



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

**“USO DE REALIDAD AUMENTADA CON CONTROL
GESTUAL DE MANOS EN APOYO AL APRENDIZAJE
DE LA LECTOESCRITURA”**

PARA OBTENER EL GRADO DE:
**MAESTRO EN SISTEMAS
COMPUTACIONALES**

PRESENTA:

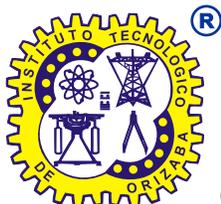
I.S.C. Raúl de Jesús Sánchez Martínez

DIRECTOR DE TESIS:

M.C. María Antonieta Abud Figueroa

CODIRECTOR DE TESIS:

Dr. Ulises Juárez Martínez



ORIZABA, VERACRUZ, MÉXICO.

OCTUBRE 2023



Orizaba, Veracruz, **19/septiembre/2023**
Asunto: Revisión de trabajo escrito

C. CUAUHTÉMOC SÁNCHEZ RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
PRESENTE.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

RAÚL DE JESÚS SÁNCHEZ MARTÍNEZ

La cual lleva el título de:

Uso de realidad aumentada con control gestual de manos en apoyo al aprendizaje de la lectoescritura

Y concluyen que se acepta.

ATENTAMENTE
Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

PRESIDENTE: M.C. MARÍA ANTONIETA ABUD FIGUEROA

ma. Antonieta Abud F.

FIRMA

SECRETARIO: DR. ULISES JUÁREZ MARTÍNEZ

Ulises Juárez Martínez

FIRMA

VOCAL: DR. HILARIÓN MUÑOZ CONTRERAS

Hilarión Muñoz Contreras

FIRMA

VOCAL SUP.: DRA. LISBETH RODRÍGUEZ MAZAHUA

Lisbeth Rodríguez Mazahua

FIRMA

TA-09-18





Orizaba, Veracruz, **18/octubre/2023**
Dependencia: **División de Estudios de
Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: I

**C. RAÚL DE JESÚS SÁNCHEZ MARTÍNEZ
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES
P R E S E N T E.-**

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

" Uso de realidad aumentada con control gestual de manos en apoyo al aprendizaje de la lectoescritura"

comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

Cuahtémoc Sánchez R.

**DR. CUAHTÉMOC SÁNCHEZ RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN**



OG-13-F06



Agradecimientos

Agradezco a Dios por las oportunidades, porque me ha sustentado y dotado la vida hasta llegar a este momento. Le agradezco también por las personas que ha puesto en mi vida, por hacerme el hombre que soy.

Gracias a mi familia, mis padres, así como mi hermana. Los múltiples sacrificios que han hecho a lo largo de mi vida no he de olvidarlos nunca. Por criarme, enseñarme el significado de la palabra responsabilidad. Me han dado una vida digna, y dentro de sus posibilidades proveyeron lo necesario para que llegara hasta este momento. Este logro les pertenece tanto a ustedes como a mí mismo. Los amo profundamente, no elegiría a nadie más como mi familia.

Gracias a Jair y Lu. Me dieron los regalos más grandes: su amistad, compasión y conocimiento; ninguna de esas cosas es barata. Ambos me dieron un sueño, los conocimientos para lograrlo y con ello, abrieron mi mundo. Creo honestamente, que tu Jair, eres lo que todo maestro debe aspirar a ser. No se confundan, ustedes también son como mi propia sangre, parte de mi familia.

Gracias a mis amigos, Ana, Adolfo y Palafox. Sin ellos probablemente no habría pasado de la licenciatura. Especialmente tú Ana, eres un ángel. No solo me dotaron su amistad, sino su aliento y ayuda. Me divertí muchísimo estando con ustedes.

Al Jardín de Niños Covadonga, por darme la oportunidad de trabajar con ellos en este proyecto.

Al CONACYT, por el apoyo que la beca significó para ayudarme a concluir este proyecto.

Al TecNM, más específicamente al Instituto Tecnológico de Orizaba, creo que los mejores años de mi vida los he pasado ahí.

También agradezco a la maestra María Antonieta Abud Figueroa, por su apoyo, su comprensión, por siempre ayudarme a resolver los problemas. Muchísimas gracias, sin usted, esta tesis no sería posible.

Contenido

Índice de Tablas.....	VIII
Índice de Figuras	IX
Resumen	XI
Abstract.....	XII
Introducción.....	XIII
Capítulo 1. Antecedentes	1
1.1 Marco teórico	1
1.1.1 Ingeniería de software educativo.....	1
1.1.2 Realidad aumentada.....	2
1.1.2.1 Herramientas de realidad aumentada	3
1.1.3 Interfaces humano-máquina.....	4
1.1.3.1 Captura de movimiento	4
1.1.3.2 Control por gestos manuales	5
1.1.4 Procesos de aprendizaje	5
1.1.4.1 Comprensión lectora	7
1.1.4.2 Trastornos de aprendizaje de la lectoescritura	8
2.1.1 Métodos de enseñanza del idioma español	9
2.1.1.1 Modelos de enseñanza discreta	9
2.1.1.2 Programa Kantor	10
1.2 Situación tecnológica, económica y operativa de la empresa.....	11
1.3 Planteamiento del problema.....	11
1.4 Objetivo General y Objetivos Específicos	12
1.4.1 Objetivo General	12
1.4.2 Objetivos Específicos	12
1.5 Justificación	13
Capítulo 2. Estado de la práctica.....	14
2.2 Trabajos relacionados.....	14
2.3 Análisis comparativo.....	20
2.4 Propuesta de solución	25
2.4.1 Análisis de las tecnologías de información	25
2.4.1.1 Lenguajes de programación	25
2.4.1.1.1 C#	26

2.4.1.2	Entornos de Desarrollo Integrado (IDE)	26
2.4.1.2.1	Visual Studio Code.....	27
2.4.1.3	<i>Software Development Kit</i> de realidad aumentada	27
2.4.1.3.1	Vuforia Engine	28
2.4.1.4	Metodologías de desarrollo de <i>software</i>	28
2.4.1.4.1	Scrum.....	28
2.4.1.5	Motor gráfico.....	29
2.4.1.5.1	Unity	29
2.4.1.6	Interfaz de control	30
2.4.1.6.1	<i>Leap Motion</i> [®]	30
Capítulo 3. Aplicación de la metodología		31
3.1	Descripción de la solución	31
3.1.1	<i>Sprint 1</i> Análisis de requerimientos.....	32
3.1.2	<i>Sprint 2</i> Diseño de la arquitectura.....	34
3.1.2.1	Diseño de la arquitectura conceptual	34
3.1.2.1.1	Patrón arquitectónico	34
3.1.2.2	Diseño de la arquitectura física.....	35
3.1.3	<i>Sprint 3</i> Diseño de los marcadores y modelos tridimensionales.....	36
3.1.4	<i>Sprint 4</i> Diseño de los niveles y <i>wireframes</i> del sistema	40
3.1.5	<i>Sprint 5</i> Grabación de los archivos multimedia.....	42
3.1.6	<i>Sprint 6</i> Desarrollo de los menús	43
3.1.7	<i>Sprint 7</i> Diseño del nivel 1	46
3.1.7.1	Configuración de <i>Leap Motion</i> [®]	47
3.1.7.2	Configuración de Vuforia Engine	49
3.1.7.3	Configuración de audios	51
3.1.7.4	Creación del panel de letras	53
3.1.7.5	Configuración de modelos tridimensionales.....	54
3.1.7.6	Programación de funcionamiento del nivel	58
3.1.8	<i>Sprint 8</i> Realización de video de ayuda.....	63
3.1.9	<i>Sprint 9</i> Diseño del nivel 2.....	64
3.1.9.1	Edición de los audios	65
3.1.9.2	Edición de los modelos tridimensionales	65
3.1.9.3	Codificación del nivel	67

3.1.10 <i>Sprint</i> 10 Diseño del nivel 3.....	70
3.1.10.1 Edición de los audios	71
3.1.10.2 Edición de los modelos tridimensionales	72
3.1.11 <i>Sprint</i> 11 Diseño del nivel 4.....	73
3.1.11.1 Edición de los audios	74
3.1.11.2 Edición de los modelos tridimensionales	74
3.1.11.3 Codificación del nivel	76
Capítulo 4. Resultados	82
4.1 <i>Sprint</i> 12 Aplicación a un caso de estudio	82
4.1.1 Caso de estudio: Aprendizaje de las letras utilizando realidad aumentada con interacción a través de gestos manuales	82
4.1.2 Caso de estudio: Jardín de Niños Covadonga	85
4.1.2.1 Planteamiento del caso de estudio	85
4.1.2.2 Características del caso de estudio.....	85
4.1.2.3 Descripción de las pruebas del caso de estudio	87
4.1.2.4 Resultados obtenidos del caso de estudio.....	88
Capítulo 5. Conclusiones y recomendaciones	101
5.1 Conclusiones	101
5.2 Recomendaciones	102
Anexos y/o apéndices	104
Productos académicos.....	104
Bibliografía.....	106

Índice de Tablas

Tabla 2.1 Análisis comparativo de los artículos del Estado del Arte	21
Tabla 3.1 Historias de usuario.....	31
Tabla 3.2 Sprint 1	32
Tabla 3.3 Funcionalidades requeridas para el sistema.....	33
Tabla 3.4 Sprint 2.	34
Tabla 3.5 Sprint 2.	37
Tabla 3.6 Marcadores y modelos respectivos.	37
Tabla 3.7 Sprint 4.	40
Tabla 3.8 Wireframes.....	40
Tabla 3.9 Sprint 5.	42
Tabla 3.10 Diálogos por niveles.	42
Tabla 3.11 Sprint 6.	44
Tabla 3.12 Funcionamiento de los menús.	44
Tabla 3.13 Sprint 7.	47
Tabla 3.14 Sprint 8.	64
Tabla 3.15 Sprint 9	64
Tabla 3.16 Sprint 10	70
Tabla 3.17 Sprint 11	73
Tabla 4.1 Sprint 12	82
Tabla 4.2 Rondas de pruebas.....	91
Tabla 4.2 Rondas de pruebas (Continuación)	92
Tabla 4.2 Rondas de pruebas (Continuación)	93
Tabla 4.2 Rondas de pruebas (Continuación)	94
Tabla 4.2 Rondas de pruebas (Continuación)	95
Tabla 4.2 Rondas de pruebas (Continuación)	96
Tabla 4.2 Rondas de pruebas (Continuación)	97

Índice de Figuras

Figura 1.1 Relación de los niveles de abstracción en las Herramientas Visuales y Herramientas de Desarrollo	4
Figura 1.2 Etapas del Proceso de Aprendizaje	7
Figura 2.1 Componentes de un IDE.....	26
Figura 2.2 Metodología Scrum.....	29
Figura 3.1 Diagrama de casos de uso.....	33
Figura 3.2 Patrón arquitectónico	34
Figura 3.3 Arquitectura física del sistema.....	36
Figura 3.4 Plantilla para sílabas.....	39
Figura 3.5 Plantilla para vocales.....	39
Figura 3.6 Menú de compilación.....	45
Figura 3.7 Plugins de Leap Motion®	48
Figura 3.8 Carpeta que contiene la escena prefabricada a utilizar para el desarrollo de los niveles.	48
Figura 3.9 Objeto Interaction Manager.	49
Figura 3.10 Detalles de Vuforia Configuration.....	50
Figura 3.11 Biblioteca de marcadores.	50
Figura 3.12 Jerarquía actual del nivel 1	51
Figura 3.13 Configuración del primer audio.	52
Figura 3.14 Audios para los objetos tridimensionales.....	52
Figura 3.15 Mockup del funcionamiento del panel de letras.	53
Figura 3.16 Panel de letras.....	53
Figura 3.17 Marcador de la letra A con sus objetos tridimensionales.....	54
Figura 3.18 Configuración para los objetos al identificar un marcador.....	55
Figura 3.19 Configuración para los objetos al perder el marcador	55
Figura 3.20 Componentes Mesh Collider y RigidBody	56
Figura 3.21 Componente Interaction Behaviour	56
Figura 3.22 Interfaces GraspBegin y GraspEnd.....	57
Figura 3.23 Elementos de ObjetosJuego dentro de marcador A.....	62
Figura 3.24 Audios instructivos modificados	65
Figura 3.25 Marcador A en el nivel dos.	66
Figura 3.26 Jerarquía del marcador A.....	67
Figura 3.27 Audios de instrucción nivel tres	71
Figura 3.28 Audios de la letra A	72
Figura 3.29 Marcador A en el nivel cuatro.....	72
Figura 3.30 Audios del nivel	74
Figura 3.31 Letras tridimensionales.....	75
Figura 3.32 Objetos con palabras	75
Figura 3.33 Bloque de palabra abeja	76
Figura 4.1 Menú Principal	83
Figura 4.2 Recuadros del video de ayuda	83
Figura 4.3 Créditos del programa	84

Figura 4.4 Menú de niveles.....	84
Figura 4.5 Funcionamiento del nivel uno.....	85
Figura 4.6 Estudiante realizando pruebas	86
Figura 4. 7 Estudiantes usando el programa.....	89

Resumen

La lectoescritura es la capacidad para leer y escribir adecuadamente, también está íntimamente relacionada a la comprensión lectora al facultar a una persona para interpretar el lenguaje a partir de un texto. En la prueba PISA 2018 México se localizó en el nivel 2 de comprensión lectora, encontrándose por debajo del promedio de los países que conforman la Organización para la Cooperación y el Desarrollo Económicos (OCDE).

En la actualidad y con el constante avance de la tecnología, distintos proyectos desarrollaron soluciones satisfactorias a problemáticas de aprendizaje relacionadas al área de las matemáticas y la medicina utilizando realidad aumentada e interfaces humano-máquina para controlar sus modelos generados. Por tanto, en el presente proyecto se desarrolló una aplicación para el apoyo al aprendizaje de la lectoescritura. Dicha aplicación utiliza realidad aumentada por medio del despliegue de modelos tridimensionales renderizados a través de un dispositivo con cámara para su despliegue a través de una pantalla en la realidad objetiva y su vez, el usuario manipula estos objetos a través de una interfaz humano-máquina enfocada en identificar movimientos de las manos como instrucciones.

El principal beneficio de la aplicación desarrollada es proporcionar una herramienta de fácil uso como auxilio a los profesores o maestros para una enseñanza interactiva.

Abstract

Literacy skills are the ability to read and write properly, they are closely related to reading comprehension by enabling a person to interpret language from a text. In the 2018 PISA test, Mexico was located at Level 2 in reading comprehension, below the average of countries belonging to the Organization for Economic Co-operation and Development (OECD).

Currently, with the constant advancement of technology, different projects have developed satisfactory solutions to learning problems related to the fields of mathematics and medicine, using augmented reality and human-machine interfaces to control their generated models. Therefore, this project presents a developed application to support literacy learning. This application utilizes augmented reality through the display of three-dimensional models rendered via a camera-equipped device for projection onto an objective reality screen. At the same time, the user manipulates these objects through a human-machine interface focused on identifying hand movements as instructions.

The main benefit of the developed application is to provide an easy-to-use tool as an aid to teachers for interactive teaching.

Introducción

La prueba PISA 2018 reveló que, entre los miembros de la OCDE, México ocupa un lugar bajo en los resultados promedio en cuanto a comprensión lectora se refiere. En estudios abordados en este documento se concluyó que la capacidad de una persona para comprender textos complejos facilita el acceso a estructuras complejas de pensamiento, ergo, la capacidad para leer y escribir es un termómetro fiable de la inteligencia de una persona. Por tanto, un promedio bajo en la comprensión lectora de una sociedad es indicativo de la decadencia de su capacidad para la resolución de problemas colectivos.

Por otro lado, las tecnologías de realidad aumentada han avanzado a pasos agigantados los últimos 10 años, a partir de su implementación en entornos de desarrollo de videojuegos. La capacidad para desplegar una imagen virtual sobre el mundo real abrió amplias posibilidades sobre todo en cuanto al aprendizaje se refiere, facilitando la creación de escenarios del mundo real en entornos virtuales, con consecuencias inexistentes, por eso, su implementación en sectores de la industria y la educación no se hizo esperar.

