arma del estudiante para que dispare a los modelos en 3D de los zombies y al lado derecho de aquel se observará el nombre del arma y debajo la cantidad de municiones que posee. Finalmente, en la parte superior derecha el estudiante puede visualizar las vidas que posee en forma de 5 barras de color roja, cada que el estudiante pierda una vida, cada barra se borrará hasta que llegue el estudiante a perder sus 5 vidas. En la Figura N°84, se presenta la arquitectura general de un nivel del videojuego.

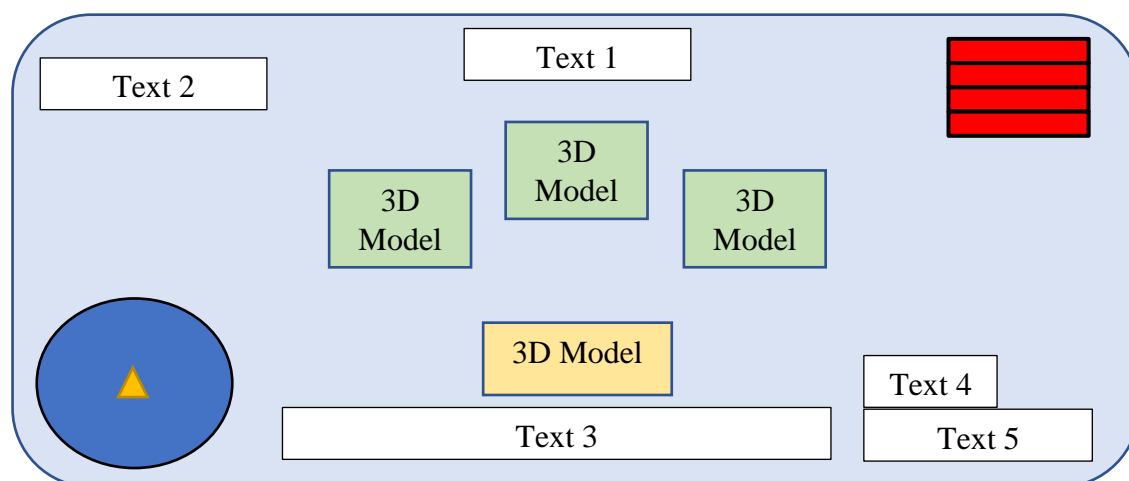


Figura N°84: Arquitectura de un nivel del videojuego.

Fuente: Elaboración propia.

4.2.3.3 Secuencia de Gameplay del videojuego

La secuencia de Gameplay del videojuego desarrollada se presenta en la Figura N°85. Como se puede observar en la Figura N°85, el videojuego inicia cuando el estudiante dentro del menú da click en el botón de “Iniciar partida”. El videojuego mostrará la escena donde se narre la historia general del videojuego con el objetivo de que el estudiante entre en el contexto del videojuego. Luego de narrar la historia, se presenta al estudiante el nivel del videojuego en el cual debe seleccionar un arma y destruir a los zombies.

Dentro de cada nivel, el estudiante tiene 5 vidas, cuando el zombie lo ataque, sus vidas disminuyen en una unidad, si sus vidas llegan a cero el estudiante debe leer una lectura y responder correctamente las 3 preguntas que lleva cada pregunta para poder regresar al nivel donde se encontró. Cada que el estudiante acierte en cada pregunta, el contador se incrementa en 1 hasta llegar a 3. En el caso de que el contador sea menor a tres en la última pregunta, el estudiante habrá perdido y es redirigido al menú principal empezando el videojuego desde el inicio.

Cada que el estudiante dispare y destruya a un zombie, el contador se incrementa en una unidad. Cuando este contador llegue a 25, el estudiante pasa al siguiente nivel hasta llegar al último nivel que es el doce. Si es estudiante pasa este último nivel, habrá terminado el videojuego por lo que aparecerá una escena final del videojuego y esa última escena redirecciona al estudiante a los créditos finales del videojuego.

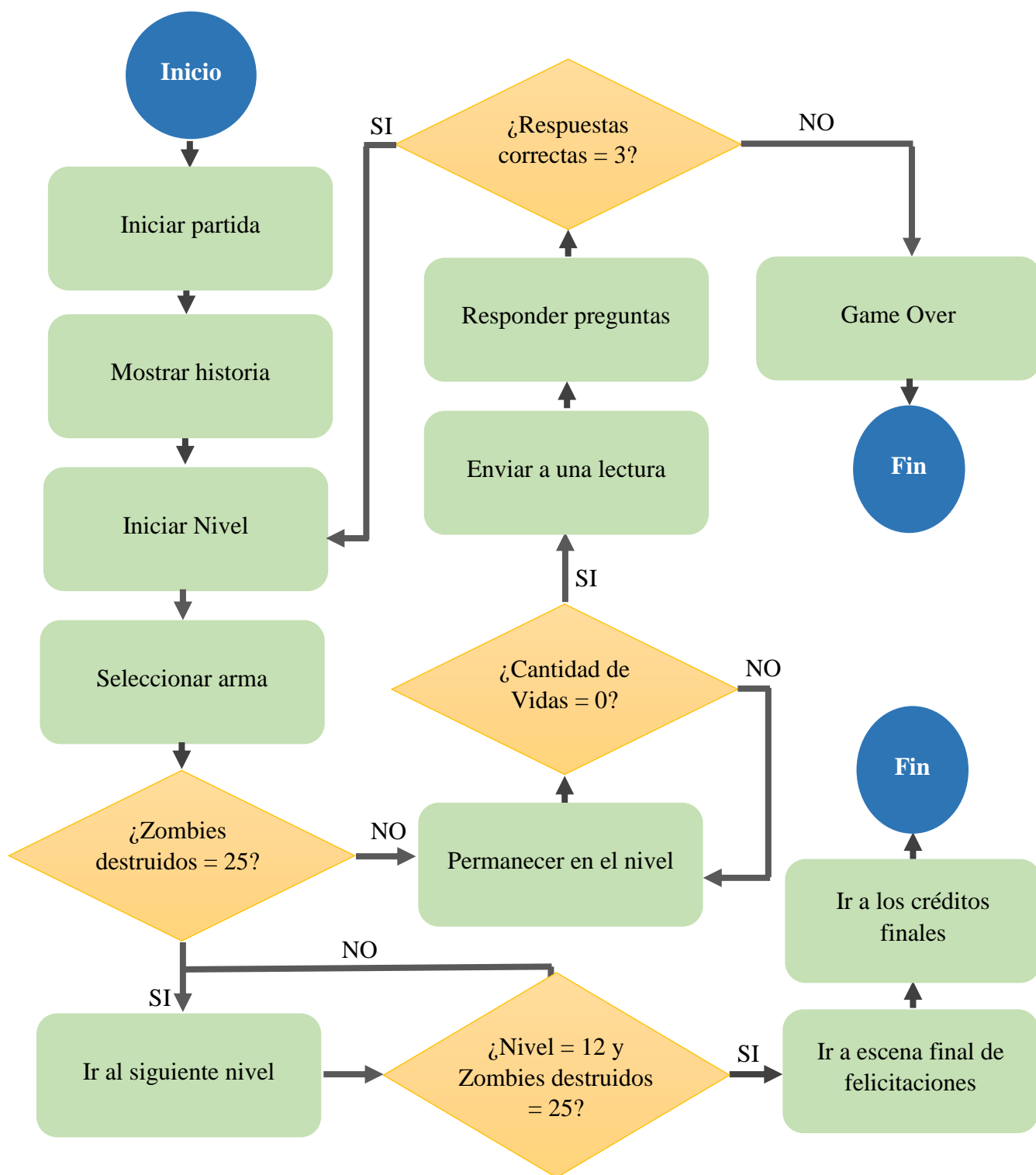


Figura N°85: Secuencia de Gameplay desarrollada.

Fuente: Elaboración propia.

4.2.3.4 Modelos en 3D descargados

Con respecto a los zombies, se descargaron 18 modelos diferentes de zombies con sus respectivas animaciones para que estas puedan ser reproducidas dentro de cada nivel. Además, se utilizaron 6 diferentes modelos de zombies para que puedan estar presentes en los 6 menús que le corresponden a cada grado de tal modo de poder diferenciar un menú de otro. Un ejemplo de los personajes descargados y que fueron utilizados dentro del videojuego se presentan en la Figura N°86.



Figura N°86: Personajes enemigos del videojuego.

Fuente: Elaboración propia.

De igual modo, con respecto a los terrenos se descargaron 12 diferentes modelos de terrenos en 3D que fueron colocados en una carpeta propia y colocados dentro del proyecto de Unity. A cada terreno se le dio solidez para que el personaje en primera persona controlado por el estudiante pueda caminar encima. Cada terreno posee diseños diferentes como árboles, arbustos, casas abandonadas, atardeceres diferentes, entre otros aspectos que hacen de cada nivel más llamativo y realista. Un ejemplo de los modelos en 3D de los terrenos utilizados dentro del videojuego se presenta en la Figura N°87.

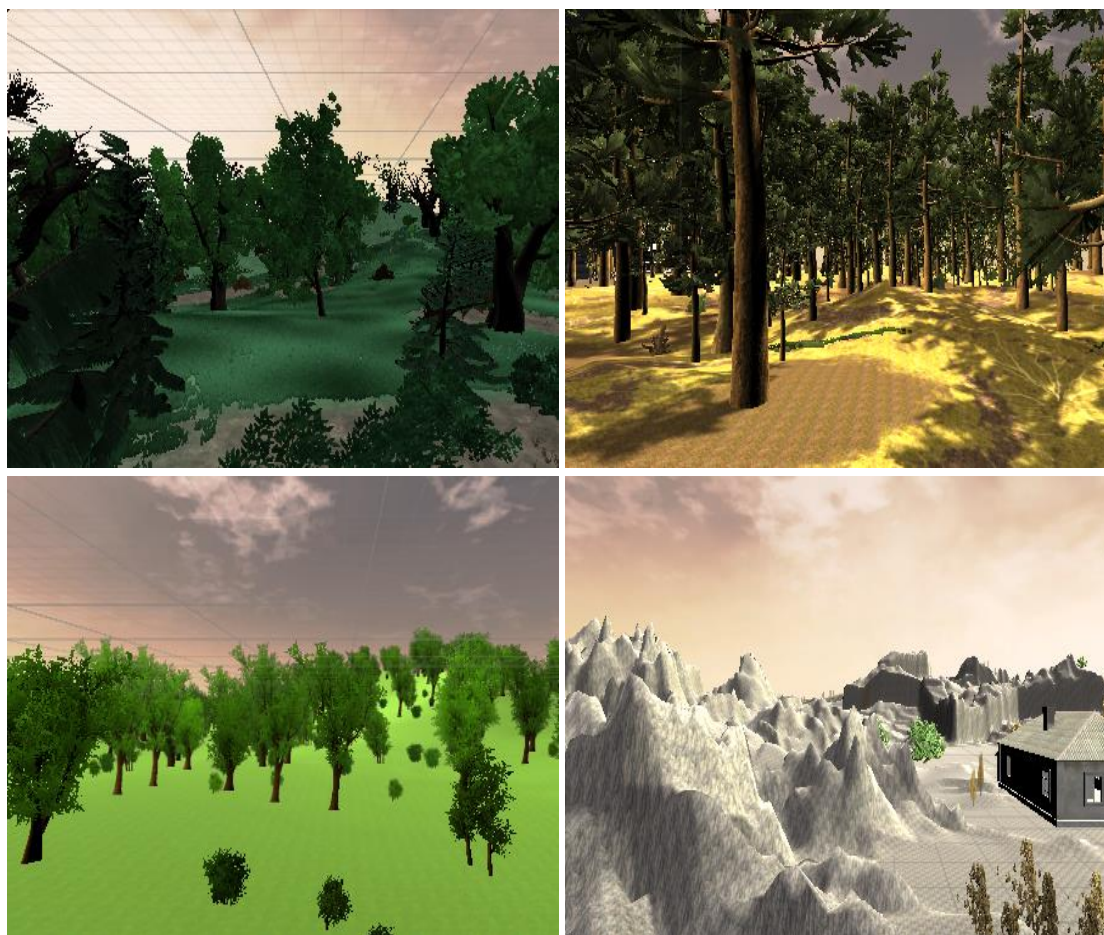


Figura N°87: Terrenos en 3D del videojuego.
Fuente: Elaboración propia.

Finalmente, se descargaron 10 diferentes modelos en 3D de armas con el objetivo de que el estudiante pueda seleccionar con los numerales el modelo que le resulte más atractivo para poder jugar. Se creó además una animación de tal modo que cuando es estudiante dispare, se eleve la parte delantera del arma y regrese a su misma posición de tal modo de simular un disparo real. Sumado a esto, cada disparo por cada arma tiene un sonido diferente y en la parte inferior derecha de la escena de cada nivel se visualiza el nombre del arma escogida de tal modo de poder diferenciar unas de otras. Un ejemplo de los modelos en 3D de armas utilizados, se presenta en la Figura N°88.



Figura N°88: Modelo de armas en 3D usados en el videojuego.

Fuente: Elaboración propia.

4.2.4 Fase de desarrollo

Habiendo desarrollado el comportamiento general del videojuego, los bosquejos iniciales y la arquitectura de las escenas, se procedió a construir las escenas del videojuego.

4.2.4.1 Desarrollo de las escenas

Con respecto al desarrollo del menú, se agregaron dos objetos tipo texto en la parte izquierda para colocar los nombres y apellidos de los estudiantes. De igual modo, se agregaron dos InputField para que el estudiante pueda ingresar sus datos y dos botones que corresponden al ingreso del estudiante al videojuego y otro botón para que el estudiante se registre. La escena del Login del videojuego, se presenta en la Figura N°89.



Figura N°89: Escena construida para el Login de videojuego.

Fuente: Elaboración propia.

Con respecto al registro de los alumnos, se creó una escena nueva para que los estudiantes puedan ingresar tanto sus nombres, apellidos y el grado al cual pertenecen. Se agregó 1 ToggleGroup con 6 Toggles de tal forma que los estudiantes puedan marcar la casilla correspondiente con respecto al grado que pertenecen. La escena construida del registro de los estudiantes se presenta en la Figura N°90.



Figura N°90: Escena construida para el registro de los estudiantes.
Fuente: Elaboración propia.

De igual modo, se construyeron 6 diferentes escenas de menú correspondientes a cada grado. El menú posee 4 diferentes botones que son para iniciar partida, para visualizar las instrucciones del videojuego, para ver las diferentes estrategias de aprendizaje para la mejora de la comprensión lectora y los créditos finales del videojuego. Un ejemplo de la escena del menú principal desarrollada se presenta en la Figura N°91.



Figura N°91: Escena construida para el menú principal del videojuego.

Fuente: Elaboración propia.

Al hacer click en el botón de instrucciones del menú principal, este redirecciona al estudiante a la escena donde se pueden visualizar a través de una imagen del teclado, los botones que el alumno debe de presionar para que pueda jugar el videojuego y pueda pasar los niveles de la mejor manera posible. La escena de las instrucciones se presenta en la Figura N°92.



Figura N°92: Escena construida para visualizar las instrucciones del videojuego.

Fuente: Elaboración propia.

El botón para iniciar partida del menú principal redirecciona al estudiante al primer nivel del videojuego. Los modelos en 3D de los terrenos fueron arrastrados a la escena del nivel y al personaje en primera persona del estudiante fue colocado encima del terreno. De igual manera, se agregaron la barra inicial de vidas en la parte superior derecha y un score en la parte superior. Para que este score se incremente, se construyó insertó un objeto en 3D para que el estudiante puede practicar su puntería dentro del nivel. Además, se agregaron 3 objetivos para que el estudiante cumpla en los primeros niveles del videojuego. La escena del primer nivel del videojuego en donde se visualiza principalmente un ejemplo del terreno utilizado, el score y barra de vidas se presenta en la Figura N°93.



Figura N°93: Nivel del videojuego y visualización del score y barra de vidas.

Fuente: Elaboración propia.

Por otro lado, se introdujo un objeto Canvas con 4 botones para la pausa del videojuego, este Canvas se activa con el botón de escape del teclado (Esc) y los 4 botones que aparecen son para regresar a la escena, para reiniciar el nivel donde se encuentra el estudiante, para ir al menú del videojuego y para regresar al Login. La visualización de la pausa del videojuego se presenta en la Figura N°94.



Figura N°94: Pausa desarrollada para el videojuego.

Fuente: Elaboración Propia.

Al superar todos los niveles y terminar el videojuego, el sistema envía al estudiante a los créditos finales del videojuego mostrando a los desarrolladores del proyecto. Luego de los créditos, aparecen 8 diferentes recomendaciones en forma de imágenes para que los estudiantes puedan cuidarse desde sus casas para prevenir el Covid-19. Estas últimas escenas se presentan en la Figura N°95.

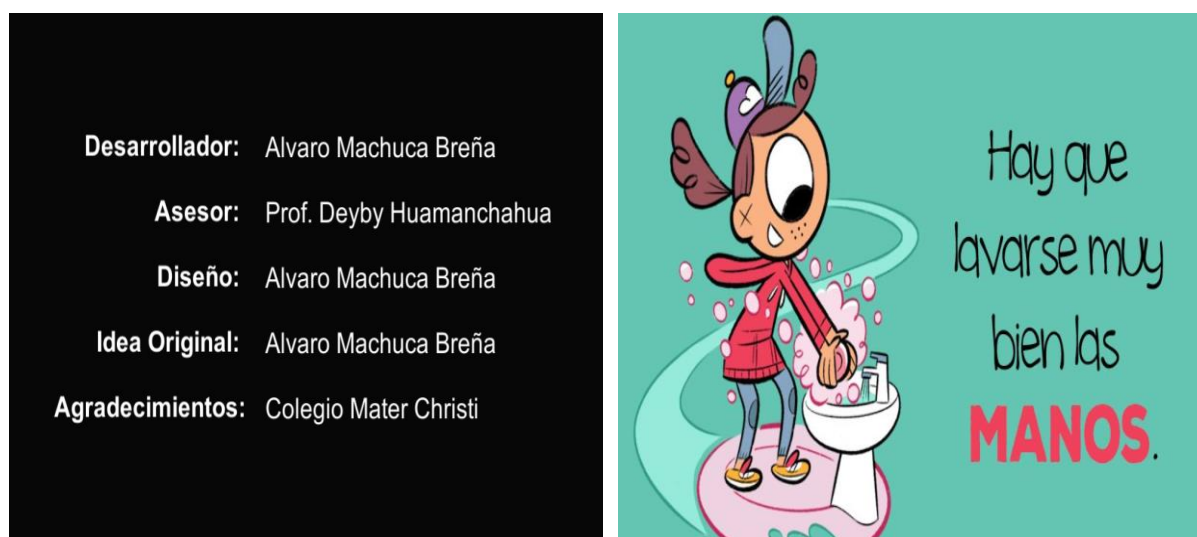


Figura N°95: Créditos finales (Izq.), Recomendaciones prevenir el Covid-19 (Der.).

Fuente: Elaboración Propia.

4.2.4.2 Implementación de realidad virtual no inmersiva

Para implementar la realidad virtual no inmersiva dentro del videojuego, se seleccionó la opción de WebGL de Unity para que el videojuego pueda ser reproducido en la pantalla de una computadora. Para realizar esto último, se cambió la configuración del proyecto seleccionando la opción de WebGL. De igual modo, para implementar y controlar a un personaje en primera persona a través del teclado, se importó la librería de “*Estandard Assets*” desde Unity como se muestra en la Figura N°96.

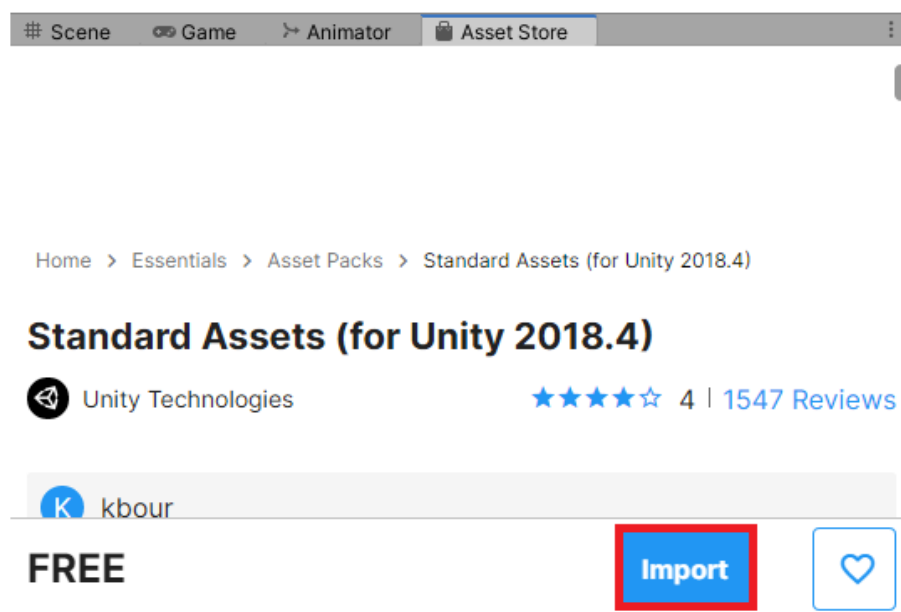


Figura N°96: Importación de Standard Assets para implementar un personaje en primera persona.

Fuente: Elaboración Propia.

Al importar la librería, aparece una carpeta nueva dentro del proyecto con el nombre de aquella librería instalada. El componente para introducir un personaje en primera persona en el prototipo desarrollado se llama “FPSController”. Este componente fue arrastrado a la escena de cada nivel del videojuego de tal modo que el estudiante pueda controlar a un personaje en primera persona tal cual como se muestra en la Figura N°97.

Este personaje importado ya viene pre configurado para el movimiento del personaje con las teclas A, D y las teclas de flechas de izquierda y derecha para el desplazamiento en el eje x. Para el desplazamiento en el eje z, está configurado para el uso de las teclas W, S y las flechas de arriba y abajo. Además, para ejecutar un salto se utiliza la tecla de “Space” y si se desea que el personaje corra se utiliza la tecla de Shift + W. Estas configuraciones se mantuvieron dentro del videojuego y forman parte de las instrucciones del videojuego.

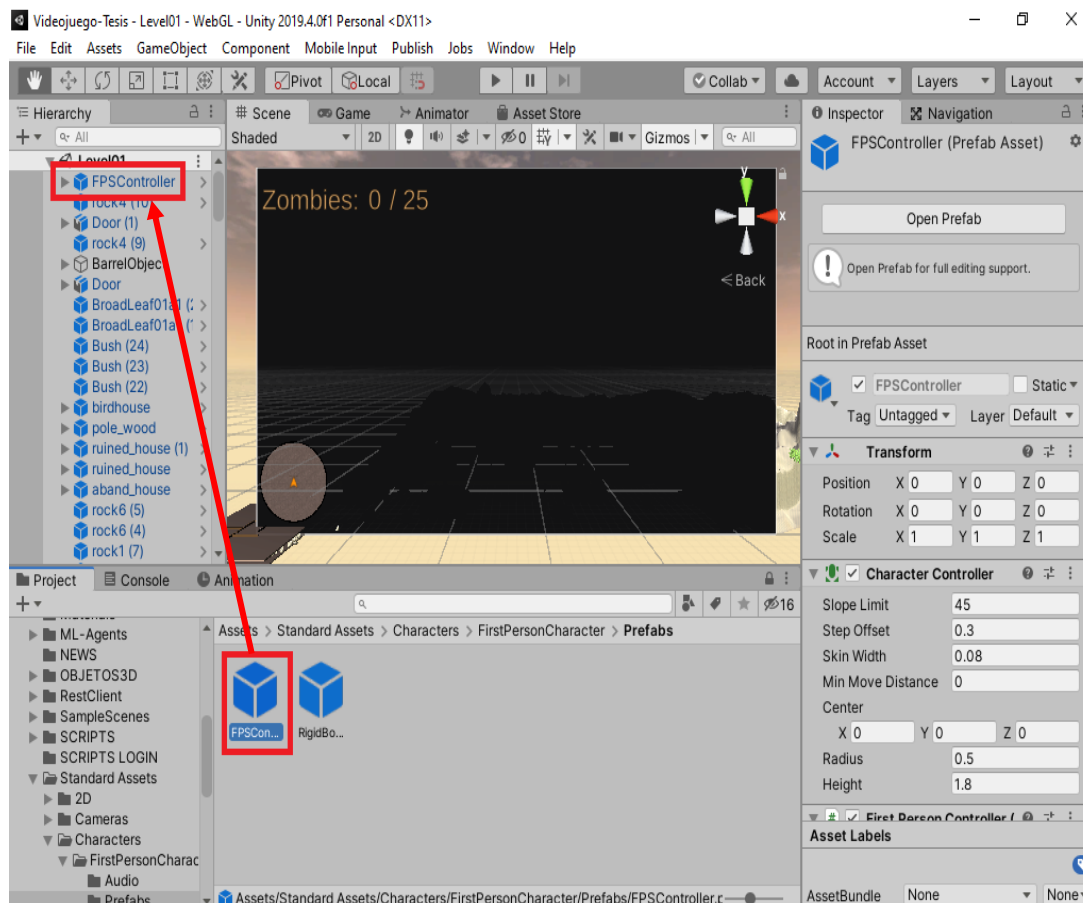


Figura N°97: Implementación de un personaje en primera persona en el videojuego.
Fuente: Elaboración Propia.

4.2.4.3 Implementación de la base de datos de Firebase

Para implementar la base de datos de Firebase, se creó un nuevo proyecto. Al acceder al nuevo proyecto, se seleccionó la opción de “Realtime Database” y se creó una nueva base de datos. En la sección de “Reglas” se cambiaron los valores de “Read” y “Write” a True para poder insertar datos desde Unity. Al crear esta base de datos, se proporciona un link tal cual como se muestra en la Figura N°98 el cual sirve para realizar la conexión con Unity.

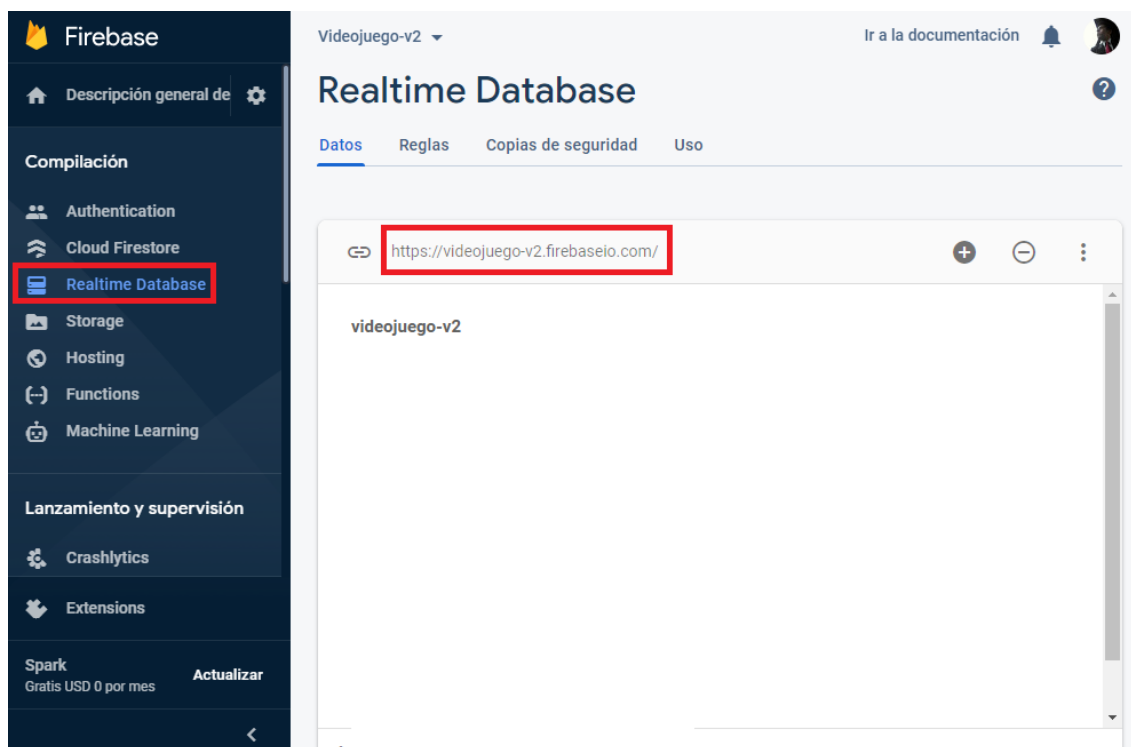


Figura N°98: Creación de una nueva base de datos.

Fuente: Elaboración propia.

Por el lado de Unity, se descargó la librería de “Rest Client for Unity”. Una vez instalada dicha librería, se procedió a crear la clase Usuario con 3 variables públicas: 2 de tipo String y 1 de tipo Integer. Las variables de tipo String corresponden al almacenamiento del nombre y apellido del estudiante. Por otro lado, la variable de tipo Integer corresponde al grado que pertenece el estudiante. Además, se creó un constructor para la creación de un objeto tipo Usuario que tenga los atributos del nombre, apellido y grado del alumno. La clase de Usuario construida se puede apreciar en la Figura N°99.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;

[Serializable]
public class Usuario
{
    public string NombreUsuario;
    public string ApellidoUsuario;
    public int GradoUsuario;

    public Usuario ()
    {
        NombreUsuario = DataBase.nombreUser;
        ApellidoUsuario = DataBase.nombreApellido;
        GradoUsuario = DataBase.GradoUser;
    }
}
```

Figura N°99: Creación de la clase Usuario.

Fuente: Elaboración propia.

Luego de la creación de la clase Usuario, se creó un script llamado “DataBase” que realiza la conexión entre Firebase y Unity. Este script completo se puede encontrar en el siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>. Las partes más resaltantes del script “DataBase” son la creación de las funciones de “EnviarPorPost” y “RecibirDeBD” descritas en el capítulo anterior. Estas funciones implementadas se presentan en la Figura N°100.

Como se puede visualizar en la Figura N°100, en ambas funciones se introducen el link que proporciona la base de datos de Firebase al momento de que fue creada. En el caso de la función de EnviarPorPost(), cuando el estudiante se encuentra en la escena del registro de usuarios, se crea un nuevo objeto de tipo Usuario y se utiliza la función RestClient.Put() para enviar la información a la base de datos en un formato tipo json.

```
public void EnviarPorPost()
{
    Usuario usuario = new Usuario();
    RestClient.Put("https://videojuego-v2.firebaseio.com/" + nombreUser + " "
+ nombreApellido + ".json", usuario);
}

public void RecibirDeBD()
{
    RestClient.Get<Usuario>("https://videojuego-v2.firebaseio.com/" +
LoginName.text + " " + LoginApellido.text + ".json").Then(onResolved: response =>
    {
        user = response;
    });
}
}
```

Figura N°100: Funciones de EnviarPorPost() y RecibirDeBD().

Fuente: Elaboración propia.

Para consultar la información a la base de datos, se utilizó la función RecibirDeBD() en donde se utiliza la funcionalidad de RestClient.Get() en donde dentro del paréntesis se colocó el link que provee la base de datos de Firebase junto con la variable tipo texto que almacena el texto del InputField del nombre en la escena del Login y junto con la variable tipo texto del InputField del apellido de la misma escena del Login. Esta función de RecibirDeBD() se encuentra dentro de la función RecibirYVerificarData() que llama a aquella función y verifica que exista un fichero con los datos que el estudiante escribe en el Login y de ser afirmativo, a través de un Switch-Case se redirecciona al estudiante al menú correspondiente en función al valor del valor del grado. Para un mayor detalle de la lógica de funcionamiento, se puede encontrar el script completo en el siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

4.2.4.4 Implementación de Deep Reinforcement Learning

Para implementar esta técnica primero se realizó la descarga de Anaconda Distribution desde su página oficial y en segundo lugar se instaló en el equipo local. Una vez instalado Anaconda, se creó un nuevo *environment* y se activó este a través del Anaconda prompt. Posteriormente, se descargó la librería de ML-Agents de Unity. Para realizar dicha actividad, se descargó la librería desde la página oficial de GitHub de Unity a través del siguiente link <https://github.com/Unity-Technologies/ML-Agents> y se descargó el “Release 1” de la librería dado cuando se desarrolló el prototipo solo se encontraba disponible esta primera versión. Habiendo descargado la librería de forma externa, se procedió a descargar esta misma librería

de forma interna usando el propio motor de videojuegos de Unity a través del Package Manager como se muestra en la Figura N°101.

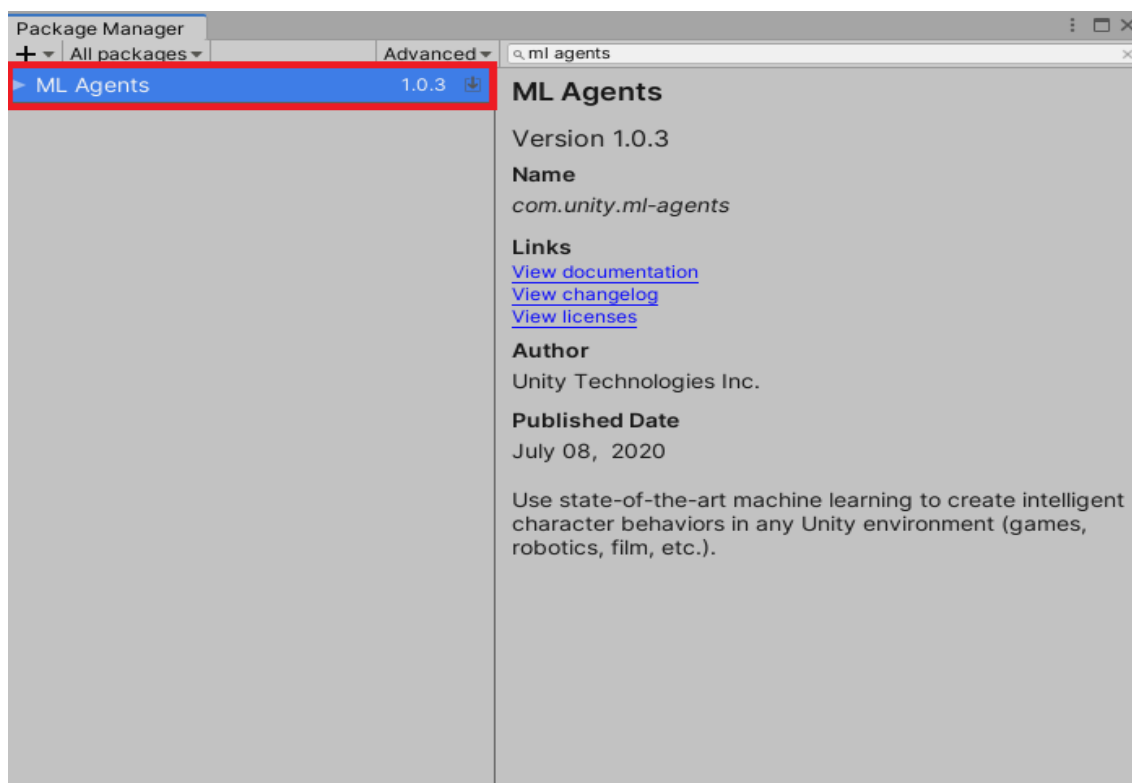


Figura N°101: Descarga de la librería ML-Agents de Unity.

Fuente: Elaboración Propia

Para desarrollar al agente inteligente se agregó un nuevo script llamado “AgentXZ” y se importó la librería ML-Agents escribiendo “Using Unity.MLAgents;” y “Unity.MLAgents.Sensors;”. Al implementar esta librería, permite la utilización de la clase Agent. El script completo del agente inteligente se puede encontrar en el siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

Con respecto a las entradas de la red, estas se encuentran dentro de la función CollectObservations(). Las entradas a la red fueron 4. Las dos primeras corresponden a la posición del agente con respecto al eje x y con respecto al eje z. Las dos últimas corresponden a la posición del objetivo con respecto al eje x y con respecto al eje z. El código para enviar las señales a la red neuronal se presenta en la Figura N°102.