A su vez, se crearon medios para interactuar directamente con estos escenarios, para facilitar su análisis, por medio de interfaces que permiten la comunicación entre los operadores humanos y el entorno virtual, generalmente a través de la captura de movimiento.

Por lo anteriormente mencionado, en el presente proyecto de tesis se creó una aplicación que utilizando realidad aumentada controlada medio de una interfaz humano-máquina, facilita el aprendizaje de la lectoescritura, como apoyo al desarrollo de dichas habilidades.

Para facilitar la comprensión del lector, este proyecto de tesis se encuentra organizado en cinco capítulos:

- Primer capítulo: llamado “Antecedentes”, se presenta el marco teórico, planteamiento del problema, objetivo general y objetivos específicos, así como la justificación del proyecto.

- Segundo capítulo: llamado “Estado de la práctica”, se presenta los trabajos de naturaleza similar al proyecto presente, destacando brevemente los objetivos, desarrollos y resultados de estos, así como también el análisis comparativo de los proyectos estudiados. También se describe la propuesta de solución describiendo las tecnologías y herramientas empleadas en la elaboración de este sistema.
- Tercer capítulo: llamado “Aplicación de la metodología”, se describe el proceso de elaboración del sistema.
- Cuarto capítulo: llamado “Resultados”, se muestra la funcionalidad del sistema y su aplicación en un caso de estudio.
- Quinto capítulo: llamado “Conclusiones y recomendaciones” se presentan las conclusiones y recomendaciones para el sistema desarrollado.

Capítulo 1. Antecedentes

En este capítulo se presentan diversos conceptos que resultan relevantes para la realización del proyecto presentado. Se describe también la problemática a resolver, el objetivo general, los objetivos específicos y la justificación del proyecto de tesis.

1.1 Marco teórico

A continuación, se describen términos relacionados con el tema de investigación.

1.1.1 Ingeniería de software educativo

La ingeniería de software se basa en la búsqueda y utilización de marcos comunes, tanto teóricos como de trabajo, para el desarrollo en conjunto de un programa informático de aplicación general o específica. En el caso de la Ingeniería de *software* enfocada al ámbito educativo, hay una inexistencia generalizada tanto de marcos teóricos como de desarrollo. A pesar de esto, las metodologías de desarrollo de software más usadas para los programas educativos son las más tradicionales: cascada, prototipo, desarrollo en espiral, desarrollo iterativo, programación extrema y desarrollo incremental [1].

El ciclo de desarrollo tradicional por lo general es el siguiente: análisis, diseño, implementación y pruebas. El análisis está centrado en la recolección de requisitos para el correcto funcionamiento del programa, que por lo general implica reuniones pautadas con el cliente (quien está solicitando el producto) para determinar el posible funcionamiento del programa en conjunción. En el caso del *software* enfocado en la educación, la enorme cantidad de requisitos causa ambigüedad en lo solicitado, y por extensión, contradicciones en el funcionamiento del programa. La etapa de diseño utiliza el análisis previo, y en él se realizan los procesos específicos dependiendo de la metodología de desarrollo seleccionada, es el proceso que más tiempo y recursos requiere. Las etapas de implementación y pruebas dependen de los requisitos específicos del cliente quien seleccionó el sistema operativo o herramienta en la que se ejecutará el programa [1].

Este proceso se ejecuta acompañado de la documentación de los pasos realizados: en el caso del análisis se diagrama todo el funcionamiento del programa y el posible

maquetado, así como los datos y sus relaciones, en el caso del diseño, dependiendo de la metodología utilizada se generan documentos explicando a detalle el funcionamiento de cada uno de los componentes, módulos o partes del *software* y también en el mismo código se realizan comentarios que permiten el entendimiento de cada uno de los elementos. En cuanto a la etapa de implementación y pruebas se realiza una bitácora de pruebas que fomenta la retroalimentación para realizar las correcciones debidas al programa en caso de que estas sean necesarias.

El correcto uso de la ingeniería de *software* permite a los desarrolladores adaptarse rápidamente a los cambios realizados durante un proceso de desarrollo ágil, facilita la comunicación en el grupo de trabajo, aun realizándose cada proceso en zonas geográficas totalmente diferentes. También asegura que se cumplan con los requisitos funcionales y no funcionales de manera rápida, facilitando el trabajo de los desarrolladores al tenerse un control exacto de las áreas ya cubiertas y funcionales del programa desarrollado [2].

1.1.2 Realidad aumentada

Realidad aumentada es un término que se aplica de forma general a una variedad de tecnologías visuales capaces de combinar información gráfica y alfanumérica para desplegarlas sobrepuestas a la visión del usuario del mundo real [3]. El sistema de Realidad Aumentada (RA) tiene sus bases en la tecnología de Realidad Virtual. Utilizando dispositivos de interacción humano-máquina, tales como *Leap Motion*[®] y *Tap Strap*[®] (que serán abordados más adelante), se obtiene información que la computadora interpreta y despliega de forma sobrepuesta a la realidad ante los ojos del usuario. La implementación del programa depende de las escenas generadas por la capacidad tecnológica del dispositivo en el que se despliega el programa. Gracias a esto, objetos no existentes en el entorno real se generan y exitosamente se integran en la visión de la realidad, asistidos por un dispositivo computacional. De esta forma, el usuario tiene una vívida experiencia del mundo virtual, en el mundo real, a través de la interfaz [4].

Se descubrió que esta tecnología, al dar al estudiante una guía de carácter visual y aparentemente físico, permite que los contenidos explicados en un aula sean vistos

desde una perspectiva más física, lo cual acerca la realidad existente fuera del aula a un entorno más educativo. Esto potencia el aprendizaje de conceptos y favorece el desarrollo de las habilidades cognitivas, espaciales, de concentración, fortalece la memoria inmediata y a largo plazo, tanto en sus formas visuales como auditivas, además de favorecer el razonamiento. También se observan cambios en la actitud de los estudiantes hacia el proceso del aprendizaje puesto que el interactuar de forma aparentemente directa refuerza el interés y el desarrollo del pensamiento inquisitivo al darles un objeto virtual con el cual interactuar [5].

1.1.2.1 Herramientas de realidad aumentada

A pesar de que la RA es actualmente utilizada de forma amplia en distintos campos como la medicina, la educación y la publicidad, el tiempo y la experiencia técnica que son requeridos para crear aplicaciones es uno de los motivos que su uso extensivo se ve lastrado [6]. Es por eso que se desarrollaron herramientas de RA que facilitan ampliamente su desarrollo, teniendo principalmente dos vertientes:

- Orientadas a programadores: requieren que el diseñador tenga conocimientos de programación al utilizar bibliotecas y lenguajes de desarrollo. En este trabajo, esta vertiente se llama Herramientas de Desarrollo.
- Orientadas a no programadores: cuentan con una menor abstracción y la capacidad de programación se oculta al desarrollador. En este trabajo, esta vertiente se llama Herramientas Visuales.

A su vez, estas dos categorías se fragmentan cada una en bajo y alto nivel. En las Herramientas de Desarrollo, su nivel depende de las bibliotecas y los lenguajes utilizados, entre mayor es el nivel de abstracción de los mismos, más alto será el nivel de la herramienta. Por el contrario, Herramientas Visuales de bajo nivel requieren escritura de código, aun en poca cantidad, mientras que las herramientas de alto nivel requieren un alto dominio de diseño visual [6]. La relación de la abstracción con los niveles de las herramientas se observa en la Figura 1.1, cuyo contenido se basa en [6].

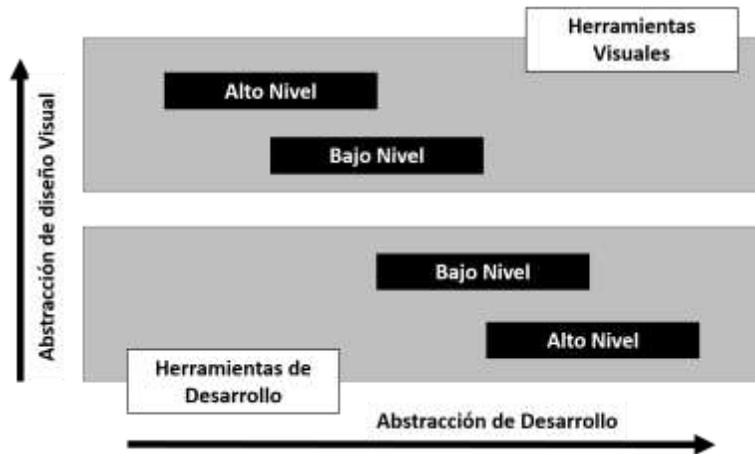


Figura 1.1 Relación de los niveles de abstracción en las Herramientas Visuales y Herramientas de Desarrollo

1.1.3 Interfaces humano-máquina

Las interfaces humano-máquina son tecnologías concernientes al diseño, evaluación y despliegue de sistemas computacionales para el uso humano utilizando fenómenos del entorno del mundo real. Es una parte de las ciencias computacionales y como tal es en sí misma una disciplina cuyo principal objetivo es crear software que establezca una estructura de comunicación entre los usuarios humanos y las máquinas que utilizan, usando procesos de ingeniería para su diseño e implementación. Para esto, por medio de entrenamiento, las interfaces obtienen información del comportamiento de su usuario y lo traducen al lenguaje de la máquina para su ejecución [7].

Las interfaces humano-máquina se desarrollaron rápidamente en los años recientes, cada vez más investigadores se interesan en utilizarlas y desarrollar técnicas para implementarlas en el sector educativo [8].

1.1.3.1 Captura de movimiento

La captura de movimiento registra movimientos en vivo por medio del rastro de puntos clave en un objeto en el espacio, los traduce a términos matemáticos utilizables por un software para obtener un modelo tridimensional que represente dichos movimientos [9]. La mayoría de estos sistemas asumen una estructura cinética de tipo humanoide representando un número específico de coyunturas con un rango de movimiento determinado [10].

1.1.3.2 Control por gestos manuales

Comparada con las técnicas utilizadas en el mundo físico, el reconocimiento gestual se considera una alternativa razonable, al permitir una interacción aparentemente directa con los objetos creados utilizando las manos, como en un entorno de la vida diaria. El reconocimiento de control gestual es una interpretación matemática del movimiento humano por un dispositivo de cómputo. La interpretación gestual que goza de mayor popularidad es el reconocimiento facial, reconocimiento de voz, rastreo visual, aunque los gestos manuales tienen mayor eficiencia y aceptación en su utilización para programas educativos [8].

Los gestos manuales son uno de los aspectos del control gestual, y son un medio natural de comunicación utilizados en un amplio rango de aplicaciones. Se clasifican en gestos y posturas, siendo el primero una serie de posiciones tal como el movimiento pendular del índice al indicar negación, y el posterior a una posición estable de la mano como cruzar los dedos índice y medio [11]. Tienen un campo muy amplio de investigación debido a la forma natural en la que facilitan la comunicación. Existen dos aproximaciones para la investigación y uso en interfaces humano-máquina:

- Sensores basados en dispositivos de ropa inteligente.
- Sensores basados en visión de cámara.

Estas tecnologías permiten el desarrollo de software con aplicaciones en la medicina, la traducción de lenguaje de señas, la domótica, la realidad virtual y aumentada. El control de los objetos por medio de las manos es natural, flexible y ultimadamente sencillo [11]. Entre los sistemas de interfaz más utilizados para la identificación de movimientos manuales se encuentran *Tap Strap*[®] y *Leap Motion*[®] siendo el primero basado en ropa inteligente y el posterior en visión de cámara.

1.1.4 Procesos de aprendizaje

El concepto de aprendizaje está ligado a la capacidad de adquirir conocimiento y comprender conceptos, adquisición de habilidades complejas y el desarrollo de estrategias para la resolución de problemas. El aprendizaje se sitúa en un contexto

particular y por tanto no es general para todas las situaciones, distintos métodos de aprendizaje no necesariamente otorgan los mismos resultados en todas las situaciones [12].

Las experiencias de aprendizaje son manifestaciones del proceso de pensamiento de una persona cuando se enfrenta a una tarea específica, así como la forma en la que ésta procesa la información relacionada a dicha tarea. La constante repetición de patrones específicos de resolución de problemas de manera periódica y correcta facilita el aprendizaje de dichos patrones, aumentando la sensación de confianza en el estudiante. La creación de confianza en el estudiante determina el éxito o el fracaso del proceso de aprendizaje implementado [12].

El aprendizaje es comúnmente representado como un ciclo de 4 etapas, las cuales se encuentran ilustradas en la Figura 1.2, basada en la Figura 1 encontrada en [13]. En la primera etapa se tiene una experiencia concreta que servirá en la segunda etapa como modelo de observación y para la obtención de postulados, estas observaciones servirán en la tercera etapa que generalizará los postulados y los planteará como hipótesis, para que en la cuarta etapa esta hipótesis sea puesta a prueba, ya sea con la experiencia concreta anteriormente obtenida o con una nueva que plantee un problema similar [13]. Para tener un aprendizaje efectivo, el estudiante debe contar con las siguientes habilidades cognitivas:

- Capacidad de adquirir información concreta.
- Capacidad para observar y reflexionar.
- Capacidad para conceptualizar ideas abstractas.
- Capacidad para experimentar de forma activa.



Figura 1.2 Etapas del Proceso de Aprendizaje

1.1.4.1 Comprensión lectora

La comprensión del lenguaje escrito está íntimamente relacionada a la comprensión del habla, aunque es posible que la comprensión lectora sobrepase a la del habla. En primera instancia es necesario que el estudiante en su etapa temprana de aprendizaje, aprenda a interpretar signos e identificar palabras. En cuanto los infantes logran decodificar las palabras, el factor que delimita la comprensión lectora se vuelve la asociación de las palabras habladas con su simbología escrita. Por tanto, está establecido que la comprensión lectora es el producto de la identificación de palabras y la comprensión del habla [14].

La comprensión se desarrolla conforme el lector establece una imagen mental de un texto escrito, dicha imagen es un modelo que ejemplifica el tema del texto. Los temas se determinan a través de la extracción de ideas a partir de oraciones, contrastadas las unas con las otras, que son complementadas con inferencias necesarias para obtener una idea coherente. Se considera que todo modelo mental adquirido por el lector es una galería de oraciones específicas del texto y de ideas generadas a partir de la lectura [14].

Se ha demostrado que la experiencia en la lectura determina la facilidad con la que un género literario es comprendido, siendo que, en edades tempranas, la narrativa goza de una mayor facilidad de comprensión al situarse en tiempos y espacios determinados. Esto se debe a que el aprendizaje a través de estructuras narrativas se basa en la comprensión de las motivaciones y la capacidad de empatizar, que es adquirida de forma típica a la edad de 3 años [14].

Por tanto, el proceso de aprendizaje de la escritura debe estar profundamente relacionado con el desarrollo y la comprensión de estructuras narrativas, al menos en edades tempranas, de manera que facilite la decodificación de palabras complejas y su asimilación para el fortalecimiento de la comprensión lectora [14].

1.1.4.2 Trastornos de aprendizaje de la lectoescritura

Los trastornos de aprendizaje son desórdenes en el desarrollo integral que aparecen durante el periodo de formación de un niño, imposibilitan la adquisición de conocimiento dificultando la comprensión e internalización de conceptos [15]. Los niños desarrollan una comprensión básica de los aspectos de su lenguaje materno conforme aumenta su capacidad de estructurar palabras y oraciones cada vez más complejas, el desarrollo del lenguaje constituye uno de los aspectos más importantes de la formación intelectual, por ejemplo, se ha descubierto que la diferencia de la estructura en la que se utilizan los caracteres chinos en su respectivo idioma, de los caracteres latinos en el idioma inglés, predispone estructuras de pensamiento con base en el desarrollo lingüístico de una persona [16].

El correcto aprendizaje del idioma nativo se basa en dos principios: oír y escribir. Hay una relación directa entre ambas actividades, la primera hace uso de una actitud pasiva y requiere de una concentración plena, que es comprometida sobre todo en las edades más tempranas al intervenir con la falta de conceptos espacio-temporales del infante, es decir, el infante no comprende el concepto del tiempo, toda información que reciba en un instante o todo objeto que está a su vista, deja de existir en cuanto sale de su percepción, en cuanto se deja de hablar o el objeto se oculta. La segunda actividad, escuchar, tiene una aplicación más efectiva al ser una actitud proactiva que hace uso de las capacidades motrices y visuales del infante. Por medio de la escritura y un estímulo auditivo el infante comprende y domina a mayor velocidad la estructura del lenguaje [17].