```
public override void CollectObservations(VectorSensor sensor)
{
    // ** Entradas de la red neuronal **

    // Posición del agente
    sensor.AddObservation(transform.localPosition.x);
    sensor.AddObservation(transform.localPosition.z);

    //Posición del objetivo
    sensor.AddObservation(Objetivo.transform.localPosition.x);
    sensor.AddObservation(Objetivo.transform.localPosition.z);
}
```

Figura N°102: Entradas a la red neuronal.

Fuente: Elaboración Propia.

Luego de enviar la información a la red neuronal, esta devuelve sus 2 salidas. El valor de la primera neurona de la capa de salida se obtiene a través de la variable `vectorAction[0]` y el segundo valor de la red se obtiene a través la variable `vectorAction[1]`. Las salidas de la red son transformadas: si la salida es 2, se transforma su valor a 1. En el caso que sea 0 o 1, el valor final es el negativo de la salida. De este modo, se obtiene un desplazamiento equitativo sobre todo el terreno tanto para el eje x y para el eje z. Las salidas transformadas de la red son sumadas a la posición actual del agente con respecto al eje x y con respecto al eje z de tal modo que se consigue un movimiento del modelo en 3D del zombie. Al iniciarse este movimiento, se reproduce una animación de movimiento para darle un mayor realismo al desplazamiento del agente. Para asignarle esta animación, se utilizó la función `Animation().Play("Walking")` para que se reproduzca la animación con ese nombre. De igual modo, se captura la distancia inicial entre el agente y el cubo y luego del desplazamiento producto de las salidas de la red se calcula la distancia final. Si la distancia final es menor a la inicial, se le otorga al agente un refuerzo positivo de +1, de lo contrario se le otorga un refuerzo negativo de -1. La lógica para la ejecución de las acciones del agente y el otorgamiento de las recompensas se presentan en la Figura N°103.

```
public override void OnActionReceived(float[] vectorAction)
{
    Vector3 controlSignal = Vector3.zero;
    controlSignal.x = vectorAction[0];
    controlSignal.z = vectorAction[1];

    if (vectorAction[0] == 2)
    {
        controlSignal.x = 1;
    }
    else
    {
        controlSignal.x = -vectorAction[0];
    }
    if (vectorAction[1] == 2)
    {
        controlSignal.z = 1;
    }
    else
    {
        controlSignal.z = -vectorAction[1];
    }

    float distanceInitial = Vector3.Distance(Objetivo.transform.localPosition,
transform.localPosition);

    float newX = transform.localPosition.x + (controlSignal.x * speed *
Time.deltaTime);

    newX = Mathf.Clamp(newX, -7f, 7f);

    float newZ = transform.localPosition.z + (controlSignal.z * speed *
Time.deltaTime);

    newZ = Mathf.Clamp(newZ, -7f, 7f);

    this.GetComponent<Animation>().Play("Walking");

    transform.LookAt(Objetivo.transform);

    float distanceFinal = Vector3.Distance(Objetivo.transform.localPosition,
transform.localPosition);

    // ** Otorgamiento de los refuerzos positivos y negativos **
    if (distanceFinal < distanceInitial)
    {
        SetReward(1.0f);
    }
    else
    {
        SetReward(-1.0f);
        EndEpisode();
    }
}
```

Figura N°103: Otorgamiento de refuerzos positivos y negativos al agente.

Fuente: Elaboración Propia.

Además, es preciso mencionar que previamente se debe de asignarle a los modelos en 3D de cada zombie el componente de “Behavior Parameters” para poder configurar la red neuronal desde Unity. Como se puede observar en la Figura N°104, para asignarle el número de entradas a la red en el campo de “Space Size”. En el campo de “Branches Size” corresponden a las salidas de la red. En este caso es igual a 2 dado que la red tiene 2 salidas y los campos “Branch 0 Size” y “Branch 1 Size” corresponden a los valores para cada salida de la red. En este caso los valores son discretos empezando desde cero. Al ser el valor de 3, la red tendrá como salida para cada neurona los valores de 0, 1 y 2. Luego estas salidas son transformadas como se explicó anteriormente.

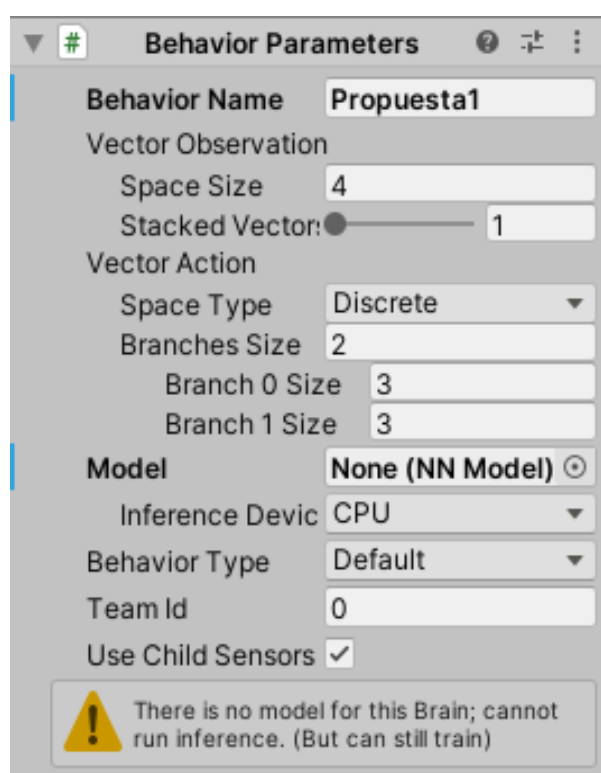


Figura N°104: Configuración de la red neuronal desde Unity para el entrenamiento del agente. Fuente: Elaboración Propia.

Teniendo creado y configurado el agente inteligente, se procedió a crear el ambiente de entrenamiento. Este ambiente es una independiente al videojuego pero que representa a este en su forma más básica. Dado que el videojuego es en el fondo un terreno por donde transita un personaje en primera persona y que debe ser perseguido por los zombies, para el ambiente de entrenamiento se agregó un cubo en 3D y se le dio la forma de un plano que represente al terreno del videojuego y se agregó un nuevo cubo 3D que hizo la función de un objetivo de tal modo que el agente persiga a este cubo de tal modo que represente este objeto al personaje en

primera persona del estudiante. Al centro del ambiente, se colocó al agente de tal modo que pueda perseguir a este cubo. El ambiente de entrenamiento desarrollado se presenta en la Figura N°105.

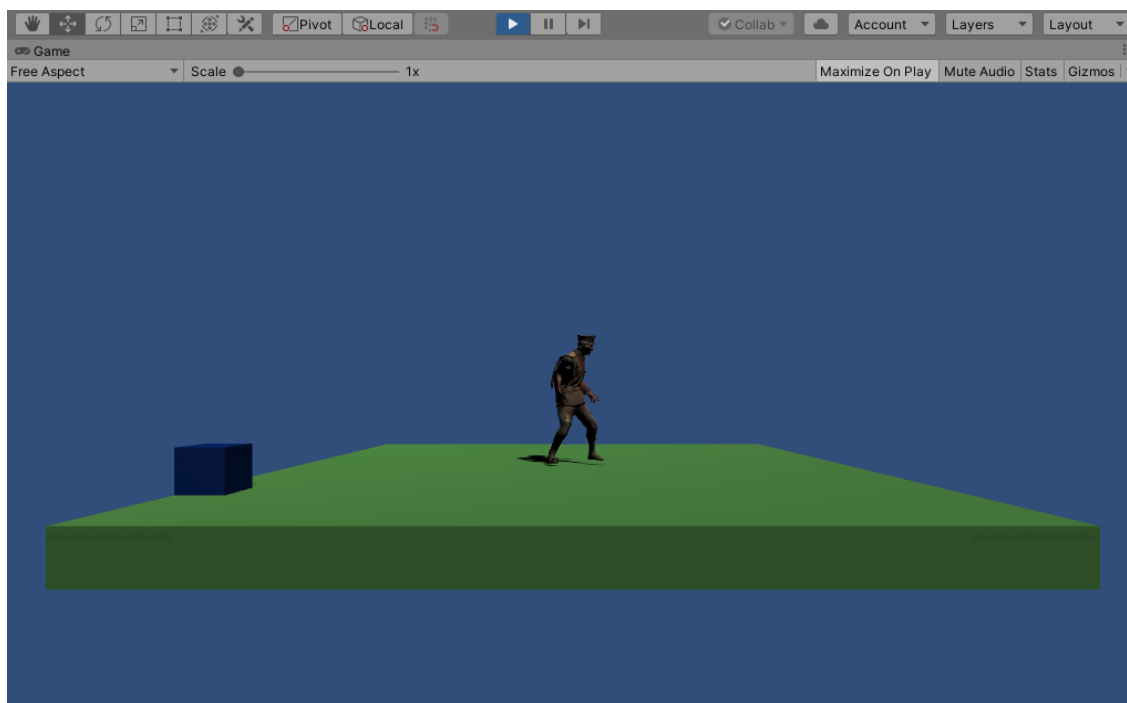
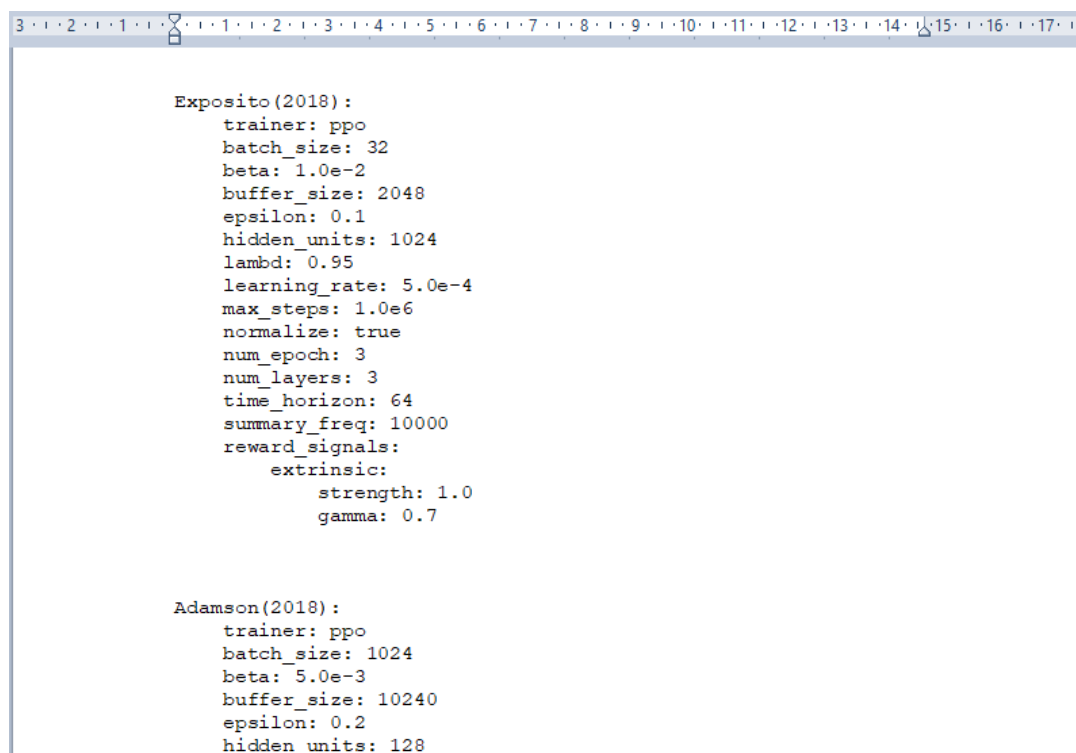


Figura N°105: Ambiente de entrenamiento construido para el entrenamiento del agente.
Fuente: Elaboración Propia.

4.2.4.4.1 Desarrollo de experimentos

Para entrenar a los agentes inteligentes, se abrió el archivo `Trainer_config.yaml` y se insertaron los 7 modelos definidos en el capítulo anterior tal cual como se muestra en la Figura N°106.

De la Figura N°106, se puede observar la aplicación del algoritmo de *Proximal Policy Optimization* dado que el hiperparámetro “Trainer” tiene como valor PPO. Este algoritmo permite aligerar el entrenamiento recortando las nuevas políticas tomadas por el agente aplicando la Ecuación (53) estableciendo un intervalo mínimo y máximo de la longitud de las políticas.



```
Exposito(2018):  
  trainer: ppo  
  batch_size: 32  
  beta: 1.0e-2  
  buffer_size: 2048  
  epsilon: 0.1  
  hidden_units: 1024  
  lambda: 0.95  
  learning_rate: 5.0e-4  
  max_steps: 1.0e6  
  normalize: true  
  num_epoch: 3  
  num_layers: 3  
  time_horizon: 64  
  summary_freq: 10000  
  reward_signals:  
    extrinsic:  
      strength: 1.0  
      gamma: 0.7  
  
Adamson(2018):  
  trainer: ppo  
  batch_size: 1024  
  beta: 5.0e-3  
  buffer_size: 10240  
  epsilon: 0.2  
  hidden_units: 128
```

Figura N°106: Modelos de DRL insertados en el archivo Trainer_config.yaml.

Fuente: Elaboración Propia.

El nombre del modelo de DRL fue colocado en el campo de “Behavior Name” que forma parte del componente de “Behavior Parameters” de tal modo que se localice aquel nombre del modelo dentro del archivo Trainer_config.yaml. Para iniciar con cada experimento se abrió el Anaconda prompt y cambió la ruta a la carpeta donde se encuentra la librería de los ML-Agents y además se activó el *environment* de Anaconda creado anteriormente. Para realizar los 7 experimentos, a cada entrenamiento de un agente se le colocó un ID determinado. El ID es este caso es el nombre “Modelo” seguido del número 1 en el caso que sea el primer entrenamiento, 2 en el caso que sea el segundo y así sucesivamente.

Al iniciar el entrenamiento, se detecta el modelo a entrenar a través del componente de “Behavior Parameters” y se envía un mensaje para iniciar con el entrenamiento presionando el botón de “Play” de la escena de entrenamiento. Al regresar a la escena de Unity y darle “Play” a la escena del ambiente de entrenamiento, se inicia un primer experimento con el nombre del modelo que se encuentra dentro del componente de “Behavior Parameters”. En la consola aparece el nombre del modelo a entrenar y se visualizan sus respectivos hiperparámetros. Luego de la presentación del modelo, se empiezan a visualizar las iteraciones conforme va avanzando el tiempo del modelo. De igual modo, se visualiza la recompensa promedio que el

agente ha obtenido por cada episodio, la desviación estándar de la recompensa y el tiempo demorado en culminar un determinado episodio tal cual como se muestra en la Figura N°107. El mismo procedimiento se repitió para el resto de los modelos hasta completar los 7 experimentos planteados.

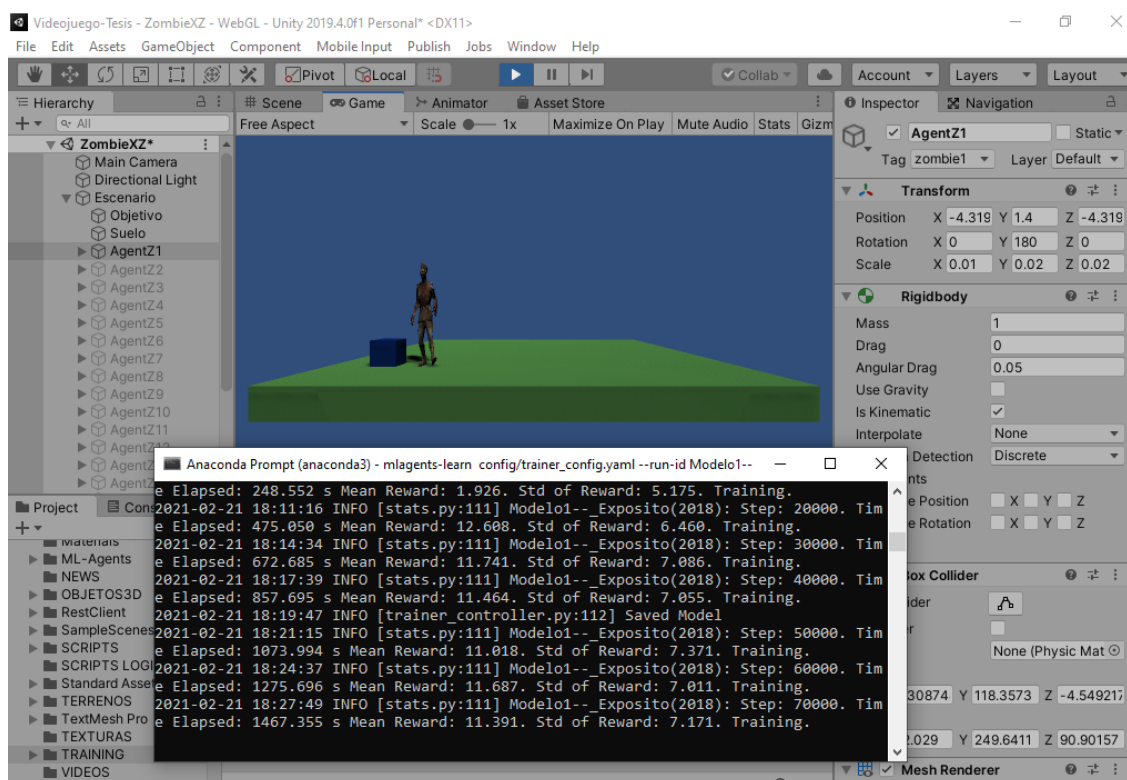


Figura N°107: Entrenamiento de agentes inteligentes.

Fuente: Elaboración Propia.

Al finalizar los entrenamientos de los agentes inteligentes, cada modelo entrenado fue colocado dentro de Unity tal cual como se muestra en la Figura N°108. El mejor modelo entrenado fue la propuesta 3 y fue arrastrado hacia cada modelo de los zombies tal como se muestra en la misma Figura N°108.

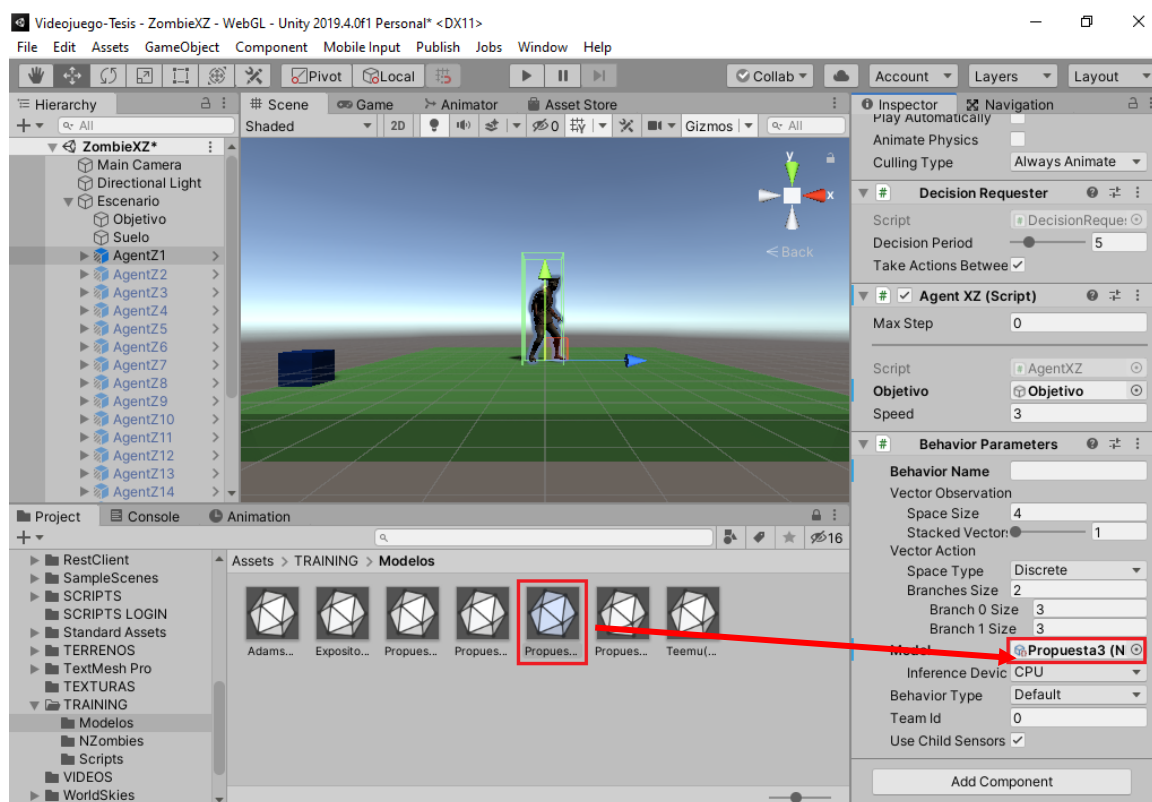


Figura N°108: Implementación del mejor modelo entrenado al agente.

Fuente: Elaboración Propia.

Cuando cada modelo en 3D de los zombies tenía ya implementado el mejor modelo de DRL, aquellos fueron arrastrados a la escena de los 12 niveles del videojuego. Cada nivel posee alrededor de 30 diferentes modelos de zombies entrenados para perseguir un objetivo. Finalmente, en el campo disponible en donde se coloca el modelo en 3D que el zombie debe perseguir, se arrastró el componente de primera persona controlado por el estudiante de tal forma que al dar “Play” al nivel, el zombie persigue al personaje del jugador como se muestra en la Figura N°109. Además, a cada modelo de cada zombie se le asignó el script de ataque descrito anteriormente. Mediante estas configuraciones, se consigue a una Inteligencia Artificial que persiga al jugador y que ataque ante una determinada distancia hasta quitarle todas sus vidas como se muestra en la Figura N°110.

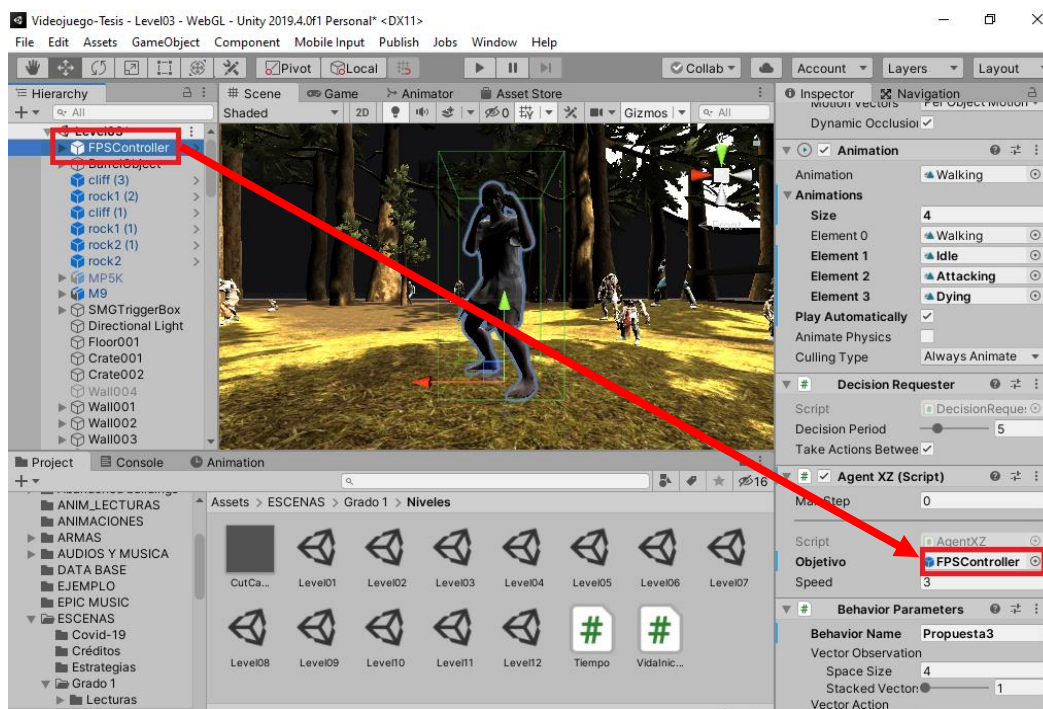


Figura N°109: Asignación del personaje controlado por el jugador al objetivo del gente.
 Fuente: Elaboración Propia.

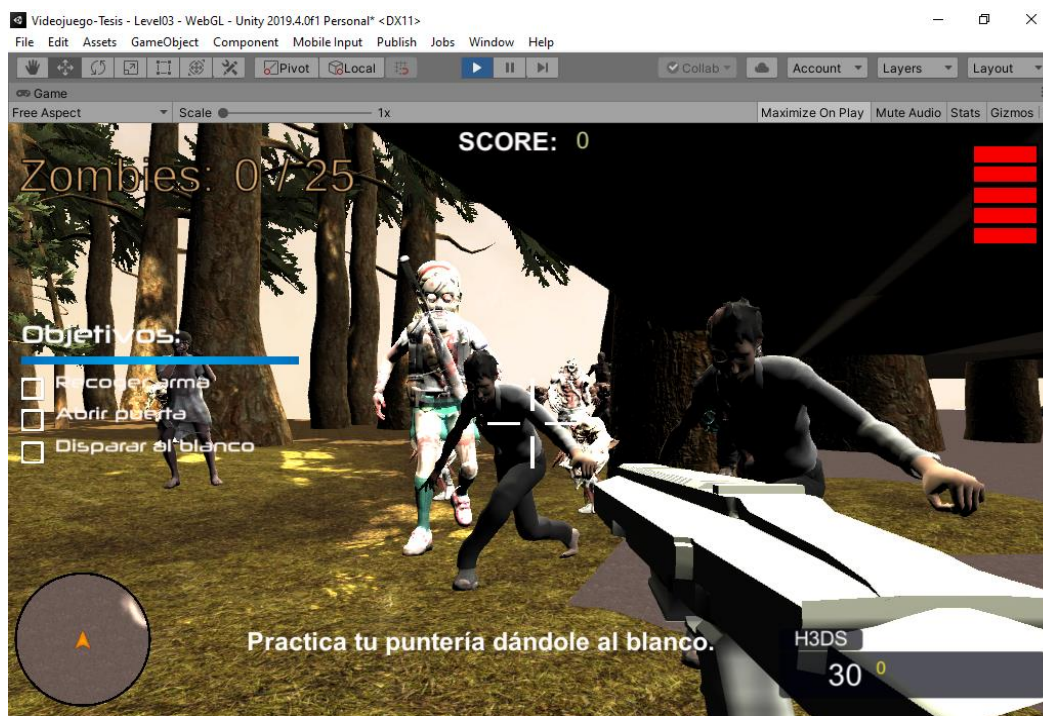


Figura N°110: Agente inteligente implementado en el videojuego desarrollado.
 Fuente: Elaboración Propia.

4.2.4.5 Implementación de técnicas de aprendizaje

Con respecto a la narración de historias y aparición de textos dentro del videojuego, al iniciar el estudiante una nueva partida el videojuego narra una historia para que el jugador pueda leer y entre en contexto con la historia del videojuego. Posteriormente, al momento de iniciar el nivel en la parte inferior aparecen oraciones que complementan la historia narrada al inicio, pero también ayuda al estudiante brindando consejos para que pueda pasar el nivel correspondiente de la mejor manera posible. La implementación de la primera estrategia para que los estudiantes puedan ejercitar su nivel de lectura, se presenta en la Figura N° 111.

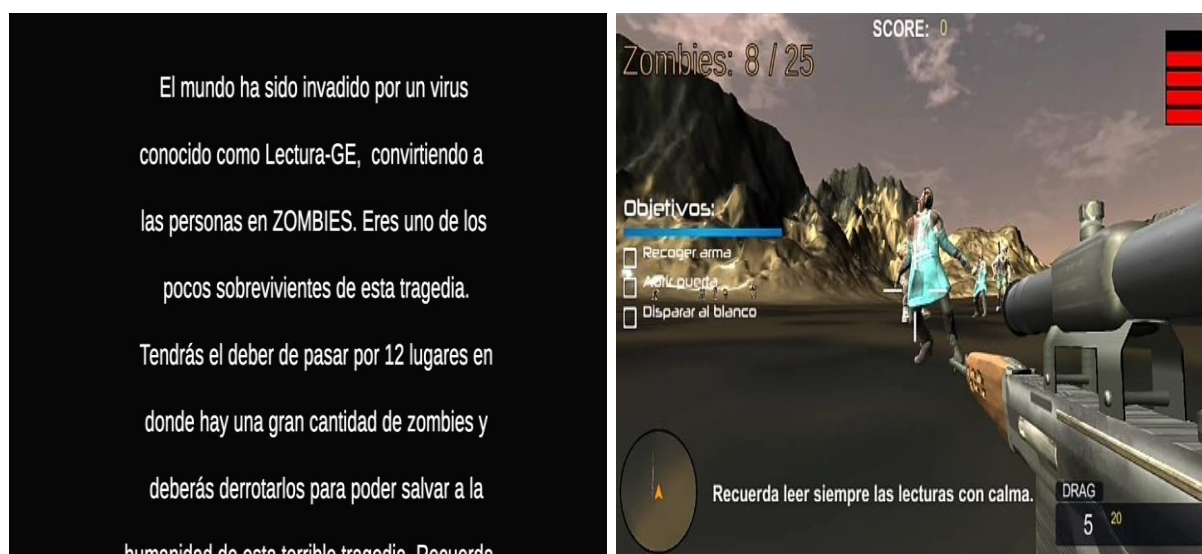


Figura N° 111: Narración de historia (izq.) y oraciones dentro de cada nivel (der.)
Fuente: Elaboración Propia.

La segunda técnica para ejercitar el nivel de comprensión lectora de los estudiantes consiste en la aparición de lecturas dentro del videojuego las cuales los usuarios deben de leer cuando pierdan todas sus vidas. Cuando el usuario pierde sus 5 vidas producto del ataque de los zombies, es redirigido a una lectura correspondiente a su grado de tal modo que debe de responder 3 preguntas con respecto a la lectura leída y responderlas adecuadamente para regresar al nivel donde perdió. Un ejemplo de una escena con una lectura de las clases virtuales con su respectiva preguntase presenta en la Figura N° 112.



Figura N°112: Aparición de lecturas (izq.) y preguntas de opción múltiple (der.)

Fuente: Elaboración Propia.

La última técnica para ejercitar el nivel de comprensión lectora en los estudiantes consiste en la aparición de estrategias de lectura en el videojuego. Esta sección de estrategias aparece cuando el usuario hace click en el botón de “Estrategias” que forma parte del menú general. Para construir estas escenas, se agregaron dos objetos tipo Texto: uno en la parte superior que muestra el número de la estrategia y el otro objeto al centro en donde se visualiza la estrategia. Adicionalmente, esta escena contiene dos botones que son para dirigirse a la siguiente estrategia y el otro es para regresar al menú. Algunos ejemplos de estas escenas construidas se presentan en la Figura N°113.

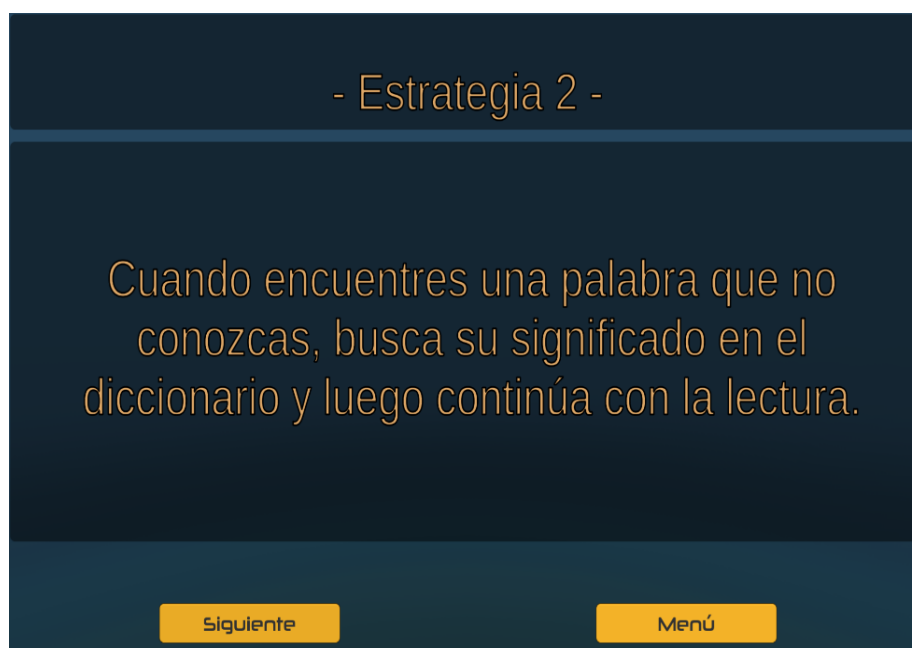


Figura N°113: Sección de estrategias de lectura dentro del videojuego.
Fuente: Elaboración Propia.

4.2.4.6 Clasificación de contenido para cada grado

Para clasificar el contenido del videojuego se creó un objeto vacío con un Tag que lleva el nombre de un número del 1 al 6 de y posee un script con la función DontDestroyOnLoad que se puede visualizar de forma completa a través del enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>. Este script se le asigna a este objeto vacío de tal forma que este identificador se mantiene en todas las escenas. Diferentes scripts desarrollados poseen el código presente en la Figura N°114. Para cada acción de regresar al menú principal si es que se encuentra un estudiante en la sección de estrategias o instrucciones del videojuego, así como si pierde el videojuego, se evalúa si la última escena posee el identificador asignado. Para buscar si una escena posee un Tag respectivo se utiliza la función FindGameObjetctWithTag() y se coloca entre los paréntesis el nombre del tag que se desea buscar. Una vez el tag fue ubicado, con un condicional redirige al estudiante al menú correspondiente al grado que corresponde. Esta misma lógica se aplica para que al estudiante le aparezcan solamente las lecturas correspondientes a su grado teniendo en cuenta además el número de la escena donde se encuentra cada lectura de tal manera de poder redireccionar al usuario correctamente al texto respectivo.

```
IEnumerator End()
{
    GameObject menu1 = GameObject.FindGameObjectWithTag("1");
    GameObject menu2 = GameObject.FindGameObjectWithTag("2");
    GameObject menu3 = GameObject.FindGameObjectWithTag("3");
    GameObject menu4 = GameObject.FindGameObjectWithTag("4");
    GameObject menu5 = GameObject.FindGameObjectWithTag("5");
    GameObject menu6 = GameObject.FindGameObjectWithTag("6");

    //Grado1: 4
    //Grado2: 405
    //Grado3: 406
    //Grado4: 407
    //Grado5: 408
    //Grado6: 409

    if (menu1)
    {
        yield return new WaitForSeconds(7);
        SceneManager.LoadScene(4);
    }

    if (menu2)
    {
        yield return new WaitForSeconds(7);
        SceneManager.LoadScene(405);
    }

    if (menu3)
    {
        yield return new WaitForSeconds(7);
        SceneManager.LoadScene(406);
    }

    if (menu4)
    {
        yield return new WaitForSeconds(7);
        SceneManager.LoadScene(407);
    }

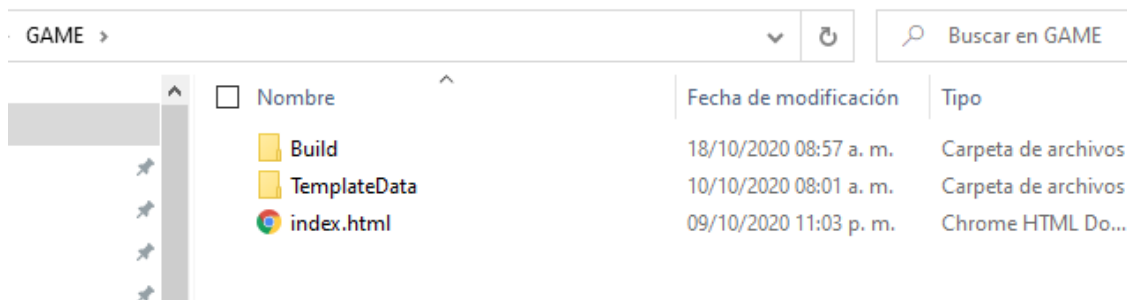
    if (menu5)
    {
        yield return new WaitForSeconds(7);
        SceneManager.LoadScene(408);
    }