Un trastorno de aprendizaje en el ámbito de la lectoescritura deviene por lo general en analfabetismo funcional y problemas con la estructuración de discursos e ideas complejas, pero el uso de estrategias de aprendizaje que impliquen la utilización de

las habilidades motrices del infante y sobre todo, de la interacción física con elementos ajenos a su propio cuerpo facilita el dominio del idioma nativo [17].

2.1.1 Métodos de enseñanza del idioma español

Se reportan dos procesos principales que intervienen en la lectura: el reconocimiento de las palabras y el acceso al léxico (comprensión de las palabras). Los lectores experimentados realizan un análisis de los rasgos físicos y trazos que visualizan, a través del cual buscan en su memoria y reconocen los patrones visuales que identifican como palabras y relacionan con sonidos. Estos sonidos se asocian con conceptos que también se encuentran almacenados en su memoria. [18]

Se reportan distintas teorías sobre el desarrollo de la lectura y la escritura, por un lado se encuentran los modelos discretos (también llamados secuenciales), que conciben el aprendizaje de la lectoescritura como una serie de etapas que utilizan distintas estrategias de enseñanza, y por otro lado los modelos continuos que sostienen que no existen fases en el proceso de aprendizaje de la lectoescritura. [18]

En el caso de este proyecto se hace énfasis en los modelos discretos.

2.1.1.1 Modelos de enseñanza discreta

La idea de que empezar a leer implica superar etapas, proporciona bases para diferenciar un aprendizaje normal de uno lento o acelerado, permitiendo el análisis de casos especiales en los que los estudiantes tengan problemas de aprendizaje [18]. Este proceso de desarrollo de las habilidades de la lectoescritura que proponen los modelos de enseñanza discreta se divide principalmente en cuatro etapas:

- Etapa 0. Que abarca el periodo en el que los niños adquieren el lenguaje oral, aprenden a identificar objetos con sonidos. También identifican los objetos con conceptos y se familiarizan con la representación física de las letras. Aprenden a identificar logotipos [18].
- Etapa 1. Los niños identifican los símbolos de las letras con sus fonemas, aprenden a identificar los símbolos independientemente de la posición en la

que se encuentra, en una palabra. Aprenden a interpretar símbolos como fonemas y a transformar palabras impresas en sonidos específicos [18].

- Etapa 2. Los lectores empiezan a adquirir fluidez en su lectura, ahora son capaces de interpretar automáticamente los símbolos en conjunto como palabras [18].
- Etapa 3. Los lectores han dominado por completo los mecanismos del proceso de lectura y ahora son capaces de adquirir nueva información a través de esta, así como de expresar sus ideas a partir de la escritura [18].

2.1.1.2 Programa Kantor

En los programas tradicionales para la enseñanza de la lectoescritura, se propone que los alumnos aprendan a identificar las letras a través de ejercicios de trazado y reconocimiento de las mismas. Se pretende que el alumno identifique las letras de forma independiente del contexto, el fondo y la forma en la que estas se muestran, por eso se requiere que los ejercicios presentados al alumno le permitan discriminar las propiedades definitorias de una letra, debido a que en el abecedario hay letras que guardan familiaridad gráfica con otras, como en el caso de la m y la n, la p y la q. [19]

El método Kantor se basa en los principios de Teoría Interconductual, y han sido aplicados con éxito en la Escuela Activa Integral de la ciudad de Culiacán desde hace más de 20 años [19]. Se especifica que el método de enseñanza de las letras se divide en tres etapas:

- La enseñanza de las vocales.
- La enseñanza de las consonantes por medio de las sílabas.
- La enseñanza de las mayúsculas.

A su vez, la enseñanza de las silabas se divide en diversas secciones:

- Consonantes que tienen fonema similar: b-v; y-i; c-s-z; ll-y.
- La letra ñ.

- La letra g que dependiendo de la vocal suena diferente.
- Consonante con diferente grafía y fonema, así como de grafía similar y fonema diferente dependiendo de su ubicación: r-rr.
- Consonante con doble grafía y un solo fonema: ch.
- Consonante sin fonema: h.
- Consonante de uso poco frecuente.

1.2 Situación tecnológica, económica y operativa de la empresa

El Instituto Tecnológico de Orizaba es una institución pública que depende de la Secretaría de Educación Pública SEP y pertenece al Tecnológico Nacional de México. Se encuentra ubicado en Oriente 9, Colonia Emiliano Zapata, en la ciudad de Orizaba, Veracruz. Esta institución ofrece carreras de licenciatura, maestría y doctorado. En el área de maestrías ofrece los posgrados de Maestría en Ingeniería Electrónica, Maestría en Ingeniería Industrial, Maestría en Ciencias en Ingeniería Química, Maestría en Ingeniería Administrativa y la Maestría en Sistemas Computacionales. Asimismo, ofrece un Doctorado en Ciencias de la Ingeniería.

1.3 Planteamiento del problema

Las técnicas de Realidad Aumentada (RA) permiten la aplicación de elementos virtuales sobre una representación de la realidad física. La RA se utiliza con éxito para proporcionar la asistencia necesaria al usuario para llevar a cabo procedimientos difíciles o comprender problemas complejos lo cual es muy útil en procesos educativos. La principal ventaja de la RA sobre los métodos de enseñanza más tradicionales es que los alumnos realmente ven y escuchan información digital complementaria, además de que manipulan intuitivamente la información virtual, lo que les permite repetir una parte específica del aumento tantas veces como deseen.

La efectividad de la aplicación de la RA en la educación depende de:

- La forma del material de estudio, de modo que fomente el interés de los estudiantes por el contenido: visualización en 3D, RA, modelos animados en 3D;

- El contenido de calidad de imágenes y modelos 3D;
- La precisión de los detalles desde el punto de vista científico: desarrollo de ciertos conocimientos y habilidades, y consecución de un resultado concreto de la investigación.
- La efectividad en la forma de interacción.

Relativo a este último punto, los avances tecnológicos ofrecen hoy en día nuevos mecanismos de interacción humano-computadora, siendo el control de gestos con las manos el cual se realiza a través del análisis de imágenes o bien utilizando dispositivos de interacción como *Leap Motion*[®] y *Tap Strap*[®].

En este proyecto se buscó incorporar técnicas de realidad aumentada con interacción a través de gestos con las manos para lograr una interfaz que permita a los estudiantes examinar y experimentar con el material de enseñanza virtual de una manera natural.

1.4 Objetivo General y Objetivos Específicos

A continuación, se exponen el objetivo general del proyecto de tesis y sus objetivos específicos.

1.4.1 Objetivo General

Desarrollar una aplicación de realidad aumentada con interacción basada en gestos con las manos para apoyar el aprendizaje de la lectoescritura.

1.4.2 Objetivos Específicos

- Analizar trabajos relacionados para conocer el estado actual del conocimiento sobre el tema.
- Analizar las plataformas existentes para el desarrollo de aplicaciones con Realidad Aumentada y elegir una para el proyecto.
- Analizar el uso de *Leap Motion*[®] y *Tap Strap*[®] para la captura de gestos con las manos y seleccionar el más apropiado para el proyecto.
- Definir la arquitectura y el modelo de entrenamiento para el sistema.

- Diseñar y desarrollar las funcionalidades del sistema.
- Recopilar un conjunto de datos para el entrenamiento del sistema.
- Diseñar y ejecutar los experimentos para al menos un caso de estudio.
- Describir los resultados y conclusiones obtenidas tras la implementación del caso de estudio.

1.5 Justificación

En 2018 los estudiantes de México obtuvieron en promedio un puntaje de 420 en la prueba PISA de comprensión lectora que se encuentra en el segundo nivel de competencia, esto en contraste con el 487 del promedio de la OCDE, siendo esto un nivel 5. Solo el 1 por ciento de los estudiantes demostró un rendimiento superior al promedio. En el nivel actual de la población estudiantil mexicana, se tiene la capacidad de identificar ideas principales en un texto de longitud moderada y recopilar información de los textos cuando se indica explícitamente [20].

El presente proyecto es de gran importancia para la población estudiantil, puesto que, al enfocarse en el desarrollo de las habilidades de la lectoescritura, su objetivo es el desarrollo de una herramienta que captura imágenes del mundo real en vivo, proporcionando modelos de lectura que dotan al usuario de una experiencia vívida por medio de tecnología de realidad aumentada y permite la interacción con los mismos a través de interfaces humano-máquina.

Capítulo 2. Estado de la práctica

En la actualidad, el uso de tecnologías de realidad aumentada y de interfaces humano-máquina para el aprendizaje se encuentra en aumento. No es de sorprenderse, porque continuamente se busca que las tecnologías de computación sirvan de apoyo a las actividades humanas y no hay nada más cercano a esto que la interacción entre el mundo real, y modelos virtuales proyectados sobre la realidad.

A fin de identificar las tecnologías con mayor eficiencia en la actualidad para el desarrollo de proyectos similares, y así realizar por cuenta propia una propuesta de solución, en este capítulo, se presentan resúmenes de trabajos de desarrollo que tienen relación directa, indirecta, parcial o total con el proyecto propuesto en este documento.

2.2 Trabajos relacionados

La geometría es un área sumamente importante de las matemáticas, y como tal, muchos estudiantes califican sus experiencias en el aprendizaje de dicha área como poco placenteras y difíciles. En [8] se propuso un marco de trabajo para el aprendizaje del área matemática de la geometría usando un *software* de realidad aumentada e interfaces humano-máquina de reconocimiento de gestos manuales. En el desarrollo de su investigación se utilizó fundamentalmente el dispositivo *Leap Motion*[®] para el reconocimiento de los gestos, pues su SDK (*Software Development Kit*) incluye una serie de funciones que permite el reconocimiento de los movimientos de la mano. Se utilizaron las bibliotecas de *ARToolkit* para el reconocimiento de los marcadores de posición de los modelos de figuras geométricas, así como para la captura de video a través de una cámara y la renderización de los modelos anteriormente mencionados en el video capturado. El software desarrollado permite la creación de objetos sólidos en tres dimensiones y la modificación de sus propiedades con interacción directa sobre el modelo renderizado. Posteriormente se comparó con *software* de modelado 3D GeoGebra y Cabri3D obteniendo resultados beneficiosos en términos de renderización e interacción con los modelos.

En [21] se discute la importancia de la ubicación espacial para la comprensión de conceptos matemáticos complejos y se desarrolló una aplicación Android basada en realidad aumentada para la enseñanza de geometría vectorial llamada *cleARmaths*. Para su desarrollo se usó la plataforma de desarrollo de videojuegos Unity que permite la captura de imagen y la renderización en tiempo real, gracias al cual se estructuró un código de comportamiento en clases que heredan de la superclase *Monobehaviour*, que permite la creación de componentes con la propiedad de Objetos de Juego. Para el respectivo rastreo de los marcadores de imagen se utilizó el SDK Vuforia. Dichos marcadores de imagen son códigos QR (*Quick Response*) diseñados para compilar ecuaciones, se generan a partir de una página web codificada en Java. Se usaron ejercicios comparativos, probados en aula real para verificar la facilidad de uso de la aplicación en el aula, siendo encontrada de fácil uso por la población de prueba en comparación con GeoGebra y Construct3D.

En [22] se desarrolló una red neuronal artificial que reconoce las expresiones humanas como instrucciones, para el aprendizaje de anatomía. Se propone un proceso de entrenamiento de la aplicación utilizando computación en la nube para su interacción con distintos dispositivos, facilitando el proceso de desarrollo. Para el desarrollo del proyecto se utilizó la cámara de profundidad *RealSense*, que con sus funciones de proyector de láser infrarrojo permite detectar la forma de los objetos escaneados para crear un modelo 3D con el método de representación de superficies basado en puntos clave. También se usó para programar los movimientos manuales que sirven para manipular los modelos desplegados a través de la captura de movimiento y calculando las coordenadas por píxeles. Mientras que para mostrar los modelos de la red neuronal se aplicó la biblioteca de Python *TensorFlow*, que permitió la conexión con el dispositivo HoloLens, que cuenta con la tecnología para rastrear los movimientos de las manos a través de su cámara integrada y su pantalla interactiva permite desplegar la red neuronal sobre la imagen real de fondo, sin necesidad de directamente grabar, solo renderizar.

En [23] se presenta una herramienta para el control de dispositivos IoT (*Internet of Things*, Internet de las Cosas) a través de gestos de las manos, usando RA. La capacidad de la RA de crear interacción inmersiva, impulsa ampliamente la capacidad de los dispositivos IoT para el control visual de los modelos interactivos. En esta herramienta propuesta *MagicHand*, se creó un prototipo de programa para la comunicación entre estos dispositivos y una interfaz de IoT, facilitando la interacción a través del sonido y sistema de iluminación. Para la interpretación de los movimientos manuales del usuario se implementó una red neuronal artificial. *MagicHand* también fue equipado con una cámara detectora de objetos que es capaz de evaluar códigos QR para la creación de objetos virtuales, además de identificar figuras geométricas para el análisis de sus formas. Los dispositivos IoT están actualmente limitados a interfaces web y dispositivos móviles, por lo cual *MagicHand* se puso a prueba en dos dispositivos móviles con resultados satisfactorios.

En la aplicación de Telemedicina (prestación de servicios médicos a distancia), el funcionamiento correcto de dispositivos médicos es elemental, así como la correcta interacción entre el médico y el paciente. En [24] se expone el uso de RA en la creación de una aplicación de telemedicina para el entrenamiento de usuarios sin experiencia al realizar estudios de electrocardiograma. Se colocan puntos de control a un maniquí que, conectados al dispositivo, permiten el reconocimiento espacial por medio de una cámara. Además de contar con cajas de texto e indicaciones audibles para auxiliar al usuario para realizar la secuencia de la prueba. También se agregaron comandos de voz para que el usuario mejore la operatividad del sistema. Al probarse la aplicación en 10 voluntarios sin experiencia previa, quedó demostrado que el uso de la RA favoreció ampliamente el aprendizaje de los mismos, pues se realizó un proceso de prueba en la aplicación y uno posterior con pacientes reales, en el cual el margen de diferencia en la colocación de los electrocardiogramas fue de 4 milímetros.

El objetivo de [25] fué el desarrollo de una aplicación de RA con captura de movimiento para el entrenamiento en la creación de Canang Sari (ofrenda hindú).

Dicha aplicación ofrece información acerca de distintos tipos de *canang* e imágenes de ejemplo. Para su desarrollo se utilizó el SDK de Vuforia, que combina la realidad objetiva y el objeto virtual a través del uso de teléfonos inteligentes y la lectura de códigos QR. Las imágenes de ejemplo se desarrollaron a través de *Adobe Illustrator* e implementados en RA a través de Unity. Se utilizaron marcadores multi-objetivo, para el desarrollo e identificación de diferentes tipos de Canang. Para su desarrollo se aplicó la metodología en cascada.

En Indonesia, la Educación del Carácter es una materia del sistema educativo enfocada en el aprendizaje de la ética, valores morales y la educación cívica. En [26] se tomó en cuenta que las generaciones más jóvenes de Indonesia crecieron expuestos a Tecnologías de Información y Comunicación, y por tanto la implementación de las mismas a través de la RA beneficiaría ampliamente el aprendizaje de la materia anteriormente mencionada. Se desarrolló una aplicación que a través de RA estimula a los estudiantes de la materia educación del carácter utilizando estímulos audio-visuales. Para el desarrollo del programa se realizó revisión de la literatura existente en el momento, que culminó en la utilización de las herramientas Unity y Vuforia como plataformas para el proyecto. Como resultado se obtuvo una aplicación Android con la capacidad de capturar marcadores de movimiento, lo cual facilitó la función de los maestros para la impartición de la materia.