    if (menu6)
    {
        yield return new WaitForSeconds(7);
        SceneManager.LoadScene(409);
    }
}
```

Figura N°114: Búsqueda de Tag correspondiente para redireccionar a escenas específicas.
Fuente: Elaboración propia.

4.2.4.7 Implementación del servicio de hosting

Finalizado el desarrollo del videojuego, se procedió a exportar proyecto. Se creó una carpeta para almacenar los 3 archivos exportados que son el Index.html, la carpeta “TemplateData” que almacena todos los íconos e imágenes que el archivo Index.html posee y la carpeta “Build” que contiene todos Assets del videojuego tal cual como se muestra en la Figura N°115.



Nombre	Fecha de modificación	Tipo
Build	18/10/2020 08:57 a. m.	Carpeta de archivos
TemplateData	10/10/2020 08:01 a. m.	Carpeta de archivos
index.html	09/10/2020 11:03 p. m.	Chrome HTML Do...

Figura N°115: Archivos creados por Unity en la exportación.

Fuente: Elaboración propia.

Teniendo el archivo exportado, se procedió a instalar Node.js desde su página oficial. Luego, se procedió a abrir el Node.js command prompt para realizar la instalación de los componentes de Firebase. Posteriormente, se ejecutó el símbolo del sistema y se cambió la ruta hacia la carpeta donde se encuentran los archivos exportados del videojuego.

Cuando se selecciona el servicio de hosting de Firebase, se seleccionó el proyecto con el cual se iba a trabajar que fue el mismo creado al inicio para la base de datos en tiempo real. Al momento que se selecciona el proyecto, dentro de la carpeta donde se exportó el videojuego, se crearon automáticamente nuevos archivos que corresponden a las funcionalidades de Firebase. En la carpeta llamada “Public” se colocaron los 3 archivos exportados de Unity.

El resultado final luego de ejecutar el comando “Firebase deploy” en la consola fue el siguiente link <https://videojuego-v2.web.app/> el cual es el link final que se coloca en el navegador y el videojuego desarrollado se reproduce en la computadora. La visualización del videojuego en la computadora se presenta en la Figura N°116.

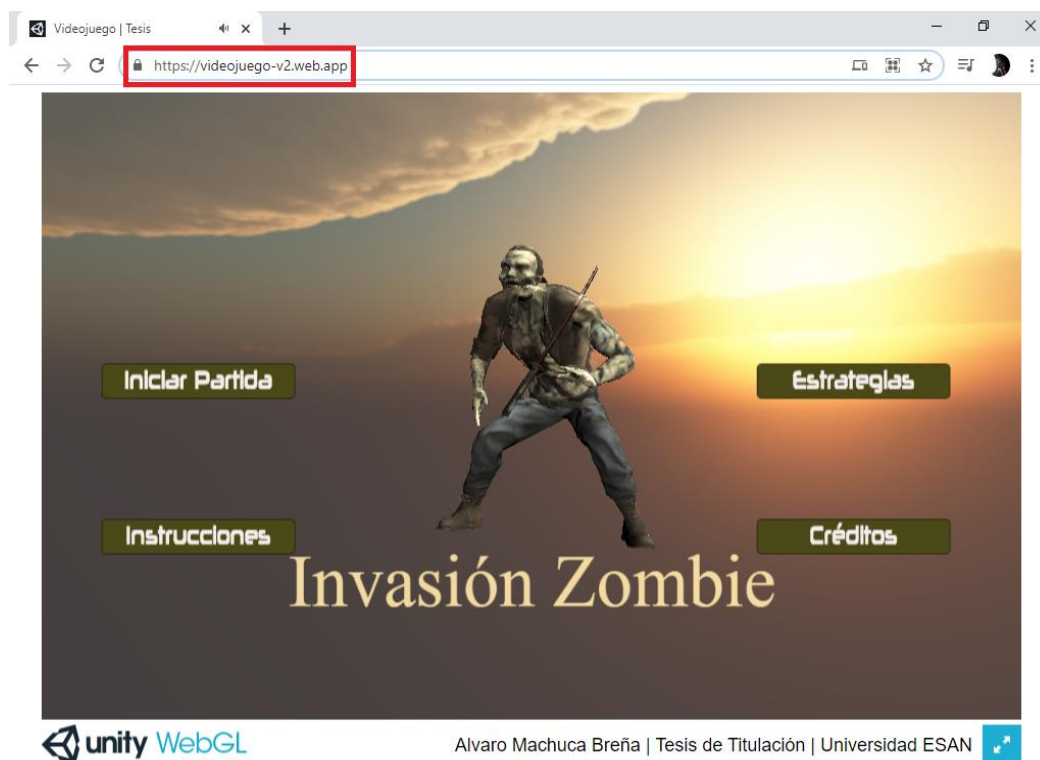


Figura N°116: Reproducción del videojuego final en la computadora.
Fuente: Elaboración propia.

4.2.4.8 Pruebas de funcionalidad

Se realizaron las verificaciones con respecto a las principales escenas del videojuego con el objetivo de que los scripts funcionen correctamente. Con respecto a la base de datos, se probó ingresando a un nuevo usuario tal cual como se muestra en la Figura N°117. Cuando se hace click en el botón de “Guardar” se activa la función de `EnviarPorPost()` y se envía la información a la base de datos de Firebase.

Si el envío de información funciona correctamente, se debe de crear un fichero nuevo con el nombre que el estudiante ha enviado. De igual modo, al desplegar el fichero se deben de visualizar los atributos de la clase `Usuario` de forma que se visualice el nombre, apellido y grado al cual pertenece el estudiante tal cual como se muestra en la Figura N°118. De igual modo, se comprobó que a través del Login, el videojuego redireccionaba al menú correspondiente llamando a la función `RecibirDeBD()` desarrollada.



Figura N°117: Prueba de envío de información a la base de datos.
Fuente: Elaboración propia.

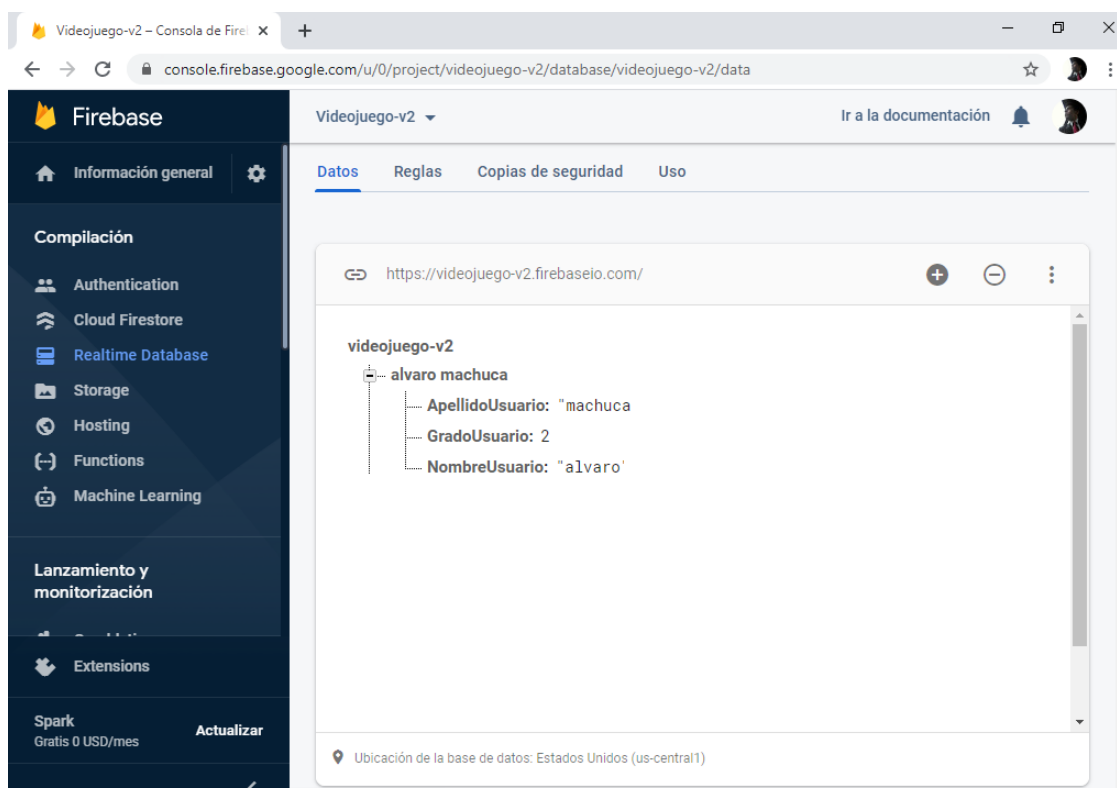


Figura N°118: Verificación de almacenamiento de información en la base de datos.
Fuente: Elaboración propia.

Luego de haber realizado la revisión individual por escenas y haber realizado el deploy respectivo utilizando el servicio de hosting de Firebase, se procedió a realizar una última revisión general del videojuego.

El videojuego fue compartido con 3 personas de prueba para determinar si había alguna diferencia o dificultad de carga. No obstante, el videojuego se comportó de manera adecuada para cada usuario. Las personas que realizaron la prueba tenían una conexión a internet de 23 Mbps, 25 Mbps y 15 Mbps. Estos usuarios afirmaron que el tiempo de carga en sus computadoras fue de 40 segundos, 35 segundos y 70 segundos respectivamente, por lo que en promedio se puede afirmar que el videojuego tiene un tiempo de carga aproximado de 48 segundos en una conexión a internet de aproximada de 21 Mbps. No obstante, el tiempo de carga varía en función a la velocidad de internet que se posea.

De igual modo, de forma local se realizaron las pruebas de las escenas del videojuego tanto de los niveles, aparición de lecturas, estrategias de lectura, así como de instrucciones y escenas finales de los créditos. No se detectaron errores de carga de modelos en 3D y los scripts implementados funcionaron de manera adecuada para cada escena del videojuego.

4.2.5 Fase de despliegue

Una vez finalizado el videojuego y habiendo realizado las pruebas de funcionalidad, el prototipo fue compartido a los profesores a través del link creado. Los profesores pudieron jugar el videojuego desde sus computadoras y quedaron satisfechos con el producto desarrollado dando su aprobación para la utilización del videojuego en las clases virtuales de los estudiantes. En esta ocasión, no se estableció una coordinación con los salones de cómputo del colegio dado que los niños jugaron desde sus hogares dada las circunstancias producto de la pandemia del Covid-19. Más bien los docentes compartieron el link a los estudiantes para que estos puedan acceder desde sus hogares. Para obtener una evidencia de que los estudiantes están usando el videojuego, se solicitó además una foto de cada niño jugando el videojuego desarrollado.

4.2.5.1 Retroalimentación del proyecto realizado

El videojuego construido fue desarrollado en función a las especificaciones brindadas por el docente del curso de lectura. En segundo lugar, se hizo una descripción de los recursos que se poseían para el desarrollo de este y se realizó un cronograma de trabajos para poder llevar a cabo cada fase de la metodología en cascada. Se cumplieron las fechas establecidas en el cronograma de trabajo culminando el videojuego a finales de setiembre con la validación de

los docentes. Con respecto a la fase del modelado, se desarrollaron los casos de uso del videojuego a construir de forma que se pueda representar y detallar el comportamiento del sistema con el usuario. De igual modo, se realizaron los bosquejos iniciales del videojuego, así como la arquitectura del prototipo.

En la etapa de desarrollo, se crearon las escenas diseñadas en la etapa anterior y además se entrenaron a los agentes inteligentes de tal forma de poder implementar al mejor agente dentro del videojuego. Se realizaron algunas pruebas previas al entrenamiento con los propios ejemplos que provee Unity de tal forma que se puedan entender los resultados de la impresión en la consola, así como los resultados de los gráficos provenientes del Tensorboard. Además, se hizo la construcción de la base de Firebase y se realizó la conexión respectiva con Unity a través de la librería de “Rest Client For Unity” para el envío de información y para realizar la consulta respectiva a la base de datos. También, se implementaron las 3 técnicas de lectura para que los estudiantes puedan ejercitar y mejorar su nivel de comprensión lectora. Finalmente, se realizaron las pruebas de funcionalidad respectiva por cada escena de tal forma de poder determinar que cada script desarrollado cumple con su función correspondiente dentro de una escena en particular. Además, se hizo la verificación respectiva con respecto a la funcionalidad de la base de datos registrando un nuevo usuario e ingresando al videojuego con el usuario creado. Una vez verificado el correcto funcionamiento, se realizó el deploy del videojuego con el servicio de hosting de Firebase y se hizo una última prueba general del prototipo en funcionamiento.

Por último, se realizó la entrega correspondiente al cliente a través del link del videojuego de tal forma que se pudo verificar que el producto desarrollado funciona correctamente dando la aprobación correspondiente para que el videojuego pueda ser utilizado en las aulas virtuales de clase.

4.3 Protocolo de aprendizaje

Una vez desarrollado el videojuego y habiendo aprobado los docentes su uso dentro de las clases de lectura, se procedió a desarrollar el protocolo de aprendizaje con el objetivo de establecer una distribución del tiempo entre las clases tradicionales de comprensión lectora y el uso del videojuego.

Con respecto a las fechas de entrenamiento, se definió el inicio el lunes 5 de octubre hasta el viernes 6 de noviembre. Los estudiantes llevan el curso de comprensión lectora 2 veces por semana con duración de cada una de 45 minutos. Se estableció un tiempo de 35 minutos

destinados a las clases de lectura y los últimos 10 minutos corresponden al uso del videojuego por parte de los estudiantes.

Del mismo modo, con respecto a las evaluaciones, el primer día de las clases correspondientes a cada grado, se realizó el pretest y el último día de las evaluaciones se realizó el postest ambos por un periodo de 40 minutos. Además, seguido del postest los últimos 5 minutos se realizó la encuesta de satisfacción con el objetivo de determinar si el videojuego fue del agrado de los estudiantes pertenecientes al grupo experimental para cada grado.

Los estudiantes de primer grado de primaria llevan sus clases los lunes y martes, los del segundo grado llevan las clases de lectura los martes y jueves. De igual modo, los alumnos del tercer grado de primaria tienen sus clases de lectura los lunes y martes y los del cuarto grado de primaria los martes y jueves. Finalmente, los estudiantes de quinto y sexto grado de primaria llevan sus clases de lectura los miércoles y viernes. El protocolo de aprendizaje desarrollado se presenta en la Tabla 31 en donde se aprecia los días de entrenamiento por semanas para cada grado dado que el curso se ve 2 veces por semana, el tiempo destinado para las clases tradicionales y el uso del videojuego, el total de semanas de entrenamiento y las fechas destinadas para las evaluaciones del pretest y postest.

Tabla 31: Protocolo de aprendizaje desarrollado para el entrenamiento de los estudiantes de primaria

ENTRENAMIENTO DE LOS ESTUDIANTES					EVALUACIONES	
Grado	Días de entrenamiento	Clases tradicionales	Uso del videojuego	Total de semanas	Pretest(*)	Postest (*)(**)
1ro	Lunes y martes	35 minutos	10 minutos	5	05-10-2020	03-11-2020
2do	Martes y jueves	35 minutos	10 minutos	5	06-10-2020	05-11-2020
3er	Lunes y martes	35 minutos	10 minutos	5	05-10-2020	03-11-2020
4to	Martes y jueves	35 minutos	10 minutos	5	06-10-2020	05-11-2020
5to	Miércoles y viernes	35 minutos	10 minutos	5	07-10-2020	06-11-2020
6to	Miércoles y viernes	35 minutos	10 minutos	5	07-10-2020	06-11-2020

(*) Duración de la evaluación: 40 minutos

(**) Se realizó además la encuesta de satisfacción. Duración: 5 minutos

Fuente: Elaboración Propia.

4.4 Entrenamiento de los estudiantes

Los docentes compartieron a link a los estudiantes de la sección “B” de cada grado y estos alumnos accedieron al videojuego para poder jugarlo en los 10 últimos minutos de cada clase hasta completar las 5 semanas. No se presentaron inconvenientes por los estudiantes al entrar al videojuego o inconvenientes con respecto a la visualización de un contenido del videojuego en particular. En la Figura N°119 se pueden apreciar a los estudiantes de 1ro a 3ro de primaria jugando el videojuego y en la Figura N°120 se puede visualizar a estudiantes de 4to a 6to grado de primaria de igual manera jugando el videojuego. Además, los profesores desarrollaron los test respectivos antes y después de la intervención para cada grado. Los pretest y postest completos se pueden descargar a través del siguiente enlace que redirecciona a un repositorio creado en GitHub <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>. De igual modo, al final el postest, los estudiantes realizaron la encuesta de satisfacción presentada en el capítulo anterior y que se puede visualizar en el Anexo 5.



Figura N°119: Entrenamiento de estudiantes - Primer grado (izq.), Segundo grado (centro), Tercer grado (Der.).
Fuente: Elaboración propia.



Figura N°120: Entrenamiento de estudiantes - Cuarto grado (izq.), Quinto grado (centro), Sexto grado (Der.).
Fuente: Elaboración propia.

CAPITULO V: ANÁLISIS Y DISCUSIÓN DE RESULTADOS

Habiendo realizado el desarrollo completo del videojuego a través de la metodología en cascada y luego de que los estudiantes realizaron el entrenamiento a través del uso del videojuego y siguiendo el protocolo de aprendizaje establecido, como última etapa del proyecto de investigación, se procedió a reportar los resultados obtenidos con respecto al videojuego desarrollado a través de la metodología en cascada, los resultados de los experimentos de entrenamiento de los agentes inteligentes a través de los 7 modelos de DRL planteados y los resultados tanto cuantitativos como cualitativos del entrenamiento de los estudiantes con el objetivo de determinar si hubo una mejora significativa. Todos los resultados descritos anteriormente serán presentados y analizados en el presente capítulo.

5.1 Medición de resultados – Metodología aplicada al videojuego

Con respecto a la fase de comunicación, se realizaron 2 actividades y tuvo una duración general de una semana. En primer lugar, se estableció una reunión con el docente de lectura y se realizó una recolección todos los requerimientos para el videojuego. En segundo lugar, estos requerimientos fueron clasificados en funcionales y no funcionales. Los resultados de esta etapa se presentan en la Tabla 32.

Tabla 32: Resultados de la metodología en cascada – Fase de Comunicación

Requerimientos	Cantidad
Funcionales	13
No funcionales	4
Total de requerimientos recopilados	17

Fuente: Elaboración Propia.

En relación con la fase de planeación, se verificó contar con los recursos necesarios para poder desarrollar el videojuego. Se hizo una descripción de las principales características que poseía del recurso a utilizar y se realizó un cronograma de trabajo para el desarrollo de cada etapa de la metodología, así como establecer las fechas para cada etapa. El total de actividades que se desarrollaron en esta etapa fueron 2 y en total tuvo una duración de una semana. Los resultados de esta etapa son principalmente las características del cronograma de trabajo desarrollado. El total de actividades por cada etapa de la metodología en cascada, así como el tiempo establecido como cada etapa, se presenta en la Tabla 33.

Tabla 33: Resultados de la metodología en cascada – Fase de Planeación

Fase	Cantidad de Actividades	Tiempo establecido
Comunicación	2	1 semana
Planeación	2	1 semana
Modelado	10	22 semanas
Desarrollo	13	21 semanas
Despliegue	2	2 días
Total	29	45 semanas con 2 días

Fuente: Elaboración Propia.

En referencia a la fase de modelado, se desarrolló el diagrama de casos de usos del videojuego a construir que consta de 6 diferentes casos de uso y se realizaron las especificaciones de estos con el fin de plasmar el comportamiento que iba a tener el videojuego con el usuario. La temática del videojuego fue de aventuras y ciencia ficción dado que este videojuego coloca en un contexto particular al estudiante. Este contexto es la aparición de virus en el mundo que convirtió a las personas en zombies y el estudiante es uno de los sobrevivientes que debe liberar a 12 diferentes ciudades de estos zombies para salvar el mundo. De igual modo, este videojuego es de tipo educativo enfocándose en el tema de la lectura comprensiva con 3 diferentes técnicas de aprendizaje para que los estudiantes puedan ejercitar su nivel de comprensión lectora.

Posteriormente, se procedió a desarrollar los bosquejos y arquitectura de las escenas del videojuego. Se realizaron en total 8 bosquejos diferentes con sus 8 respectivas arquitecturas. Las escenas que se modelaron fueron la escena del Login, Registro de los usuarios, Menú principal, de un nivel, de las lecturas, preguntas de opción múltiple, escena final de felicitaciones y la escena de estrategias de lectura. Seguidamente se procedió a realizar la secuencia de GamePlay del videojuego y, además, se hizo la descarga respectiva de los modelos en 3D de los personajes enemigos, de los terrenos dentro del videojuego y de las armas disponibles para los estudiantes. Se descargaron 18 modelos diferentes para los enemigos, 12 modelos de terrenos en 3D y 10 diferentes modelos de armas. Adicionalmente, de 5 links se obtuvo la música y sonidos para el videojuego desarrollado. Finalmente, se obtuvo el material educativo que son las lecturas utilizadas para grado. En total se recopilieron 72 lecturas para el videojuego. Son 12 lecturas para cada uno de los 6 grados que se encuentran en cada uno de

los 12 niveles y son 3 preguntas de opción múltiple para cada grado. Un resumen de los resultados obtenidos para la fase de modelado se presenta en la Tabla 34.

Tabla 34: Resultados de la metodología en cascada – Fase de Modelado

Actividad realizada	Total
Casos de usos desarrollados	6 casos de usos desarrollados
Bosquejos desarrollados	8 bosquejos desarrollados
Arquitecturas de escenas desarrolladas	8 arquitecturas de escenas desarrolladas
Personajes enemigos en 3D descargados	18 personajes en 3D descargados
Terrenos en 3D descargados	12 terrenos en 3D descargados
Modelos en 3D de armas descargados	10 armas en 3D descargados
Links para la música y sonidos	5 links utilizados
Lecturas para el videojuego	72 (12 lecturas para cada grado)

Fuente: Elaboración Propia.

En relación con la fase de desarrollo, se desarrollaron en total 431 escenas para el videojuego y 107 diferentes scripts para su funcionamiento. El total de niveles que posee el videojuego son 12. Cada nivel presenta el mismo grado de complejidad dado cada nivel posee en los agentes enemigos el mismo modelo de DRL seleccionado. El total de modelos en 3D implementados en el videojuego fueron de 40 diferentes modelos en donde se encuentran los personajes enemigos, los terrenos y los modelos de armas. Del mismo modo, en este videojuego se aplicaron 3 diferentes tecnologías. En primer lugar, se aplicó realidad virtual no inmersiva para el control del videojuego a través del uso de una pantalla de computadora y el uso del teclado. Para aplicar esta tecnología se exportó el videojuego a través de la opción de WebGL, así como se utilizó la librería de “*Standard Assets*” de Unity para introducir un personaje en primera persona que pueda controlar el usuario. La segunda tecnología que se aplicó fue la base de datos en tiempo real de Firebase. Para aplicar esta tecnología se creó un nuevo proyecto en Firebase y luego una nueva base de datos. Al realizar esta última actividad, Firebase proporcionó un link para realizar la conexión. Por el lado de Unity, se descargó la librería “*Rest Client for Unity*” y a través de su función de `RestClient.Put()` y `RestClient.Get()` se envió la envió y consultó a la base de datos respectivamente colocando dentro de los paréntesis de las funciones GET y POST el link que provee Firebase junto con los textos que el estudiante ingresa en los `InputField` de sus nombres, apellidos y el grado al cual corresponde si corresponde al registro de los estudiantes y solamente sus nombres y apellidos si corresponde

al Login. La última tecnología aplicada fue el servicio de Hosting de Firebase con el objetivo de realizar el Deploy respectivo y colocar el videojuego en los servidores de Firebase de tal modo que al final se obtuvo un link final el cual fue colocado en los navegadores de tal modo que el videojuego pudo reproducirse en las computadoras de los estudiantes. Adicionalmente, se aplicó la técnica de *Deep Reinforcement Learning* con el objetivo de entrenar a 7 diferentes agentes a través de 7 modelos establecidos de DRL y poder seleccionar al mejor agente el cual fue implementado en el videojuego. Para implementar esta técnica se descargó la librería ML-Agents de Unity y también se descargó Anaconda Distribution. Dentro de Unity se creó un ambiente de entrenamiento junto con un script para construir el agente inteligente y para que este pueda recibir las salidas de la red neuronal y para que además pueda recibir los refuerzos positivos y negativos. Por otro lado, dentro del videojuego se implementaron 3 técnicas diferentes para ejercitar y poder mejorar el nivel de comprensión lectora en los estudiantes. La primera técnica consiste en la aparición de textos dentro del propio videojuego a través de la narración de una historia y la aparición de oraciones dentro de los niveles. El total de oraciones que el videojuego posee son 21 que aparecen cada 20 segundos en cada nivel ayudando al estudiante a pasar este nivel de la mejor manera. La segunda técnica para ejercitar el nivel de comprensión lectora es la aparición de lecturas que el estudiante deberá leer para poder regresar al nivel donde perdió sus 5 vidas. En total se introdujeron 72 lecturas dentro del videojuego y cada lectura con 3 preguntas de opción múltiple diferentes. Como última estrategia se introdujo una sección de estrategias de lecturas en donde se plasmaron 15 diferentes estrategias para que el estudiante pueda tenerlas en cuenta y pueda aplicar esas estrategias a las lecturas del videojuego y en sus clases de lectura. Con respecto al total de música y sonidos, se introdujeron en total 51 diferentes audios dentro del videojuego y se crearon 6 diferentes menús para cada grado de educación primaria. El resumen de los resultados de la fase de desarrollo se presenta en la Tabla 35.

Tabla 35: Resultados de la metodología en cascada – Fase de Desarrollo

Actividad realizada	Total
Total de escenas construidas	431 escenas
Scripts desarrollados	107 scripts
Niveles del videojuego	12 niveles
Modelos en 3D implementados	40 modelos en 3D
Tecnologías aplicadas	3 tecnologías diferentes aplicadas
Técnica de Inteligencia Artificial aplicada	1 técnica (Deep Reinforcement Learning)
Modelos de DRL entrenados	7 modelos diferentes
Agentes inteligentes implementados	1 agente inteligente implementado
Técnicas para el aprendizaje de la comprensión lectora	3 técnicas aplicadas para el aprendizaje de la comprensión lectora
Oraciones implementadas	21 oraciones diferentes
Lecturas implementadas	72 diferentes lecturas
Estrategias de lectura implementadas	15 estrategias diferentes de lectura
Música y sonidos implementados	51 diferentes audios implementados
Menús construidos	6 menús construidos para cada grado

Fuente: Elaboración Propia.

Finalmente, con respecto a la fase de despliegue, tuvo esta una duración de dos días en donde se realizó la entrega final del producto final. Los profesores ingresaron desde las computadoras de sus casas al videojuego y dieron su aprobación para poder utilizar el videojuego en las clases virtuales. No se obtuvieron observaciones con el videojuego desarrollado dado que se construyó el videojuego respetando los requerimientos brindados por parte del docente.

5.2 Medición de Resultados – Experimentos de DRL

En esta sección, se presentan los resultados generales de los modelos de DRL así como los resultados del mejor modelo seleccionado.

5.2.1 Resultados generales de los modelos de DRL desarrollados

En relación con la gráfica de *Cumulative Reward* del entrenamiento de los 7 agentes inteligentes, esta se presenta en la Figura N°121.

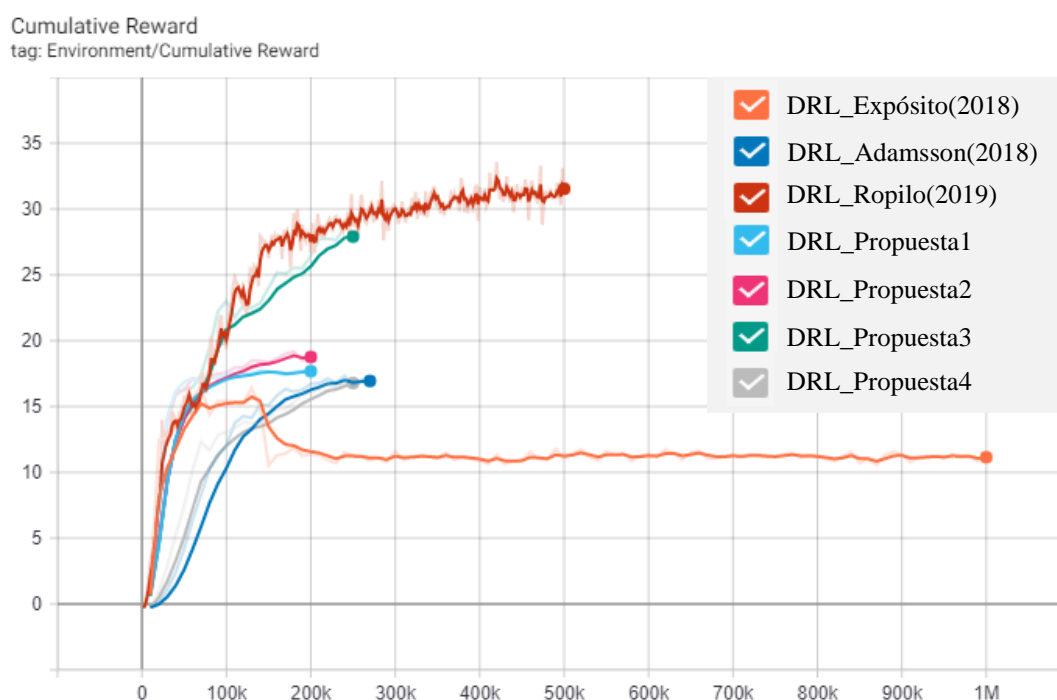


Figura N°121: Resultados de los modelos de DRL desarrollados – Cumulative Reward.
Fuente: Elaboración propia.

En la anterior gráfica se puede apreciar la mayoría de los agentes empiezan con un *Reward* un poco mayor a cero lo cual es un buen indicio que el entrenamiento empieza de forma correcta dado que el *Reward* inicial empieza siendo positivo. Conforme avanza el entrenamiento, el *Reward* se va acumulando, aplicando la Ecuación (41). Para determinar si el entrenamiento fue exitoso, la gráfica debe de ir en aumento dado que este comportamiento quiere decir que el agente está aprendiendo cada vez más a ejecutar la tarea se seguimiento encomendada. Analizando las tendencias de las gráficas, todas tienden a incrementarse, pero el modelo de Expósito (2018) tiende a ser casi constante cuando llega a un *Reward* acumulado de un poco más de 10 en un poco más de 100 000 iteraciones mientras que los demás siguen en aumento. Los dos modelos que más resaltan son el modelo de Ropilo (2019) y el modelo de la propuesta 3 dado que tienen los *Rewards* más elevados. Se aprecia que el modelo de la propuesta 3 tiene un incremento más estable que el modelo de Ropilo (2019) dado que este último su *Reward* tiene muchas subidas y bajadas conforme avanza el entrenamiento siendo menos estable. Los 4 modelos restantes llegaron a un *Reward* mayor a 15 pero menor a 20 en un poco menos de 300 000 iteraciones.

Con respecto a la gráfica de *Episode Length* del entrenamiento de los 7 agentes inteligentes, esta se puede visualizar en la Figura N°122.

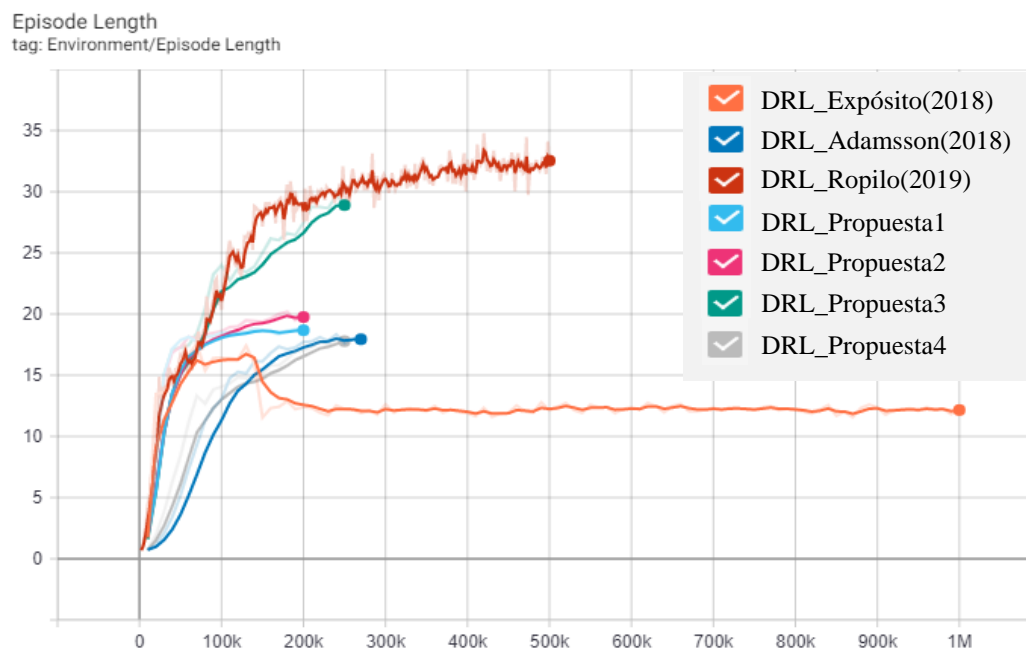


Figura N°122: Resultados de los modelos de DRL desarrollados – Episode Length.

Fuente: Elaboración propia.

En la Figura N°122, se puede apreciar que el promedio de acciones que suele realizar un agente tiene a ir en incremento. Obviamente, todos empiezan en un punto superior a cero dado que no hay número de acciones que puedan ser negativas. La gráfica tiende a incrementarse debido a la interacción que existe entre el agente y el ambiente de entrenamiento desarrollado. El objetivo del agente es acumular la mayor cantidad de recompensa posible por lo que cada vez trata realizar la mayor cantidad de acciones posibles de tal modo de incrementar su recompensa por episodio. En el caso del modelo de Expósito (2018) ha tenido un menor promedio de acciones llegando a un valor aproximado de 12. Los modelos que han tenido mayor interacción con el ambiente de entrenamiento han sido el modelo de Ropilo (2019) y la propuesta 3 con un promedio de acciones aproximado de 32 y 28 respectivamente. El modelo de Adamsson (2018) con las otras 3 propuestas han tenido una exploración media con el ambiente de entrenamiento construido llegando a un promedio de acciones aproximado de 17 hasta 20 acciones en general por episodio.

Haciendo referencia a la gráfica de *IsTraining* para el entrenamiento de los 7 agentes inteligentes, esta se presenta en la Figura N°123.

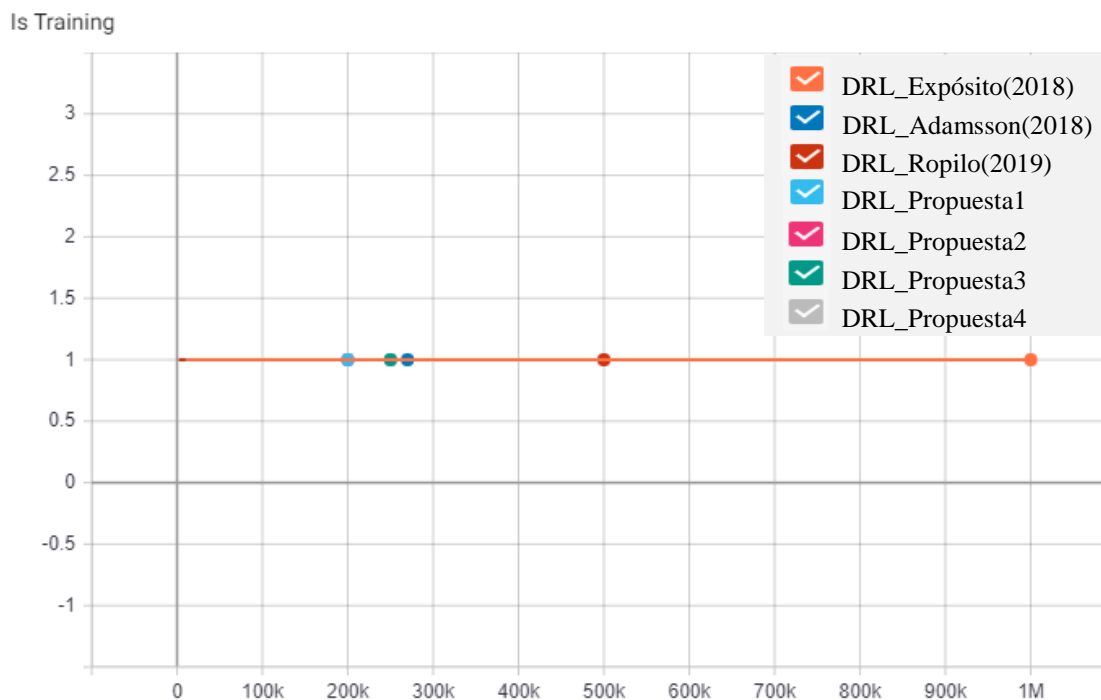


Figura N°123: Resultados de los modelos de DRL desarrollados – *IsTraining*.
Fuente: Elaboración propia.

En la gráfica del *IsTraining* se aprecia que todos los entrenamientos han sido continuos dado que el valor para cada modelo se ha mantenido en 1. No se realizaron interrupciones a ningún modelo de DRL. También se aprecia que el modelo de Expósito (2018) ha tenido un mayor número de iteraciones hasta llegar a un millón de iteraciones. El segundo modelo con mayor número de iteraciones es el de Ropilo (2019) con 500 000 iteraciones. Luego de este modelo le sigue el de Adamsson (2018) con 270 000 iteraciones. La propuesta 3 y propuesta 4 tienen un número de iteraciones de 250 000 y los 2 modelos con menor número de iteraciones son la propuesta 1 y propuesta 2 llegando a 200 000 iteraciones.

Con respecto a la gráfica del *PolicyLoss* en el entrenamiento de los 7 agentes inteligentes, esta gráfica se puede visualizar en la Figura N°124.

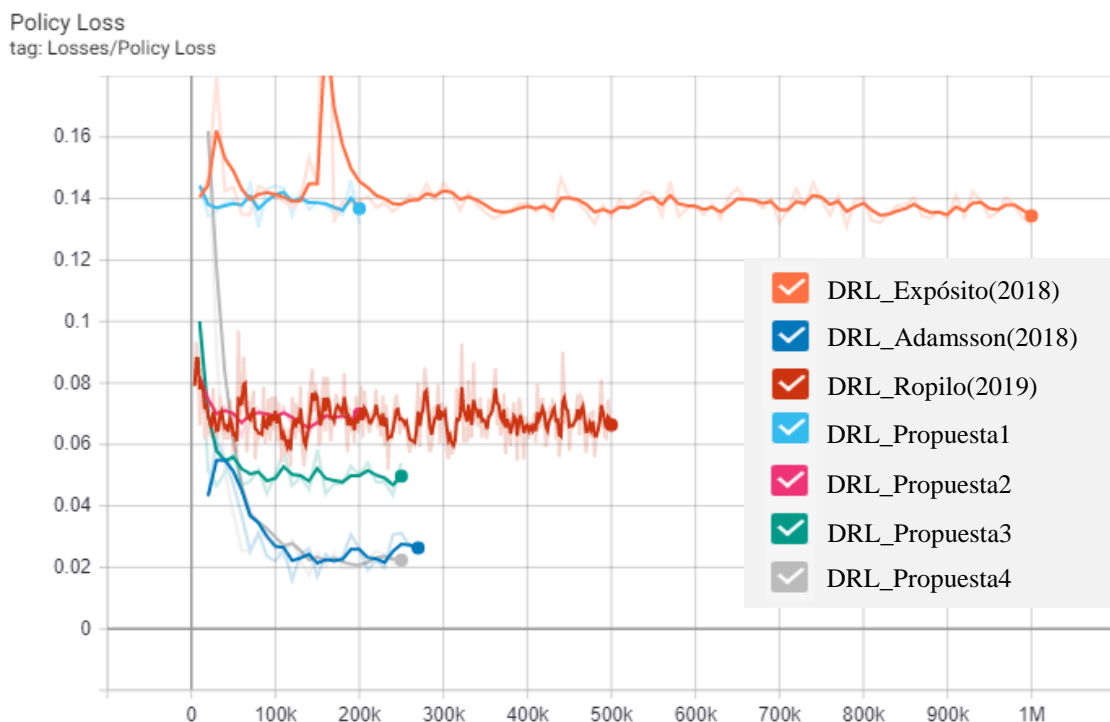


Figura N°124: Resultados de los modelos de DRL desarrollados – PolicyLoss.

Fuente: Elaboración propia.

De la Figura N°124, se puede observar que la tendencia del *PolicyLoss* para cada modelo ha sido diferente. También es preciso mencionar que en esta gráfica en particular se aplica la Ecuación (47). Para un entrenamiento exitoso, el *PolicyLoss* debería de decrecer y tener un comportamiento casi constante dado que esto implica que los conocimientos del agente con respecto a determinar la acción de desplazamiento ya se han consolidado. En primer lugar, se puede apreciar que en la gráfica del *PolicyLoss* el valor final para cada modelo ha ido en un intervalo de 0.02 hasta 0.14 lo cual es un aspecto positivo para todos los modelos dado que la idea es tener un muy pequeño para el *PolicyLoss*. Los modelos que tienen un mayor valor para el *PolicyLoss* son el modelo de Expósito (2018) y la propuesta 1 ambos con un valor casi de 0.14. El modelo de DRL de Adamsson (2018) tuvo un *PolicyLoss* inicial de un poco más de 0.04 y llegó a un valor final un poco superior de 0.02 lo cual el segundo modelo de DRL con mejor valor para el *PolicyLoss*. El modelo de Ropilo (2019) ha tenido una tendencia muy inestable dado que su valor de *PolicyLoss* ha estado en constantes subidas y bajadas. No obstante, se mantenido dentro del rango de 0.06 a 0.08. La propuesta 2 ha sido muy estable dado que se ha mantenido en un valor aproximado de 0.07. La propuesta 3 y la propuesta 4 son los modelos que más se han ajustado a la teoría y los que han tenido mejor desenvolvimiento. La propuesta 3 tiene una tendencia a la baja y mostró un valor casi constante cuando llegó al valor de 0.05 antes de llegar a las 100 000 iteraciones. La propuesta 4 ha sido el que obtuvo el

menor *PolicyLoss* con un valor un poco mayor a 0.02 de igual modo tendiendo a ser constante antes de llegar a las 100 000 iteraciones.

En relación con la gráfica de *ValueLoss* para los 7 agentes entrenados, esta se puede apreciar en la Figura N°125.

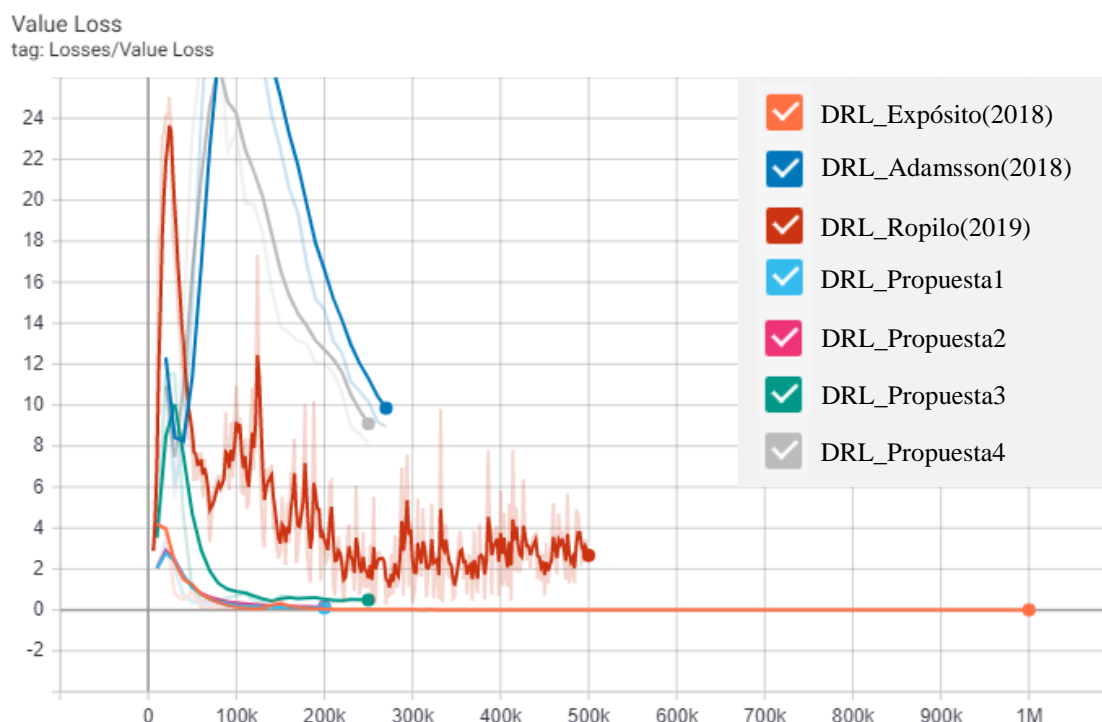


Figura N°125: Resultados de los modelos de DRL desarrollados – ValueLoss.
Fuente: Elaboración propia.

En la Figura N°125 se puede observar las tendencias del *ValueLoss* para cada modelo y también se aplicó para su cálculo la Ecuación (46). Como se mencionó en el capítulo anterior, en un entrenamiento exitoso, el *ValueLoss* debe tener dos comportamientos: debe de ir al alza y luego disminuir formando una especie de montaña. Esto es debido a que la red neuronal al inicio está empezando con el entorno y espera recibir una recompensa por sus salidas (Adamsson, 2018). No obstante, esta recompensa esperada se contrapone con la recompensa real que recibe al desplazarse el agente y dado que el aprendizaje se da a través de la experiencia, la diferencia puede ir incrementándose por lo que la tendencia va a tender a la alza (Adamsson, 2018). Conforme la red va aprendiendo con el pasar de las iteraciones, va aproximándose cada vez más a la recompensa esperada por lo que la diferencia tiene que empezar a disminuir (Adamsson, 2018). Si se visualiza la gráfica de *ValueLoss*, se puede apreciar que los 7 modelos tienen el comportamiento descrito anteriormente. El modelo de Expósito (2018) tiene un *ValueLoss* máximo que llega hasta un aproximado de 4 en un poco

menos de 50 000 iteraciones luego empieza a decrecer hasta llegar a un *ValueLoss* de 0 aproximadamente y tiende a ser constante. Este comportamiento es debido el modelo ya no tiene suficiente poder para realizar una predicción adecuada y determinar una recompensa esperada. Este comportamiento se puede complementar con la gráfica de *Cumulative Reward* dado que en aquella gráfica el modelo de Expósito (2018) tiene una recompensa que se vuelve constante en vez de ir en aumento. El modelo de Adamsson (2018) ha sido uno de los modelos que más ha tardado en detectar la tarea a realizar dado que el error máximo ha sido mayor 24 y superando las 100 000 iteraciones luego sufrió el decremento hasta llegar a 10. El aprendizaje de Ropilo (2019) fue mucho mejor y más rápido dado que llegó a un *ValueLoss* casi de 24 en menos de 50 000 iteraciones y luego disminuyó hasta llegar a un *ValueLoss* un poco mayor a 2. La propuesta 1 y propuesta 2 tuvieron un comportamiento similar. Ambos tuvieron un *ValueLoss* máximo de un valor casi de 2.5 y luego decayeron hasta un valor un poco superior a cero, pero menor a 1. La propuesta 3 ha tenido un aprendizaje muy adecuado dado que llegó a un valor máximo de 10 en menos de 50 000 iteraciones y luego descendió a un valor casi constante un poco mayor a cero estando cerca a las 100 000 iteraciones. La propuesta 4 tuvo un comportamiento similar al modelo de Adamsson (2018) dado que el modelo ha tenido un aprendizaje más lento teniendo un pico superior a 24 y llegando a un valor mínimo de 8 luego de 250 000 iteraciones.

Referente a los resultados del gráfico de *Entropy*, estos se presentan en la Figura N°126 y para el cálculo de este valor se utilizó la Ecuación (49).

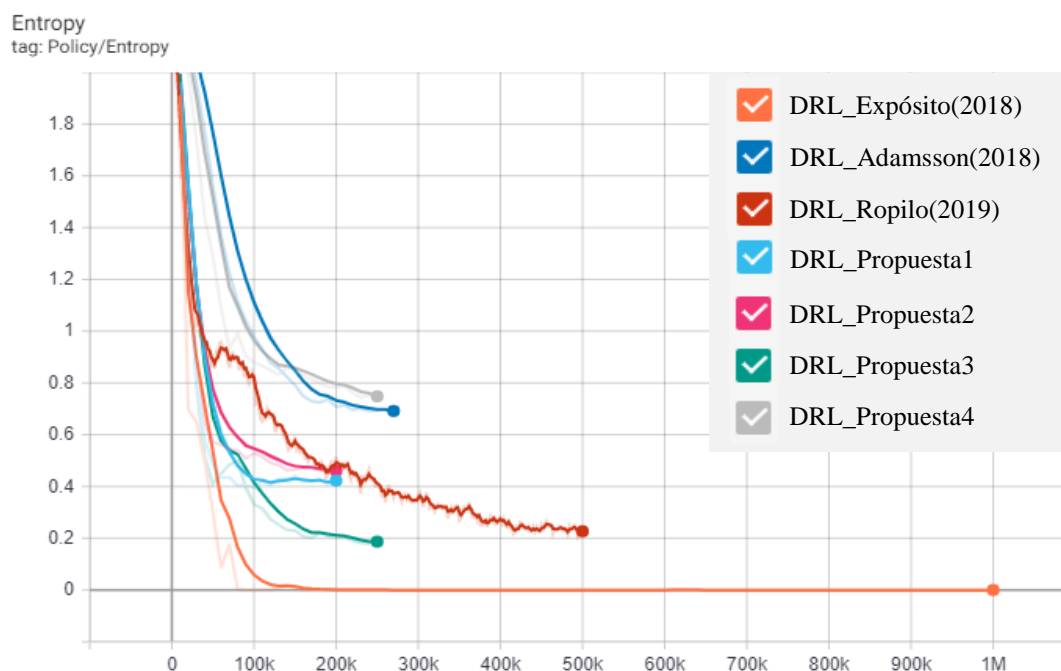


Figura N°126: Resultados de los modelos de DRL desarrollados – Entropía.

Fuente: Elaboración propia.

La gráfica de *Entropy* representa la aleatoriedad de las decisiones que toma el agente para realizar sus acciones. En un entrenamiento exitoso, la gráfica de la entropía debe de decrecer a lo largo del entrenamiento. En los resultados de los 7 modelos de DRL entrenados, se puede apreciar que los comportamientos de las gráficas decrecen por lo que en general los agentes han aprendido a realizar acciones fijas para realizar sus movimientos. En el caso del modelo de Expósito (2018), se vuelve a apreciar que la gráfica tiende a llegar a cero y ser constante cuando llega a las 100 000 iteraciones. Esto es debido a que el agente no ha llegado a tener una interacción más profunda con el ambiente de entrenamiento, sino que se ha detenido en una determinada recompensa volviéndose casi constante como se visualizó en la gráfica de *Cumulative Reward*. Con este comportamiento, el agente ya fija en las 100 000 iteraciones las acciones que debe de realizar para realizar la tarea encomendada pero el modelo no es escalable. El modelo de Adamsson (2018) tuvo un decremento hasta llegar a un valor un poco mayor de 0.6 al llegar a una cantidad de iteraciones mayor a 250 000. El modelo de Ropilo (2019) de igual manera tuvo un decremento a lo largo del entrenamiento, pero menos inestable que el resto de los modelos dado que la entropía como se visualiza posee subidas y bajadas. Al final del entrenamiento llegó a un valor un poco mayor a 0.2 al llegar a las 500 000 iteraciones. La propuesta 1 y la propuesta 2 tuvieron un comportamiento similar. Ambos al final del entrenamiento tuvieron un valor de entropía un poco mayor a 0.4 y llegando a las 200 000

iteraciones. El modelo de la propuesta 3 tuvo un comportamiento más adecuado dado que llegó a un valor de entropía más aceptable de un poco menor a 0.2 luego de las 200 000 iteraciones. La propuesta 4 es el que ha tenido un mayor valor para la entropía llegando casi a 0.8 luego de 200 000 iteraciones.

En relación con la gráfica de *Extrinsic Reward*, esta se puede apreciar en la Figura N°127.

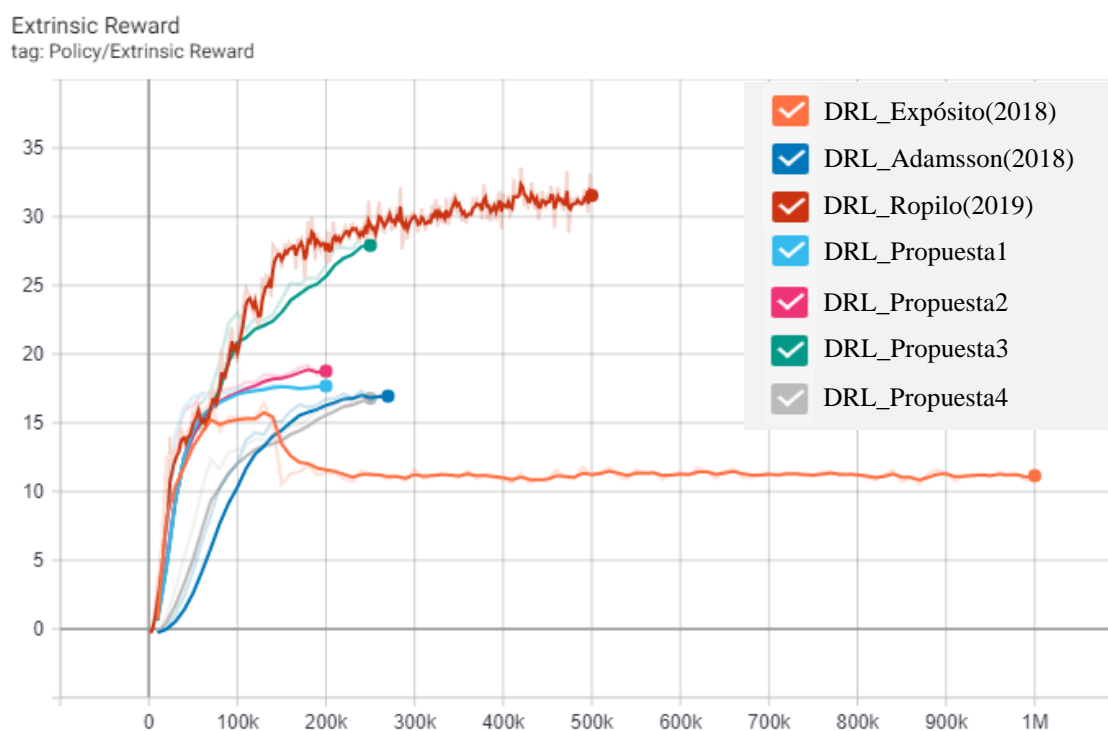


Figura N°127: Resultados de los modelos de DRL desarrollados – Extrinsic Reward.

Fuente: Elaboración propia.

Como se detalló en el anterior capítulo, la gráfica de *Extrinsic Reward* es similar a la gráfica de *Cumulative Reward* dado que esta gráfica presenta las recompensas que son otorgadas al agente y que son generadas a través del ambiente de entrenamiento creado. En el caso del modelo de Expósito (2018) no tuvo una interacción tan alta con el ambiente de entrenamiento dado que solo llegó a un *Extrinsic Reward* máximo de un poco más de 15 y luego disminuyó luego de 200 000 iteraciones y se mantuvo casi constante. Con respecto al modelo de Adamsson (2018) se puede observar que llega a un valor de *Extrinsic Reward* aproximado de 17 en una cantidad de iteraciones menor a 300 000. Para el modelo de Ropilo (2019) el *Extrinsic Reward* tiende a ir en aumento con altas y bajas en el camino haciendo que el aprendizaje sea poco estable llegando a un valor de un poco más de 30 luego de 500 000 iteraciones. La propuesta 1 y propuesta 2 son muy semejantes llegando a un valor de *Extrinsic Reward* menor a 20 luego de 200 000 iteraciones. La propuesta 3 ha tenido un entrenamiento

muy estable llegando un valor de *Extrinsic Reward* casi de 30 en un poco más de 250 000 iteraciones. La propuesta 4 también ha tenido un entrenamiento estable pero no ha llegado a un *Extrinsic Reward* elevado dado que su valor fue de un poco más de 15 en 250 000 iteraciones aproximadamente.

Acerca de la gráfica de *Extrinsic Value Estimate* para cada modelo de DRL entrenado, esta se presenta en la Figura N°128.

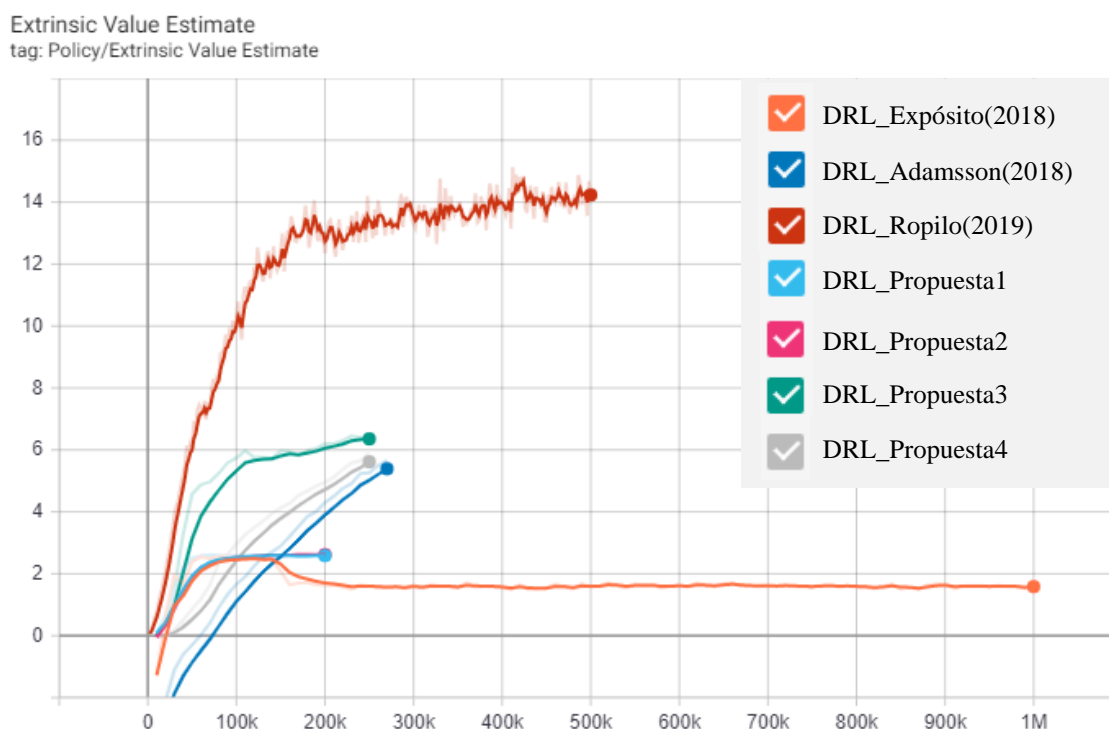


Figura N°128: Resultados de los modelos de DRL desarrollados – Extrinsic Value Estimate. Fuente: Elaboración propia.

Para un entrenamiento exitoso del *Extrinsic Value Estimate* este valor debe de incrementar con el pasar del entrenamiento siempre y cuando el *Cumulative Reward* también incremente. Con respecto al modelo de Expósito (2018), el valor del *Extrinsic Value Estimate* se eleva hasta ser un poco mayor a 2 pero luego tiende a ser constante luego de 100 000 iteraciones. Este comportamiento se complementa con su gráfica de *Cumulative Reward* dado que la recompensa acumulada también se mantiene constante llegando al mismo número de iteraciones. Con respecto al modelo de Adamsson (2018), la gráfica del *Extrinsic Value Estimate* tiende a ir en aumento hasta llegar a un valor un poco inferior a 6 luego de realizar más de 200 000 iteraciones. El modelo de Ropilo (2019) tiene el más alto valor para el *Extrinsic Value Estimate* llegando a 14 luego de 500 000 iteraciones. No obstante, se puede apreciar que el incremento de la gráfica ha sido inestable porque ha tenido altas y bajas cuando ha llegado