Aquellos que estudian el idioma inglés suelen pasarse por alto la importancia de comprender la fonética en el proceso de aprendizaje. La falta de comprensión del sonido de la lengua provoca errores de pronunciación que afectan la comunicación, tanto oral como escrita. Por eso, en [27] se describe un modelo de enseñanza basado en RA para facilitar el proceso de aprendizaje de la fonética en el idioma inglés. Para el desarrollo del modelo se utilizó la aplicación Aurasma que recientemente cambió su nombre a *HP Reveal*. Esta tecnología que combina objetos virtuales y videoclips en un entorno real y al mismo tiempo virtual revela lo importante que es el estímulo físico en el aprendizaje de conceptos sencillos. Se utilizaron marcadores de imágenes de uso libre descargadas de Internet en

contraste con todos los proyectos mencionados anteriormente que optaron en su mayoría por generación a través de códigos QR, que fueron editadas en forma de tarjetas relámpago. Al mostrarlas en la pantalla del dispositivo móvil, se observa un modelo 3D de la imagen en la tarjeta, las palabras, así como una grabación de la pronunciación correcta.

En la provincia de Aceh al norte de Indonesia la falta de conocimiento sobre los objetos tradicionales de la región, así como de su arquitectura, por el público general permiten la disminución de los valores culturales de la zona. Uno de los tantos esfuerzos realizados para mejorar la percepción de los acehnenses (habitantes de la región de Aceh), sobre su propia cultura se describió en [28], con la realización de una aplicación de RA para la visualización de construcciones y objetos acehnenses tradicionales en animación 3D. El método usado es de rastreo de marcadores, mientras que el software usado es Vuforia, Blender 3D y Unity. En la aplicación desarrollada, los usuarios visualizan animaciones 3D de arquitectura de Aceh, así como objetos tradicionales de la región. La herramienta desarrollada se despliega en dispositivos Android, con una capacidad para la identificación de marcadores de hasta 60 cm de distancia de la cámara. Para el rastreo de marcadores se utilizó la biblioteca de ARToolkit para poder generar un modelo virtual que sea desplegable en el mundo real por medio de la identificación de coordenadas en relación a su distancia con la cámara.

El uso de la tecnología en el aprendizaje del idioma inglés como segundo idioma, tomando en cuenta las habilidades de la escritura, se explora ampliamente. En [29] se diseñó una aplicación de realidad aumentada para la escritura ubicua consciente del contexto, enfocada en incrementar la motivación y la memoria a largo plazo. Se compararon las escrituras obtenidas y las percepciones de los estudiantes a través de la aplicación contra las obtenidas a través del método clásico de clases. Los dos estilos tuvieron diferencias en la percepción de los estudiantes, demostrando que la aplicación desarrollada favoreció la memoria a largo plazo, la motivación y la percepción de mejoría en el proceso de la escritura. Se determinó que la herramienta desarrollada debía incorporarse en cursos a distancia de aprendizaje

del idioma inglés para favorecer los resultados del aprendizaje. También se incluyó un procedimiento en cinco pasos para ayudar a los estudiantes del idioma inglés como lengua extranjera a integrar la escritura ubicua en sus métodos de aprendizaje. Para el desarrollo de la aplicación utilizaron el *software* Unity.

La RA permite a los usuarios interactuar con objetos virtuales con un objeto real porque tiene la capacidad de mezclarlos en un entorno intermedio. De manera que para proveer una interacción natural e intuitiva en un entorno de RA las manos fueron exploradas como medio de interacción. En [30] se propuso la creación de un entorno virtual en el que objetos sean manipulados por gestos de la mano en RA. Para ello se realizó un estudio de los trabajos disponibles enfocados en la manipulación a través de gestos manuales, luego se utilizó *Leap Motion*[®] para capturar las expresiones manuales del usuario que permitirán la manipulación de los objetos en la aplicación de RA desarrollada. La fase de evaluación se condujo tomando en cuenta la percepción del usuario, gracias a lo cual se determinó que logra su propósito de producir experiencias satisfactorias en la manipulación de modelos 3D a través de la manipulación gestual a través del uso de *Leap Motion*[®]. Para la implementación de RA utilizaron el SDK Vuforia.

Numerosos estudios sitúan a la RA en el contexto de la educación y muy pocos de estos demostraron su impacto en cuanto al aprendizaje de la escritura se refiere, en [31] se evaluó el uso de RA en un contexto universitario comparando la calidad de la escritura en distintos grupos estudiantiles, según su habilidad. Antes del experimento se expuso a un grupo de 32 estudiantes a una actividad de escritura sin realidad aumentada, mientras otro grupo tuvo la asignación de la prueba haciendo uso de una herramienta de RA. Después se realizaron entrevistas individuales con un grupo de 5 estudiantes para determinar su actitud con la actividad realizada en RA. Después de revisar las pruebas se determinó que las calificaciones obtenidas por los estudiantes con promedio más bajo obtuvieron puntaje similar a los estudiantes con mayor promedio, lo cual indica que la RA favorece el desempeño de las personas en cuanto al desempeño al escribir. Para

el desarrollo de los modelos de realidad aumentada se utilizó la aplicación *HP Reveal*, conocida como Aurasma.

En [32] se desarrolló un sistema que permite la comunicación entre dos trabajadores a través de gestos manuales. Este sistema usa realidad aumentada para desplegar frente a los ojos de un trabajador la mano de su auxiliar, permitiendo que el primero observe los gestos del último para recibir instrucciones del mismo. Para determinar la utilidad de este sistema se realizaron pruebas comparativas en situaciones de trabajo remoto. Para el despliegue del sistema utilizaron las interfaces *HoloLens* y *Leap Motion*[®]. Se usaron dos modelos de pruebas: el primero por medio de instrucciones de líneas indicadas con las manos y el segundo con instrucciones de gestos de manos, aunque los resultados fueron muy parecidos se determinó que la instrucción por medio de trazado de líneas fue levemente más comprensible para el trabajador que los gestos manuales.

2.3 Análisis comparativo

A continuación, en la tabla 2.1 se muestra un análisis comparativo de los artículos consultados para verificar el estado del arte citado en la sección anterior.

Tabla 2.1 Análisis comparativo de los artículos del Estado del Arte

Artículo	Problema	Contribución	Tecnologías	Resultado
Le y Kim. [8]	Las experiencias poco placenteras y difíciles de los estudiantes con el aprendizaje de la geometría.	Marco de trabajo para el aprendizaje de la geometría usando realidad aumentada	<i>Leap Motion</i> ® y <i>ARToolkit</i>	Se desarrolló un marco de trabajo que es útil para estudiar geometría de forma funcional que mostró mejores resultados que procesos de enseñanza.
Schutera et al. [19]	La dificultad de comprender conceptos matemáticos relacionados con cálculos vectoriales.	Aplicación de Realidad Aumentada en la comprensión de modelos matemáticos de 3 dimensiones	Unity y Vuforia	La aplicación demostró facilitar al estudiante la comprensión de objetos tridimensionales en una manera entretenida.
Karambakhsh et al. [22]	La aplicación de Realidad Aumentada para mejorar los procesos de enseñanza en las instituciones médicas.	Una Red Neuronal Artificial para el reconocimiento de gestos de las manos como instrucciones	Cámara de profundidad <i>RealSense</i> , biblioteca de Python <i>Tensorflow</i> , gafas de Realidad Aumentada <i>HoloLens</i>	La Red Neuronal es capaz de detectar con mayor facilidad nuevos patrones de movimiento además de demostrar la utilidad de la realidad aumentada.
Sun et al. [23]	Se presenta una herramienta de Realidad Aumentada utilizando control de gestos manuales a través de dispositivos IoT, cuyo nombre es <i>MagicHand</i>	La implementación de una Red Neuronal entrenada a través de un dispositivo IoT que permite interactuar con modelos 3D.	Red Neuronal Artificial, <i>HoloLens</i>	Se detectó a través de evaluación cuantitativa que <i>MagicHand</i> detecta con exactitud instrucciones manuales que permiten la interacción con modelos en tres dimensiones, con mayor rapidez que otros productos similares en el mercado.

Tabla 2.1 Análisis comparativo de los artículos del Estado del Arte (continuación)

Artículo	Problema	Contribución	Tecnologías	Resultado
Sun et al. [23]	Se presenta una herramienta de Realidad Aumentada utilizando control de gestos manuales a través de dispositivos IoT, cuyo nombre es <i>MagicHand</i>	La implementación de una Red Neuronal entrenada a través de un dispositivo IoT que permite interactuar con modelos 3D.	Red Neuronal Artificial, <i>HoloLens</i>	Se detectó a través de evaluación cuantitativa que <i>MagicHand</i> detecta con exactitud instrucciones manuales que permiten la interacción con modelos en tres dimensiones, con mayor rapidez que otros productos similares en el mercado.
Bifulco et al. [24]	En la aplicación de telemedicina el uso correcto de los equipos médicos es elemental, la RA ofrece un soporte válido para el entrenamiento en el uso de estos dispositivos a larga distancia.	Desarrollo de una aplicación de RA para la capacitación de usuarios no entrenados en el uso de dispositivos médicos.	Monitor virtual de retina, cámara HD Logitech	Durante las pruebas se concluyó que la herramienta era efectiva en la enseñanza a usuarios sin entrenamiento en la aplicación de pruebas de electrocardiograma, pudiendo ser adaptada para el entrenamiento de otros dispositivos médicos.
Putra y Putra. [23]	Desarrollar una aplicación de RA para la enseñanza de preparación de ofrendas Canang Sari.	La creación de modelos 3D a partir de marcadores multi-objetivo, además de proveer imágenes e información de apoyo para el entendimiento del modelo mostrado.	Unity, Vuforia.	La aplicación realiza las lecturas de los modelos a través de códigos QR que almacenan también la información de cada variante. Esta aplicación se ejecuta en diferentes dispositivos <i>Android</i> .

Tabla 2.1 Análisis comparativo de los artículos del Estado del Arte (continuación)

Artículo	Problema	Contribución	Tecnologías	Resultado
Sarosa et al. [24]	Desarrollo de una aplicación de RA enfocada en auxiliar a la enseñanza de valores cívicos y éticos.	Aplicación que proporciona información audiovisual a través de un motor de realidad aumentada utilizando un dispositivo móvil	Vuforia, Unity, Blender 3D	El uso de Vuforia y Unity proveen amplias ventajas para la creación de entornos de RA, como la disminución de costos y tiempo, además de eliminar la necesidad de monitores visuales de retina.
Nugraha et al. [25]	En el aprendizaje del idioma inglés, los estudiantes suelen pasar por alto la importancia de la comprensión de las palabras en el proceso de la adquisición de vocabulario.	Una aplicación de realidad aumentada para facilitar el proceso de aprendizaje de la fonética del idioma inglés.	HP Reveal, antes Aurasma.	Se demostró que el uso de realidad aumentada como alternativa de medios de aprendizaje para el vocabulario del idioma inglés tiene la capacidad de proveer mayor entendimiento y mantener la atención de niños aprendiendo inglés.
Mursyidah et al. [26]	En la provincia de Aceh la falta de conocimiento sobre su propia cultura inhibe el desarrollo de la misma así como su publicitación, disminuyendo el enriquecimiento cultural de la zona.	Aplicación de RA para la exploración de construcciones tradicionales de Indonesia así como la visualización de objetos con valor cultural e información de los mismos.	Unity, Vuforia, Blender3D, ARToolkit.	La herramienta se despliega en dispositivos Android, con una capacidad para identificar marcadores hasta 60 cm de distancia de la cámara.

Tabla 2.1 Análisis comparativo de los artículos del Estado del Arte (continuación)

Artículo	Problema	Contribución	Tecnologías	Resultado
Lin et al. [27]	Pocos estudios sobre el aprendizaje del idioma inglés como lengua extranjera incorporan tecnología del entorno.	Una aplicación de RA para la escritura ubicua consciente del contexto que se enfoca en incrementar la memoria a largo plazo y la motivación del estudiante.	Unity.	Se determinó que la herramienta era efectiva en el aprendizaje del idioma inglés, sobretodo en modelos de enseñanza a distancia.
SyafiqahSafiee e Ismail. [28]	Proveer un medio de interacción natural e intuitivo a través de los gestos manuales para la manipulación de objetos 3D en entornos de RA.	Aplicación que permite la interacción del usuario con el objeto virtual a través del uso de <i>Leap Motion</i> ® para la captación de gestos manuales.	<i>Leap Motion</i> ®, Vuforia.	Se logró la interacción con el objeto virtual generado a través del uso de <i>Leap Motion</i> ®, permitiendo también la modificación de dicho objeto, para su uso en el diseño de interiores.
Allagui. [29]	Determinar el impacto potencial del uso de la RA en el desempeño de los estudiantes en sus procesos de escritura.	Se estableció que incluso aplicaciones educativas de RA con bajo presupuesto pueden proveer motivación e influir en la creación de nuevas ideas por parte de los estudiantes durante su proceso de escritura al estimular su sentido espacial.	<i>HP Reveal</i> , antes Aurasma	Se logró determinar que gracias al uso de RA, los estudiantes con menor promedio fueron capaces de obtener puntajes similares en pruebas escritas que los estudiantes de mayor promedio, implicando que la interacción con objetos virtuales estimula los sentidos y creatividad de los estudiantes.

Tabla 2.1 Análisis comparativo de los artículos del Estado del Arte (continuación)

Artículo	Problema	Contribución	Tecnologías	Resultado
Yamada y Chandrasiri. [30]	Desarrollar un sistema de comunicación a distancia entre trabajadores utilizando instrucciones a través de gestos manuales.	Se codificó un módulo para la visualización de gestos de un trabajador en el <i>HoloLens</i> de otro, permitiendo la comunicación a distancia.	<i>HoloLens</i> , <i>Leap Motion</i> ®.	Se determinó que el uso de un código específico de trazos no traducibles por el sistema, pero comprensibles para el trabajador es más eficiente que la lectura de gestos realizada a través del SDK de <i>Leap Motion</i> ®, aunque la diferencia entre una y otra es mínima.

Después de realizar el análisis comparativo se concluye que las aplicaciones de RA se ocuparon en diversos entornos relacionados con la educación, y que la mayoría de estas ocasiones utilizan control gestual para la interacción con la RA, aunque no siempre se ocupan los mismos dispositivos. Basándose en toda esta información se desarrolló la propuesta de solución que se expone a continuación.

2.4 Propuesta de solución

A continuación, se describen las tecnologías, los marcos de trabajo y lenguajes de programación considerados para la realización del programa presentado en este trabajo de tesis.

2.4.1 Análisis de las tecnologías de información

En esta sección se presenta un análisis de todas las tecnologías disponibles para el desarrollo de la solución propuesta en este capítulo.

2.4.1.1 Lenguajes de programación

En este apartado se describe el lenguaje de programación que, por su compatibilidad con los motores de desarrollo, es la alternativa viable para el desarrollo de la solución al problema planteado.

2.4.1.1.1 C#

C# es un lenguaje orientado a objetos y a componentes, diseñado para construir aplicaciones robustas que se ejecutan en .NET. Como su nombre lo indica tiene su origen en la familia del lenguaje C, por lo cual es similar a los demás lenguajes derivados de este último, como C++ o Java. Anders Hejlsberg lo diseñó en el año 2000 y actualmente cuenta con un estándar ISO, el ISO/IEC 23270.

Su diseño permite acceder a bajo nivel, permitiendo trabajar con punteros de memoria y elementos físicos de los dispositivos en los que se trabaja. Al diseñarlo para su uso en .NET, que es una plataforma utilizada para crear aplicaciones, C# es ideal para el diseño de aplicaciones robustas y ligeras en dicha plataforma. Es un lenguaje que permite la creación y uso de componentes de forma práctica y sencilla, además de evolucionar continuamente para la implementación de nuevos paradigmas de programación [33].

2.4.1.2 Entornos de Desarrollo Integrado (IDE)

Un IDE es un ambiente de desarrollo integrado por un conjunto de herramientas necesarias para el desarrollo de aplicaciones, su propósito principal es aumentar la productividad en el desarrollo por medio de la integración antes mencionada de las herramientas. En la Figura 2.1 se muestran los componentes de un IDE.