a un valor de 12. La propuesta 1 y propuesta 2 son muy semejantes llegando a un valor un poco mayor a 2 luego de realizar 200 000 iteraciones. La propuesta 3 ha tenido un incremento estable hasta llegar a un valor un poco mayor a 6 luego de un poco más de 200 000 iteraciones. Es preciso mencionar que este modelo tiene el segundo valor de *Extrinsic Value Estimate* más alto. La propuesta 4 tuvo un incremento de igual manera estable hasta llegar a un valor menor a 6 luego de 200 000 iteraciones.

Finalmente, la gráfica del *Learning Rate* para cada uno de los 7 modelos de DRL entrenados se presenta en la Figura N°129.

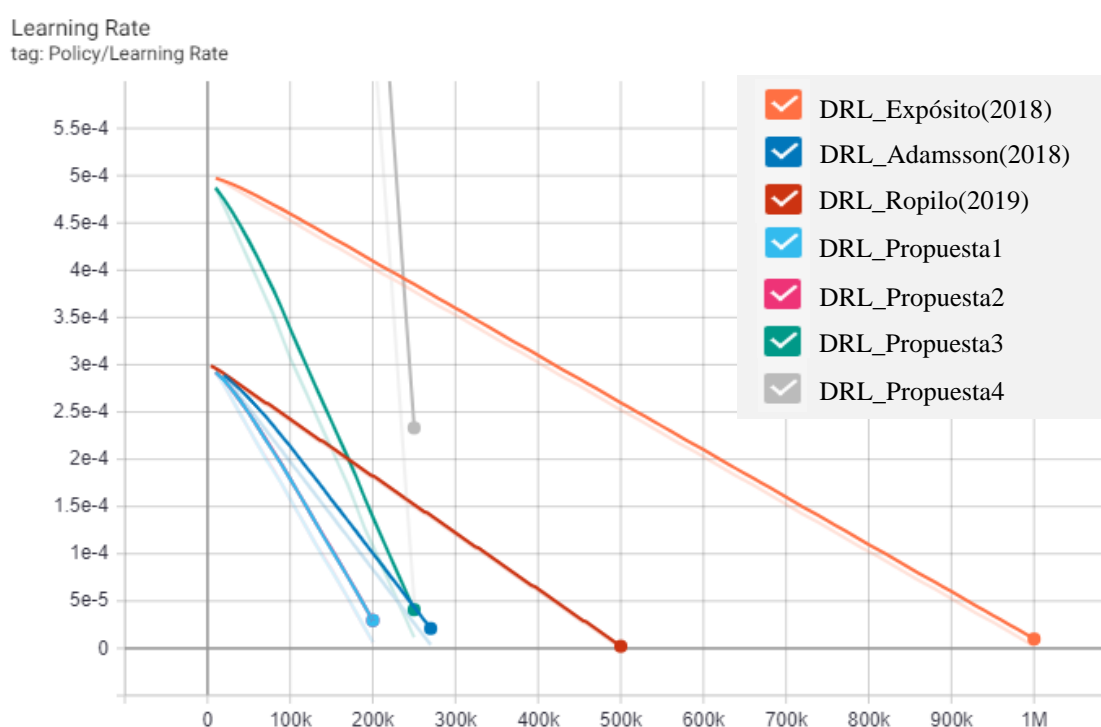


Figura N°129: Resultados de los modelos de DRL desarrollados – Learning Rate.

Fuente: Elaboración propia.

La gráfica del *Learning Rate* en un entrenamiento exitoso debe de ir decreciendo con el avance del aprendizaje del agente y para la actualización de este *Learning Rate* para cada modelo se ha aplicado la Ecuación (64).

El modelo de Expósito (2018) tiende a decrecer hasta llegar casi a cero luego de 1 millón de iteraciones. El modelo de Adamsson (2018) de igual manera tiende a decrementarse hasta llegar casi a cero luego de entrenar hasta casi llegar a las 300 000 iteraciones. El modelo de Ropilo (2019) ha tenido un valor final muy cercano a cero luego de 500 000 iteraciones. El modelo de la propuesta 1 y propuesta 2 tienen casi el mismo comportamiento con respecto a

su tasa de aprendizaje llegando a un valor inferior a $5e-5$ luego de realizar 200 000 iteraciones. La propuesta 3 ha tenido una disminución de la tasa de aprendizaje hasta llegar casi a $5e-5$ luego de realizar un poco más de 250 000 iteraciones. La propuesta 4 tuvo de igual manera un decremento de la tasa de aprendizaje llegando a un valor menor de $2.5e-4$ luego de haber realizado 250 000 iteraciones.

Habiendo desarrollado un análisis del comportamiento de cada modelo de DRL por cada gráfica obtenida, en la Tabla 36 se presenta un resumen de las gráficas descritas anteriormente con el objetivo de tener una visión general del entrenamiento de cada agente de tal forma de poder visualizar los resultados finales obtenidos.

Los resultados de las gráficas completas con su respectiva leyenda de todos los modelos de DRL entrenados para la implementación del agente inteligente en el videojuego desarrollado se pueden encontrar en el repositorio creado en GitHub a través del siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

Tabla 36: Resultados generales del entrenamiento de los modelos de Deep Reinforcement Learning

Modelo	Cumulative Reward	Episode Length	Is Training	Policy Loss	Value Loss	Entropy	Extrinsic Reward	Extrinsic Value Estimate	Learning Rate	Step	Time (h)
Adamsson (2018)	17.01	18	1	0.02487	8.949	0.6832	17.01	5.666	3.9357e-6	270K	3h 29m 5s
Ropilo (2019)	31.13	32.1	1	0.06363	2.244	0.2178	31.13	14.18	3.499e-7	500K	6h 54m 17s
Expósito (2018)	11.28	12.3	1	0.1317	7.3078e-3	1.3213e-5	11.28	1.616	2.2117e-6	1M	11h 52m 9s
Propuesta 1	17.75	18.75	1	0.1318	0.09728	0.4305	17.75	2.585	6.6706e-6	200K	2h 28m 29s
Propuesta 2	18.84	19.82	1	0.07273	0.1436	0.456	18.84	2.626	6.5488e-6	200K	2h 13m 54s
Propuesta 3	28.05	29.02	1	0.05389	0.475	0.1862	28.05	6.39	1.1743e-5	250K	3h 29m 40s
Propuesta 4	17.14	18.14	1	0.02179	8.094	0.7303	17.14	5.904	4.866e-5	250K	3h 28m 36s

Fuente: Elaboración Propia.

En la Tabla 36 se puede visualizar en la columna de *Cumulative Reward* que el mayor valor es de 31.13 del modelo de Ropilo (2019). No obstante, para que aquel llegue a ese valor tuvo que demorarse un total de 6h 54m 17s como se muestra en la última columna y realizando un total de 500 000 iteraciones. Por otro lado, el segundo modelo con un *Cumulative Reward* elevado es la propuesta 3 con un valor de 28.05. Además, visualizando el tiempo total de entrenamiento este modelo se puede visualizar que fue de 3h 29m 40s es decir que fue el doble de eficiente que el primero dado que en la menor cantidad de tiempo el agente pudo comprender la actividad que le fue encomendada. Con respecto al *Value Loss*, *Policy Loss* y *Entropy*, los resultados fueron mejores en la propuesta 3 que en el modelo de Ropilo (2019) dado que los valores fueron menores y muy cercanos a cero. Los resultados presentados de la Tabla 36 sumados con la interpretación individual de cada gráfica para cada modelo, se puede llegar a la conclusión que la propuesta 3 ha sido el modelo que ha presentado una mayor estabilidad durante todo su entrenamiento y a la vez ha sido eficiente dado que en un periodo de 3 horas y media ha obtenido un *Cumulative Reward* elevado y valores de *Value Loss*, *Policy Loss* y *Entropy* muy pequeños en comparación con modelo de Ropilo (2019). Con todos los resultados presentados, se ha determinado que el mejor modelo de DRL fue la propuesta 3.

Continuando con el análisis de la Tabla 36, se puede apreciar que el tercer modelo con un *Cumulative Reward* considerable fue la propuesta 2 con un valor de 18.84 en un tiempo de 2h 13m 54s realizando en total 200 000 iteraciones. No obstante, este modelo sigue inferior a la propuesta 3 dado que el valor de su *Cumulative Reward* es inferior al del mejor modelo.

También se puede observar casi un triple empate entre los modelos de DRL de Adamsson (2018), la propuesta 1 y la propuesta 4 teniendo un *Cumulative Reward* de 17.01, 17.75 y 17.14 respectivamente. No obstante, en base a los resultados obtenidos se puede apreciar que la propuesta 1 tuvo un entrenamiento más eficiente dado que en primer lugar de los 3 modelos mencionados, este es el que obtuvo un *Cumulative Reward* mayor y porque para lograr este valor lo hizo en un tiempo mucho menor que fue de 2h 28m 29s realizando 200 000 iteraciones.

Finalmente, se puede apreciar que el modelo de Expósito (2018) fue modelo que obtuvo el menor *Cumulative Reward* de los 7 entrenados llegando a un valor de 11.28 a pesar de haber realizado un mayor número de iteraciones que el resto llegando a 1 millón de iteraciones demorándose un total de 11h 52m 9s. Esto demuestra que este modelo tuvo un entrenamiento muy poco eficiente comparado con los anteriores.

5.2.2 Resultados del mejor modelo de DRL seleccionado

Luego de haber determinado que el mejor modelo de DRL fue la propuesta 3, en este apartado se detallarán los resultados obtenidos de este modelo en particular.

5.2.2.1 Resultados del entrenamiento del agente por episodio

Durante el entrenamiento de cada modelo de DRL, a través del Anaconda Prompt se van imprimiendo los resultados que se van obteniendo por cada episodio del entrenamiento hasta llegar al total de iteraciones establecido. En la Tabla 37 se presentan los resultados por cada episodio de entrenamiento del modelo de DRL de la propuesta 3.

Como se puede observar en la Tabla 37, el aprendizaje del agente ha ido de menos a más dado que el *Cumulative Reward* en cada episodio fue incrementándose. También se aprecia que en los 3 primeros episodios es en donde el agente más ha tenido una interacción con el ambiente de entrenamiento dado que ha tenido un mayor incremento de recompensa. También se observa que cuando ha llegado el modelo a un *Cumulative Reward* de 22, este ha empezado a tener incrementos mínimos hasta llegar a su valor final de 28.04.

Con respecto al *Episode Length* se observa que en cada episodio tiende a ir en aumento, este es un comportamiento natural y acorde al *Cumulative Reward* dado que si este último va incrementándose como se mencionó anteriormente, la cantidad promedio de acciones que realiza en agente también se incrementarán en cada episodio. Se observa que el valor final del *Episode Length* fue de 29.02.

En referencia al *Is Training* se aprecia que el modelo no ha sido interrumpido durante el entrenamiento, sino que se ha mantenido durante ejecutando durante todo el proceso de aprendizaje.

En relación con el *Policy Loss* este se ha mantenido en valores que fluctúan entre 0.05 0.04 lo cual es un buen indicador dado que el *Policy Loss* debe tener un ir decreciendo hasta tener un valor cercano a cero. Luego se 250 000 iteraciones, el *Policy Loss* tuvo un valor de 0.053.

Tabla 37: Resultados del mejor modelo de DRL por episodio de entrenamiento

Cumulative Reward	Episode Length	Is Training	Policy Loss	Value Loss	Entropy	Extrinsic Reward	Extrinsic Value Estimate	Learning Rate	Step
0.608661623	1.609285342	1	0.100110285	3.532767	1.9937043	0.608661623	0.01090754	0.000487193	10000
6.785588752	7.805628848	1	0.051598594	11.458759	1.3115714	6.785588752	0.5807493	0.000467957	20000
14.05457464	15.03370787	1	0.046574447	11.536068	0.7880183	14.05457464	1.5703981	0.000448709	30000
16.09782609	17.11231884	1	0.0508741	4.674582	0.5136688	16.09782609	3.2704577	0.000429448	40000
16.25912409	17.26459854	1	0.057447135	0.9238353	0.40954936	16.25912409	4.572818	0.000410202	50000
16.44833948	17.45202952	1	0.046715245	0.47691527	0.4503605	16.44833948	4.869234	0.00039096	60000
17.44163424	18.44357977	1	0.04802059	0.4377381	0.48503885	17.44163424	4.9669476	0.000371699	70000
19.42398287	20.4496788	1	0.05192609	0.42312804	0.49793217	19.42398287	5.239552	0.000352439	80000
22.24757282	23.24514563	1	0.044041798	0.61240983	0.39481407	22.24757282	5.580694	0.000329967	90000
23.0025	23.9975	1	0.050272405	0.7030416	0.3308417	23.0025	5.748666	0.000307454	100000
21.7719715	22.72921615	1	0.058203217	0.7748745	0.31438816	21.7719715	5.9708743	0.000288171	110000
22.69059406	23.75247525	1	0.046522427	0.4317204	0.27444622	22.69059406	5.7590823	0.000268927	120000
22.49388753	23.44254279	1	0.049307946	0.3565255	0.2524538	22.49388753	5.755942	0.000249616	130000
22.88059701	23.90298507	1	0.045519594	0.29342622	0.2318556	22.88059701	5.725433	0.000230353	140000
24.00519481	24.9974026	1	0.05830951	0.69147223	0.22912177	24.00519481	5.961762	0.000211077	150000
25.16847826	26.16847826	1	0.044015337	0.69843477	0.2065869	25.16847826	5.946984	0.000191814	160000
25.07588076	26.05691057	1	0.047101367	0.51376873	0.20221138	25.07588076	5.8141565	0.000172523	170000

25.54269972	26.58953168	1	0.04721936	0.62661964	0.21944414	25.54269972	5.9770064	0.000150003	180000
25.5	26.46703297	1	0.05292231	0.6212842	0.2074985	25.5	6.042271	0.000127502	190000
26.47293447	27.46438746	1	0.049930792	0.42727575	0.20592439	26.47293447	6.206593	0.000108189	200000
27.70535714	28.74404762	1	0.053981047	0.4005064	0.20381008	27.70535714	6.204232	8.89e-05	210000
27.74480712	28.7388724	1	0.047785565	0.4294831	0.18986669	27.74480712	6.300393	6.96e-05	220000
27.66172107	28.67062315	1	0.047763553	0.57462937	0.18088545	27.66172107	6.4445286	5.04e-05	230000
28.65337423	29.68711656	1	0.043565795	0.48917314	0.1782378	28.65337423	6.395732	3.10e-05	240000
28.04504505	29.02102102	1	0.053889487	0.47497678	0.1862279	28.04504505	6.390074	1.17e-05	250000

Fuente: Elaboración Propia.

Con respecto al *Value Loss* este debe de incrementarse y luego empezar a disminuir conforme avanza el entrenamiento. De la Tabla 37 presentada se puede apreciar que el *Value Loss* alcanza un valor máximo igual a 11.53 al llegar a las 30 000 iteraciones. Luego empieza a decrecer. También se observa que en las siguientes 10 000 iteraciones tiene una caída bastante significativa pasando de 11.53 a 4.67. Luego su decrecimiento empieza a ser más lento hasta llegar a un valor final de 0.47 después de haber entrenado durante 250 000 iteraciones.

En referencia al valor de *Entropy* que representa la aleatoriedad de las decisiones del agente, este valor tiene que disminuir conforme avanza el entrenamiento. Para el caso de este mejor modelo, se puede apreciar que ha empezado en un valor muy pequeño que llega casi a 2 lo cual es un buen indicador de que el agente está ejecutando las acciones adecuadas para poder obtener una mayor recompensa. Luego este valor empieza a decrecer hasta llegar a un valor mínimo de 0.18 luego de 250 000 iteraciones.

En relación con el valor del *Extrinsic Reward* su comportamiento es muy similar al del *Cumulative Reward* dado que conforme avance el entrenamiento su valor tiene que ir incrementándose. Se puede apreciar en la Tabla 37 que su valor empieza siendo positivo y va incrementándose hasta llegar a un valor final de 28.04 luego de las 250 000 iteraciones.

Sobre el valor de *Extrinsic Value Estimate* debe ir en aumento su valor si es que el valor del *Cumulative Reward* también lo hace. Para esta ocasión el valor del *Extrinsic Value Estimate* ha ido en aumento y se ha mantenido en un intervalo de 0 a 6.5 aproximadamente. Se puede apreciar en la Tabla 37 que el valor se ha mantenido en constante crecimiento para cada episodio llegando a un valor final de 6.39 luego de completar la totalidad de iteraciones establecidas.

De igual forma, el *Learning Rate* ha disminuido con el avance de cada episodio lo cual es un buen indicador de que el agente está aprendiendo a ejecutar la acción de desplazamiento encomendada. El valor final para el *Learning Rate* fue de $1.17e-05$ luego de las 250 000 iteraciones.

En la última columna se presente el *Step* o la cantidad de iteraciones que se han realizado para cada episodio. Como se visualiza en la Tabla 37, el *Step* va de 10 000 en 10 000. Esto es debido a que se ha colocado en el hiperparámetro de *Summary_freq* este valor por lo que cada episodio del entrenamiento fue avanzado de 10 000 en 10 000 hasta llegar al valor establecido en el hiperparámetro de *Max_steps* que fue de 250 000 iteraciones. Finalmente, se aprecia que

en total el número de episodios que tuvo el entrenamiento de este modelo de DRL fue de 25 episodios en total.

La visualización del resto de los resultados de los otros modelos de DRL por impresión en cada episodio se pueden encontrar en el siguiente enlace que redirecciona al repositorio de GitHub creado <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

5.2.2.2 Tabla de entradas y salidas de la red

La arquitectura de la red neuronal definida tiene 4 neuronas en la capa de entrada. La primera neurona corresponde a la posición inicial del agente inteligente con respecto al eje x (Input 1), la segunda neurona de la capa de entrada corresponde a la posición del agente con respecto al eje z (Input 2). Por otro lado, la tercera y cuarta neurona de la capa de entrada corresponde a la posición del modelo en 3D que es el objetivo con respecto al eje x y con respecto al eje z respectivamente (Input 3 e Input 4). La capa de salida de la red tiene 2 neuronas. La primera neurona corresponde al desplazamiento con respecto al eje x del agente (Output 1) y la segunda neurona corresponde al desplazamiento con respecto al eje z del agente (Output 2). Para poder realizar la acción de desplazarse, las salidas de la red son transformadas. En el caso que salga 2, el valor de la salida se almacena en una variable y se transforma a 1. En el caso que sea 0 o 1, el valor final es el negativo de estos de modo que el agente pueda desplazarse de forma equitativa siendo 1 unidad adicional a la derecha, 1 unidad adicional a la izquierda o 0 para no desplazarse. La misma idea corresponde con el desplazamiento al eje vertical. En la Tabla 38, se presenta las entradas a la red neuronal a inicios del entrenamiento. La visualización de la tabla completa se puede visualizar en el repositorio de GitHub creado para visualizar todos los resultados del entrenamiento a través del siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

Tabla 38: Entradas y salidas iniciales de la red neuronal durante el entrenamiento

Input 1	Input 2	Input 3	Input 4	Output 1	Output 2
0	0	0.819116	-0.5736279	2	2
0.04277864	0.04277864	0.819116	-0.5736279	2	2
0.0850914	0.0850914	0.819116	-0.5736279	2	2
0.1264973	0.1264973	0.819116	-0.5736279	2	2
0.1666024	0.1666024	0.819116	-0.5736279	2	2
0.2050758	0.2050758	0.819116	-0.5736279	2	2
0.2416597	0.2416597	0.819116	-0.5736279	2	2
0.2761724	0.2761724	0.819116	-0.5736279	2	2
0.3085047	0.3085047	0.819116	-0.5736279	2	2
0.3386129	0.3386129	0.819116	-0.5736279	2	2
0.3665083	0.3665083	0.819116	-0.5736279	0	2
0	0	0.728788	-0.6847394	0	0

Fuente: Elaboración Propia.

5.2.2.3 Matriz de pesos

La mejor arquitectura de red tiene 512 capas ocultas con 3 neuronas en cada una de las capas. Dado que la cantidad de capas ocultas es muy elevada haciendo que la red tenga una alta profundidad, en la Tabla 39 se presenta los pesos iniciales de las 6 primeras capas ocultas y en la Tabla 40 se presentan los pesos iniciales para la capa de salida de la red que está conformada por dos neuronas como se había mencionado anteriormente.

Para la actualización de los pesos de la red, se calculó el error por cada neurona de la capa de salida que corresponde a la Ecuación (22) para finalmente realizar la sumatoria y obtener el error total. Para aplicar el pase en *Backward* y actualizar los pesos de la red, se aplicó la regla de la cadena que corresponde a la Ecuación (13), Ecuación (14), Ecuación (15), Ecuación (16), Ecuación (17), Ecuación (18), Ecuación (19), Ecuación (20), Ecuación (21), Ecuación (23), Ecuación (24), Ecuación (25), Ecuación (26), Ecuación (27), Ecuación (28), Ecuación (29), Ecuación (30), Ecuación (31), Ecuación (32) y termina con la Ecuación (33).

Tabla 39: Matriz de pesos iniciales antes del entrenamiento

N° de Capa Oculta	Bias	Input 1	Input 2	Input 3	Input 4
1	0.5	0.00802312	0.00768366	0.01628495	0.01560155
	0.5	0.01546976	0.02323052	0.00533405	0.01239621
	0.5	0.02612852	0.00899565	0.02486474	0.00333628
2	0.5	0.01225784	0.00112126	0.01285166	0.00877245
	0.5	0.00391316	0.01863172	0.01211322	0.00890255
	0.5	0.00999541	0.00027994	0.03406748	0.02921366
3	0.5	0.02063347	0.01401791	0.02108494	0.00760104
	0.5	0.00497174	0.02732993	0.02044281	0.01158945
	0.5	0.00712158	0.04064227	0.01913105	0.00731577
4	0.5	0.04346265	0.02126769	0.01527735	0.01224611
	0.5	0.00693242	0.01217269	0.00703456	0.01697678
	0.5	0.00523174	0.00609291	0.00178075	0.00185855
5	0.5	0.01785531	0.01785523	0.02136331	0.00109821
	0.5	0.02942896	0.01700293	0.00855352	0.01009646
	0.5	0.00545525	0.01833517	0.00568144	0.02519705
6	0.5	0.00473731	0.01221312	0.02267666	0.02641983
	0.5	0.01103476	0.01207715	0.01509365	0.01940042
	0.5	0.02583437	0.01752489	0.01256022	0.03033209

Fuente: Elaboración Propia.

Tabla 40: Matriz de pesos de la capa de salida antes del entrenamiento

Capa de salida	Bias	W1	W2	W3
Output 1	0.5	0.01162972	0.02760642	0.01228843
Output 2	0.5	0.02681382	0.02155051	0.01599036

Fuente: Elaboración Propia.

Al finalizar el entrenamiento, se obtuvieron los pesos finales de la red aplicando las mismas ecuaciones. En la Tabla 41, se presentan los pesos de las primeras 6 capas ocultas finales al terminar con el entrenamiento y en la Tabla 42 se presentan los pesos finales para la capa de salida. La visualización de los pesos completos para las 512 capas ocultas de la red neuronal se pueden encontrar en el repositorio de GitHub a través del siguiente enlace <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

Tabla 41: Matriz de pesos actualizados al final el entrenamiento

N° de Capa Oculta	Bias	Input 1	Input 2	Input 3	Input 4
1	2.09444109	1.53662665	2.00348832	-2.00348832	-2.36690678
	1.34077224	2.32520513	0.35703607	1.38828708	2.45398972
	2.21235805	-3.88901681	2.30657402	-2.27836035	1.40354754
2	4.96323733	1.00254351	1.88333932	2.84125842	3.40010831
	6.81058329	-3.95754445	1.79519057	2.88193027	-1.53433913
	2.07487661	1.57544315	0.58554278	2.33548954	1.72865242
3	1.15577327	1.13218501	2.73027106	1.43349439	0.47466942
	1.99221838	2.05809530	1.06077697	1.79811791	1.22751454
	2.29329546	2.04647128	2.77778492	2.51538511	2.89755766
4	1.58464089	-3.22866589	1.51014002	0.81755994	1.69462059
	2.23737094	-2.19128374	0.31784957	1.85287558	2.87386112
	2.58296654	2.71405671	-1.11371148	1.30616259	1.41655272
5	1.09671028	1.24045017	1.57962553	0.82904648	1.79079436
	7.95494858	0.34993569	2.40685418	1.06183302	2.49581188
	2.06069928	1.63352665	1.01578367	2.53071561	1.46180373
6	8.10346287	3.10206469	2.35031638	-1.85233788	0.40502369
	2.02096701	1.81381349	0.36158888	1.60620492	2.91445512
	3.05021461	0.85550489	2.34872866	1.38516687	-1.53329130

Fuente: Elaboración Propia.

Tabla 42: Matriz de pesos de la capa de salida después del entrenamiento

Capa de salida	Bias	W1	W2	W3
Output 1	6.58480637	1.73646971	2.92724669	1.78901781
Output 2	4.39908467	1.51526742	1.85096916	1.54400580

Fuente: Elaboración Propia.

5.3 Medición de Resultados – Entrenamiento de los estudiantes

En este apartado se presentarán los resultados obtenidos luego del entrenamiento de los estudiantes de primaria. Para cada grado se presentarán en primer los resultados cuantitativos descriptivos. Luego se presentarán los resultados cuantitativos inferenciales en donde se aplicó la prueba T para muestras independientes de tal forma de poder determinar si hubo una mejora significativa en cada grado. Finalmente, se presentarán los resultados cualitativos de tal manera de poder verificar si el videojuego fue del agrado de los estudiantes de primaria.

5.3.1 Análisis cuantitativo descriptivo de las evaluaciones

En esta sección se presentarán los resultados del análisis cuantitativo descriptivo de para cada grado de primaria.

5.3.1.1 Análisis cuantitativo descriptivo – Primer Grado

En la Tabla 43, se puede observar el resumen estadístico con respecto a las notas del pretest tanto para el grupo de control como para el grupo experimental en donde se puede apreciar que el promedio de notas del grupo de control es mayor en 0.45 puntos con respecto al grupo experimental. También se puede apreciar que la desviación estándar del grupo de control es superior al del grupo experimental de lo cual se puede afirmar que las notas del grupo de control tienen una dispersión mayor al del grupo experimental. Por último, se puede apreciar que la mediana en ambos grupos tiene un valor de 13.

Tabla 43: Resumen estadístico del pretest para el primer grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	11	13.45	1.3684	13
Experimental	9	13.00	0.8660	13

Nota. N = Número total de la muestra.
Fuente: Elaboración Propia.

En la Tabla 44 se puede visualizar los resultados del posttest para ambos grupos. Se puede observar que en ambos grupos el promedio de notas se ha incrementado. Además, en ambos grupos la desviación estándar es muy semejante. También, se puede observar que la mediana en el grupo experimental fue mayor a la del grupo de control.

Tabla 44: Resumen estadístico del posttest para el primer grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	11	14.00	1.7888	14
Experimental	9	15.77	1.7159	16

Nota. N = Número total de la muestra.
Fuente: Elaboración Propia.

En la Tabla 45 se puede visualizar los resultados de los indicadores para el grupo de control en el pretest del primer grado. Se puede observar que los estudiantes poseen un mejor desenvolvimiento en el nivel literal. Una de las razones es porque los niños a esta edad temprana no tienen desarrollado un elevado de poder realizar un análisis inferencial o crítico sino más bien textual.

Tabla 45: Resultados de indicadores del pretest para el grupo de control – Primer grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	6	5	11
Ordena secuencia de hechos o acciones en el texto.	Literal	5	6	11
Analiza hechos presentes en el texto.	Inferencial	3	8	11
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	8	11
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	9	11

Fuente: Elaboración Propia.

Por otro lado, en la Tabla 46 se puede observar los resultados del pretest para el grupo experimental del primer grado. En este grupo en particular que posee un nivel de comprensión literal muy similar al del grupo de control, un bajo nivel en comprensión inferencial y solamente un estudiante logró responder adecuadamente una pregunta del nivel crítico. No obstante, en general se puede apreciar que poseen dificultades tanto a nivel inferencial como crítico y resultados muy similares con respecto al nivel de comprensión literal.

Tabla 46: Resultados de indicadores del pretest para el grupo experimental – Primer grado

GRUPO EXPERIMENTAL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	3	6	9
Ordena secuencia de hechos o acciones en el texto.	Literal	4	5	9
Analiza hechos presentes en el texto.	Inferencial	3	6	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	2	7	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	1	8	9

Fuente: Elaboración Propia.

Pasadas las 5 semanas de entrenamiento para el grupo de control, en la Tabla 47 se puede visualizar los resultados obtenidos. Se aprecia que ha habido una mejora en las respuestas brindadas por los estudiantes. El nivel de comprensión literal tuvo un incremento al igual que el nivel inferencial. Es importante mencionar que este grupo no tuvo contacto con el videojuego desarrollado, sino que estos estudiantes siguieron la metodología tradicional de las clases virtuales de lectura.

Tabla 47: Resultados de indicadores del postest para el grupo de control – Primer grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	8	3	11
Ordena secuencia de hechos o acciones en el texto.	Literal	7	4	11
Analiza hechos presentes en el texto.	Inferencial	4	7	11
Interpreta información expuesta en el texto	Inferencial	3	8	11

o reacciones en los personajes.				
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	9	11

Fuente: Elaboración Propia.

Para el caso del grupo experimental, en la Tabla 48 se presentan los resultados finales a través de la combinación de las clases de comprensión lectora y el uso del videojuego. Se aprecia que los estudiantes obtuvieron un incremento mucho mayor con respecto al grupo de control en el nivel literal de la comprensión lectora. También se vio un incremento en el nivel inferencial de los estudiantes y con respecto al nivel crítico, un estudiante adicional obtuvo una respuesta correcta para la pregunta redactada. En general se puede apreciar un correcto aprendizaje en el grupo experimental a través del uso del videojuego por parte de los estudiantes.

Tabla 48: Resultados de indicadores del postest para el grupo experimental – Primer grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	7	2	9
Ordena secuencia de hechos o acciones en el texto.	Literal	8	1	9
Analiza hechos presentes en el texto.	Inferencial	5	4	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	6	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	7	9

Fuente: Elaboración Propia.

5.3.1.2 Análisis cuantitativo descriptivo – Segundo Grado

Con respecto al segundo grado de primaria, en la Tabla 49 se puede visualizar los resultados del pretest tanto para el grupo de control como el grupo experimental. Los datos finales muestran unas notas bajas en ambos grupos, pero principalmente con respecto al grupo experimental. De igual manera, se presenta una mayor dispersión de datos del grupo experimental que el grupo de control. También se puede apreciar que la mediana del grupo de control es mayor en 1.5 con respecto al grupo experimental.

Tabla 49: Resumen estadístico del pretest para el segundo grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	8	12.00	1.6035	12.5
Experimental	10	10.90	1.9119	11

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 50 se puede contemplar los resultados del postest tanto para el grupo de control como para el grupo experimental. Se puede apreciar que la nota final del postest en el grupo experimental es mayor a la del grupo de control. También se puede apreciar que las notas del grupo de control son más dispersas a las del grupo experimental. Finalmente se puede visualizar que la mediana del grupo experimental es mayor en dos unidades a la del grupo de control.

Tabla 50: Resumen estadístico del postest para el segundo grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	8	13.25	1.4880	13
Experimental	10	14.70	1.2516	15

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 51 se puede observar los resultados de los indicadores del pretest para el grupo de control. Los estudiantes de este grado han demostrado un mejor desenvolvimiento respondiendo las preguntas del nivel literal de comprensión lectora. Con respecto al nivel inferencial, más de la mitad no lograron responder las preguntas correspondientes a este nivel de la comprensión lectora y solamente dos estudiantes pudieron responder las preguntas de nivel crítico.

Tabla 51: Resultados de indicadores del pretest para el grupo de control – Segundo grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	4	4	8
Ordena secuencia de hechos o acciones en el texto.	Literal	5	3	8
Analiza hechos presentes en el texto.	Inferencial	3	5	8
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	2	6	8
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	6	8

Fuente: Elaboración Propia.

En la Tabla 52 se puede visualizar resultados del pretest para el grupo experimental del segundo grado de primaria. Se puede apreciar que casi la mitad de los participantes respondieron adecuadamente las preguntas del nivel literal de la comprensión lectora. Por otro lado, se puede apreciar que fueron muy pocos los participantes que pudieron responder las preguntas de nivel inferencial y dos estudiantes del grupo experimental pudieron resolver la pregunta planteada del nivel crítico de forma adecuada.

Tabla 52: Resultados de indicadores del pretest para el grupo experimental – Segundo grado

GRUPO EXPERIMENTAL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	4	6	10
Ordena secuencia de hechos o acciones en el texto.	Literal	5	5	10
Analiza hechos presentes en el texto.	Inferencial	3	7	10
Interpreta información expuesta en el texto	Inferencial	4	6	10

o reacciones en los personajes.				
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	8	10

Fuente: Elaboración Propia.

En la Tabla 53 se puede observar los resultados del grupo de control luego del entrenamiento de los estudiantes. Luego de las 5 semanas se puede apreciar que casi la totalidad de los estudiantes pudieron responder adecuadamente las preguntas de nivel literal de comprensión lectora. De igual manera se puede visualizar que respondieron de manera adecuada un mayor número de estudiantes las preguntas del nivel inferencial y dos estudiantes lograron responder adecuadamente las preguntas de nivel crítico.

Tabla 53: Resultados de indicadores del postest para el grupo de control – Segundo grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	6	2	8
Ordena secuencia de hechos o acciones en el texto.	Literal	7	1	8
Analiza hechos presentes en el texto.	Inferencial	4	4	8
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	5	8
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	6	8

Fuente: Elaboración Propia.

En la Tabla 54 se puede visualizar los resultados obtenidos del grupo experimental luego de las 5 semanas de entrenamiento a través de las clases virtuales y el uso del videojuego. Se puede observar que la mayoría de los estudiantes lograron responder de manera adecuada las preguntas de nivel inferencial. Por otro lado, se puede examinar que un poco más de la mitad de los estudiantes pudieron responder adecuadamente las preguntas del nivel inferencial de la

comprensión lectora. Finalmente, se puede contemplar que hubo un incremento de los estudiantes que lograron responder adecuadamente las preguntas correspondientes al nivel crítico.

Tabla 54: Resultados de indicadores del postest para el grupo experimental – Segundo grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	8	2	10
Ordena secuencia de hechos o acciones en el texto.	Literal	7	3	10
Analiza hechos presentes en el texto.	Inferencial	6	4	10
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	7	3	10
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	3	7	10

Fuente: Elaboración Propia.

5.3.1.3 Análisis cuantitativo descriptivo – Tercer Grado

Con respecto al aprendizaje de los estudiantes de los alumnos de tercero de primaria, en la Tabla 55 se pueden visualizar los resultados estadísticos del pretest para cada grupo en donde se puede visualizar que el promedio de notas del grupo de control es superior al del grupo experimental. También se puede observar que la dispersión de los datos es superior en el grupo experimental que en grupo de control y la mediana del grupo de control es superior en una unidad a la mediana del grupo experimental.

Tabla 55: Resumen estadístico del pretest para el tercer grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	9	12.66	1.8708	13
Experimental	11	11.45	2.0181	12

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 56, se puede observar los resultados obtenidos para ambos grupos del tercer grado luego de las 5 semanas de entrenamiento. Se aprecia que el promedio de las notas obtenidas del grupo experimental es superior al grupo de control a través del uso del videojuego en sus clases. También se aprecia que la dispersión de las notas obtenidas con respecto a la media es superior en el grupo de control en comparación al grupo experimental. Además, se puede apreciar que la mediana de notas del grupo experimental es superior a la del grupo de control.

Tabla 56: Resumen estadístico del posttest para el tercer grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	9	13.66	1.1180	13
Experimental	11	14.72	1.0090	15

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 57, se puede observar los resultados de los indicadores iniciales para el grupo de control. Se aprecia que 4 estudiantes respondieron adecuadamente las preguntas de nivel literal. También se puede visualizar que tres estudiantes respondieron correctamente las preguntas de nivel inferencial y solamente dos alumnos tuvieron una respuesta aceptable en las preguntas de nivel crítico.

Tabla 57: Resultados de indicadores del pretest para el grupo de control – Tercer grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	4	5	9
Ordena secuencia de hechos o acciones en el texto.	Literal	4	5	9
Analiza hechos presentes en el texto.	Inferencial	3	6	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	6	9
Enjuicia las actitudes y conductas de los personajes o	Crítico	2	7	9

protagonistas en el
texto

Fuente: Elaboración Propia.

En la Tabla 58, se visualizan los resultados de los indicadores del pretest del grupo experimental. Se aprecia que casi la mitad de los estudiantes han respondido adecuadamente las preguntas del nivel literal. También se aprecia que menos de la mitad de los estudiantes respondieron adecuadamente las preguntas de nivel inferencial y solamente 3 estudiantes de los 11 alumnos respondieron correctamente las preguntas de nivel crítico.

Tabla 58: Resultados de indicadores del pretest para el grupo experimental – Tercer grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	5	6	11
Ordena secuencia de hechos o acciones en el texto.	Literal	6	5	11
Analiza hechos presentes en el texto.	Inferencial	4	7	11
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	8	11
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	3	8	11

Fuente: Elaboración Propia.

En la Tabla 59, se pueden ver los resultados de los indicadores luego de las 5 semanas de entrenamiento en el postest para el grupo de control. En dicha tabla se puede apreciar que ha habido un incremento en las preguntas de nivel literal de la comprensión lectora. También se puede visualizar una mayor cantidad de respuestas correctas en el nivel inferencial de la comprensión lectora y 5 estudiantes lograron responder adecuadamente las preguntas de nivel crítico de la lectura del postest.

Tabla 59: Resultados de indicadores del postest para el grupo de control – Tercer grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	6	3	9
Ordena secuencia de hechos o acciones en el texto.	Literal	7	2	9
Analiza hechos presentes en el texto.	Inferencial	6	3	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	4	5	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	5	4	9

Fuente: Elaboración Propia.

En la Tabla 60 se pueden observar los resultados del grupo experimental del tercer grado de primaria. Se aprecia que una mayor cantidad de estudiantes pudieron responder adecuadamente las preguntas de nivel literal e inferencial. También se puede observar que un poco más de la mitad de los alumnos pudieron responder adecuadamente las preguntas del nivel crítico.

Tabla 60: Resultados de indicadores del postest para el grupo experimental – Tercer grado

GRUPO EXPERIMENTAL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	8	3	11
Ordena secuencia de hechos o acciones en el texto.	Literal	9	2	11
Analiza hechos presentes en el texto.	Inferencial	7	4	11
Interpreta información	Inferencial	7	4	11

expuesta en el texto o reacciones en los personajes.				
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	6	5	11

Fuente: Elaboración Propia.

5.3.1.4 Análisis cuantitativo descriptivo – Cuarto Grado

En relación con el entrenamiento de los estudiantes del cuarto grado de primaria, en la Tabla 61 se puede visualizar los resultados generales obtenidos con respecto al pretest. Se aprecia en la Tabla 61 que el grupo de control tiene un promedio de notas mayor al grupo experimental. También se aprecia que la dispersión de las notas obtenidas del pretest del grupo de control es menor con respecto al del grupo experimental. De igual modo, se puede apreciar que la mediana del grupo de control es mayor en una unidad con respecto a la mediana del grupo experimental.

Tabla 61: Resumen estadístico del pretest para el cuarto grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	7	13.00	1.1547	13
Experimental	9	11.77	1.2018	12

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 62 se pueden observar los resultados luego de las 5 semanas de entrenamiento para cada grupo del cuarto grado de primaria. En esta ocasión se puede visualizar que el promedio de las notas del grupo experimental es mayor a las del grupo de control. También se aprecia que la dispersión de los datos del grupo experimental es mayor a la del grupo de control y se visualiza por último que la mediana del grupo de control es inferior en una unidad a la del grupo experimental.

Tabla 62: Resumen estadístico del postest para el cuarto grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	7	14.71	0.7559	15
Experimental	9	15.88	1.0540	16

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 63 se pueden observar los resultados de los indicadores del grupo de control para el cuarto grado de primaria. Se puede visualizar en los resultados que la mayoría de los estudiantes pudo resolver satisfactoriamente correctamente las preguntas del nivel literal. No obstante, menos de la mitad de los estudiantes pudo responder de manera adecuada las preguntas de nivel inferencial. Con respecto al nivel crítico, se observó un mayor número de estudiantes que respondieron las preguntas de este nivel correctamente.

Tabla 63: Resultados de indicadores del pretest para el grupo de control – Cuarto grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	4	3	7
Ordena secuencia de hechos o acciones en el texto.	Literal	4	3	7
Analiza hechos presentes en el texto.	Inferencial	3	4	7
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	2	5	7
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	4	3	7

Fuente: Elaboración Propia.

En la Tabla 64 se visualizan los resultados de los indicadores del grupo experimental en el pretest. En dicha tabla se puede observar que casi la mitad de los estudiantes pudieron resolver adecuadamente las preguntas del nivel literal de la comprensión lectora. También se puede apreciar que la tercera que la tercera parte de los alumnos pudo responder correctamente las

preguntas del nivel inferencial y solamente dos estudiantes pudieron responder las preguntas de nivel crítico.

Tabla 64: Resultados de indicadores del pretest para el grupo experimental – Cuarto grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	4	5	9
Ordena secuencia de hechos o acciones en el texto.	Literal	5	4	9
Analiza hechos presentes en el texto.	Inferencial	3	6	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	6	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	2	7	9

Fuente: Elaboración Propia.

En la Tabla 65 se pueden visualizar los resultados para el grupo de control luego del entrenamiento. Se puede observar un cambio en su nivel de comprensión literal dado que la mayoría de los estudiantes acertaron correctamente las preguntas planteadas. De igual modo se puede apreciar que casi la totalidad del grupo de control contestaron correctamente la pregunta de nivel inferencial y un estudiante adicional respondió adecuadamente las preguntas de nivel crítico.

Tabla 65: Resultados de indicadores del postest para el grupo de control – Cuarto grado

Indicadores	Nivel	GRUPO DE CONTROL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	6	1	7
Ordena secuencia de hechos o acciones en el texto.	Literal	7	0	7
Analiza hechos presentes en el texto.	Inferencial	5	2	7

Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	5	2	7
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	5	2	7

Fuente: Elaboración Propia.

En la Tabla 66 se pueden visualizar los resultados del postest para el grupo experimental de este cuarto grado. Se aprecia que los estudiantes mejoraron en su nivel literal de comprensión lectora. También se puede observar que un mayor número de estudiantes lograron responder adecuadamente las preguntas de nivel inferencial y un poco de la mitad de los alumnos lograron responder correctamente las preguntas de nivel crítico.

Tabla 66: Resultados de indicadores del postest para el grupo experimental – Cuarto grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	8	1	9
Ordena secuencia de hechos o acciones en el texto.	Literal	8	1	9
Analiza hechos presentes en el texto.	Inferencial	6	3	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	7	2	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	5	4	9

Fuente: Elaboración Propia.

5.3.1.5 Análisis cuantitativo descriptivo – Quinto Grado

En la Tabla 67 se pueden visualizar los resultados del pretest para el quinto grado de primaria. Se aprecia que existe una diferencia mínima en el promedio de notas del grupo de control y experimental. También se puede observar que la dispersión de los datos del grupo

experimental es mayor a la del grupo de control. Por último, se puede visualizar que la mediana en ambos grupos tiene un valor de 13.

Tabla 67: Resumen estadístico del pretest para el quinto grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	9	13.66	1.2247	13
Experimental	10	13.10	1.6633	13

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 68 se pueden observar los resultados obtenidos luego del entrenamiento para ambos grupos de quinto grado. En la Tabla 68 se puede visualizar que el promedio de notas del grupo experimental fue superior al del grupo de control. También se aprecia que la dispersión en los datos es mayor en el grupo experimental con respecto al grupo de control. Además, se puede apreciar que la mediana del grupo experimental es mayor en una unidad a la mediana del grupo de control.

Tabla 68: Resumen estadístico del posttest para el quinto grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	9	14.66	1.0000	15
Experimental	10	16.00	1.0540	16

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 69 se puede visualizar los resultados obtenidos en el pretest para el grupo de control de quinto grado. Se aprecia que los estudiantes parten con un nivel de comprensión literal relativamente alto dado que solamente la tercera parte no lograron responder las preguntas adecuadamente. También se aprecia que los estudiantes poseen un nivel intermedio en la comprensión lectora de nivel inferencial y también se aprecia que 4 estudiantes de los 9 involucrados en el grupo lograron responder adecuadamente las preguntas de nivel crítico.

Tabla 69: Resultados de indicadores del pretest para el grupo de control – Quinto grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	6	3	9
Ordena secuencia de hechos o acciones en el texto.	Literal	6	3	9
Analiza hechos presentes en el texto.	Inferencial	5	4	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	6	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	4	5	9

Fuente: Elaboración Propia.

En la Tabla 70 se puede observar los resultados del grupo experimental en el pretest. En dicha tabla se puede visualizar que la mitad de los estudiantes pudieron responder adecuadamente las preguntas de nivel literal. También se aprecia que casi la mitad de los participantes pudieron responder correctamente las preguntas de nivel inferencial y que solamente 4 estudiantes pudieron responder adecuadamente las preguntas de opción múltiple.

Tabla 70: Resultados de indicadores del pretest para el grupo experimental – Quinto grado

GRUPO EXPERIMENTAL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	5	5	10
Ordena secuencia de hechos o acciones en el texto.	Literal	6	4	10
Analiza hechos presentes en el texto.	Inferencial	5	5	10
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	4	6	10

Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	4	6	10
--	---------	---	---	----

Fuente: Elaboración Propia.

En la Tabla 71 se pueden observar los resultados luego del entrenamiento de las 5 semanas para el grupo de control. Se aprecia que casi la totalidad de los estudiantes pudieron responder adecuadamente las preguntas de nivel literal e inferencial. También se observa una mejora en el nivel crítico dado que 6 estudiantes pudieron recorrer adecuadamente este tipo de preguntas.

Tabla 71: Resultados de indicadores del postest para el grupo de control – Quinto grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	8	1	9
Ordena secuencia de hechos o acciones en el texto.	Literal	8	1	9
Analiza hechos presentes en el texto.	Inferencial	7	2	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	6	3	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	6	3	9

Fuente: Elaboración Propia.

En la Tabla 72 se pueden observar los resultados obtenidos para el grupo experimental del quinto grado. En dicha tabla se puede visualizar que hubo una mejora importante con respecto al nivel literal de la comprensión lectora. De igual modo, ha puede apreciar un mejor rendimiento de los estudiantes en el nivel inferencial de la comprensión lectora. En el caso del nivel crítico ha habido un incremento en la cantidad de estudiantes que lograron responder la pregunta, pero este aumento no ha sido tan alto como en los niveles literal e inferencial.

Tabla 72: Resultados de indicadores del postest para el grupo experimental – Quinto grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	9	1	10
Ordena secuencia de hechos o acciones en el texto.	Literal	10	0	10
Analiza hechos presentes en el texto.	Inferencial	7	3	10
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	8	2	10
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	6	4	10

Fuente: Elaboración Propia.

5.3.1.6 Análisis cuantitativo descriptivo – Sexto Grado

En la tabla 73, se pueden observar los resultados generales del pretest por parte de los estudiantes de sexto grado de primaria. En dicha tabla se puede apreciar que los estudiantes del grupo de control obtuvieron un promedio de notas superior a la del grupo experimental. También se aprecia que la dispersión de los datos del grupo experimental es menor a la del grupo de control. Por último, se visualiza que la mediana del grupo de control es mayor en 0.5 a la del grupo experimental.

Tabla 73: Resumen estadístico del pretest para el sexto grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	10	14.70	1.3374	14.5
Experimental	9	13.88	0.7817	14

Nota: N = Número total de la muestra.

Fuente: Elaboración Propia.

Por otro lado, en la Tabla 74 se visualizan los resultados generales de las notas luego del entrenamiento para ambos grupos del sexto grado de primaria. Se puede apreciar que el grupo experimental tuvo un promedio de notas mayor en comparación al grupo de control. También

se visualiza que la dispersión de los datos es menor en el grupo experimental con respecto al grupo de control. Finalmente, se aprecia un incremento de la mediana en ambos grupos siendo mayor la mediana del grupo experimental.

Tabla 74: Resumen estadístico del postest para el sexto grado

Grupo	N	Prom. Notas	Desv. Estándar	Mediana
Control	10	15.60	1.2649	15.5
Experimental	9	16.77	1.0929	17

Nota. N = Número total de la muestra.

Fuente: Elaboración Propia.

En la Tabla 75 se pueden visualizar los resultados de los indicadores para el grupo de control en el pretest. En este grupo de estudiantes se puede observar que la mitad pudieron responder adecuadamente las preguntas tanto de nivel literal como de nivel inferencial de la comprensión lectora. También se puede apreciar que de igual modo la mitad de estos alumnos pudieron responder adecuadamente las preguntas de nivel crítico que poseía el pretest.

Tabla 75: Resultados de indicadores del pretest para el grupo de control – Sexto grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	6	4	10
Ordena secuencia de hechos o acciones en el texto.	Literal	7	3	10
Analiza hechos presentes en el texto.	Inferencial	5	5	10
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	6	4	10
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	5	5	10

Fuente: Elaboración Propia.

En la Tabla 76 se visualizan los resultados obtenidos por el grupo experimental en el pretest. Se puede observar que la mayoría de los estudiantes pudieron responder adecuadamente las preguntas de nivel literal. No obstante, menos de la mitad de los estudiantes pudieron responder correctamente las preguntas de nivel inferencial y de nivel crítico de la comprensión lectora.

Tabla 76: Resultados de indicadores del pretest para el grupo experimental – Sexto grado

GRUPO EXPERIMENTAL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	5	4	9
Ordena secuencia de hechos o acciones en el texto.	Literal	7	2	9
Analiza hechos presentes en el texto.	Inferencial	4	5	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	3	6	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	4	5	9

Fuente: Elaboración Propia.

En la Tabla 77 se pueden se visualizar los resultados luego del entrenamiento para el grupo de control. Se puede observar un incremento de estudiantes que respondieron correctamente las preguntas de nivel literal, así como del nivel inferencial. También se aprecia que una mayor cantidad de estudiantes pudieron responder adecuadamente las preguntas de nivel crítico.

Tabla 77: Resultados de indicadores del postest para el grupo de control – Sexto grado

GRUPO DE CONTROL				
Indicadores	Nivel	Si lo lograron	No lo lograron	Total
Reconoce personajes del texto.	Literal	8	2	10
Ordena secuencia de hechos o acciones en el texto.	Literal	9	1	10
Analiza hechos presentes en el texto.	Inferencial	6	4	10

Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	8	2	10
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	7	3	10

Fuente: Elaboración Propia.

Por último, en la Tabla 78 se pueden observar los resultados del grupo experimental luego de las 5 semanas de entrenamiento a través de las clases y el uso del videojuego. En este grupo se puede apreciar que hubo un aumento de estudiantes que respondieron correctamente las preguntas del nivel literal. También una mayor cantidad de estudiantes pudieron analizar de una mejor manera los textos y respondieron adecuadamente las preguntas de nivel inferencial y del nivel crítico.

Tabla 78: Resultados de indicadores del postest para el grupo experimental – Sexto grado

Indicadores	Nivel	GRUPO EXPERIMENTAL		Total
		Si lo lograron	No lo lograron	
Reconoce personajes del texto.	Literal	8	1	9
Ordena secuencia de hechos o acciones en el texto.	Literal	9	0	9
Analiza hechos presentes en el texto.	Inferencial	7	2	9
Interpreta información expuesta en el texto o reacciones en los personajes.	Inferencial	6	3	9
Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto	Crítico	6	3	9

Fuente: Elaboración Propia.

5.3.2 Análisis cuantitativo inferencial de las evaluaciones

En esta sección se presentarán los resultados obtenidos del análisis cuantitativo inferencial realizado para cada grado siguiendo la secuencia de actividades planteada en la Figura N°67 del capítulo anterior. Para la aplicación de la prueba T de student para muestras independientes se estableció la hipótesis nula e hipótesis alterna. La hipótesis nula (H_0) afirma que no existe una diferencia significativa entre las medias de las notas del grupo de control grupo experimental y la hipótesis alterna (H_1) corresponde a la hipótesis general planteada al inicio del proyecto, pero para realizar este análisis estadístico, esta hipótesis fue transformada a una hipótesis estadística de tal forma que la hipótesis alterna es que si existe una diferencia significativa entre las medias de las calificaciones del grupo de control y grupo experimental. Para validar o rechazar la hipótesis nula se aplicó los criterios de la Ecuación (54) y Ecuación (55). El nivel de significancia establecido para este proyecto de investigación fue de 0.05. Luego de establecer la hipótesis nula, hipótesis alterna y el nivel de significancia, se procedió a realizar la prueba de normalidad con el objetivo de determinar que los datos provengan de una distribución normal. Seguidamente, se procedió a calcular el tamaño, media y varianza para cada muestra de cada grupo por cada grado aplicando de esta manera la Ecuación (56). Posteriormente, se aplicó la prueba de Levene con el objetivo de determinar que las varianzas sean homogéneas. A continuación, se procedió a calcular la diferencia de medias (D_m) aplicando la Ecuación (57), la varianza común (V_c) aplicando la Ecuación (58) y el error estándar de la diferencia de medias (ESM) aplicando para este último la Ecuación (59). Con los datos anteriores encontrados, se calculó el valor del estadístico T- student (t) aplicando la Ecuación (60) y el valor de los grados de libertad aplicando la Ecuación (61). Con estos dos últimos valores calculados, se procedió a verificar el valor del P-Value y de este modo validar o rechazar la hipótesis nula. El mismo procedimiento descrito anteriormente fue aplicado al pretest y posttest de cada grado del nivel primario. La aplicación de esta secuencia de actividades será presentada en la siguiente sub sección para el primer grado de primaria como ejemplo y finalmente se presentarán los resultados generales obtenidos para cada grado de primaria.

5.3.2.1 Ejemplo del análisis cuantitativo inferencial aplicado al primer grado

A continuación, se presenta la aplicación de la prueba T de student para muestras independientes en el pretest del primer grado de primaria:

- **Paso 1:** Formulación de la hipótesis nula y alterna.

La hipótesis nula y alterna son las siguientes:

H_0 : No existe una diferencia significativa entre las medias de las calificaciones del grupo de control y grupo experimental.

H_1 : Sí existe una diferencia significativa entre las medias de las calificaciones del grupo de control y grupo experimental.

- **Paso 2:** Establecimiento de nivel de significancia.

El nivel de significancia (α) establecido es de 0.05.

- **Paso 3:** Aplicación de la prueba de normalidad.

Las hipótesis para contrastar en la Prueba de normalidad son las siguientes:

H_0 : Los datos provienen de una distribución normal.

H_1 : Los datos no provienen de una distribución normal.

El nivel de significancia establecido para realizar la prueba de normalidad es de $\alpha = 0.05$.

Dado que la muestra es inferior a 30, se utilizó el test de Shapiro-Whilk. Realizando las operaciones respectivas a través del software SPSS, se obtienen los resultados presentados en la Tabla 79.

Tabla 79: Prueba de normalidad de Shapiro-Wilk – Pretest de Primer Grado

Grupo	Estadístico	GL	P-valor
Control	0.957	11	0.735
Experimental	0.823	9	0.137

Nota. GL = Grados de libertad.

Fuente: Elaboración Propia.

Se aprecia en la Tabla 79 que el P-valor del grupo de control es de 0.735 y P-valor del grupo experimental es de 0.137. Aplicando el criterio de decisión de la Ecuación (54) y Ecuación (55) y utilizando el nivel de significancia establecido se acepta la H_0 para ambos grupos y se puede afirmar que los datos del pretest tanto del grupo de control como del grupo experimental provienen de una distribución normal.

- **Paso 4:** Cálculo del tamaño, media y varianza para cada muestra.

En la Tabla 80 se puede visualizar el tamaño, la media y varianza tanto del grupo de control como del grupo experimental.

Tabla 80: Tamaño, media y varianza – Pretest del Primer Grado

Grupo	N	Media	Varianza
Control	11	13.45	1.873
Experimental	9	13.00	0.750

Fuente: Elaboración Propia.

- **Paso 5:** Aplicación de la Prueba de Levene.

Para la prueba de Levene, se formularon las siguientes hipótesis:

H_0 : La varianza de los datos del pretest son iguales.

H_1 : Existe una diferencia significativa en las varianzas de ambos grupos.

El nivel de significancia establecido es de 0.05.

Los resultados de la Prueba de Levene se pueden visualizar en la Tabla 81.

Tabla 81: Prueba de Levene – Pretest del Primer Grado

Supuesto	P-valor
Se asumen igualdad de varianzas	0.249
No se asumen varianzas iguales	-

Fuente: Elaboración Propia.

En la Tabla 81 se puede visualizar que el P-valor es de 0.249 para la Prueba de Levene. Utilizando el criterio de decisión de la Ecuación (54) y Ecuación (55) y el nivel de significancia establecido, se acepta la H_0 por lo que se afirma que la varianza de los datos del pretest son iguales.

- **Paso 6:** Cálculo de la diferencia de medias y varianza común.

Aplicando la Ecuación (57) utilizando los resultados de la Tabla 80 se obtiene una diferencia de medias de 0.45. Por otro lado, aplicando la Ecuación (58) para la varianza común utilizando los resultados de la Tabla 80 se obtiene un valor de 1.3738.

- **Paso 7:** Cálculo del error estándar de la diferencia de medias.

Aplicando la Ecuación (59) se obtiene un error estándar de diferencia de medias de 0.5230.

- **Paso 8:** Cálculo del estadístico T de student.

Aplicando la Ecuación (60) se obtiene un valor del T de student de 0.863. De igual modo, aplicando la Ecuación (61) para los grados de libertad se obtiene un valor de 18. Los resultados completos provenientes del SPSS se pueden visualizar en la Tabla 82.

Tabla 82: Prueba T para muestras independientes – Pretest del Primer Grado

Supuesto	t	GL	P-Valor
Se asumen varianzas iguales	0.863	18	0.400
No se asumen varianzas iguales	0.903	17.07	0.379

Fuente: Elaboración Propia.

- **Paso 9:** P-valor correspondiente al estadístico T de student

Debido a que en el resultado de la prueba de Levene realizado anteriormente se obtuvo igualdad de varianzas, se utilizó solamente la primera fila de la Tabla 82. En dicha fila se observa que el P-valor correspondiente al valor del estadístico T de student asumiendo varianzas iguales es de 0.400.

- **Paso 10:** Aceptación o rechazo de la hipótesis nula.

El P-valor encontrado fue de 0.400 y utilizando el criterio de decisión planteado en la Ecuación (54) y Ecuación (55) junto con el nivel de significancia establecido al inicio, se acepta la hipótesis nula H_0 por lo que se afirma que no existe una diferencia significativa en las notas obtenidas en el pretest realizado por los estudiantes de primero de primaria.

Luego de las 5 semanas de entrenamiento para ambos grupos de primero de primaria y habiendo realizado estos estudiantes el postest, se presenta a continuación al análisis inferencial para esta prueba en particular.

- **Paso 1:** Formulación de la hipótesis nula y alterna.

La hipótesis nula y alterna para determinar si hay una diferencia significativa en el postest son las siguientes:

H_0 : No existe una diferencia significativa entre las medias de las calificaciones del grupo de control y grupo experimental.

H_1 : Sí existe una diferencia significativa entre las medias de las calificaciones del grupo de control y grupo experimental.

- **Paso 2:** Establecimiento de nivel de significancia.

El nivel de significancia (α) establecido es de 0.05.

- **Paso 3:** Aplicación de la prueba de normalidad.

Las hipótesis para contrastar en la Prueba de normalidad son las siguientes:

H_0 : Los datos provienen de una distribución normal.

H_1 : Los datos no provienen de una distribución normal.

El nivel de significancia establecido para realizar la prueba de normalidad es de $\alpha = 0.05$.

Dado que la muestra es inferior a 30, se utilizó el test de Shapiro-Whilk. Realizando las operaciones respectivas a través del software SPSS, se obtienen los resultados presentados en la Tabla 83.

Tabla 83: Prueba de normalidad de Shapiro-Wilk – Postest de Primer Grado

Grupo	Estadístico	GL	P-valor
Control	0.907	11	0.223
Experimental	0.951	9	0.696

Nota. GL = Grados de libertad.

Fuente: Elaboración Propia.

Se aprecia en la Tabla 83 que el P-valor del grupo de control es de 0.223 y P-valor del grupo experimental es de 0.696. Aplicando el criterio de decisión de la Ecuación (54) y Ecuación (55) y utilizando el nivel de significancia establecido se acepta la H_0 para ambos grupos y se puede afirmar que los datos del postest tanto del grupo de control como del grupo experimental provienen de una distribución normal.

- **Paso 4:** Cálculo del tamaño, media y varianza para cada muestra.

En la Tabla 84 se puede visualizar el tamaño, la media y varianza tanto del grupo de control como del grupo experimental.

Tabla 84: Tamaño, media y varianza – Posttest del Primer Grado

Grupo	N	Media	Varianza
Control	11	14.00	3.200
Experimental	9	15.77	2.944

Fuente: Elaboración Propia.

- **Paso 5:** Aplicación de la Prueba de Levene.

Para la prueba de Levene, se formularon las siguientes hipótesis:

H_0 : La varianza de los datos del posttest son iguales.

H_1 : Existe una diferencia significativa en las varianzas de ambos grupos.

El nivel de significancia establecido es de 0.05.

Los resultados de la Prueba de Levene se pueden visualizar en la Tabla 85.

Tabla 85: Prueba de Levene – Posttest del Primer Grado

Supuesto	P-valor
Se asumen igualdad de varianzas	0.821
No se asumen varianzas iguales	-

Fuente: Elaboración Propia.

En la Tabla 85 se puede visualizar que el P-valor es de 0.821 para la Prueba de Levene. Utilizando el criterio de decisión de la Ecuación (54) y Ecuación (55) y el nivel de significancia establecido, se acepta la H_0 por lo que se afirma que la varianza de los datos del posttest son iguales.

- **Paso 6:** Cálculo de la diferencia de medias y varianza común.

Aplicando la Ecuación (57) utilizando los resultados de la Tabla 84 se obtiene una diferencia de medias de -1.77. Por otro lado, aplicando la Ecuación (58) para la varianza común utilizando los resultados de la Tabla 84 se obtiene un valor de 3.0862.

- **Paso 7:** Cálculo del error estándar de la diferencia de medias.

Aplicando la Ecuación (59) se obtiene un error estándar de diferencia de medias de 0.7896.

- **Paso 8:** Cálculo del estadístico T de student.

Aplicando la Ecuación (60) se obtiene un valor del T de student de -2.251. De igual modo, aplicando la Ecuación (61) para los grados de libertad se obtiene un valor de 18. Los resultados completos provenientes del SPSS se pueden visualizar en la Tabla 86.

Tabla 86: Prueba T para muestras independientes – Postest del Primer Grado

Supuesto	t	GL	P-Valor
Se asumen varianzas iguales	-2.251	18	0.037
No se asumen varianzas iguales	-2.261	17.49	0.037

Fuente: Elaboración Propia.

- **Paso 9:** P-valor correspondiente al estadístico T de student

Debido a que en el resultado de la prueba de Levene realizado anteriormente se obtuvo igualdad de varianzas, se utilizó solamente la primera fila de la Tabla 86. En dicha fila se observa que el P-valor correspondiente al valor del estadístico T de student asumiendo varianzas iguales es de 0.037.

- **Paso 10:** Aceptación o rechazo de la hipótesis nula.

El P-valor encontrado fue de 0.037 y utilizando el criterio de decisión planteado en la Ecuación (54) y Ecuación (55) junto con el nivel de significancia establecido al inicio, se rechaza la hipótesis nula H_0 por lo que se afirma que existe una diferencia significativa en las notas obtenidas en el postest realizado por los estudiantes de primero de primaria.

En la presente sección se ha realizado la secuencia de actividades y aplicación de las ecuaciones correspondientes de la prueba T de Student para muestras independientes aplicado al primer grado de primaria como ejemplo. No obstante, para visualizar la aplicación y cálculos de la prueba T de Student para los grados de segundo hasta sexto de primaria se puede consultar al siguiente link <https://github.com/alvaromachuca22/Tesis-Ingenieria-ESAN>.

5.3.2.2 Resumen de los resultados obtenidos para cada grado

En la Tabla 87, se presenta el resumen de los resultados de la aplicación de la prueba T de student para muestras independientes para cada grado de educación primaria.

En primer lugar, en la Tabla 87 con respecto al pretest se puede observar que no se ha detectado una mejora significativa al inicio de la intervención, estos resultados significan que

cada grupo tanto de control como experimental de cada grado han partido en las mismas condiciones.

Por otro lado, en la Tabla 87 cada grado ha obtenido una mejora significativa en el posttest a favor del grupo experimental. La mejora mínima fue de 1.06 puntos que corresponde al grupo experimental de tercer grado de primaria. La segunda nota mínima es la que le corresponde al grupo experimental del cuarto y sexto grado de primaria, ambos con una mejora de 1.17 puntos. La tercera nota mínima corresponde al grupo experimental del quinto grado de primaria con un valor de 1.34 puntos. Las dos mejores significativas corresponden a los grados de primero y segundo grado de primaria. La segunda mejor mejora significativa corresponde al grupo experimental del segundo grado de primaria con un valor de 1.45 puntos y la mejora significativa más alta corresponde al grupo experimental del primer grado de primaria con un valor de 1.77 puntos.

Tabla 87: Resumen de los resultados del análisis estadístico inferencial para cada grado de educación primaria

PRUEBA T DE STUDENT PARA MUESTRAS INDEPENDIENTES						
Grado	Evaluación	Grupo	Prom. De notas	P-valor	Nivel de significancia	Mejora significativa
1°	Pretest	Control	13.45	0.400	0.05	No encontrada
		Experimental	13.00			
	Postest	Control	14.00	0.037	0.05	1.77 puntos (*)
		Experimental	15.77			
2°	Pretest	Control	12.00	0.212	0.05	No encontrada
		Experimental	10.90			
	Postest	Control	13.25	0.039	0.05	1.45 puntos (*)
		Experimental	14.70			
3°	Pretest	Control	12.66	0.184	0.05	No encontrada
		Experimental	11.45			
	Postest	Control	13.66	0.039	0.05	1.06 puntos (*)
		Experimental	14.72			
4°	Pretest	Control	13.00	0.059	0.05	No encontrada
		Experimental	11.77			
	Postest	Control	14.71	0.026	0.05	1.17 puntos (*)
		Experimental	15.88			
5°	Pretest	Control	13.66	0.414	0.05	No encontrada
		Experimental	13.10			
	Postest	Control	14.66	0.012	0.05	1.34 puntos (*)
		Experimental	16.00			
6°	Pretest	Control	14.70	0.131	0.05	No encontrada
		Experimental	13.88			
	Postest	Control	15.60	0.045	0.05	1.17 puntos (*)
		Experimental	16.77			

(*) Mejora significativa a favor del grupo experimental.

5.3.3 Análisis cualitativo de la encuesta de satisfacción

Los resultados del análisis cualitativo se obtuvieron por la encuesta de satisfacción completada por los estudiantes de primaria. La variable a analizar es la variable independiente que corresponde a la utilización de videojuego desarrollado para las clases de comprensión lectora de estudiantes de educación primaria. La dimensión de esta variable corresponde a la satisfacción del uso del videojuego por parte de los alumnos de educación primaria que posee como indicador la experiencia de uso del videojuego. Los resultados de la satisfacción de los alumnos a través del uso del videojuego se presentan en la Tabla 88.

Tabla 88: Satisfacción encontrada de los estudiantes de primaria por uso del videojuego

Grado	BUENA		REGULAR		MALA		Total
	M	F	M	F	M	F	
1°	4	3	0	2	0	0	9
2°	6	1	1	2	0	0	10
3°	7	2	1	0	0	1	11
4°	4	2	3	0	0	0	9
5°	5	1	1	2	0	1	10
6°	3	3	2	1	0	0	9

Nota. M = Masculino, F = Femenino.

Fuente: Elaboración Propia.

Los resultados de la Tabla 88 presentada muestran que la mayoría de los niños y niñas del grupo experimental de cada grado se sintieron más a gusto con el videojuego desarrollado. En el caso de las niñas del primer, segundo y quinto grado mostraron una satisfacción regular con respecto al uso del videojuego. Por otro lado, una estudiante del tercer grado y otra del quinto grado no se sintieron a gusto con el uso del videojuego. No obstante, en la mayoría de los grados obtuvieron respuesta favorable con respecto al videojuego que fue construido para ellos.

También se recibieron comentarios muy positivos de los dos docentes de comprensión lectora afirmando que este videojuego permitió que sus estudiantes pudieran estar más participativos dentro de sus clases virtuales y además los ayudó a poder brindar estrategias de lectura de una mejor manera dado que el videojuego a través de la sección de estrategias que se desarrolló los estudiantes pueden tener siempre disponible gran variedad de técnicas de lectura que pueden aplicar a sus propias tareas que se les deja para la casa.

Del mismo modo, en la Figura N°130 se presenta de forma gráfica la experiencia del uso del videojuego por parte de los grupos experimentales que participaron del proyecto en donde

se puede observar el total de estudiantes por cada grado que respondieron a través de la encuesta de satisfacción su experiencia con uso del videojuego desarrollado para ellos.

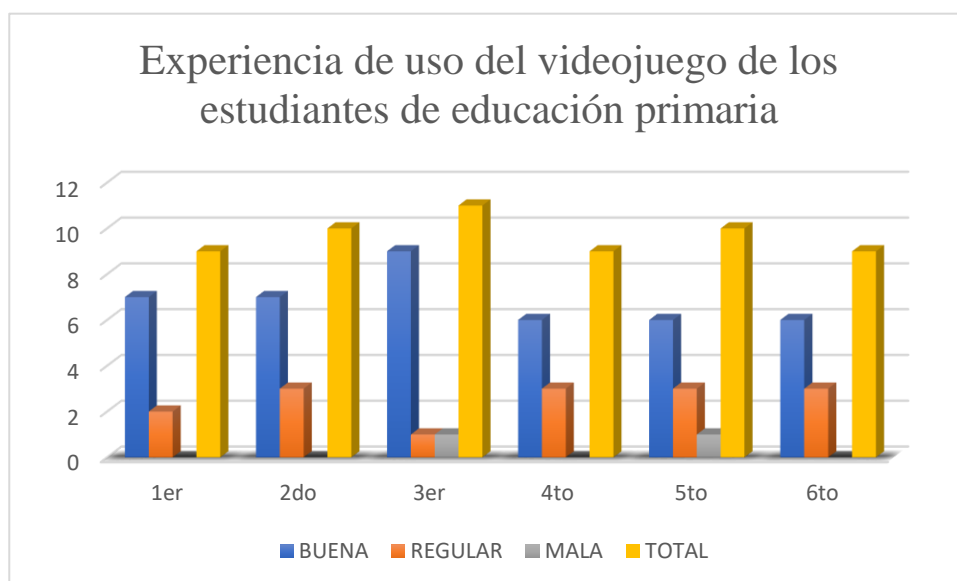


Figura N°130: Resultados del análisis cualitativo – Experiencia de uso del videojuego.
Fuente: Elaboración propia.

En Figura N°130 presentada se puede observar que en general se obtuvieron resultados favorables con respecto a las opiniones de los estudiantes sobre su experiencia del uso del videojuego. En la gráfica se visualiza que el primer, segundo y tercer grado de primaria mostraron una aprobación mayor con respecto al videojuego. Por otro lado, con respecto al cuarto, quinto y sexto grado se aprecia que en general dieron una opinión positiva del videojuego desarrollado, pero también se observa que 3 estudiantes por cada uno de estos tres grados afirmaron tener una experiencia regular con respecto al videojuego y solamente un estudiante de tercer y otro de quinto grado afirmaron que el prototipo desarrollado no fue de su agrado.

La segunda pregunta de la encuesta de satisfacción corresponde a qué fue lo que les pareció más agradable a los estudiantes por cada grado. Algunas de las respuestas por parte de los estudiantes de primero de primaria fueron las siguientes:

“Los personajes”, “Las lecturas y preguntas”, “Los niveles”.

Los estudiantes del segundo grado tuvieron las siguientes respuestas en la encuesta de satisfacción con respecto a lo que más les agrado en el videojuego:

“Los zombies que atacaban”, “Las preguntas y las lecturas”, “Los disparos y las pistolas”.

Con respecto a los estudiantes de tercer grado de primaria, sus respuestas fueron las siguientes:

“Las lecturas del videojuego”, “Las armas del videojuego”, “Los diferentes niveles del videojuego”.

En relación con los estudiantes del cuarto grado de primaria, redactaron las siguientes respuestas:

“Los terrenos que tiene el juego”, “Los zombies y las lecturas que el videojuego tiene”, “La cantidad de armas que se puede intercambiar”.

Acerca de las respuestas de los estudiantes de quinto grado de primaria, se obtuvieron las siguientes:

“Las lecturas que tiene el videojuego”, “Las imágenes sobre el cuidado en la casa”, “La música que el videojuego tiene”.

Por último, las respuestas de los estudiantes del sexto grado de primaria sobre los aspectos que más les agradaron del videojuego se presentan a continuación:

“Las lecturas se ven muy claras y las preguntas son interactivas”, “Las animaciones de los zombies”, “Las estrategias de lectura que el videojuego enseña”.

En general se puede apreciar que a los estudiantes les agradaron los personajes de los modelos en 3D de los zombies que fueron descargados, las lecturas y las preguntas que deben de responder con respecto a la lectura leída, los modelos en 3D de las armas que fueron utilizadas dentro del videojuego y la sección de estrategias que el prototipo contiene. También se obtuvieron respuestas resaltando la música que el videojuego posee, los consejos que se brinda para cuidarse dentro de sus hogares para prevenir el Covid-19 así como los diferentes diseños de los terrenos que el videojuego posee.

La tercera pregunta que contiene la encuesta de satisfacción es con el objetivo de recopilar las sugerencias que los estudiantes proponen para poder mejorar el videojuego desarrollado.

Las sugerencias que proponen los estudiantes del primero de primaria son las siguientes:

“Poner más niveles”, “Poner más preguntas”, “Poner más armas”.

Por otro lado, las recomendaciones que proponen los estudiantes de según grado de primaria son:

“Poner más zombies”, “Poner poderes nuevos”, “Nuevos niveles”.

De igual modo, las recomendaciones que brindaron los estudiantes de tercer grado son las siguientes:

“Poner nuevas armas”, “Quiero cambiar de personaje”, “Poner más lecturas”.

Asimismo, las sugerencias que brindaron los estudiantes del cuarto grado de primaria para poder mejorar el videojuego fueron las siguientes:

“Agregar más vidas”, “Poner atajos nuevos”, “Poner zombies más grandes”.

Los estudiantes del quinto grado también brindaron sugerencias para poder hacer del videojuego más atractivo, las cuales fueron las siguientes:

“Poner niveles con más zombies”, “Poder disparar más rápido”, “Agregar monedas para comprar objetos”.

Finalmente, las sugerencias del sexto grado de primaria para mejorar el videojuego se presentan a continuación:

“El videojuego debería mostrar las respuestas correctas al final de la lectura”, “Poner una selección de personajes”, “Que el videojuego tenga niveles de dificultad”.

En general con respecto a esta pregunta, se aprecia que los estudiantes desean que el videojuego posea más niveles, pero también agregar más enemigos dentro de cada nivel. Además, se aprecia que los estudiantes desean que el videojuego pueda tener una mayor cantidad de lecturas para cada grado y agregar una tienda virtual para poder comprar nuevos objetos como armas o adquirir nuevos poderes.

La última pregunta de la encuesta de satisfacción corresponde a lo que los estudiantes comentaban con sus padres sobre el videojuego que habían jugado.

Algunos de los comentarios de los estudiantes de primer grado de primaria que les comentaban a sus padres sobre el videojuego fueron los siguientes:

“Que era divertido”, “Que era bueno”, “Que estaba jugando”.

Los comentarios que escribieron los alumnos del segundo de primaria se presentan a continuación:

“Les decía que el videojuego era bueno”, “Que me encontraba feliz”, “Que estaba jugando un videojuego”.

De igual modo, las respuestas que brindaron los estudiantes de tercero de primaria fueron los siguientes:

“Que era un videojuego divertido”, “Que aprendo mejor con el videojuego”, “Que el videojuego es entretenido”.

Los estudiantes de cuarto de primaria también brindaron sus respuestas sobre lo que les comentaban a sus padres sobre el videojuego. Estos comentarios se presentan a continuación:

“Les comentaba que estaba jugando un videojuego de mi colegio”, “Que el videojuego tenía zombies”, “Que estaba en clases jugando”.

Del mismo modo, los estudiantes de quinto grado de primaria brindaron sus comentarios sobre lo que les comentaban a sus padres. Algunas de estas respuestas fueron las siguientes:

“Les decía que el videojuego era genial”, “Que el videojuego enseñaba a leer”, “Que estaba jugando un videojuego para aprender a leer mejor”.

Finalmente, los comentarios de los estudiantes de sexto grado de primaria sobre lo que comentaban a sus padres del videojuego se presentan a continuación:

“Que el videojuego era mejor que las clases”, “Que la profesora nos compartió un videojuego para mejorar nuestra forma de leer”, “Que el videojuego mostraba estrategias para leer mejor”.

En general, se aprecia que los estudiantes tuvieron comentarios positivos sobre el videojuego que comentaban a sus padres. Se aprecia además que los estudiantes entendieron el objetivo general del videojuego que era ayudarlos a poder leer mucho mejor mostrando lecturas dentro de este con sus respectivas preguntas y estrategias para que los alumnos puedan aplicarlas en sus clases de comprensión lectora pero también en las lecturas que posee el videojuego.

Por otro lado, los estudiantes mostraron sentirse a gusto mientras jugaban el videojuego dado que dentro de sus comentarios afirmaban sentirse a gusto o contentos jugando el videojuego desarrollado para ellos.

Los resultados del análisis cualitativo muestran en general que los estudiantes tuvieron un nivel de satisfacción alta con respecto al uso del videojuego principalmente en los estudiantes de primer, segundo y tercer grado. Los estudiantes además brindaron sugerencias para poder realizar el videojuego más atractivo y mostraron comentarios positivos sobre el videojuego.

5.4 Discusión de los resultados

En primer lugar, con respecto a la metodología implementada para el desarrollo del videojuego, se siguieron los procedimientos establecidos en Méndez (2006) en donde se estableció una población de metodologías de desarrollo de software para luego reducir esta población en función al producto específico de software que se desea desarrollar que en este caso es un videojuego. Para reducir la población de metodologías, se recopiló las metodologías aplicadas en los papers de investigación de videojuegos en donde las más resaltantes fueron la metodología SCRUM en representación de las metodologías ágiles y metodología en cascada en presentación de las metodologías tradicionales. Por último, a partir de la matriz de evaluación propuesta en Méndez (2006) se obtuvo como resultado que la metodología que más se ajustaba al presente proyecto de investigación fue la metodología en cascada.

La metodología en cascada se divide en 5 fase que son la fase de comunicación, planeación, modelado, desarrollo y despliegue. En relación con la primera fase, se hizo una recopilación de los requerimientos del docente llegando en total a recopilar 17 requerimientos de los cuales fueron 13 funcionales y 4 no funcionales. En relación a la fase de planeación se realizó un cronograma de trabajo para establecer el tiempo establecido para cada fase de la metodología. Para la primera y segunda fase se estableció un tiempo de 1 semana. Para la fase de modelado y construcción se estableció un tiempo de 22 y 21 semanas respectivamente dado que son las fases que involucran un mayor número de actividades y para la fase de despliegue se estableció un tiempo de 2 días. En total, el tiempo para desarrollar todo el prototipo fue de 45 semanas con 2 días. Con respecto a los resultados obtenidos de la tercera fase de modelado se realizaron en total 6 diferentes casos de uso para modelar el comportamiento del prototipo con respecto a su interacción con el estudiante. Además, se realizaron 8 bosquejos diferentes y 8 diseño de arquitecturas de las principales escenas del videojuego, se descargaron en total 40 modelos en 3D diferentes, se extrayeron la música y sonidos de 5 links diferentes de la plataforma de Youtube y se seleccionaron 72 lecturas diferentes que el videojuego iba a contener. En relación a los resultados de la fase de desarrollo, se construyeron en total 431 diferentes escenas y 107 diferentes scripts para darle al videojuego un comportamiento determinado. Se desarrollaron en total 12 diferentes niveles para el videojuego en donde cada nivel se introdujo un terreno diferente. También se implementaron 3 tecnologías diferentes que fueron de realidad virtual no inmersiva, base de datos en tiempo real y servicio de hosting. Para darle un mayor realismo al videojuego se implementó la técnica de *Deep Reinforcement Learning* en donde se entrenaron 7 modelos de DRL de los cuales se escogió el mejor agente entrenado el cual fue

implementado dentro de cada nivel del videojuego. Por otro lado, se implementaron 3 técnicas diferentes para que el estudiante pueda ejercitar su nivel de comprensión lectora. En relación con la primera técnica, se introdujeron textos dentro del videojuego a través de la narración de una historia y la aparición de oraciones. Con respecto a estas oraciones se introdujeron 21 diferentes oraciones. La segunda técnica fue la implementación de lecturas dentro del videojuego de forma que se introdujeron 72 lecturas diferentes siendo 12 para cada grado y por último una sección de estrategias de lectura en donde los estudiantes pudieron visualizar diferentes estrategias para poder aplicarlas dentro del videojuego pero también en sus clases virtuales. Además, se implementaron 51 diferentes archivos de sonidos y música y 6 menús diferentes para cada grado. Por último, se hizo la entrega correspondiente del videojuego a los docentes y estos dieron su aprobación para poder utilizar el videojuego dentro de sus clases virtuales. No se detectaron errores con respecto a la visualización del contenido o reproducción de las escenas del videojuego.

En segundo lugar, los resultados obtenidos de los experimentos de DRL demuestran que el modelo de DRL de Expósito (2018) fue el modelo menos eficiente dado que llegó a un *Cumulative Reward* mayor de un poco más de 15 pero luego este valor empezó a disminuir hasta llegar a 11.28 manteniéndose constante en las primeras 100 000 iteraciones para mantenerse así durante todo el entrenamiento. Este resultado demuestra que el agente ya no tuvo suficiente capacidad para entender la tarea que se le fue asignada por lo que mantuvo un *Cumulative Reward* casi constante durante todo el entrenamiento. Los resultados del comportamiento en la gráfica de *Cumulative Reward* tienen un impacto en el resto de las gráficas provenientes del Tensorboard. En el caso del *Episode Length* que muestra el promedio de acciones realizada por el agente durante un periodo de entrenamiento, para este mismo modelo fue en aumento hasta llegar a un valor de 12.3 manteniéndose de igual manera casi constante durante todo el entrenamiento. La gráfica de *IsTraining* mantuvo un valor de 1 debido a que el modelo no fue detenido en ningún momento. La tendencia del *Policy Loss* tuvo a mantener constante pasando las 200 000 iteraciones llegando a un valor de un poco menor a 0.1317. El *Value Loss* llegó a un valor $7.3078e-3$ en las 100 000 primeras iteraciones y se mantuvo en este valor formando casi una línea recta. Esta gráfica del *Value Loss* tuvo este comportamiento debido al comportamiento del *Cumulative Reward*. Dado que el *Cumulative Reward* tuvo un valor casi constante, el modelo no tuvo el suficiente poder predictivo para tener un aproximado del nuevo *Reward* obtenido por lo que el *Value Loss* se mantuvo constante luego de las 100 000 iteraciones. El valor de *Entropy* al final del entrenamiento fue de $1.3213e-$

5 manteniéndose casi constante de igual manera luego de las 100 000 iteraciones. La entropía representa la aleatoriedad de las acciones del agente, si el *Reward* obtenido se mantuvo casi constante durante el entrenamiento quiere decir que el agente no ha interactuado a profundidad con el ambiente de entrenamiento por lo que la gráfica de *Entropy* tiende de igual manera a ser constante. El valor del *Extrinsic Reward* al final el entrenamiento fue de 11.28 igual al *Cumulative Reward* esto se debe a que la recompensa que va generando el ambiente de entrenamiento pasan al agente por lo que los valores van a tender a ser iguales. Por otro lado, el valor del *Extrinsic Value Estimate* fue de 1.616 llegando a un valor de 2 a inicios del entrenamiento y luego de las 100 000 iteraciones llegando al valor de 1.616 mencionado anteriormente y manteniéndose constante durante el entrenamiento. Por último, el *Learning rate* llegó a un valor de $2.2117e-6$ al final de todo el entrenamiento. Todo el entrenamiento de este modelo de Expósito (2018) tuvo una duración de 11h 52m 9s y realizando un total de 1 millón de iteraciones.

Con respecto al modelo de Adamsson (2018), este presentó un *Cumulative Reward* de 17.01 el cual es una valor aceptable y tuvo un comportamiento muy estable en donde el *Reward* iba de menos a más. Luego de las 200 000 iteraciones el *Cumulative Reward* empezó a ser casi constante con pequeños incrementos en su valor. El resultado final del *Episode Length* fue de 18 que representa el promedio de acciones realizadas por el agente hasta completar un episodio. La gráfica de *IsTraining* se mantuvo con un valor de 1 durante todo el entrenamiento. Esto es debido a que no se realizaron interrupciones para el entrenamiento de este agente en particular. La gráfica del *Policy Loss* tendió a decrecer antes de las 100 000 iteraciones y llegó a un valor final de 0.02487 que también es un valor aceptable dado que este tiene que ser de preferencia cercano a cero. La gráfica del *Value Loss* tendió a ir en aumento hasta llegar de igual manera a las 100 000 iteraciones para luego empezar a decrecer hasta llegar a un valor de 8.949 al final del entrenamiento. La gráfica de *Entropy* empezó en un valor superior a 1.8 y fue decreciendo hasta llegar a un valor final de 0.6832 luego de 200 000 iteraciones. La tendencia de la gráfica de *Extrinsic Reward* fue ir en aumento hasta superar las 200 000 iteraciones para luego mantenerse constante hasta llegar a un valor final de 17.01. La gráfica del *Extrinsic Value Estimate* se mantuvo incrementándose hasta el final del entrenamiento llegando a un valor final de 5.666. En último lugar, la gráfica de *Learning Rate* tuvo un comportamiento decreciente hasta el final del entrenamiento y fue de $3.9357e-6$. El tiempo total utilizado para entrenar este modelo fue de 3h 29m 5s llegando a realizar un total de 270 000 iteraciones. En general este

modelo ha sido estable llegando a un *Cumulative Reward* bastante considerable en un corto periodo de tiempo de entrenamiento.

En relación con el modelo de Ropilo (2019), este fue uno de los dos mejores modelos de este proyecto de investigación dado que obtuvo un *Cumulative Reward* bastante elevado que fue de 31.13 al finalizar del entrenamiento. No obstante, la tendencia de esta gráfica de *Cumulative Reward* presenta muchas subidas y bajadas de este valor lo cual indica inestabilidad en el aprendizaje del agente. A diferencia del resto de modelos, este ha sido el único que ha presentado este tipo de comportamiento, pero pese a la inestabilidad llegó a un valor elevado de *Cumulative Reward*. De igual manera la gráfica del *Episode Length* llegó a un valor elevado de 32.1 de igual modo presentando inestabilidad en la tendencia de la gráfica con subidas y bajadas durante todo el entrenamiento. La gráfica de *IsTraining* se mantuvo en un valor de 1 durante todo el aprendizaje del agente dado que el modelo no fue interrumpido en ninguna iteración. La gráfica de *Policy Loss* tendió a decrecer hasta llegar a las 50 000 iteraciones y a lo largo del entrenamiento también presentó inestabilidad en esta gráfica llegando a un valor final de 0.06363. La gráfica del *Value Loss* tuvo un incremento normal sin inestabilidad, pero cuando llegó a su valor máximo para luego decrecer presentó inestabilidad antes de llegar a las 100 000 iteraciones manteniéndose estos incrementos y decrementos a partir de las 100 000 iteraciones hasta el final del entrenamiento llegando a un valor final de 2.244. La gráfica de *Entropy* tuvo un comportamiento decreciente de acuerdo con la teoría, pero de igual manera presentó pequeñas subidas y bajadas de la entropía durante el entrenamiento hasta llegar a un valor final de 0.2178. La gráfica de *Extrinsic Reward* al ser similar a la de *Cumulative Reward* fue en aumento, pero presentó de igual manera inestabilidad durante el aprendizaje del agente hasta llegar a un valor de 31.13 al finalizar el entrenamiento. La gráfica de *Extrinsic Value Estimate* tiende a ser creciente y casi constante hasta llegar a las 100 000 iteraciones, pero luego continúa el crecimiento, pero de nuevo presentando subidas y bajadas en el valor hasta llegar al final del entrenamiento obteniendo un valor de 14.18. El *Learning Rate* se mantuvo en constante disminución a lo largo del entrenamiento sin presentar ningún tipo de inestabilidad en la actualización del valor llegando a $3.499e-7$ finalizando el entrenamiento. Este modelo en general posee valores finales bastante aceptables, pero, la forma en cómo se ha llegado a estos valores no es la adecuada dado que durante todo el proceso el agente no ha tenido un aprendizaje adecuado, partiendo de menos a más, sino que ha presentado inestabilidad a lo largo de su aprendizaje teniendo muchos picos de subidas y bajadas. El tiempo total de entrenamiento fue de 6h 54m 17s y se realizó en total 500 000 iteraciones.

En cuanto a la propuesta 1 de modelo de DRL, se obtuvo un *Cumulative Reward* de 17.75. En las primeras 50 000 iteraciones, el *Cumulative Reward* tendió a ir en aumento, pero, luego de esa cantidad de realizada de iteraciones empezó a tener un valor casi constante llegando al 17.75 al final el entrenamiento. La gráfica de *Episode Length* tuvo un comportamiento similar tendiendo a incrementarse en las primeras 50 000 iteraciones y llegándose a estabilizar superando esta cantidad de iteraciones llegando a un valor final de 18.75. La tendencia de la gráfica de *IsTraining* ha sido siempre la línea recta manteniéndose en el valor de 1 lo que significa que el entrenamiento de este agente no ha sido interrumpido. La gráfica de *Policy Loss* se mantuvo casi constante desde un inicio y no presentó un decremento significativo a lo largo del entrenamiento llegando a un valor final de 0.1318. La gráfica del *Value Loss* presentó un incremento mínimo al iniciar el entrenamiento, pero luego fue decreciendo hasta llegar a un valor final de 0.09728. La gráfica de *Entropy* presentó una disminución durante todo el entrenamiento hasta antes de llegar a las 100 000 iteraciones para luego mantener un valor casi constante superior a 0.4 hasta llegar a un valor final de 0.4305. La gráfica de *Extrinsic Reward* tuvo un comportamiento creciente hasta llegar a las 100 000 iteraciones, pero luego tendió a ser su valor casi constante llegando a un valor final de 17.75. La gráfica de *Extrinsic Value Estimate* demuestra que el valor tendió a incrementarse hasta llegar a superar el valor de 2 en las 100 000 primeras iteraciones para volverse luego casi constante hasta llegar a un valor final de 2.585. Por último, la gráfica de *Learning Rate* se mantuvo en constante decremento durante todo el entrenamiento hasta llegar a un valor final de $6.6706e-6$. Todo el modelo tuvo un tiempo total de entrenamiento de 2h 28m 29s llegando a las 200 000 iteraciones.

En relación con los resultados de la propuesta 2, la gráfica de *Cumulative Reward* tendió a incrementarse hasta casi a las 150 000 iteraciones, luego la gráfica tendió casi a ser constante hasta llegar a un valor de 18.84. La gráfica de *Episode Length* tendió a incrementarse hasta superar las 150 000 iteraciones, pero, luego tendió a ser constante hasta llegar a un valor final de 19.82. La gráfica de *IsTraining* tendió a ser una línea recta horizontal manteniéndose en el valor de 1 durante todo el entrenamiento demostrando que no hubo interrupciones en el aprendizaje del agente. La gráfica de *Policy Loss* tendió a decrecer en las primeras 50 000 iteraciones, pero, luego mantuvo un valor casi constante hasta llegar a un valor final de 0.07273. La gráfica de *Value Loss* tendió a incrementarse muy poco y luego disminuir durante todo el entrenamiento llegando a un valor final de 0.1436. La gráfica de *Entropy* que representa la aleatoriedad de las acciones del agente tendió a disminuir hasta llegar a superar un poco más de las 100 000 iteraciones. Luego de haber realizado esta cantidad de iteraciones, la entropía

mantuvo un valor casi constante llegando a un valor final de 0.456. La gráfica de *Extrinsic Reward* tendió a ir en aumento hasta superar las 150 000 iteraciones hasta llegar a un valor final de 18.84. La gráfica de *Extrinsic Value Estimate* tuvo un incremento hasta llegar a las primeras 50 000 iteraciones y luego obtuvo un valor casi constante hasta llegar a un valor final de 2.626. La gráfica de *Learning Rate* tendió a decrecer de forma constante hasta obtener un valor final de $6.5488e-6$. El tiempo total utilizado para entrenar a este modelo fue de 2h 13m 54s el cual es el menor tiempo de todos los modelos entrenados y se realizaron 200 000 iteraciones.

En referencia a la propuesta 4, se obtuvo un *Cumulative Reward* empezó a ir en aumento hasta superar las 200 000 iteraciones y luego tendió a ser constante hasta llegar a un valor final de 17.14. La gráfica de *Episode Length* tuvo un comportamiento muy similar al *Cumulative Reward* creciendo hasta las 200 000 iteraciones y llegando a un valor final de 18.14. La gráfica de *IsTraining* se mantuvo en un valor de 1 durante todo el entrenamiento debido a que el aprendizaje del agente no se detuvo en ningún momento. La gráfica de *Policy Loss* tendió a decrecer hasta llegar a un aproximado de las 100 000 iteraciones y luego tomó un valor casi constante hasta llegar a un valor de 0.02179. La gráfica de *Value Loss* tendió a incrementarse hasta superar las 200 000 iteraciones y luego empezó a decrecer hasta llegar a un valor final de 8.094. La gráfica de *Entropy* mostró que el valor empezó a decrecer durante todo el entrenamiento hasta llegar a un valor final de 0.7303. La gráfica de *Extrinsic Reward* tendió a ir en aumento hasta superar las 200 000 iteraciones y llegó a un valor final de 17.14. La gráfica de *Extrinsic Value Estimate* muestra que el valor tendió a ir en aumento hasta llegar a un valor de 5.904. Por último, la gráfica de *Learning Rate* muestra que el valor de la tasa de aprendizaje tendió a decrecer durante todo el entrenamiento hasta llegar a un valor final de $4.866e-5$. El tiempo total empleado para entrenar este modelo de DRL fue de 3h 28m 36s y realizando 250 000 iteraciones en total.

El mejor modelo de DRL desarrollado fue la propuesta 3 obteniendo un *Cumulative Reward* de 28.05. Esta gráfica muestra que el valor del *Cumulative Reward* se mantuvo en constante crecimiento sin presentar periodos de inestabilidad. La gráfica de *Episode Length* de igual manera se mantuvo en constante crecimiento hasta llegar a un valor de 29.02 al final del entrenamiento. La gráfica de *IsTraining* se mantuvo constante con un valor de 1 en todo entrenamiento debido a que el modelo no fue interrumpido en ninguna iteración. La gráfica de *Policy Loss* muestra que este valor empezó en el valor de 0.1 y comenzó a disminuir hasta las primeras 50 000 iteraciones hasta llegar a un valor superior a 0.04 manteniéndose casi constante

hasta llegar a un valor final de 0.05389. La gráfica de *Value Loss* muestra un incremento de este valor hasta llegar a un valor de 10 en las primeras 20 000 iteraciones para luego decrecer hasta antes de llegar a las 100 000 iteraciones para luego mantener un valor casi constante hasta llegar a un valor de 0.475. La gráfica de *Entropy* durante las 150 000 primeras iteraciones tendió a decrecer su valor hasta llegar a un valor aproximado de 0.2 para luego mantenerse casi constante hasta llegar a un valor final de 0.1862. La gráfica de *Extrinsic Reward* muestra que este valor tendió a estar en constante crecimiento durante todo el entrenamiento hasta llegar a un valor final de 28.05 al final del entrenamiento. La gráfica de *Extrinsic Value Estimate* muestra que este valor tendió a incrementarse hasta las 150 000 iteraciones para luego mantenerse casi constante hasta llegar a un valor final de 6.39. Por último, la gráfica de *Learning Rate* muestra que este valor estuvo decreciendo constantemente hasta el final del entrenamiento hasta llegar a un valor final de $1.1743e-5$. El tiempo total utilizado para entrenar a este modelo fue de 3h 29m 40s y se realizó un total de 250 000 iteraciones.

En tercer lugar, en relación con los resultados del análisis estadístico, los estudiantes del primer grado de primaria demostraron que en el grupo de control el 72% de los alumnos lograron mejorar su nivel de comprensión literal, un 36% mejoraron su nivel de comprensión inferencial y un 18% mejoraron su nivel crítico de comprensión lectora. Por otro lado, los estudiantes del grupo experimental de este mismo grado demostraron que el 77% de los alumnos pudieron mejorar su nivel de comprensión literal, un promedio del 55% lograron mejorar su nivel inferencial y 22% de los estudiantes mejoraron su nivel crítico.

Con respecto al segundo grado de primaria, el 75% de los estudiantes que formaron parte del grupo de control pudieron mejorar su nivel de comprensión literal, el 50% pudieron mejorar su nivel de comprensión literal y el 25% de los alumnos mejoraron su nivel de comprensión crítico. Por otro lado, los estudiantes que formaron parte del grupo experimental y que entrenaron a través del videojuego mejoraron un 80% de este grupo su nivel de comprensión literal, un 70% de estos estudiantes pudieron mejorar su nivel de comprensión inferencial y el 30% pudieron mejorar su nivel de comprensión crítico.

En relación con los estudiantes del tercer grado de educación primaria, el 65% de los alumnos pertenecientes al grupo de control demostraron una mejora en su nivel de comprensión literal, el 66% de los estudiantes del mismo grupo demostraron un mejor desempeño en su nivel inferencial y el 55% mostraron una mejora en el nivel crítico. Por otra parte, los estudiantes del grupo experimental mostraron un 81% de estos una mejora en el nivel literal de la comprensión

lectora. Además, un 63% de este grupo mostró un mejor desenvolvimiento en su nivel de comprensión inferencial y un 54% de estos alumnos mostró una mejora en su nivel de comprensión crítico.

Acerca de los alumnos del cuarto grado de primaria pertenecientes al grupo de control, el 85% de los estudiantes mostraron una mejora en su nivel de comprensión literal. También, el 71% de estos alumnos demostraron un mejor desenvolvimiento en su nivel de comprensión inferencial y el 72% de los alumnos mostraron una mejora en el nivel crítico de comprensión lectora. Por otro lado, los estudiantes pertenecientes al grupo experimental presentaron el 89% de estos una mejora en su nivel de comprensión literal, un 78% mostraron una mejora un su nivel inferencial de comprensión lectora y el 55% de los alumnos mostraron una mejora en su nivel crítico.

Sobre los estudiantes del grupo de control del quinto grado, el 88% de estos mostraron una mejora en su nivel literal de comprensión lectora. Además, el 77% de los estudiantes de este grupo logró obtener una mejora en su nivel inferencial y el 67% lograron mejorar en su nivel crítico de comprensión lectora. Por otra parte, el 90% de los alumnos del grupo experimental que entrenaron a través del videojuego mostraron una mejora en su nivel literal de comprensión lectora, el 80% pudieron mejorar su nivel inferencial de comprensión lectora y el 60% de este grupo obtuvieron una mejora en el nivel crítico.

Finalmente, en relación con los estudiantes del grupo de control pertenecientes al sexto grado de primaria, el 80% de los alumnos obtuvieron una mejora en su nivel literal de comprensión lectora. Además, el 60% de los estudiantes de primaria mostraron una mejora en su nivel inferencial de comprensión lectora y el 70% de los alumnos pudieron obtener una mejora en su nivel crítico de lectura. Por otro lado, el 88% de los estudiantes pertenecientes al grupo experimental pudieron obtener una mejora en su nivel literal de comprensión lectora. También, el 78% de los alumnos de este mismo grupo pudo obtener una mejora en su nivel inferencial de lectora y el 66% de los alumnos pudo obtener un mejor desenvolvimiento en su nivel crítico de comprensión lectora.

CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES

En el presente capítulo, se presentan las conclusiones y recomendaciones del proyecto de investigación.

6.1 Conclusiones

A partir de los hallazgos encontrados, se evidencia que el uso del videojuego desarrollado pudo mejorar el nivel de comprensión lectora en los estudiantes de educación primaria. Los alumnos del grupo experimental de cada grado mostraron al final del entrenamiento una mejora significativa en su nivel de comprensión lectora en comparación de los alumnos del grupo de control. Los resultados de la mejora encontrada por cada grado ordenados de mayor a menor se presentan en la Tabla 89.

Tabla 89: Resumen de las mejoras significativas encontradas en el nivel comprensión lectora de los estudiantes de primaria ordenados de mayor a menor

Grado	Mejora significativa
1°	1.77 puntos
2°	1.45 puntos
5°	1.34 puntos
6°	1.17 puntos
4°	1.17 puntos
3°	1.06 puntos

Fuente: Elaboración Propia.

Los resultados presentados en la Tabla 89 demuestran que se ha cumplido con el objetivo general (O.G) del proyecto el cual fue desarrollar un videojuego que permita mejorar el nivel de comprensión lectora en estudiantes de primaria y permiten, además, aceptar la hipótesis general (H.G) planteada que establece que el uso del videojuego desarrollado posibilita la mejora del nivel de comprensión lectora en estudiantes de primaria. Los estudiantes de primer, segundo y quinto grado presentaron un mejor desenvolvimiento a través del uso del videojuego obteniendo una mayor mejora en su nivel de comprensión lectora que fue específicamente de 1.77 puntos, 1.45 puntos y 1.34 puntos respectivamente.

De igual manera, los resultados de la Tabla 89 guardan relación con lo que se sostiene en Castro (2015), Huerta (2017), Moreno, Vahos, y Mazo (2019) y Cheng y Tsai (2019) quienes afirman que la implementación de un videojuego dentro de un salón de clases posibilita la mejora de diversas habilidades en desarrollo de los estudiantes, en este caso en particular de la comprensión lectora. Estos autores expresan además que la implementación de un videojuego en un salón de clase no solamente posibilita la mejora de una determinada habilidad sino

también el aprendizaje de un tema que para los estudiantes se les resulta difícil entender de un determinado curso. Lo anterior mencionado, es acorde a lo que este estudio plantea dado que en este proyecto de investigación se tocó el tema de la lectura comprensiva que forma parte del curso de comprensión lectora.

No obstante, es preciso resaltar que aquellos autores implementaron sus prototipos en un contexto diferente al presentado en este proyecto. Este trabajo de investigación fue realizado bajo la pandemia del Covid-19 y que dado el peligro que existe de contagio entre las personas, los colegios tuvieron que replantear su manera de enseñanza y cambiar de una metodología tradicional a una nueva metodología virtual. Dado que la forma de enseñanza ha sido diferente a un contexto normal de enseñanza presencial, existe la posibilidad que el impacto de las clases virtuales y el uso del videojuego para mejorar el nivel de comprensión lectora en los estudiantes no haya sido tan elevado en comparación si se hubiera seguido una metodología presencial de enseñanza. La idea de este proyecto de investigación fue utilizar las tecnologías existentes de tal forma de crear un nuevo producto que pueda ser jugado desde el hogar de los estudiantes de tal manera que el aprendizaje sea de acuerdo con lo establecido por el colegio y por el gobierno peruano.

Además, los resultados del análisis estadístico cuantitativo de las notas de los estudiantes demuestran que ambos grupos de cada grado obtuvieron una mejora en los tres niveles de la comprensión lectora. Con respecto a los grupos de control, el 70% de los alumnos de cada grado tuvieron una mejora en su nivel inferencial, un 60% obtuvo una mejora en su nivel literal y con respecto al nivel crítico se obtuvo un mejor desenvolvimiento en las respuestas de los estudiantes de grados mayores dado que el 70% de estos pudieron contestar adecuadamente estas preguntas en comparación con los primeros grados donde solamente el 30% de los estudiantes pudieron tener un mejor juicio al analizar y responder estas preguntas del nivel crítico.

No obstante, los alumnos pertenecientes al grupo experimental de cada grado y que entrenaron a través del uso del videojuego tuvieron mejores resultados principalmente en su nivel literal e inferencial de comprensión lectora. Los resultados demuestran que más del 80% de los estudiantes de cada grado tuvieron una mejora en los niveles literal e inferencial de la comprensión lectora de modo que las respuestas correctas de los estudiantes en las preguntas de estos dos niveles fueron mayores a las del pretest. Una de las causas que podría explicar estos resultados es que el videojuego contiene principalmente preguntas de nivel literal donde

los estudiantes tienen que recordar aspectos puntuales que contiene explícitamente la lectura de tal manera de poder responder correctamente todas las preguntas y de igual manera contiene preguntas de nivel inferencial dentro de las preguntas de opción múltiple como qué otro título se le podría poner a la lectura o qué podría significar un determinado concepto dentro de la lectura. Además, el videojuego narra una historia manteniendo al alumno dentro de un contexto con oraciones apareciendo en cada nivel que complementan esta narración del inicio por lo que lo que los estudiantes mantienen su nivel literal principalmente en constante entrenamiento. Se detectó además un incremento de estudiantes que pudieron redactar de una mejor manera las respuestas para las preguntas de nivel crítico. Un aproximado del 20% de los estudiantes de los primeros grados pudieron contestar adecuadamente estas preguntas y un aproximado del 60% de los alumnos de los estudiantes de grados mayores pudieron responder correctamente las preguntas de este nivel.

Los resultados descritos permiten concluir que el uso del videojuego tuvo un mayor impacto en el nivel literal e inferencial de la comprensión lectora de los estudiantes del grupo experimental pero fue menor en el nivel crítico en comparación al grupo de control debido a que el videojuego no tiene secciones para poner práctica este nivel mientras que en las clases tradicionales mediante la lectura de los estudiantes y el intercambio de ideas se puede ejercitar más este nivel en los alumnos de primaria.

Con respecto al primer objetivo específico (O.E 1) en donde se planteó identificar las tecnologías y técnicas de Inteligencia Artificial que se integran para el desarrollo funcional del videojuego, se identificaron las tecnologías aplicadas en los antecedentes de investigación y se adaptaron al presente proyecto. Las tres tecnologías seleccionadas fueron la base de datos en tiempo real de Firebase, el servicio de hosting de Firebase y la realidad virtual no inmersiva. El primero fue con el objetivo de poder registrar los datos de los estudiantes y comprobar que estén jugando el videojuego dado que al establecer una educación a distancia se pierde el control de lo que los estudiantes realizan desde sus hogares. Esta es una de las principales consecuencias de la educación a distancia, pero, estableciendo este tipo de educación, se protege la salud tanto de docentes como de los estudiantes. La segunda tecnología aplicada fue con el objetivo de que el videojuego pueda ser jugado desde cualquier computadora con conexión a internet y que el prototipo pueda ser compartido de una manera simple con los estudiantes. De este modo, se protege tanto al alumnado como a los docentes del Covid-19 dado que el videojuego lo jugaron desde sus hogares evitando reuniones de forma presencial para realizar el entrenamiento. Por último, la tecnología de realidad virtual no inmersiva fue

utilizada para poder controlar este videojuego a través del teclado y mouse de los estudiantes que son periféricos más comunes que ellos utilizan para sus clases a distancia.

En relación con la técnica de Inteligencia Artificial aplicada, se implementó la técnica de *Deep Reinforcement Learning* cuyo objetivo fue hacer del videojuego más atractivo y realista entrenando a una Inteligencia Artificial enemiga para que pueda combatir con los usuarios y quitarles vida para que los estudiantes puedan ser redirigidos a las lecturas del videojuego. En la mayoría de los comentarios de los estudiantes en la encuesta de satisfacción afirmaron que les había parecido más agradable los enemigos que tenía el prototipo y que en lo posible se puedan incluir muchos más en cada nivel lo que da indicios que el comportamiento del agente entrenado para perseguir y atacar a cada estudiante fue la correcta.

Para implementar la técnica de *Deep Reinforcement Learning (DRL)*, se tuvo en consideración los modelos de DRL de Expósito (2018), Ropilo (2019) y Adamsson (2018) de tal manera que se pudieron establecer 4 nuevos modelos de *Deep Reinforcement Learning* para poder entrenar en total 7 modelos de DRL (3 modelos de los autores mencionados anteriormente y 4 nuevos modelos de DRL). Dado que los valores de los hiperparámetros de los modelos de estos 3 autores fueron plasmados en función a la consulta de expertos en este campo de Inteligencia Artificial, estos valores fueron reutilizados ordenándolos de menor a mayor para crear los 4 nuevos modelos de DRL. El optimizador utilizado fue el algoritmo de *Proximal Policy Optimization (PPO)* tal cual como lo utilizaron los 3 autores mencionados anteriormente. Este algoritmo permitió mantener a la *Policy* que va generando el agente dentro de un intervalo de tal manera de poder disminuir el costo computacional aligerando el entrenamiento del agente.

Los modelos de DRL con un *Cumulative Reward* más elevado fue el modelo de Ropilo (2019) y la propuesta 3 llegando a un valor de 31.13 y 28.05 respectivamente. No obstante, el modelo de Ropilo (2019) presento una inestabilidad notable durante todo el proceso de entrenamiento, pero la propuesta 3 mostró un aprendizaje creciente sin problemas de inestabilidad y además fue más eficiente debido a que en 250 000 iteraciones el modelo de DRL de la propuesta 3 llegó a un *Cumulative Reward* de 28.05 a comparación del modelo de Ropilo (2019) que se demoró 500 000 iteraciones. Adicionalmente, el tiempo de entrenamiento de la propuesta 3 fue inferior demorándose un aproximado de 3 horas y media en comparación con el modelo de Ropilo (2019) que se demoró casi 7 horas. También, los valores del *Policy Loss*, *Value Loss* y *Entropy* fueron muy cercanos a cero en el caso de la propuesta 3 en