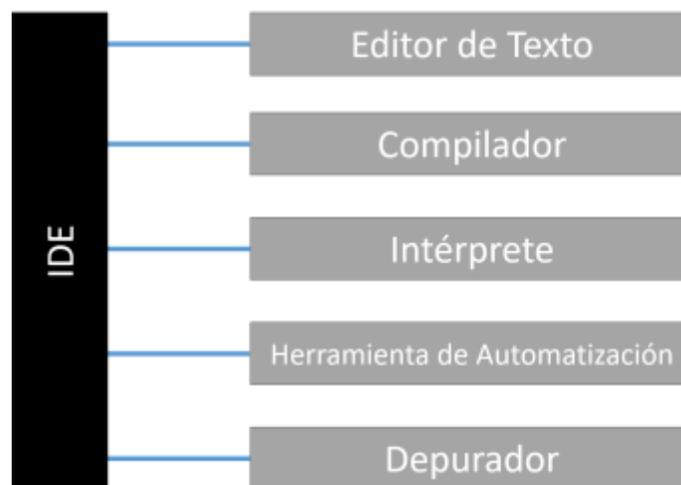


Figura 2.1 Componentes de un IDE

El editor de texto permite la escritura y lectura del código, permite distinguir las funciones de cada parte del código, haciendo distinción entre las sentencias y resaltando palabras reservadas. El compilador justamente genera un resultado en lenguaje máquina, el *bytecode* que utilizará el intérprete para realizar una a una las instrucciones especificadas en las sentencias codificadas. La herramienta de automatización permite la integración de componentes externos que ayuden a generar el código fuente de manera más sencilla, como en el caso de los marcos de trabajo. Finalmente, el depurador establece puntos de ruptura para ejecutar el código y verificar los resultados obtenidos [34].

2.4.1.2.1 Visual Studio Code

Visual Studio Code es un IDE utilizado para crear, editar, depurar código y publicar aplicaciones. Además del editor y depurador que ofrecen todos los IDE, Visual Studio Code incluye herramientas de finalización de código, compiladores, diseñadores gráficos y otras herramientas que facilitan el proceso de desarrollo de software. Es gratuito, multiplataforma y de código abierto, lo cual significa que funciona tanto en Windows, macOS y Linux.

Tiene soporte para casi todos los lenguajes de programación, incluyendo: Java, C++, C#, PHP, Go, SQL, HTML, CSS, JavaScript, TypeScript, y Ruby. Tiene un depurador incorporado, que permite que el seguimiento de errores y la ejecución del código sea rápida y sencilla. Su diseño es eficiente en comparación con otros IDE [35].

2.4.1.3 *Software Development Kit* de realidad aumentada

Un *Kit* de Desarrollo de Software es una colección de herramientas de desarrollo integradas en un único paquete instalable. Son específicas a una plataforma y sistema operativo, pueden tomar la forma de *API (Application Programming Interface, Interfaz de Programación de Aplicaciones)* como bibliotecas de funciones reutilizables para un lenguaje de programación específico. Suelen venir incluidas en IDE para plataformas específicas [36].

A continuación, se presentan los posibles SDK para la resolución del problema planteado en este proyecto.

2.4.1.3.1 Vuforia Engine

Vuforia es la plataforma para desarrollo de RA con mayor amplitud en su uso, cuenta con soporte en la mayoría de dispositivos móviles puesto que los desarrolladores pueden fácilmente integrar captura por medio de cámara a través de sus dispositivos para desplegar objetos de realidad aumentada con los que se puede interactuar de forma realista. Ofrece una amplia variedad de objetos, imágenes y entornos para la lectura de marcadores y el despliegue de modelos basados en ellos.

Permite el reconocimiento de la forma de los objetos utilizando modelos tridimensionales preexistentes en su propia biblioteca de imágenes y la proyección de RA a través de estas lecturas. Cuenta además con un *pluggin* que facilita su conexión con el motor de juegos Unity para el uso de su interfaz en el desarrollo de entornos de RA. Cuenta con una versión gratis, aunque está limitada en el uso de lectura de marcadores y áreas [37].

2.4.1.4 Metodologías de desarrollo de software

En el siguiente apartado se describen las opciones de metodologías a utilizar para el desarrollo de la aplicación propuesta en este proyecto.

2.4.1.4.1 Scrum

Scrum fue creado por el equipo *Easel Corporation* en 1993 para utilizarlo en el desarrollo de software. Combina los conceptos de control empírico de datos, el desarrollo orientado a objetos y el desarrollo iterativo e incremental. Es una metodología de desarrollo ágil, es rápido, eficaz y adaptable. Su objetivo principal es satisfacer las necesidades de los clientes a través del progreso continuo, mejora la productividad y el desarrollo de sistemas complejos.

El proceso inicia a partir de una idea general de las necesidades del cliente, se crea una lista de hipótesis a través del orden de las necesidades del cliente y después de desarrollar la característica se realiza un proceso de retroalimentación para realizar las correcciones necesarias [38].

En la Figura 2.2 se muestra el funcionamiento de la metodología Scrum.

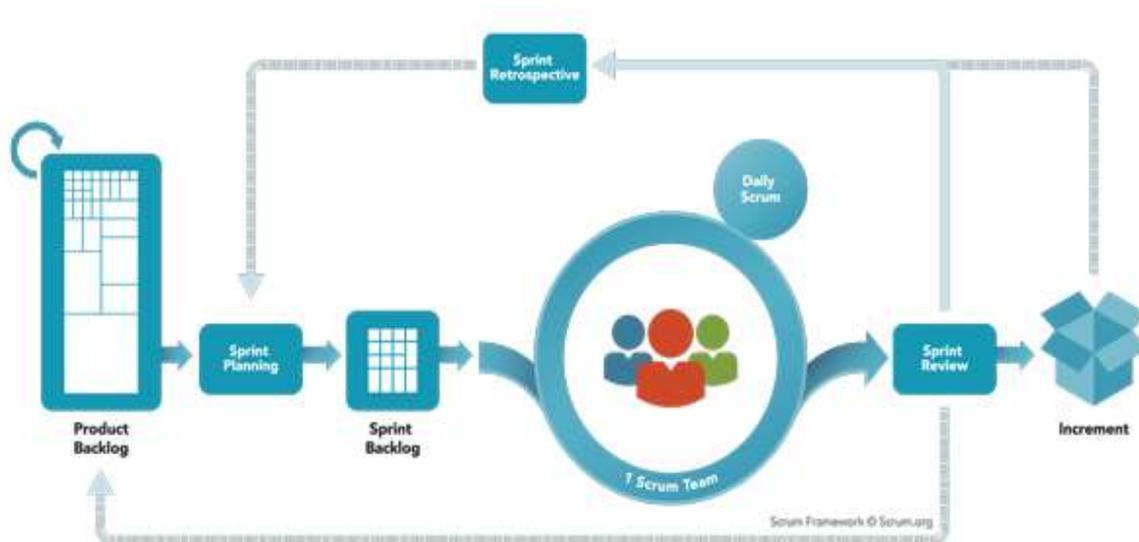


Figura 2.2 Metodología Scrum

Fuente: extraído de [39]

2.4.1.5 Motor gráfico

Los motores gráficos que se mencionan en este apartado se les llama también motores de videojuegos, y estos son referencia a rutinas de programación que incluyen un motor para renderizar gráficos 2D y 3D, otro motor que detecta colisiones y simula leyes de la física, animación, inteligencia artificial, escenarios gráficos y codificación [40].

A continuación, se enlistan y explican las opciones de motores gráficos disponibles para solucionar el problema planteado en este documento.

2.4.1.5.1 Unity

Unity3D es un motor gráfico con IDE integrado, diseñado para la creación de medios interactivos, es famoso por sus capacidades de rápida creación de prototipos y facilidad de publicación e integración de proyectos en distintos medios. Su primera versión se lanzó en 2005, siendo un *software* con herramientas profesionales para el uso de desarrolladores sin experiencia. Aunque su primera versión solo era soportada por Mac OS X, en las modificaciones posteriores se agregaron *plugins* para la exportación de proyectos a otros sistemas operativos. Aunque originalmente su IDE solo soportaba programación en lenguaje C o C++, actualmente está orientado al desarrollo en C# [41].

A pesar de tener versiones de pago enfocadas al uso profesional e industrial, cuenta con una versión gratis enfocada al desarrollo de proyectos educativos.

2.4.1.6 Interfaz de control

Las interfaces de control son las interfaces humano-máquina usadas para la interacción entre el usuario de forma directa con los objetos virtuales generados por el motor de gráficos. En el presente apartado se explicarán las opciones disponibles para desarrollar la solución del proyecto planteado.

2.4.1.6.1 Leap Motion®

Leap Motion Controller® es un dispositivo que captura el movimiento de las manos y rastrea sus movimientos, permitiendo la interacción con el contenido digital de una forma natural, está basado en un modelo de rastreo por puntos coyunturales de esqueleto, cuenta con *plugins* para su integración y uso tanto en Unity3D y *Unreal Engine* en cualquiera de sus versiones. Su SDK cuenta con un núcleo de codificación en C#, aunque las versiones más antiguas soportan C++, Java, JavaScript, Python y Objective-C, pero no reciben actualizaciones de soporte [42].

La página oficial de Ultra Leap ofrece distintos dispositivos de rastreo enfocados a diversas áreas de la industria, pero el uso de *Leap Motion*® en específico solo requiere de una licencia personal, para el uso no comercial y por tanto sin costo [42].

Capítulo 3. Aplicación de la metodología

En este capítulo se describe la aplicación de la metodología. Se exponen las historias de usuario para obtener las tareas a realizar para el diseño del sistema.

3.1 Descripción de la solución

En esta sección se presenta el desarrollo de la solución propuesta mediante el uso de Scrum. Para el desarrollo se creó una pila de producto, el cual es la fuente de requerimientos para gestión del proyecto, para esto se analizaron diferentes métodos de enseñanza siendo seleccionado para la creación de requisitos el método Kantor. La pila de producto puede observarse en la Tabla 3.1.

Tabla 3.1 Historias de usuario.

ID	Como...	Quiero...	Tareas
H01	Usuario	Visualizar modelos tridimensionales en realidad aumentada	Diseñar los marcadores.
			Diseñar los modelos tridimensionales.
			Codificar la función de realidad aumentada.
H02	Usuario	Escuchar los sonidos de las letras y relacionarlas con los modelos que visualizo.	Diseñar la periodicidad de los audios.
			Grabar los audios.
			Codificar la función de reproducción de audio al realizarse la lectura de marcador
H03	Usuario	Escuchar instrucciones que me indiquen los pasos a continuación.	Grabar instrucciones.
H04	Usuario	Interactuar con los modelos tridimensionales a través de gestos manuales.	Codificar los modelos para su interacción con <i>Leap Motion</i> [®]
			Probar la correcta interacción de los modelos en realidad aumentada con <i>Leap Motion</i> [®]
			Configurar audios de instrucciones.
H05	Usuario	Seleccionar la dificultad según mi nivel de comprensión actual.	Diseñar niveles.
			Codificar los niveles.
			Probar los niveles codificados.

Con base en la pila del producto se determinaron los *sprints* a realizar, a continuación, se muestran las iteraciones planeadas.

1. Análisis de requerimientos.
2. Diseño de la arquitectura.
3. Diseño de los marcadores y los modelos tridimensionales.
4. Diseño de los niveles y *Wireframes* del sistema

5. Grabación de los recursos multimedia.
6. Diseño de los menús.
7. Codificación del Nivel 1.
8. Realización de video de ayuda.
9. Codificación del Nivel 2.
10. Codificación del Nivel 3.
11. Codificación del Nivel 4.
12. Aplicación a un caso de estudio.

3.1.1 *Sprint* 1 Análisis de requerimientos

Las actividades realizadas en el *sprint* número uno, se muestran en la Tabla 3.2.

Tabla 3.2 *Sprint* 1

<i>Sprint</i>	Nombre	Tareas
1	Análisis de requerimientos.	Determinar los requerimientos necesarios para el sistema.
Propósito	Determinar las funcionalidades necesarias para que el sistema cumpla con las necesidades del usuario.	
Estado	Finalizado	

El programa está conformado por un conjunto de funcionalidades requeridas para cumplir con su objetivo: Desarrollar una aplicación de realidad aumentada con interacción basada en gestos con las manos para apoyar al aprendizaje de la lectoescritura. Las funcionalidades se describen en la Tabla 3.3.

Tabla 3.3 Funcionalidades requeridas para el sistema.

Funcionalidad	Descripción
Seleccionar niveles.	Se requiere que el usuario sea capaz de seleccionar el nivel de aprendizaje en el que se encuentra con relación al método Kantor.
Despliegue de modelos tridimensionales.	Se requiere que el sistema reconozca marcadores a través de la cámara web y despliegue los modelos tridimensionales diseñados sobre estos.
Reproducción de audios e instrucciones.	Se requiere que el sistema reproduzca audios que relacionen los objetos tridimensionales desplegados con el sonido de la letra del marcador que el usuario ha colocado frente a la cámara.
Interacción con modelos tridimensionales.	Se requiere que el usuario sea capaz de interactuar con los modelos tridimensionales desplegados a través de gestos con las manos, captados con <i>Leap Motion</i> ®.
Niveles conforme a pedagogía.	Se requiere que los niveles diseñados insten al usuario a la realización de ejercicios enmarcados en pedagogía para la enseñanza del idioma español en etapas tempranas.
Identificación de los sonidos y los objetos con símbolos de las letras	Se requiere que el sistema relacione el sonido de la letra del marcador detectado, con el objeto y con la representación simbólica de la letra por medio de la audición y el contacto con los objetos tridimensionales.

El objetivo es determinar los requerimientos funcionales del sistema y mostrarlos como casos de uso, los cuales se observan en la Figura 3.1. En dicho diagrama las funciones se representan desde el punto de vista del actor, facilitando la identificación de las relaciones entre el usuario y las funciones.

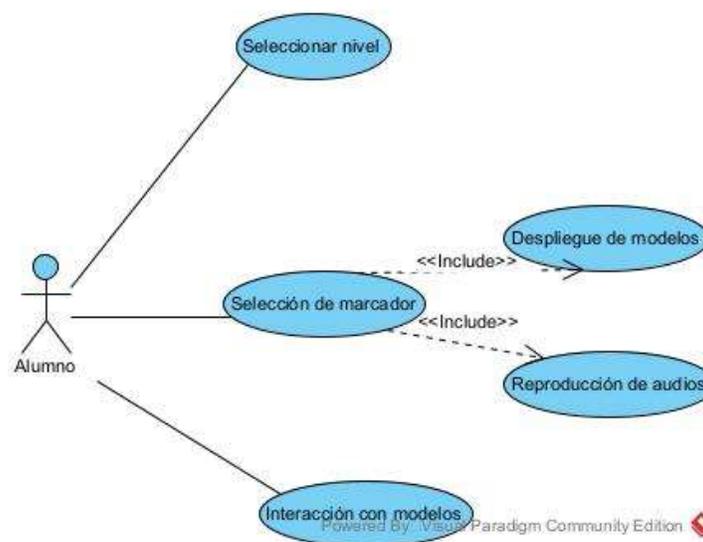


Figura 3.1 Diagrama de casos de uso

El actor alumno tiene la opción de seleccionar el nivel al que quiere acceder, debe seleccionar un marcador y ponerlo a la vista de la cámara. Además, para completar los niveles tiene también la opción de interactuar con los modelos tridimensionales desplegados a través del marcador.

3.1.2 *Sprint* 2 Diseño de la arquitectura

Las actividades que se realizaron en el *sprint* se muestran en la Tabla 3.4.

Tabla 3.4 *Sprint* 2.

<i>Sprint</i>	Nombre	Tareas
2	Diseño de la arquitectura conceptual y física.	Desarrollar el diseño conceptual del sistema. Desarrollar el diseño físico del sistema.
Propósito	Conocer la arquitectura, relaciones, componentes físicos e interacciones que tendrá el sistema.	
Estado	Finalizado	

3.1.2.1 Diseño de la arquitectura conceptual

Durante el diseño conceptual se estableció la arquitectura del sistema a través de las especificaciones planteadas durante el *sprint* número uno.

3.1.2.1.1 Patrón arquitectónico

El patrón arquitectónico es una guía que permite expresar la organización y la estructura de los componentes de un sistema de software, facilitando el desarrollo y futuro mantenimiento. En este caso el sistema se diseñó con base en el patrón arquitectónico por capas, lo cual se puede observar en la Figura 3.2.