comparación Ropilo (2019). Debido al análisis de los resultados mencionados anteriormente se llegó a la conclusión que el mejor modelo de DRL fue la propuesta 3 por presentar mayor estabilidad y eficiencia durante todo el proceso de entrenamiento del agente.

Como se mencionó anteriormente, cada tecnología y técnica de Inteligencia Artificial identificada e implementada tuvieron un impacto en la funcionalidad del sistema desarrollado. En la Tabla 90, se presenta el resumen de las tecnologías y técnicas de Inteligencia Artificial implementadas y su impacto dentro del videojuego.

Tabla 90: Impacto en funcionalidad del sistema de las tecnologías y técnicas de Inteligencia Artificial implementadas

Nombre	Tipo	Impacto en funcionalidad del sistema
Base de datos en tiempo real	Tecnología	Registro de los alumnos que ingresan al videojuego y acceso al sistema.
Realidad virtual no inmersiva	Tecnología	Control del videojuego a través de los periféricos de mouse y teclado.
Servicio de Hosting	Tecnología	Visualización del videojuego desde la computadora de los hogares de los alumnos de primaria.
Deep Reinforcement Learning	Técnica de Inteligencia Artificial	Desarrollo de enemigos inteligentes que combaten con los usuarios y brindan un mayor grado de realismo al sistema.

Fuente: Elaboración Propia.

Como se aprecia en la Tabla 90, la integración de cada tecnología y técnica de Inteligencia Artificial tuvieron un impacto particular dentro del videojuego posibilitando un correcto comportamiento del videojuego y permitió darle características especiales al prototipo desarrollado lo que permite aceptar la primera hipótesis específica (H.E 1) dado que la integración de las tecnologías y técnicas de Inteligencia Artificial implementadas posibilitaron la correcta funcionalidad del videojuego.

En cuanto al segundo objetivo específico (O.E 2) en donde se propone obtener el porcentaje de aceptación del sistema desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria, se tomó a los alumnos que utilizaron el sistema una encuesta de satisfacción cuyos resultados demostraron que el prototipo desarrollado fue del agrado de los 6 grados participantes y mucho mejor en los grados de primero, segundo y tercero de primaria.

El 71% de los participantes que usaron el videojuego afirmaron que el videojuego les había parecido bueno, el 25% sintió que la experiencia que tuvieron con el videojuego fue regular y solamente el 3% opinaron que el videojuego no fue de su agrado. Los resultados obtenidos permiten aceptar la segunda hipótesis específica (H.E 2) en donde se afirmó que el porcentaje de aceptación del prototipo desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria iba a ser mayor a 50% dado que el 71% de los participantes afirmaron que el videojuego fue de su agrado.

En relación con el tercer objetivo específico (O.E 3), en donde se propuso validar estadísticamente la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención, se elaboró un resumen de las técnicas estadísticas aplicadas en los antecedentes de investigación (Tabla 18) en donde se resaltó la aplicación de la Prueba T de student en dichos trabajos. Dado que existen dos tipos de Prueba T de student (Prueba T para muestras dependientes y prueba T para muestras independientes), se siguió el criterio de selección de prueba estadística planteado en Soto (2018). Los resultados del análisis mostraron que la prueba T de Student para muestras independientes era la prueba necesaria para detectar la mejora significativa a través de las notas de los estudiantes.

Para la aplicación de la técnica estadística de Prueba T de Student para muestras independientes se realizaron una serie de actividades y procedimientos matemáticos para poder detectar la mejora significativa para cada grado de educación primaria. Los resultados de la aplicación de la prueba T de Student se presentaron en la Tabla 87 en donde se visualizó la mejora significativa encontrada para cada grado por lo cual se comprobó la tercera hipótesis específica (H.E 3) dado que realizando una validación estadística a través de la técnica de Prueba T de Student permitió detectar la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención.

La aplicación de esta prueba T de Student en el pretest demostró que no existía una diferencia estadística en el grupo de control y grupo experimental antes de realizar la intervención lo que evidencia, desde un punto de vista estadístico, que ambos grupo de cada grado parten en las mismas condiciones. Es decir, que no hay un grupo ya sea de control o experimental que sea superior al otro grupo sino que ambos poseen conocimientos similares dentro del curso de comprensión lectora. Sin embargo, luego de las 5 semanas de entrenamiento se demostró que existe una superioridad de los grupo experimentales en su nivel de comprensión lectora en comparación con los estudiantes del grupo de control. De tal modo que

se pudo comprobar estadísticamente que el uso del videojuego había tenido una mayor impacto en el nivel de la comprensión lectora en los estudiantes que utilizaron el prototipo desarrollado que los estudiantes que siguieron una metodología tradicional.

Por último, en relación con el cuarto objetivo específico (O.E 4) se desarrolló un procedimiento pedagógico y tecnológico para complementar las clases tradicionales en los estudiantes de primaria mediante el uso del videojuego desarrollando un protocolo de aprendizaje en cual consistió en 5 semanas de entrenamiento en donde los estudiantes tuvieron 35 minutos clases teóricas y 10 minutos el uso del videojuego. Al inicio del protocolo de aprendizaje se tomó a los estudiantes el pretest y al finalizar el postest junto con la encuesta de satisfacción. Para la realización del protocolo de aprendizaje se siguieron los protocolos establecidos en Cheng y Tsaib (2019) y Didehbani et al. (2016) cuyos procedimientos pedagógicos fueron de normalmente 2 a 3 semanas de entrenamiento y el uso del videojuego fue por un periodo de 15 minutos el uso del videojuego. Al poseer un mayor número de semanas de entrenamiento en el presente proyecto, se redujo el tiempo de uso del videojuego de tal manera de encontrar una proporción adecuada durante el entrenamiento de los alumnos.

Mediante el desarrollo y aplicación del protocolo de aprendizaje se corroboró la cuarta hipótesis específica (H.E 4) en donde se estableció que el procedimiento pedagógico y tecnológico desarrollado permitirá distribuir el tiempo destinado a las clases tradicionales y el tiempo dedicado al uso del videojuego debido a que se logró distribuir adecuadamente el tiempo total de enseñanza de tal manera que los 35 minutos iniciales fueron destinados a las clases tradicionales y los últimos 10 minutos el uso del videojuego. La idea de implementar un procedimiento pedagógico y tecnológico mediante un protocolo de aprendizaje o de entrenamiento tal cual como sugiere Didehbani et al. (2016) es que el videojuego no opaque la labor del docente sino más bien que complemente las clases del educador. El videojuego desarrollado tiene dos principales funciones en el presente proyecto de investigación. La primera es contribuir con el aprendizaje de los estudiantes de educación primaria en el tema de la lectura comprensiva de forma que puedan aquellos mejorar su nivel de comprensión lectora a través del uso del videojuego y la segunda función es que el videojuego sirva como una herramienta tecnológica adicional para el docente.

6.2 Recomendaciones

Para hacer del videojuego más atractivo, es aconsejable poder recabar la opinión de los propios estudiantes de cómo ellos desearían que el videojuego sea dado que ellos serán los que

jugarán este videojuego. Realizando lo anterior mencionado, es posible que los resultados de la encuesta de satisfacción sean más favorables y se pueda incrementar el porcentaje de aceptación del videojuego por parte del alumnado. Además, siguiendo las sugerencias planteadas por los propios estudiantes en la encuesta de satisfacción, se sugiere introducir más niveles dentro del videojuego, así como la construcción de una tienda virtual en donde el estudiante pueda intercambiar su score obtenido por objetos que puedan otorgarle nuevas habilidades dentro del videojuego y así el alumno pueda estar más motivado. Es preciso resaltar que el score construido dentro de este videojuego va incrementándose, pero solo sirve para darle un mejor aspecto visual al prototipo construido, pero no se puede realizar ningún intercambio con este score obtenido.

Con respecto al desarrollo del protocolo de aprendizaje, se sugiere mantener la proporción de las clases tradicionales y el uso del videojuego establecido por Cheng y Tsaib (2019) y Didehbani et al. (2016) en donde se establece un tiempo considerable para las clases tradicionales y el uso del videojuego en los últimos 10 o 15 minutos finales de la clase de modo que el videojuego sirva como una herramienta adicional para el docente de modo que pueda hacer sus clases más didácticas pero que a su vez sirva para mejorar el nivel de comprensión lectora en los estudiantes.

Por otro lado, se sugiere aprovechar el uso de la base de datos en tiempo real de Firebase. En este proyecto solo se utilizó la base de datos para el registro de información de los alumnos y para que estos accedan al videojuego, pero también se podrían crear nuevos formularios para poder obtener más información de los involucrados del proyecto. Por ejemplo, cuántas veces al día suelen leer una lectura, si les gusta leer, si viven con papá y mamá o viven solo con uno de ellos, etc. De tal modo que se puedan obtener resultados más profundos luego del entrenamiento. Por ejemplo, de que los niños que viven con papá y mamá presentaron mejor desenvolvimiento que los estudiantes que vienen de una familia disfuncional. Al obtener este tipo de resultados se estarían detectando nuevas variables que estén impactando en el nivel de comprensión lectora de los estudiantes lo que podrían dar origen a nuevos proyectos que propongan soluciones para afrontar este problema no solamente en el campo de la tecnología sino también educacional y en el campo de la psicología.

Se sugiere, además, mantener el uso de un servicio de hosting de modo que los estudiantes puedan continuar jugando desde sus hogares mientras continúe la educación a distancia producto de la pandemia del Covid-19. Se resalta que el servicio de hosting de Firebase provee

una capacidad de 1 GB de forma gratuita, pero en caso el videojuego desarrollado exceda esa capacidad, se sugiere poder contratar un servicio de hosting profesional de modo de poder alojar el videojuego en un ambiente con mayor capacidad o ilimitada en función al plan contratado.

Con respecto al tamaño de la muestra, se sugiere poder tener una muestra mayor con la cual poder trabajar de modo que el videojuego pueda llegar a más estudiantes y puedan mejorar su nivel de comprensión lectora. Uno de los principales inconvenientes en este tipo de proyectos es poder encontrar un colegio con el cual poder trabajar dado que depende de la opinión tanto del director como de los docentes. Al momento de iniciar con la búsqueda de un colegio con el cual poder trabajar, se sugiere llevar una demo a través de un video mostrando el inicio del videojuego y cómo podría este ayudar a mejorar el nivel de comprensión lectora en los estudiantes dado que es mejor que tanto el director como docentes puedan ver una primera versión del prototipo en lugar de solo escuchar la idea del proyecto de forma que se pueda convencer a estas personas de poder realizar el proyecto en su centro educativo.

De igual modo, con respecto a la división de la muestra, se sugiere mantener de igual forma un grupo de control y un grupo experimental dado que para verificar que la nueva metodología de enseñanza a través del uso del videojuego es mejor que la metodología tradicional es necesario un grupo que aprenda a través de la forma normal de enseñanza de modo de poder comparar los resultados a través de la prueba T de muestras independientes.

En relación con los experimentos de DRL, siguiendo las sugerencias que también se establecieron en los trabajos de Expósito (2018), Ropilo (2019) y Adamsson (2018) se recomienda no sobrecargar al agente con demasiadas acciones debido a que asignarle al agente muchas tareas que el debe de aprender podría hacer que el aprendizaje se vuelva más lento y también puede darse el caso que se consuma un mayor recurso computacional. De igual modo, los valores para los refuerzos positivos y negativos deben de ser valores pequeños no mayores a 2 debido a que al darle al agente valores muy grandes, este pueda entender que en un determinado estado es cuando obtiene una recompensa mayor y al darle una recompensa muy negativa, el agente podría no realizar ninguna acción debido a que se le está brindando un castigo muy grande obteniendo un *Reward* promedio muy negativo por lo que sin ejecutar ninguna acción su *Reward* podría mantenerse más alto que al realizar una acción.

También se sugiere poder utilizar la versión más actual de la librería de ML-Agents. En este proyecto se utilizó el Release 1 porque era única versión que estaba disponible y se aconseja

también no cambiar de versión en medio del desarrollo del videojuego porque cada versión de la librería en algunos casos elimina procesos que ya no son necesarios o actualizan los scripts que esta librería posee por lo que si se trabaja con una versión de la librería de ML-Agents y se continúa con otra podría que no funcione el entrenamiento de los agentes inteligentes de forma adecuada. Adicionalmente, se sugiere poder profundizar en otros tipos de aprendizaje que también se puede utilizar con la librería de ML-Agents como *Curriculum Learning* o *Imitation Learning* en donde se pueden utilizar otros tipos de redes como redes recurrentes o redes convolucionales pero la elección de la forma de aprendizaje del agente depende qué tipo de videojuego se desee desarrollar.

Asimismo, se sugiere indagar en el componente *Curiosity* que no se implementó en este proyecto debido a que en Expósito (2018), Ropilo (2019) y Adamsson (2018) tampoco se utilizó pero sería ideal poder entrenar a un agente inteligente siguiendo lo planteado en este proyecto de investigación y también agregando este nuevo componente con el objetivo de determinar qué tanto influye dentro del aprendizaje del agente y poder hacer una comparación de ambas formas de aprendizaje. Por último, con respecto a la cantidad de experimentos, se sugiere poder entrenar una mayor cantidad de modelos de DRL con el objetivo de poder determinar el efecto de la variación de un determinado hiperparámetro dentro del aprendizaje de un agente como puede ser el caso del *Batch_Size* que se mencionó en las conclusiones presentadas en la sección anterior de modo de poder detectar qué hiperparámetros tienen más impacto que otros. Estos descubrimientos podrían aportar mucho al campo de *Deep Reinforcement Learning* dado que se estarían brindando nuevos criterios para poder crear nuevos modelos de DRL.

Finalmente, con respecto a la metodología implementada, se sugiere poder seguir la misma secuencia planteada en Méndez (2006) con el objetivo de determinar cuál es la metodología más adecuada para poder aplicar dentro de un proyecto de investigación en particular que se relacione con el desarrollo de software dado que uno de los principales inconvenientes que se suele presentar es que los investigadores aplican una metodología en particular por estar esta de moda o por cuestiones de gustos pero lo que se suele olvidar es que en el fondo una metodología es ese conjunto de actividades que va a permitir al investigador poder obtener los mejores resultados posibles, poder conocer qué actividades involucra cada etapa y qué actividades realiza una determinada persona en el caso de que el equipo sea multidisciplinario por que lo que es un aspecto importante poder establecer un criterio para poder seleccionar la metodología más adecuada y poder así obtener un producto de software de calidad.

Referencias

- Acibeiro, M. (2017). *Qué es el hosting web y para qué sirve*. Recuperado de: <https://es.godaddy.com/blog/que-es-el-hosting-web-y-para-que-sirve/>.
- Adamsson, M. (2018). *Curriculum learning for increasing the performance of a reinforcement learning agent in a static first-person shooter game*. Estocolmo: School of Electrical Engineering and Computer Science (EECS).
- AETecno. (2019). *El crecimiento imparable de la industria gamer de América Latina*. Recuperado de: <https://tecno.americaeconomia.com/articulos/el-crecimiento-imparable-de-la-industria-gamer-de-america-latina>.
- Agoros, P. (2013). *Call Of Duty: Ghosts past-gen multiplayer review (PS3)*. Recuperado de <http://legacyfuture.com/gaming/call-duty-ghosts-review-ps3xbox-360/>.
- Alonso, J. (2001). *Videojuegos 3D*. Barcelona: Universitat Oberta de Catalunya.
- Alonso, R. (2020). *¿Qué es un hosting o alojamiento web?*. Recuperado de: <https://miposicionamientoweb.es/que-es-un-hosting/#:~:text=El%20hosting%20o%20alojamiento%20web,cualquiera%20pueda%20verlo%20en%20Internet>.
- Alvarado, I. (2015). *Redes Neuronales*. Recuperado de: https://ml4a.github.io/ml4a/es/neural_networks/.
- AWS. (2020). *¿Qué es NoSQL?*. Recuperado de: <https://aws.amazon.com/es/nosql/>.
- Barambones, J. (2016). *Realidad Virtual*. Recuperado de: <http://juanbarambones.com/wp-content/uploads/2017/01/realidad-virtual-2016-V1.pdf>.
- BBC News. (2020). *Coronavirus: 6 ilustraciones para enseñar a los niños a protegerse (y para que se entretengan coloreando)*. Recuperado de: <https://www.bbc.com/mundo/noticias-52092558>.
- Bernard, M. (2018). *What Is Deep Learning AI? A Simple Guide With 8 Practical Examples*. Recuperado de: <https://www.forbes.com/sites/bernardmarr/2018/10/01/what-is-deep-learning-ai-a-simple-guide-with-8-practical-examples/?sh=33d65cb8d4ba>.
- Botella, P., Alacreu, M., & Martínez, M. (2012). *Inferencia estadística (intervalos de confianza y p-valor). Comparación de dos poblaciones (test de comparación de medias, comparación de dos proporciones, comparación de dos varianzas)*. Valencia: Universidad Cardenal Herrera.
- Business Wire. (2020). *La industria de los juegos se une para promover los mensajes de la Organización Mundial de la Salud contra COVID-19; Lanzamiento de la campaña #PlayApartTogether*. Recuperado de: <https://www.businesswire.com/news/home/20200329005040/es/>
- Camps, R., Casillas, L., Costal, D., Ginesta, M., Escofet, C., & Pérez, O. (2005). *Bases de datos*. Barcelona: Fundació per a la Universitat Oberta de Catalunya.
- Castro, A. (2015). Maximum Consequencia: un videojuego educativo para ejercitar la comprensión lectora. *Innovación educativa*, 25, 191-206. Barranquilla: Universidad del Norte.

- Cheng, K.-H., & Tsaib, C.-C. (2019). A case study of immersive virtual field trips in an elementary classroom: Students' learning experience and teacher-student interaction behaviors. *Computers & Education, 140*.
- Correo. (2010). *Perú en las pruebas pisa 2009*. Recuperado de: <https://diariocorreop.epe/opinion/peru-en-las-pruebas-pisa-2009-452930/>.
- Diario Maulee. (2020). *El vertiginoso crecimiento de la industria de los videojuegos en Chile*. Recuperado de: <https://www.maulee.cl/el-vertiginoso-crecimiento-de-la-industria-de-los-videojuegos-en-chile/>.
- Didehbani, N., Allen, T., Kandalaf, M., & Krawczyk, D. (2016). Virtual Reality Social Cognition Training for children with high functioning autism. *Computers in Human Behavior, 703-711*.
- Domínguez, P. (2017). *En qué consiste el modelo en cascada*. Recuperado de: <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>.
- Drake, J. (2008). *Proceso de desarrollo de aplicaciones de software*. Recuperado de: https://www.ctr.unican.es/asignaturas/Ingenieria_Software_4_F/Doc/M1_08_Proceso.pdf
- Eguía, J., Contreras, R., & Solano, L. (2012). *Videojuegos: conceptos, historia y su potencial como herramientas para la educación*. Recuperado de: <https://www.3ciencias.com/wp-content/uploads/2013/04/videojuegos.pdf>.
- Expósito, P. (2018). *Inteligencia Artificial de Comportamiento de Agentes no jugadores en un RPG clásico con Unity y TensorFlow*. Badajoz: Universidad de Extremadura.
- G&M News. (2020). *Perú: Un mercado prometedor para los videojuegos y los Esports*.
- García, B. (2009). *Videojuegos: Medio de ocio, cultura popular y recurso didáctico para la enseñanza y aprendizaje de las matemáticas escolares*. Madrid: Universidad Autónoma de Madrid.
- García, P., Mora, A., Castillo, P., & Pérez, I. (2019). A bibliometric study of the research area of videogames using Dimensions.ai database. *Procedia Computer Science, 162, 737-744*.
- Geronazzo, M., Vescovi, D., Nordahl, R., Serafin, S., Avanzini, F., Innocenti, E., & Ludovico, L. (2019). Mobile virtual reality for musical genre learning in primary education. *Computers & Education, 102-117*.
- González, A. (2018). *Neural networks applied to a tower defense video game*. Castellón: Universidad Jaime I.
- Gordillo, A., & Del Pilar, M. (2009). Los niveles de comprensión lectora: hacia una enunciación investigativa y reflexiva para mejorar la comprensión lectora en estudiantes universitarios. *Actualidades Pedagógicas(53)*.
- Granic, I., Lobel, A., & Engels, R. (2014). The Benefits of Playing Video Games. *American Psychologist, 69, 66-78*. Recuperado de: <https://www.apa.org/pubs/journals/releases/amp-a0034857.pdf>

- GreeLane. (2020). *Aprenda qué es un grupo de control, cómo funciona y por qué puede ser valioso*. Recuperado de: <https://www.greelane.com/es/ciencia-tecnolog%C3%ADa-matem%C3%A1ticas/ciencia/what-is-a-control-group-606107/>
- Guadalupe, C., León, J., Rodríguez, J., & Vargas, S. (2017). *Estado de la educación en el Perú. Análisis y perspectivas de la educación básica*. Cajamarca: Grupo de Análisis para el desarrollo (GRADE).
- Guru. (2021). *Reinforcement Learning: What is, Algorithms and Applications*. Recuperado de: [https://www.guru99.com/reinforcement-learning-tutorial.html#:~:text=The%20example%20of%20reinforcement%20learning,are%201\)%20Positive%20\)%20Negative](https://www.guru99.com/reinforcement-learning-tutorial.html#:~:text=The%20example%20of%20reinforcement%20learning,are%201)%20Positive%20)%20Negative)
- Han, M., Zhao, J., Zhang, X., Shen, J., & Li, Y. (2020). The reinforcement learning method for occupant behavior in building control: a review. *Energy and Built Environment*, 2 (2021), 137-148.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. Ontario: Pearson Education.
- Huerta, R. (2017). *Plataforma web para el apoyo en la mejora de la comprensión lectora en los alumnos de educación básica*. Lima: Pontificia Universidad Católica del Perú.
- Ionos. (2019). *El modelo en cascada: desarrollo secuencial de software*. Recuperado de: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Javier, L. (2013). *Lectura comprensiva y sus estrategias*. Guatemala: Calzada Aguilar Batres.
- Lampadia. (2016). *El Perú mejora en los resultados PISA*. Recuperado de: <https://www.lampadia.com/analisis/educacion/el-peru-mejora-en-los-resultados-pisa/>.
- Layona, R., Yulianto, B., & Tunardi, Y. (2018). Web based Augmented Reality for Human Body Anatomy Learning. *Computer Science*, 135, 457-464.
- Liessner, R., Schmitt, J., Dietermann, A., & Baker, B. (2018). *Hyperparameter Optimization for Deep Reinforcement Learning in Vehicle Energy Management*. Dresden: Technische Universität Dresden.
- Lillo, A. (2019). *Artificial Intelligence for Videogames with Deep Learning*. Alicante.
- Lincapié, D. (2020). *Docentes: los otros trabajadores esenciales durante el COVID-19*. Recuperado de: <https://blogs.iadb.org/educacion/es/docentestrabajadoresesenciales/>.
- López, C. (2015). El videojuego como herramienta educativa. Posibilidades y problemáticas acerca de los serious games. *Apertura*, 8(1), 1-15.
- López, S. (2020). *Firestore: qué es, para qué sirve, funcionalidades y ventajas*. Recuperado de: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>.
- Lucas, J. (2019). *Qué es NodeJS y para qué sirve*. Recuperado de: <https://openwebinars.net/blog/que-es-nodejs/>
- Matesanz, M. (2014). *La lectura en la educación primaria: Marco teórico y propuesta de intervención*. Segovia: Universidad de Valladolid.

- Maulee. (2020). *El vertiginoso crecimiento de la industria de los videojuegos en Chile*. Recuperado de: <https://www.maulee.cl/el-vertiginoso-crecimiento-de-la-industria-de-los-videojuegos-en-chile/>.
- Mazur, M. (2015). *A Step by Step Backpropagation Example*. Recuperado de: <https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/>
- Méndez, E. (2006). *Modelo de evaluación de metodologías para el desarrollo de software*. Caracas: Universidad Católica Andres Bello.
- MINEDU. (2013). *¿Cómo mejorar la Comprensión Lectora de nuestros estudiantes?* Recuperado de: http://umc.minedu.gob.pe/wp-content/uploads/2014/03/Informe-Director-Docente_4to-EIB-2013.pdf.
- MINEDU. (2015). *El Perú en PISA 2015. Informe nacional de resultados*. Recuperado de: http://umc.minedu.gob.pe/wp-content/uploads/2017/04/Libro_PISA.pdf.
- MINEDU. (2018). *Resultados de la evaluación internacional PISA 2018*. Recuperado de: <http://umc.minedu.gob.pe/wp-content/uploads/2019/12/PISA-2018-Resultados.pdf>.
- Mnih, V., Puigdomènech, A., Mirza, M., Graves, A., Lillicrap, T., Harley, T. & Kavukcuoglu, K. (2016). *Asynchronous Methods for Deep Reinforcement Learning*. Recuperado de: <http://arxiv.org/abs/1602.01783>.
- Morales, E. (2009). *El uso de los videojuegos como recurso de aprendizaje en educación primaria y Teoría de la Comunicación*. Madrid: Universidad Complutense de Madrid.
- Moreno, J., Vahos, S., & Mazo, C. (2019). Videojuego para la enseñanza del cuidado del agua. *TecnoLógicas*, 22(45), 59-72. Recuperado de: [doi:https://doi.org/10.22430/22565337.1091](https://doi.org/10.22430/22565337.1091)
- Mueller, B., & Taj, M. (2020). *La educación por televisión vive un auge por la pandemia del coronavirus*. Recuperado de: <https://www.nytimes.com/es/2020/08/17/espanol/educacion-television.html>
- Nanalyze. (2017). *Unity Technologies – The World’s Leading Game Engine*. Recuperado de: <https://www.nanalyze.com/2017/10/unity-technologies-leading-game-engine/>.
- Naslund, E. (2020). *3000 profesores comparten sus experiencias de aprendizaje remoto COVID-19*. Recuperado de: <https://blogs.iadb.org/educacion/es/docentescovid/>.
- Needleman, S. (2020). *La pandemia aumenta las ganancias de la industria de los videojuegos*. Recuperado de: <https://www.latercera.com/que-pasa/noticia/la-pandemia-aumenta-las-ganancias-de-la-industria-de-los-videojuegos/Y3DAUZMUMZEKVBAT2P33CCXPCQ/>
- OECD. (2015). *PISA 2015: Draft Reading Literacy Framework*. París: OECD.
- Oficina de Medición de la calidad de los aprendizajes. (2019). *Evaluación Pisa 2018*. Lima: Ministerio de Educación.
- Ohyver, M., Moniaga, J., Sungkawa, I., Subagyo, B., & Chandra, I. (2019). The Comparison Firebase Realtime Database and MySQL Database Performance using Wilcoxon Signed-Rank Test. *Procedia Computer Science*(157), 396-405.

- Oracle. (2020). *¿Qué es una base de datos relacional?* Recuperado de: oracle.com/ar/database/what-is-a-relational-database/.
- Orbegozo, F. (2016). *Perú sale del último lugar en la prueba PISA 2015*. Recuperado de: <https://elcomercio.pe/peru/peru-sale-lugar-prueba-pisa-2015-152124>.
- Otegui, J. (2017). La realidad virtual y la realidad aumentada en el proceso de marketing. *Dirección y Administración de Empresas*(24), 155-229.
- Pérez, A. (2007). *Desarrollo de herramientas web de gestión docente*. Recuperado de: <https://repositorio.upct.es/bitstream/handle/10317/179/pfc2475.pdf>
- Pérez, D. (2007). *¿Qué son las bases de datos?* Recuperado de: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Pierre, V. (2020). *ML-Agents Toolkit Overview*. Recuperado de: <https://github.com/Unity-Technologies/ml-agents>.
- Pressman, R. (2010). *Ingeniería de software. Un enfoque práctico*. México,D.F.: The McGraw-Hill. Obtenido de EcuRed.
- Quang, D., & Hoon, S. (2020). Proximal Policy Optimization Through a Deep Reinforcement Learning Framework for Multiple Autonomous Vehicles at a Non-Signalized Intersection. *Applied Science*,10(16), 5722. Recuperado de: <https://doi.org/10.3390/app10165722>.
- Quora. (2016). *How many game programmers use Unity?* Recuperado de: <https://www.quora.com/How-many-game-programmers-use-Unity>.
- Recuero, P. (2017). *Tipos de aprendizaje en Machine Learning: supervisado y no supervisado*. Recuperado de: <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/>.
- Richter, A., & Smarrelli, G. (2020). *Hacia un “nuevo normal” en la educación de niños y adolescentes: riesgos, retos y oportunidades*. Recuperado de: <http://hacerperu.pe/hacia-un-nuevo-normal-en-la-educacion-de-ninos-y-adolescentes-riesgos-retos-y-oportunidades/>.
- Risi, S., & Togelius, J. (2015). Neuroevolution in Games: State of Art and Open Challenges. *IEEE Transactions on Computational Intelligence and IA in Games*,9(1),25-41.
- Rivera, J. (2018). *Realidad Virtual: ¿qué es y qué tipos existen?* Recuperado de: <https://www.tudosisdigital.com/noticias/realidad-virtual-que-es-y-que-tipos-existen>.
- Roa, A. (2020). *Trucos para leer mejor y desarrollar la comprensión lectora*. Recuperado de: <https://www.conmishijos.com/educacion/lectura-escritura/trucos-para-leer-mejor-y-desarrollar-la-comprension-lectora/>.
- Romero, L. (2020). *Elegir el método de investigación adecuado*. Recuperado de: <https://www.revistacomunicar.com/wp/escuela-de-autores/elegir-el-metodo-de-investigacion-adecuado/>.

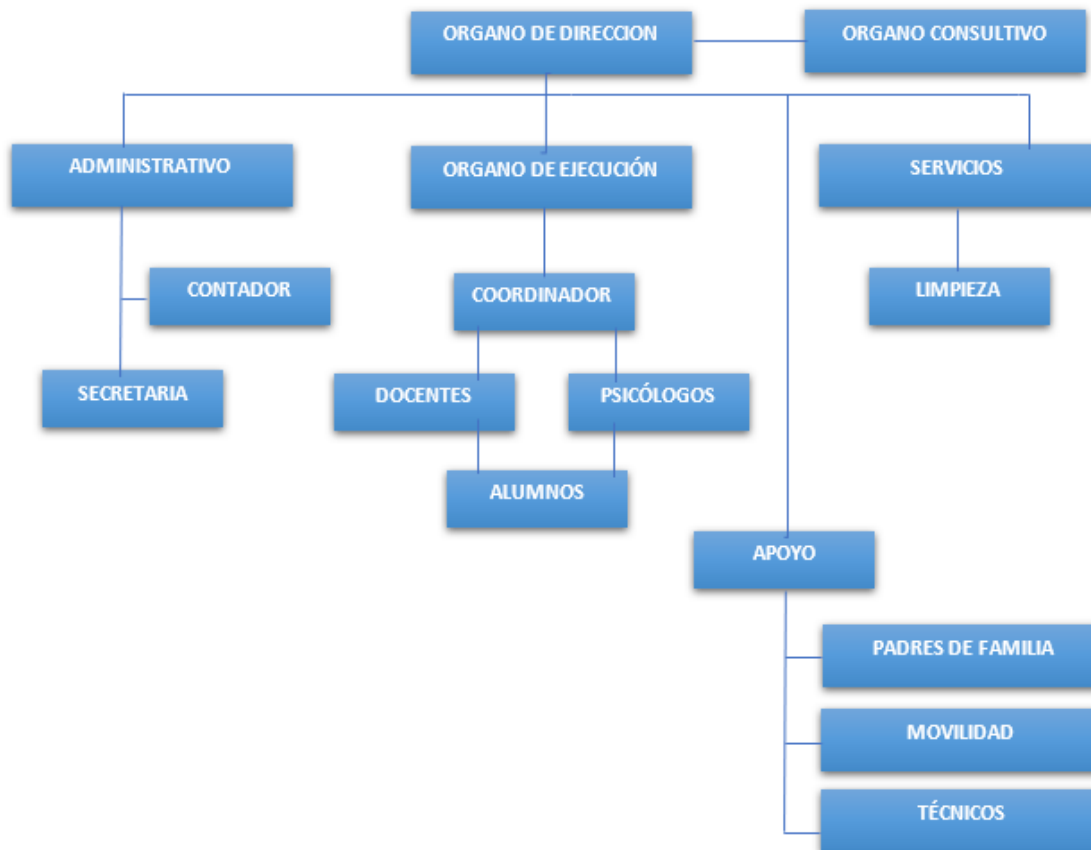
- Roncancio, A., Ortiz, M., Llano, H., Malpica, M., & Bocanegra, J. (2017). El uso de los videojuegos como herramienta didáctica para mejorar la enseñanza-aprendizaje: Una revisión del estado del tema. *Revista Ingeniería, Investigación y Desarrollo*, 36-46.
- Ropilo, T. (2019). *Teaching a machine learning agent to survive in a 2D Top-Down Environment*. Mittweida: University of Applied Sciences.
- Rubiales, A. (2020). *Explicación de las Funciones de activación en Redes Neuronales y práctica con Python*. Recuperado de: <https://rubialesalberto.medium.com/explicaci%C3%B3n-funciones-de-activaci%C3%B3n-y-pr%C3%A1ctica-con-python-5807085c6ed3>
- Rubio, M., & Berlanga, V. (2012). Cómo aplicar las pruebas paramétricas bivariadas t de Student y ANOVA en SPSS.Caso Práctico. *Revista d'Innovació i Recerca en Educació*, 5(2), 83-100.
- Ruete, B. (2020). *La industria se une a la iniciativa #PlayApartTogether contra el coronavirus*. Recuperado de: https://as.com/meristation/2020/03/30/noticias/1585568824_362876.html
- Ruiz, C., Basualdo, M., & Matich, D. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Rosario: Universidad Tecnológica Nacional.
- Ryte Wiki. (2019). *Modelo en Cascada*. Recuperado de: https://es.ryte.com/wiki/Modelo_en_Cascada#Importancia_para_la_programaci%C3%B3n.
- Sánchez, R. (2015). T-Student. Usos y abusos. *Revista Mexicana de Cardiología*, 26(1), 59-61.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. Recuperado de: <https://arxiv.org/pdf/1707.06347.pdf>.
- Shankar, S. (2020). *Types of Machine Learning Algorithms*. Recuperado de: <https://www.7wdata.be/visualization/types-of-machine-learning-algorithms-2/>.
- Shao, K., Tang, Z., Zhu, Y., Li, N., & Zhao, D. (2019). *A Survey of Deep Reinforcement Learning in Video Games*. Recuperado de:
- Soto, J. (2018). *Estadística: Test de hipótesis*. Recuperado de: <https://es.scribd.com/document/395901359/Estadistica-inferencial-TEST-DE-HIPOTESIS-por-Bioq-Jose-Luis-Soto-Velasquez-3-2>
- Tonatiuh, A. (2017). *El poder educativo de los videojuegos*.
- Trends, E. (2016). Gamificación. *Observatorio de Innovación Educativa del Tecnológico de Monterrey*, 1-36.
- Trends, E. (2017). Realidad aumentada y virtual. *Observatorio de Innovación Educativa del Tecnológico de Monterrey*, 1-36.
- Zurita, J., Villasís, M., Miranda, G., & Márquez, H. (2018). *Estudios experimentales: diseños de investigación para la evaluación de intervenciones en la clínica*. Recuperado de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2448-91902018000200178

Anexos

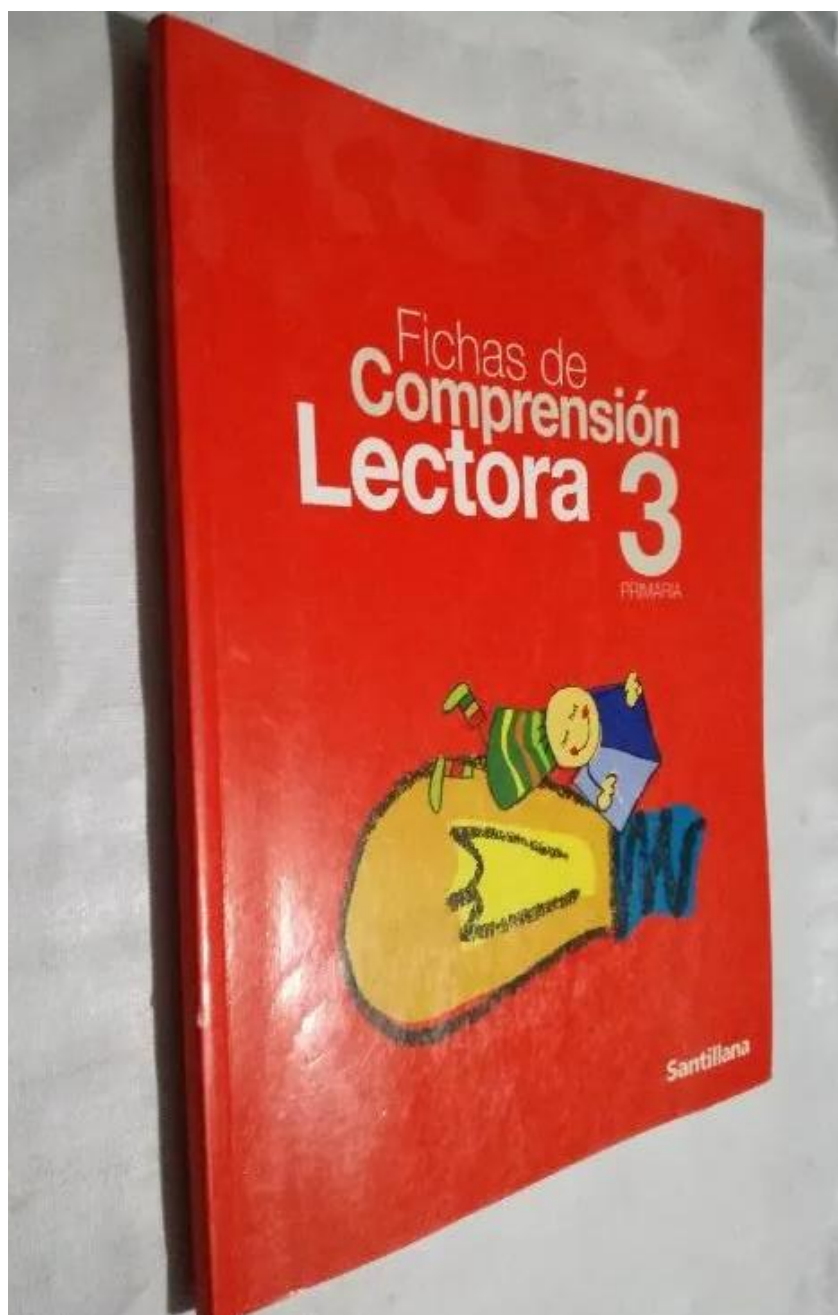
Anexo 1. Organigrama de la I.E.P Colegio “Mater Christi”



ORGANIGRAMA ESTRUCTURAL



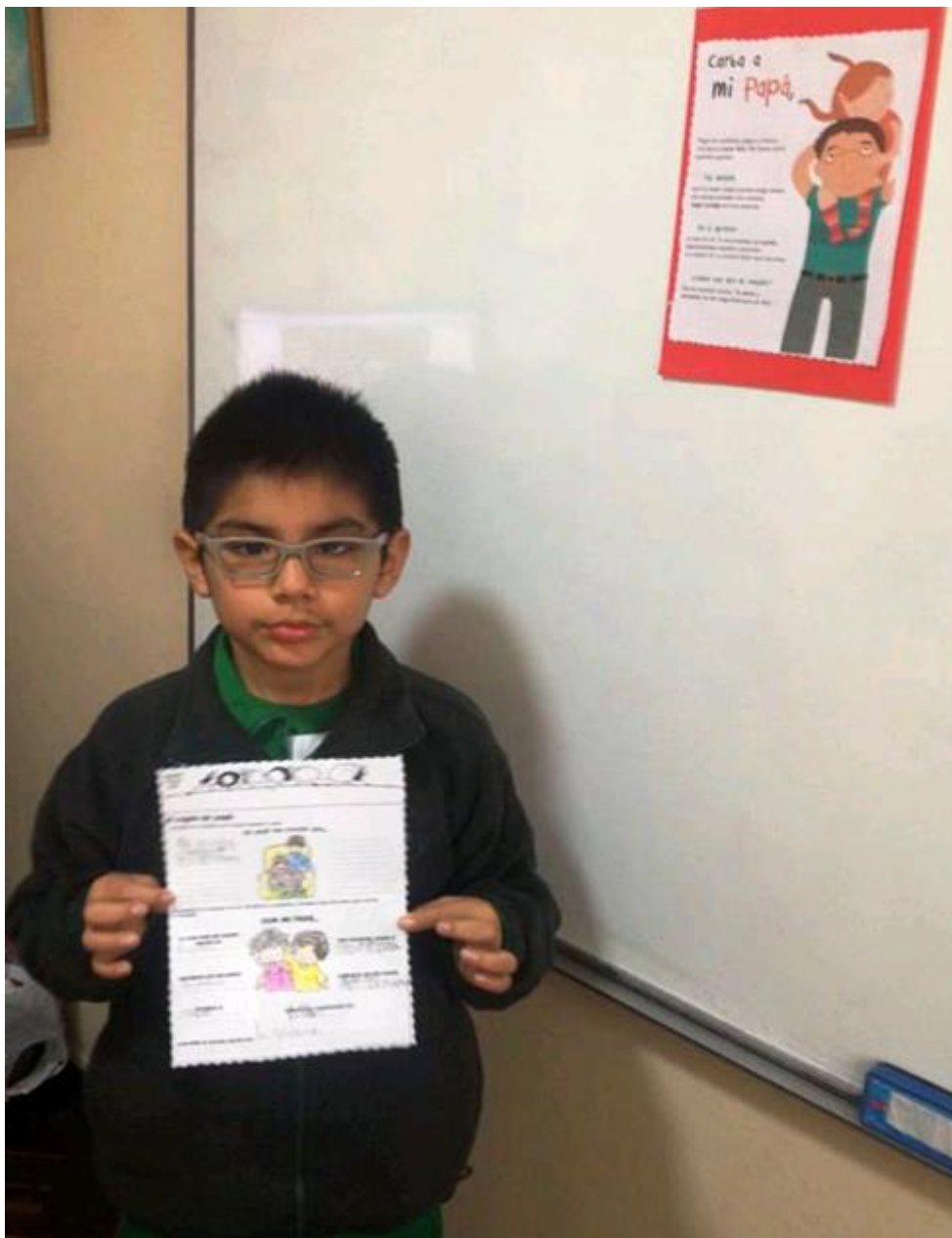
Anexo 2. Material educativo para la enseñanza de las clases de lectura



Anexo 3. Clases de lectura antes de la pandemia del Covid-19



Anexo 4. Desarrollo de ejercicios de comprensión lectora



Anexo 6. Instrumento de validez de encuesta de videojuego**INSTRUMENTO DE VALIDACIÓN DE ENCUESTA DE USO DE VIDEOJUEGO POR CRITERIO DE EXPERTO****I. DATOS GENERALES**1.1 Apellidos y nombres de experto: *GARCÍA CERRUTTI, CARMEN NATALIA*1.2 Especialidad: *PSICOLOGÍA*

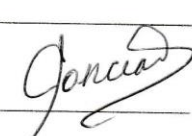
1.3 Apellidos y nombres del investigador: Machuca Breña Alvaro Rodrigo

II. ASPECTOS DE VALIDACIÓN

ITEM	SI	NO	SUGERENCIAS
1. El instrumento presenta coherencia con el problema de investigación.	X		
2. El instrumento guarda relación con los objetivos propuestos en la investigación.	X		
3. Las preguntas siguen una secuencia lógica.	X		
4. Las preguntas utilizadas son claras y comprensibles.	X		
5. El grado de dificultad de cada pregunta es aceptable para los participantes.	X		
6. En general, el instrumento permite un manejo ágil de la información.	X		

II. CALIFICACIÓN GENERAL:

CATEGORIA	Marcar con un aspa
APROBADO	X
OBSERVADO	
DESAPROBADO	

Lugar y Fecha: *PUEBLO LIBRE, 12 DE JULIO 2021*

 Firma de experto

Anexo 7. Matriz de consistencia

TITULO DE LA INVESTIGACIÓN	DESARROLLO DE UN VIDEOJUEGO PARA MEJORAR EL NIVEL DE COMPRENSIÓN LECTORA EN ESTUDIANTES DE PRIMARIA					
PROBLEMA GENERAL	PROBLEMAS ESPECÍFICOS	OBJETIVO GENERAL	OBJETIVOS ESPECÍFICOS	HIPÓTESIS GENERAL	HIPÓTESIS ESPECÍFICAS	VARIABLE INDEPENDIENTE
<p>P.G: ¿De qué manera es posible desarrollar un videojuego que mejore el nivel de comprensión lectora en estudiantes de primaria?</p>	<p>P.E 1: ¿Qué tecnologías y técnicas de Inteligencia Artificial deben de integrarse para el desarrollo funcional del videojuego?</p>	<p>O.G: Desarrollar un videojuego que permita mejorar el nivel de comprensión lectora en estudiantes de primaria.</p>	<p>O.E 1: Identificar las tecnologías y técnicas de Inteligencia Artificial que se integran para el desarrollo funcional del videojuego.</p>	<p>H.G: El uso del videojuego desarrollado posibilita la mejora del nivel de comprensión lectora en estudiantes de primaria.</p>	<p>H.E 1: La integración de las tecnologías y técnicas de Inteligencia Artificial implementadas posibilitarán la correcta funcionalidad del videojuego.</p>	<p>Descripción: Uso de videojuego desarrollado Indicadores: Experiencia de uso del videojuego Instrumento de medición: Encuesta de satisfacción Moreno, Vahos, y Mazo (2019).</p>
	<p>P.E 2: ¿De qué forma se puede obtener el porcentaje de aceptación del prototipo desarrollado mediante la experiencia de uso del videojuego de los</p>		<p>O.E 2: Obtener el porcentaje de aceptación del sistema desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria.</p>		<p>H.E 2: El porcentaje de aceptación obtenido del sistema desarrollado mediante la experiencia de uso del videojuego de los estudiantes de primaria será mayor a 50%.</p>	<p>VARIABLE DEPENDIENTE Descripción: Nivel de comprensión lectora de estudiantes. Indicadores: Nivel literal: Reconoce personajes del texto, Ordena secuencia de hechos o acciones en el texto.</p>

	<p>estudiantes de primaria?</p> <p>P.E 3: ¿Cómo validar estadísticamente la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención?</p>		<p>O.E 3: Validar estadísticamente la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención.</p>		<p>H.E 3: La validación estadística permitirá detectar la mejora de la comprensión lectora de los estudiantes de primaria a través de las notas antes y después de la intervención.</p>	<p>Nivel inferencial: Analiza hechos presentes en el texto, Interpreta información expuesta en el texto o reacciones de los personajes. Nivel crítico: Enjuicia las actitudes y conductas de los personajes o protagonistas en el texto Instrumento de medición: Pretest y Postest.</p>
	<p>P.E 4: ¿Cómo se puede generar un procedimiento pedagógico y tecnológico para complementar las clases de comprensión lectora mediante el uso del videojuego?</p>		<p>O.E 4: Generar un procedimiento pedagógico y tecnológico para complementar las clases de comprensión lectora mediante el uso del videojuego.</p>		<p>H.E 4: El procedimiento pedagógico y tecnológico desarrollado permitirá complementar las clases de comprensión lectora mediante el uso del videojuego.</p>	