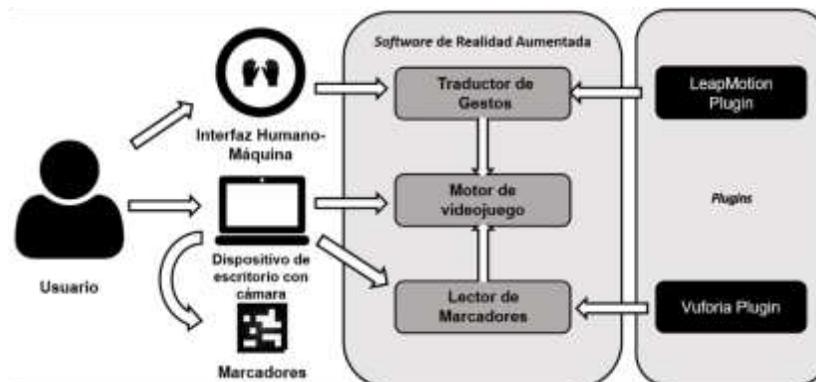


Figura 3.2 Patrón arquitectónico

Capa 1. Es la capa física, concede las interfaces físicas que permiten la comunicación del usuario con el sistema. Cuenta con el dispositivo de escritorio con cámara *Web*, además de la interfaz humano-máquina que rastrea los gestos manuales del usuario y también los marcadores que utilizará la cámara. En la siguiente sección se explicará más a detalle esta capa física.

Capa 2. Es el sistema como tal, está compuesto por tres módulos: el traductor de gestos, el lector de marcadores y el motor de videojuegos. El módulo principal es el motor de videojuegos pues utiliza los otros dos de forma simultánea, es ahí donde se muestra en tiempo real lo que el lector de marcadores identifica, despliega los modelos tridimensionales, reproduce lo que la interfaz humano-máquina captura y permite su interacción con dichos modelos, además de reproducir los archivos multimedia y ejecuta el código que permite la realización de los ejercicios a modo de juego.

Capa 3. Contiene los *plugins* que permiten el funcionamiento de los módulos de lectura de marcadores y traducción de gestos, en el caso de la interfaz humano-máquina se utiliza el *plugin* de *Leap Motion Controller*[®], que es un servicio en línea que rastrea los movimientos y en el caso del lector de marcadores se utiliza el *plugin* de *Vuforia Engine*, que identifica las imágenes que captura la cámara y las busca en la base de datos en línea que se creó previamente con los marcadores diseñados.

El uso del patrón por capas es un diseño probado y funcional que permite el desarrollo rápido y eficiente de los componentes del sistema, facilitando su mantenimiento en futuras actualizaciones.

3.1.2.2 Diseño de la arquitectura física

La arquitectura física es la capa 1 del patrón arquitectónico, debido a la naturaleza del proyecto se requiere el uso de artefactos extras a los incluidos en una computadora común. En la Figura 3.3 se observan todos los componentes necesarios para el correcto funcionamiento del sistema.

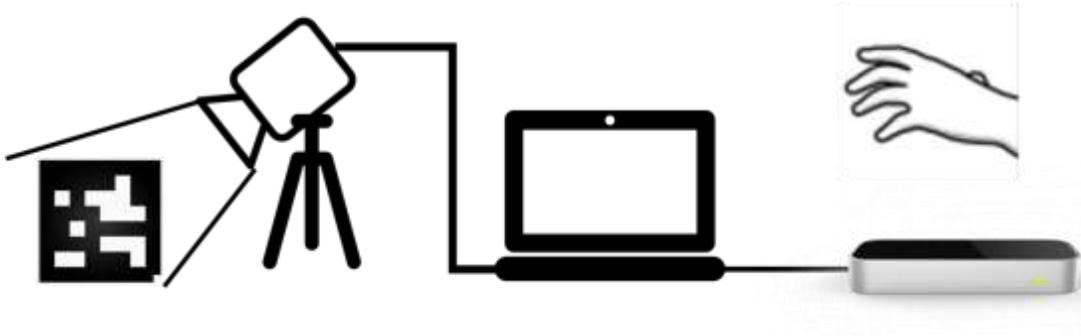


Figura 3.3 Arquitectura física del sistema

Se requieren cuatro componentes principales:

- Computadora de escritorio o laptop con dos entradas USB. Aquí se ejecutará el programa, se requiere que sea un dispositivo de este tipo ya que *Leap Motion Controller*[®] actualmente no tiene forma alguna de ser integrado a ningún dispositivo móvil que son los que acaparan el uso de aplicaciones de realidad aumentada.
- Cámara *Web*. Se pondrá sobre un tripié, apuntando al *layout* diseñado para que el usuario coloque correctamente los marcadores y estará conectada a la computadora.
- *Leap Motion Controller*[®]. Se colocará enfrente del usuario, entre él y la computadora, de manera que pueda visualizar sin estorbos la pantalla y al mismo tiempo mover las manos encima de la interfaz humano-máquina.
- Marcadores. Se colocarán frente a la cámara, en la posición indicada por el *layout*.

Estos elementos permitirán la ejecución correcta del sistema.

3.1.3 *Sprint 3* Diseño de los marcadores y modelos tridimensionales

Las actividades realizadas en el *sprint* número tres se muestran en la Tabla 3.5.

Tabla 3.5 Sprint 2.

Sprint	Nombre	Tareas
3	Diseño de los marcadores y los modelos tridimensionales.	Desarrollar los marcadores usando como base las tarjetas relámpago de la pedagogía. Diseñar los <i>layouts</i> físicos que servirán de guía para el usuario Modelar los objetos tridimensionales que se desplegarán
Propósito	Desarrollar los marcadores para la lectura del sistema, dos <i>layouts</i> que indiquen al usuario la disposición física de los mismos y los modelos tridimensionales.	
Estado	En proceso	

Debido a la limitación del tiempo, a que Vuforia Engine solo permite el uso de veinte marcadores en su versión gratis, y que el método de enseñanza Kantor especifica la enseñanza de las vocales por separado y las consonantes por medio de sílabas, se delimitó la creación de modelos y marcadores a las cinco vocales y a las consonantes b, d, f. Estas permutaciones silábicas, sus respectivos marcadores y sus respectivos modelos tridimensionales se observan en la Tabla 3.6.

Tabla 3.6 Marcadores y modelos respectivos.

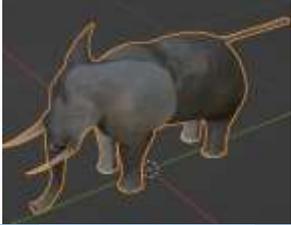
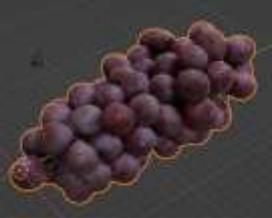
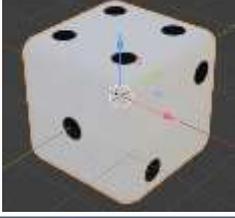
Marcadores	Modelos
	
	
	

Tabla 3.6 Marcadores y modelos respectivos (continuación).

Marcadores	Modelos
	
	
	
	
	

Por otro lado, se desarrollaron dos *layout* para el uso del usuario, que se dividen dependiendo de si se quieren repasar las vocales o las sílabas. Los *layouts* se observan en las Figuras 3.4 y 3.5.

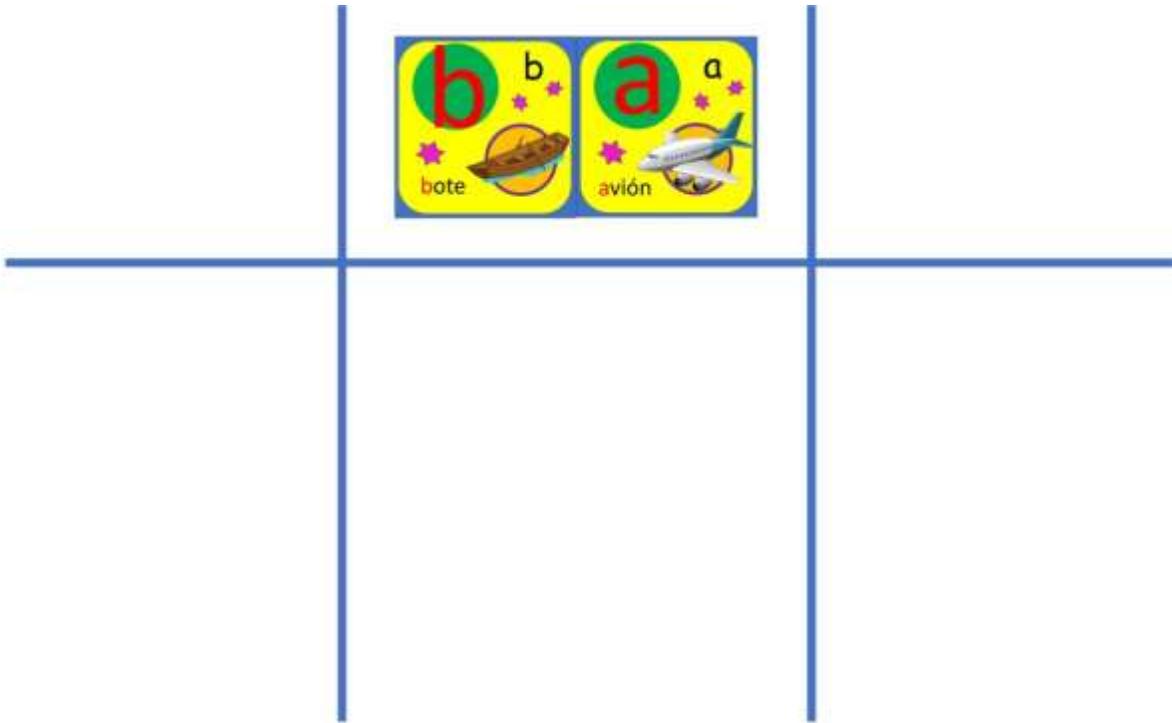


Figura 3.4 Plantilla para sílabas

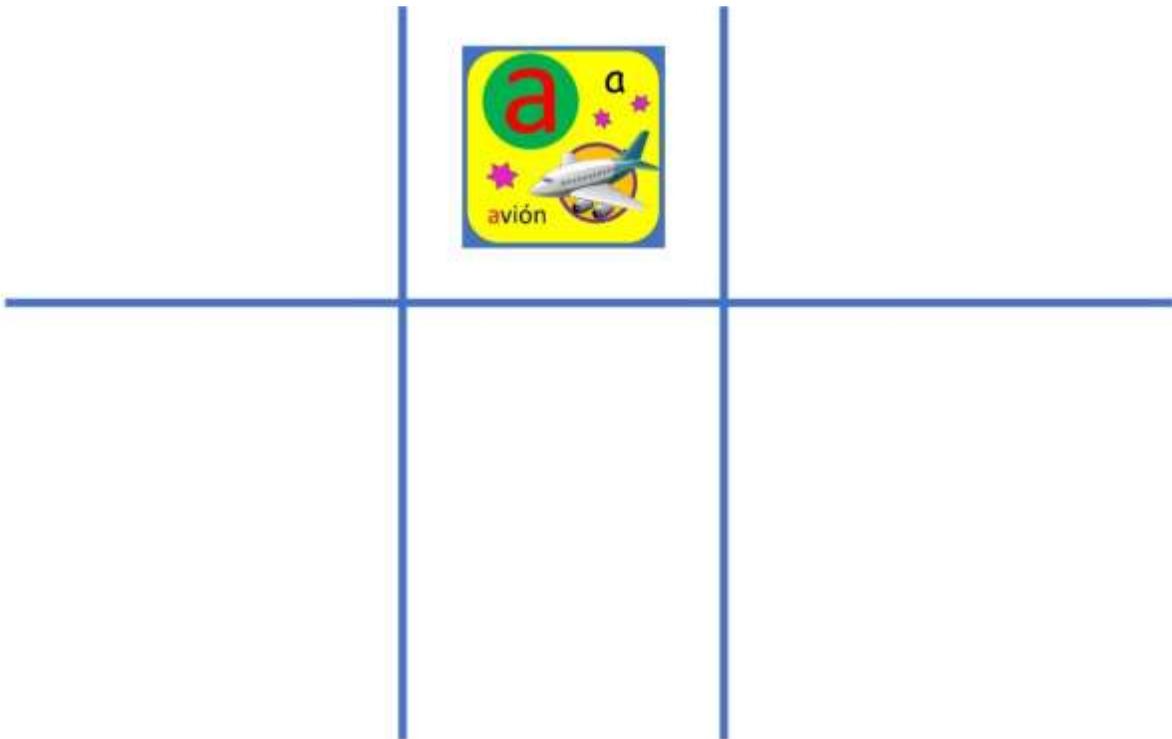


Figura 3.5 Plantilla para vocales

3.1.4 *Sprint 4* Diseño de los niveles y *wireframes* del sistema

En la Tabla 3.7 se observan las actividades que se realizaron en el *sprint* número cuatro.

Tabla 3.7 *Sprint 4*.

<i>Sprint</i>	Nombre	Tareas
4	Diseño de los niveles y <i>wireframes</i> del sistema	Diseñar los ejercicios que se realizarán en los diferentes niveles del sistema Diseñar <i>wireframes</i> que permitan visualizar el funcionamiento del sistema.
Propósito	Obtener una comprensión más efectiva de la funcionalidad del sistema	
Estado	Finalizado	

A continuación, se muestran los *wireframes* y niveles diseñados en la Tabla 3.8.

Tabla 3.8 *Wireframes*

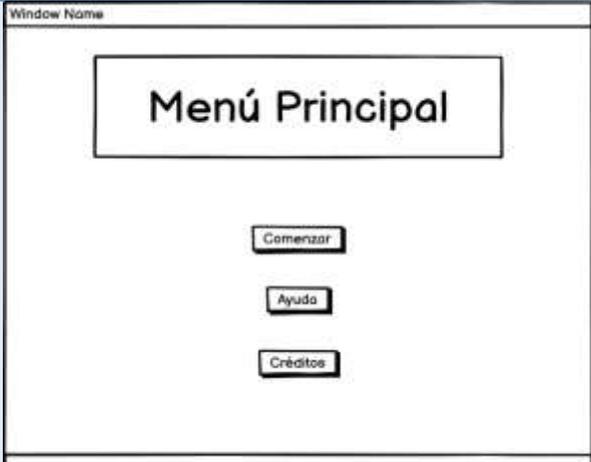
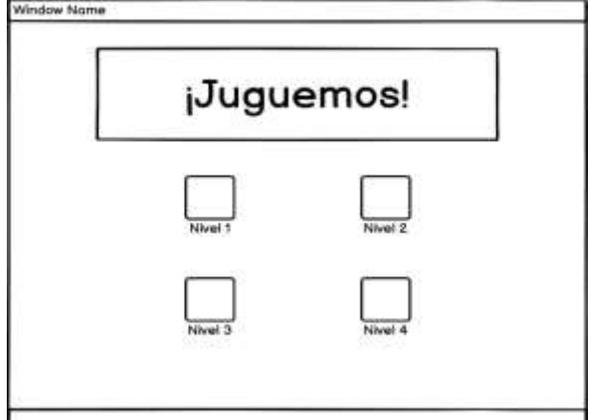
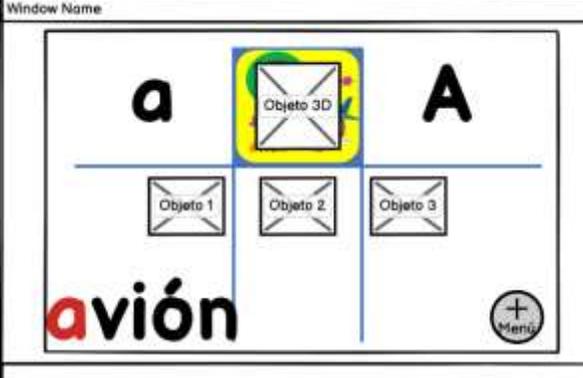
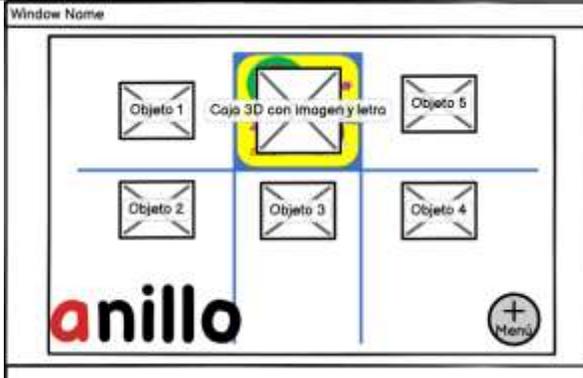
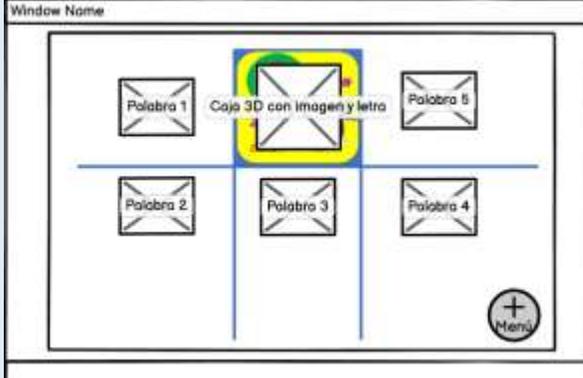
<i>Wireframe</i>	Descripción
 <p>The wireframe shows a window titled 'Menú Principal'. It features a large rectangular box at the top containing the title. Below the title, there are three buttons arranged vertically: 'Comenzar', 'Ayuda', and 'Créditos'.</p>	<p>Menú Principal</p> <p>Ventana inicial, donde el tutor puede pasar directamente a la selección de niveles, al pequeño instructivo de ayuda y a la sección de los créditos.</p>
 <p>The wireframe shows a window titled '¡Juguemos!'. It features a large rectangular box at the top containing the title. Below the title, there are four buttons arranged in a 2x2 grid, labeled 'Nivel 1', 'Nivel 2', 'Nivel 3', and 'Nivel 4'.</p>	<p>Menú de Juego</p> <p>Ventana donde se seleccionan los niveles de juego para presentar los ejercicios al alumno.</p>

Tabla 3.8 Wireframes (continuación).

Wireframe	Descripción
	<p>Nivel 1</p> <p>Al detectarse el marcador se despliega el objeto tridimensional sobre él, al igual que se reproduce un audio haciendo referencia a la letra y al objeto, además de mostrar en mayúscula y minúscula la letra que se está repasando. También al interactuar con el modelo tridimensional por medio de <i>Leap Motion</i>[®], se despliegan tres objetos que también inician con la letra o sílaba que se está repasando. El nivel se completa cuando el estudiante ha arrastrado los tres objetos al modelo principal.</p>
	<p>Nivel 2</p> <p>Al detectarse el marcador se despliega el objeto principal que al tocarse activa cinco objetos, tres que inician con la letra o sílaba que se repasa y dos con comienzo diferente. Al tocar cualquiera de los objetos se escucha un audio que liga el sonido con el nombre del objeto. El nivel se completa cuando los objetos con la letra o sílaba que se está repasando se colocan en la caja.</p>
	<p>Nivel 3</p> <p>Al detectarse el marcador se despliega el modelo principal con la letra o sílaba que se está repasando en minúsculas, también se despliegan cinco palabras tres comienzan con la letra o sílaba que se está repasando y las otras dos no, se tienen que arrastrar a la caja las palabras que tienen relación con la letra o sílaba que se está repasando. El nivel se completa cuando los tres objetos se colocan en el modelo principal.</p>
	<p>Nivel 4</p> <p>Se despliega una palabra que contiene la letra que se está repasando, suena la instrucción que pide al alumno buscar dentro de la palabra la letra que está repasando y arrastrarla al modelo tridimensional, cuando se ha hecho, se despliega otra palabra. El nivel se completa cuando se realizó con cuatro palabras.</p>

3.1.5 *Sprint 5* Grabación de los archivos multimedia

En la Tabla 3.9 se observan las actividades que se realizaron en el *sprint* número cinco.

Tabla 3.9 *Sprint 5*.

<i>Sprint</i>	Nombre	Tareas
5	Grabación de los archivos multimedia	Grabar los audios que se reproducirán
Propósito	Desarrollar los estímulos auditivos que permitirán al estudiante interactuar correctamente con la herramienta a desarrollar	
Estado	Finalizado	

Para cada uno de los niveles se grabaron una serie de instrucciones específicas que orientaran al estudiante para el uso adecuado de la aplicación, los diálogos se presentan en la Tabla 3.10.

Tabla 3.10 *Diálogos por niveles*.

<i>Nivel</i>	Dialogo
Ayuda	<p>Hola, este pequeño tutorial es para que puedas usar sin problemas este programa.</p> <p>Frente a ti tienes un pequeño recuadro gris llamado <i>Leap Motion</i>[®]. Al pasar tus manos sobre él, verás en la pantalla unas manos de colores.</p> <p>Tienes también unos pequeños recuadros de color azul y amarillo con letras en ellos, estos se llaman marcadores y los utilizarás para jugar.</p> <p>Cuentas con una hoja que tiene líneas azules y un marcador completamente vacío, esta plantilla la colocarás frente a la cámara web y te servirá para colocar los marcadores en el lugar correcto.</p> <p>Para comenzar a jugar, coloca frente a la cámara web la plantilla como se te muestra en el video. Acerca a la cámara web el marcador, hasta que salte a la vista el objeto que se ve en el marcador, en este caso un avión. Cuando el objeto aparezca, coloca el marcador en su posición correspondiente en la planilla.</p> <p>Colocando las manos sobre <i>Leap Motion</i>[®], sujeta y mueve el objeto tridimensional ¡No lo sueltes!</p> <p>Como puedes escuchar, se escucha la letra con la que comienza el nombre del objeto. ¡Repite en voz alta sin soltar el objeto!</p> <p>Cuando sueltes el objeto notarás que han aparecido más objetos que inician con la misma letra que estas repasando. Sujeta cada uno de ellos para escuchar cómo se llaman.</p> <p>Una vez que hayas repetido los nombres de los objetos, arrástralos hasta el objeto sobre el marcador, en este caso el avión. Cuando hayas arrastrado todos los objetos ¡Felicidades! habrás terminado el nivel.</p>

Tabla 3.10 Diálogos por niveles (Continuación)

Nivel	Dialogo
Nivel 1	¡Hola! Coloca la plantilla y los marcadores que utilizarás frente a la cámara.
	Sujeta el objeto que ha aparecido y repite conmigo.
	Han aparecido más objetos que inician con la letra que estás repasando, muévelos y repite conmigo sus nombres.
	Ahora toma los tres objetos y arrástralos hacia el marcador para completar el nivel.
	¡Excelente!
	¡Muchas Felicidades! ¡Haz completado el Primer nivel!
Nivel 2	¡Hola! Coloca la plantilla y los marcadores que utilizarás frente a la cámara.
	Sujeta el objeto que ha aparecido y repite conmigo.
	Alrededor del marcador han aparecido objetos, sujétalos y repite conmigo sus nombres. Aquellos que empiecen con el mismo sonido de la letra que estas repasando, arrástralos a la caja para completar el nivel.
	¡Excelente!
	¡Ay! Inténtalo de nuevo.
	¡Muchas Felicidades! ¡Haz completado el Segundo nivel!
Nivel 3	¡Hola! Coloca la plantilla y los marcadores que utilizarás frente a la cámara.
	Alrededor del marcador han aparecido palabras, algunas empiezan con la misma letra que estas repasando, toca cada una de las palabras para escucharlas y aquellas que empiecen con el mismo sonido de la letra que estas repasando, arrástralas al marcador para completar el nivel
	¡Excelente!
	¡Ay! Inténtalo de nuevo.
	¡Muchas Felicidades! ¡Haz completado el Tercer nivel!
Nivel 4	¡Hola! Coloca la plantilla y los marcadores que utilizarás frente a la cámara.
	Ha aparecido una palabra que contiene varias veces la letra que estás repasando. Búscalas y arrástralas al marcador para completar el nivel.
	¡Excelente!
	¡Ay! Inténtalo de nuevo.
	¡Muchas Felicidades! ¡Haz completado el Cuarto nivel!

3.1.6 Sprint 6 Desarrollo de los menús

En la tabla 3.11 se describen las actividades que se realizaron en el *sprint* número seis.

Tabla 3.11 Sprint 6.

Sprint	Nombre	Tareas
6	Desarrollo de los menús: principal y de selección de nivel	Diseño de los menús para visualización de pantallas secundarias
Propósito	Diseñar el menú principal, el menú de selección de niveles, el menú de créditos y la pantalla de visualización de video de ayuda	
Estado	Finalizado	

Para el diseño de los menús se utilizaron como base los *wireframes* que se desarrollaron en el punto 3.1.4. En la Tabla 3.12 se visualizan los menús que se desarrollaron, así como los detalles del funcionamiento.

Tabla 3.12 Funcionamiento de los menús.

Wireframe	Descripción
	<p>Menú Principal</p> <p>Ventana inicial, que incluye los botones “Jugar”, “Ayuda”, “Créditos” y “Salir”. El botón jugar redirige el programa al Menú de Niveles, el botón Ayuda redirige a un video tutorial que sirve de apoyo para el padre de familia o maestra que orienta al estudiante durante la ejecución del programa, el botón Créditos redirige a los agradecimientos por el apoyo durante el desarrollo del proyecto y el botón Salir termina la ejecución del programa.</p>
	<p>Menú de Niveles</p> <p>Ventana donde se selecciona el nivel de dificultad para el estudiante. También tiene un botón que redirige al menú principal en caso de desearse.</p>
	<p>Sección de Ayuda</p> <p>Ventana donde se visualiza el video tutorial orientado al maestro o padre de familia que acompaña al estudiante, cuenta además con un botón que te regresa al menú principal.</p>

Tabla 3.12 Funcionamiento de los menús (continuación)

Wireframe	Descripción
	<p>Sección de Créditos</p> <p>Sección que da los agradecimientos correspondientes al Tecnológico Nacional de México, Campus Orizaba y al CONACYT. Cuenta con un botón que regresa al usuario al menú principal.</p>

Dentro del editor de Unity3D, se seleccionaron las escenas creadas que se desean compilar, en este caso las cuatro ventanas que se mostraron en la Tabla 3.12. En el menú de compilación se le asigna un número a cada una de estas escenas, tal como se observa en la Figura 3.6, gracias a esto, a cada uno de los botones, tanto del menú principal, como del menú de niveles y el botón rojo que redirige al menú de inicio que está en la escena de Ayuda, Créditos y Niveles, se le asigna un valor, del cero al siete, dependiendo de a donde se desea que el programa navegue según se haga *click* en cualquiera de estos botones.

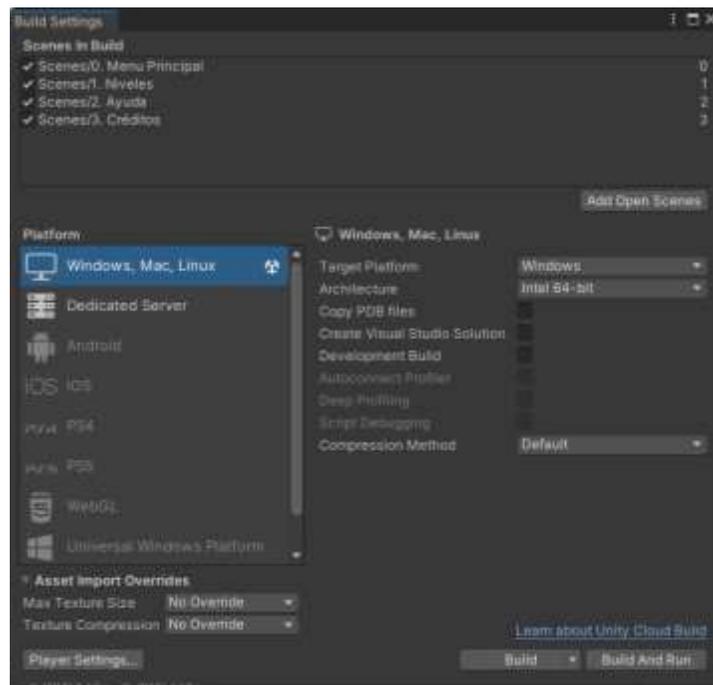


Figura 3.6 Menú de compilación

Dentro de la carpeta `Assets` se creó otra carpeta llamada `códigos`, para tener en un solo lugar los *scripts* generados para el funcionamiento correcto. Dentro un archivo llamado `SeleccionarNivel()`, cuyo código se puede observar en el Listado 3.1. y luego se incrustó en la cámara de los menús.

Listado 3.1 Clase `SeleccionarNivel()`

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5. using UnityEngine.UI;
6.
7. public class SeleccionarNivel : MonoBehaviour
8. {
9.     public GameObject PanelDeCarga;
10.
11.     public void seleccionarNivel(int nombreNivel){
12.         StartCoroutine(CargarNivel(nombreNivel));
13.     }
14.
15.     IEnumerator CargarNivel(int nombreNivel){
16.         AsyncOperation cargar = SceneManager.LoadSceneAsync(nombreNivel);
17.         PanelDeCarga.SetActive(true);
18.         while (!cargar.isDone){
19.             yield return null;
20.         }
21.     }
22. }
```

Como se puede observar, se creó una corrutina, que de forma asíncrona espera la llegada de un número en específico, cuando lo recibe, toma ese número y lo usa para cargar la escena, de esta manera, cuando un botón sea apretado, el método usará su número asignado para navegar el programa al nivel seleccionado.

También se configuró la escena para que mientras se carga el menú o el nivel seleccionado, aparezca un panel de carga para indicar al usuario que el programa no se ha trabado, puesto que los niveles uno al cuatro, tardan un tiempo promedio de tres minutos en cargar por completo.

3.1.7 *Sprint 7* Diseño del nivel 1

En la tabla 3.13 se describen las actividades que se realizaron en el *sprint* número siete.

Tabla 3.13 Sprint 7.

<i>Sprint</i>	Nombre	Tareas
7	Diseño del nivel 1	Configuración de <i>Leap Motion</i> [®]
		Configuración de Vuforia Engine
		Configuración de los audios
		Creación de panel de letras
		Configuración de los modelos tridimensionales
		Programación de funcionamiento de nivel
Propósito	Diseñar el nivel 1	
Estado	Finalizado	

A continuación, se detallarán los pormenores de la configuración necesaria para el diseño del nivel 1, cabe mencionar que los puntos 3.1.7.1 y 3.1.7.2 son configuraciones generales que aplicaron tanto en este *sprint* como en los siguientes tres.

Este nivel funciona de la siguiente forma: se presenta un marcador frente a la cámara, que al reconocer la figura muestra un objeto tridimensional encima de la misma, cuando el usuario sujeta dicha figura, de fondo se reproduce el nombre de la misma. Cuando el modelo tridimensional se suelta, tres objetos más aparecen en escena, tienen la misma funcionalidad que el primer modelo, al sujetarlas se reproducen audios con sus respectivos nombres. Una vez que el estudiante ha sujetado los tres modelos y repitió sus nombres, debe arrastrar cada uno de estos objetos al objeto principal sobre el marcador, ganando un punto por cada uno de estos modelos. Cuando el usuario acumula tres puntos, el juego se termina.

3.1.7.1 Configuración de *Leap Motion*[®]

Para configurar *Leap Motion Controller*[®], previamente se debe descargar de la página oficial los *plugins* necesarios para su funcionamiento que incluyen escenas prefabricadas, estos *plugins* se observan en la Figura 3.7. Dichos ejecutables se arrastraron a la pantalla del editor que descargó las bibliotecas necesarias.

Nombre	Fecha de modificación	Tipo	Tamaño
Tracking Examples	11/01/2022 03:42 p. m.	Unity package file	2,368 KB
Tracking Preview Examples	11/01/2022 03:42 p. m.	Unity package file	3,317 KB
Tracking Preview	11/01/2022 03:42 p. m.	Unity package file	454 KB
Tracking	11/01/2022 03:42 p. m.	Unity package file	14,014 KB

Figura 3.7 Plugins de Leap Motion®.

En Unity3D, dentro del panel del proyecto, en la carpeta `Assets` se creó otro folder llamado `ThirdParty` que contiene las escenas prefabricadas de *Leap Motion*®. Se navegó a través de las carpetas `Ultraleap`, `Tracking`, `Examples`, `Core Examples` y se seleccionó la escena prefabricada llamada `Capsule Hands (Desktop)`.unity, esto se observa en la Figura 3.8.



Figura 3.8 Carpeta que contiene la escena prefabricada a utilizar para el desarrollo de los niveles.

Dicha escena contiene los objetos `EventSystem` que organiza los códigos y los objetos presentes en la escena, `Desktop Rig` que simula al *Leap Motion*®, y `Hand Models` que permite al usuario observar el movimiento de sus manos mientras es rastreado en tiempo real. Luego se procedió a eliminar la `Main Camera` de la jerarquía de desarrollo, debido a que es innecesaria para el diseño del proyecto, así como el piso predeterminado que está bajo el `Desktop Rig`. Después, desde la carpeta `Tracking`, se navegó a través de las carpetas `Interaction Engine`, `Runtime` y `Prefabs`, donde se encontró un objeto llamado `Interaction Manager`, tal y como se observa en la Figura 3.9.

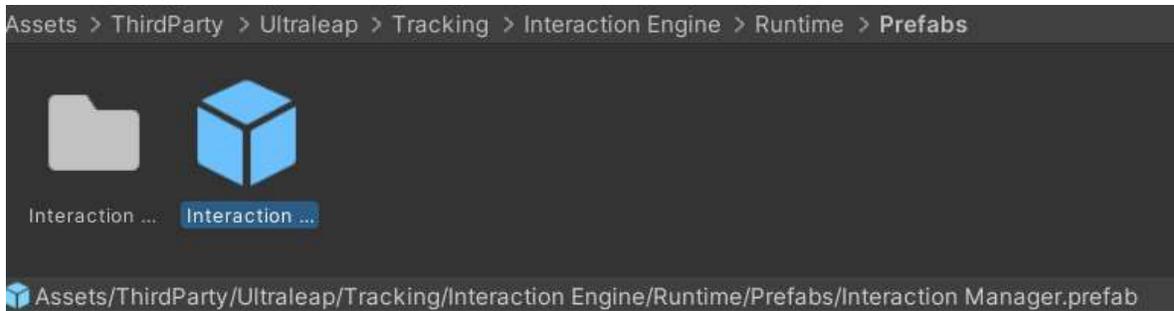


Figura 3.9 Objeto Interaction Manager.

`Interaction Manager` se arrastró a la jerarquía de la escena, pues contiene las rutinas necesarias para el rastreo de las manos del usuario, su proyección en el entorno virtual y la manipulación de los objetos a través del objeto `Hand Models`, aunque para esto se realizaron configuraciones sobre los modelos que se detallarán en la sección 3.1.7.3.

Finalmente se guardó la escena con un nombre diferente para no sobrescribir la que existe en `Core Examples`. Se almacenó en la subcarpeta `Scenes` dentro del directorio principal del folder `Assets`.

3.1.7.2 Configuración de Vuforia Engine

En primera instancia, de la página de Vuforia se descargó el paquete SDK que sirve de *plugin* para Unity3D. El ejecutable descargado se arrastró a la carpeta `Assets` dentro del panel del proyecto y se descargaron las bibliotecas necesarias para el correcto funcionamiento de Vuforia. Dentro de la carpeta `Assets`, se creó una subcarpeta llamada `Resources`, donde se encontró un objeto de rutinas llamado `Vuforia Configuration`. Al hacer *click* en el objeto, en el panel inspector muestra su configuración como se observa en la Figura 3.10, donde se ingresó un código que se provee en la página de descarga del *plugin*, iniciando sesión y después de crear una licencia.

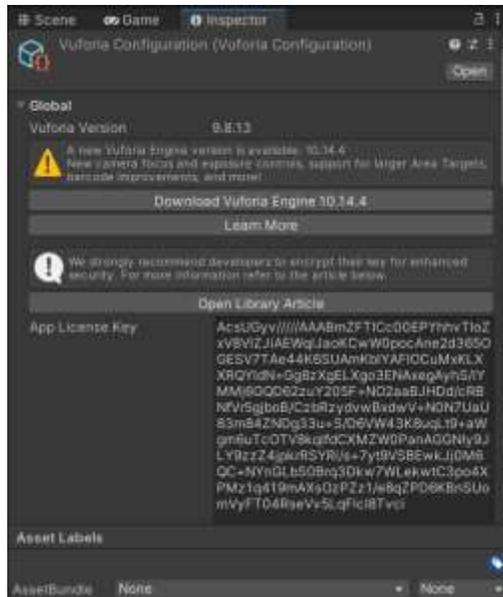


Figura 3.10 Detalles de Vuforia Configuration.

Dentro de la misma página de Vuforia, se creó una biblioteca de *Target Manager* que se nombró Kantor, donde se subieron los marcadores que se utilizaron en la implementación de este proyecto. Esto se observa en la Figura 3.11. Gracias a la licencia que se creó, se tiene conexión con esta biblioteca.

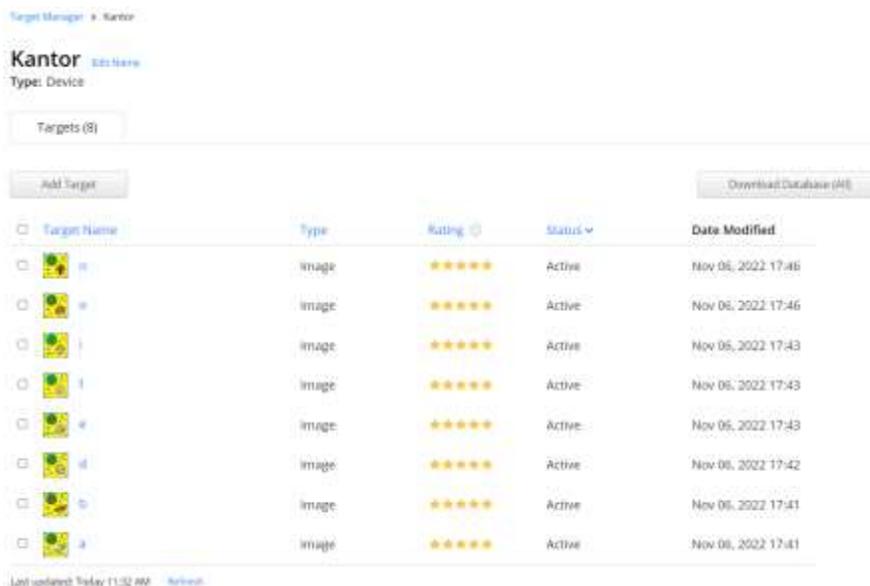


Figura 3.11 Biblioteca de marcadores.

Dentro de la carpeta *Assets* se creó una carpeta llamada *Marcadores* donde se agregaron las imágenes que se van a utilizar. También, dentro del panel de la

jerarquía se generó una AR Camera, que permite la captura de video a través de una *Webcam*. De igual forma, se arrastró dentro de este objeto recientemente creado *HandModels* y *Desktop Rig*, de esta forma, cuando la cámara se desplace por el universo virtual, no se perderá de vista el modelo de las manos que interactuará con los objetos tridimensionales. La organización de los objetos hasta este punto se observa en la Figura 3.12.



Figura 3.12 Jerarquía actual del nivel 1

3.1.7.3 Configuración de audios

Primero se realizó la configuración de los audios de instrucción, se crearon cinco objetos en la jerarquía de AR Camera con los nombres:

1. Instrucción inicial
2. Sujetar objeto marcador
3. Sujetar otros objetos
4. Excelente
5. Felicidades

De la carpeta dentro de *Assets* que contiene los audios, se asignaron las pistas correspondientes, como se observa en la Figura 3.13, el audio uno, dentro de sus características conserva la seleccionada la opción de *Play On Awake*, que hace que, una vez iniciado el nivel, de inmediato se reproduzca este sonido. Por el otro lado a las demás fuentes de audio se les deseleccionó esta opción, pues solo serán reproducidas cuando se cumplan ciertos escenarios dentro del nivel.

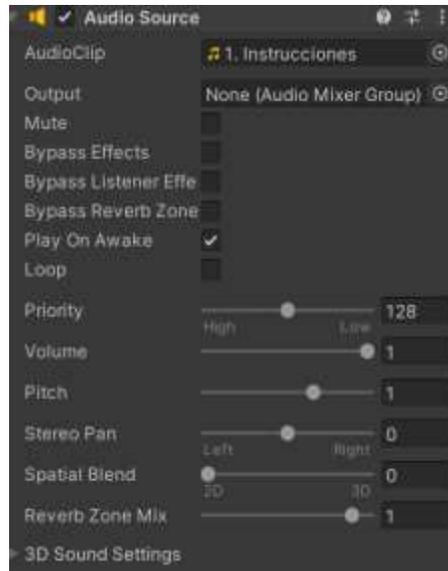


Figura 3.13 Configuración del primer audio.

Luego, se crearon ocho objetos que sirvieron para clasificar los audios de cada marcador, dentro de estos, se crearon cuatro fuentes de audio y se les asignó el audio correspondiente para su funcionamiento, al contrario que en la Figura 3.13, se deseleccionó la opción *Play On Awake* y se seleccionó la opción *Loop*, de esta manera, el audio no se reproducirá al iniciar el nivel y cuando lo haga, se reproducirá una y otra vez. Los audios creados se presentan en la Figura 3.14.

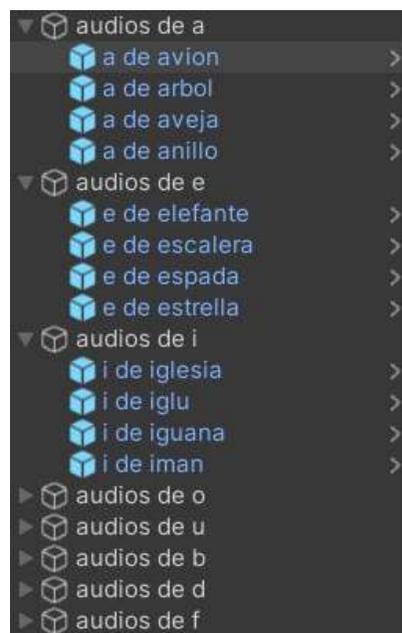


Figura 3.14 Audios para los objetos tridimensionales.

3.1.7.4 Creación del panel de letras

Dentro de la misma jerarquía de AR Camera, se creó un objeto Canvas de dos dimensiones. Este objeto se proyectará de forma predeterminada como fondo, dentro se crearon objetos para cada uno de los marcadores y tres secciones de texto, como ejemplo, una vez que se detecte el marcador de la letra a utilizar, se desplegará la misma en mayúsculas, minúsculas y la palabra avión. También se colocó un texto que indicara al usuario que para salir del juego se debe apretar la barra espaciadora. Esto se observa en la Figura 3.15, que usa como ejemplo la letra A.



Figura 3.15 Mockup del funcionamiento del panel de letras.

También, se creó una etiqueta para actualizar el conteo de los puntos del usuario, de manera que cuando se complete la dinámica sea más fácil obtener los resultados de una variable global. Por otro lado, se creó un cartel de felicitaciones que se desplegará cuando se cumplan con las condiciones de finalización de nivel. Los objetos creados se visualizan en la Figura 3.16.

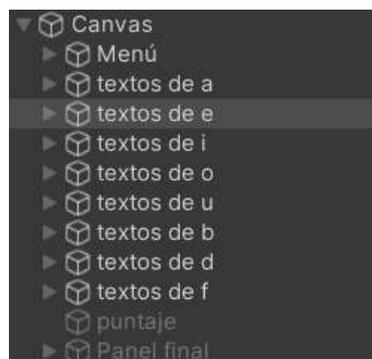


Figura 3.16 Panel de letras.

3.1.7.5 Configuración de modelos tridimensionales

En el panel de jerarquía se creó un *Image Target* y se le asignó la imagen que servirá de marcador de lectura, para propósitos de esta explicación se describirá haciendo énfasis en la letra A. Dentro de la jerarquía del marcador a se introdujeron los objetos que se utilizarán para las mecánicas del juego que están relacionados con dicho marcador, son objetos cuyos nombres inician con la letra a: avión, abeja, anillo y árbol. El avión se colocó encima del marcador, debido a que es la imagen que representa la letra a, esto se presenta en la Figura 3.17.

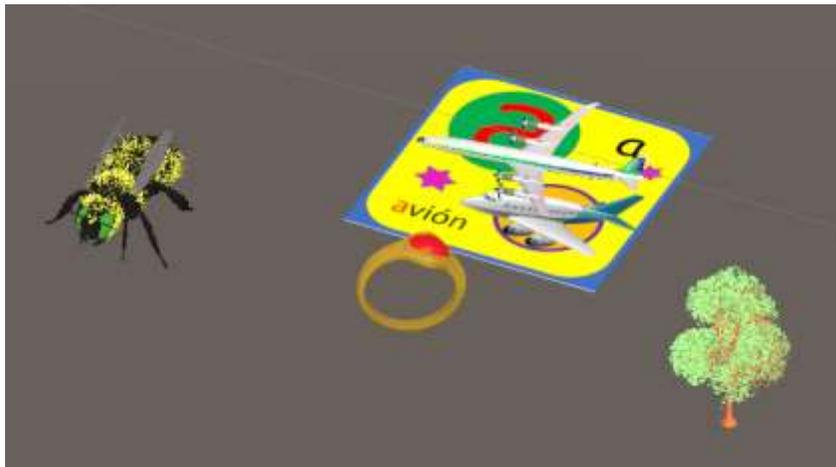


Figura 3.17 Marcador de la letra A con sus objetos tridimensionales

Al seleccionar el marcador, dentro del panel de instrucción se muestran las interfaces de los métodos `onTargetFound()` y `onTargetLost()`. En la primera interfaz, desde la jerarquía, se arrastraron los cuatro objetos y dentro del funcionamiento se especificó que solo el objeto principal, en el caso de la letra A el avión, fuera visible al momento de identificarse el marcador, se arrastró también el objeto del panel de letras y se activó su visibilidad al igual que el avión. Por otro lado, se arrastraron también las instrucciones uno y dos. Aunque la uno se programó para detenerse en cuanto se detectara el marcador, la dos se configuró para reproducirse. Esto se observa en la Figura 3.18.

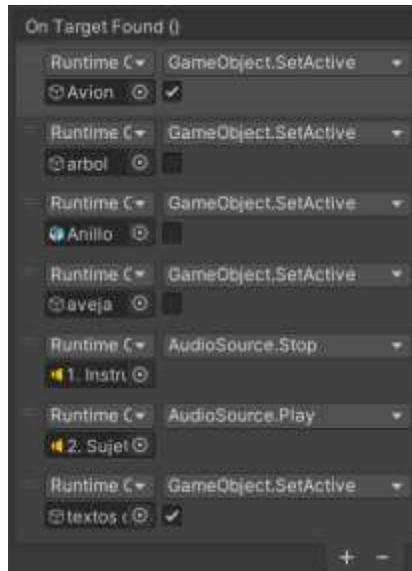


Figura 3.18 Configuración para los objetos al identificar un marcador

En la segunda interfaz se configuraron los objetos para desaparecer y los audios para detenerse una vez que el marcador se pierda de la vista de la cámara, lo que se observa en la Figura 3.19.

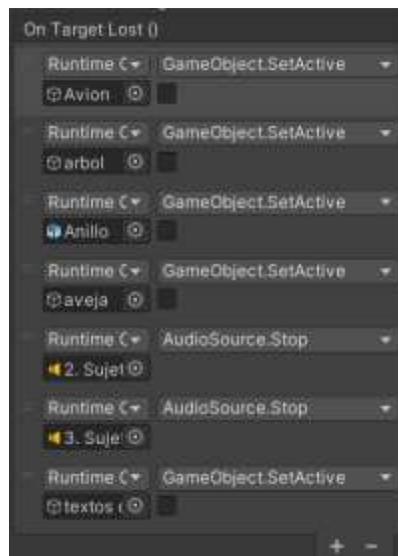


Figura 3.19 Configuración para los objetos al perder el marcador

Después se pasó a la configuración de los objetos como tal, en primera instancia a los cuatro se les agregaron los componentes `Mesh Collider` (Figura 3.20), e `Interaction Behaviour` (Figura 3.21). `Mesh Collider` trae consigo un segundo componente llamado `Rigidbody`, dentro del cual, a todos los objetos se les

desmarcó la casilla *Gravity*, para evitar que los objetos, al estar en un entorno de realidad aumentada, caigan al vacío.

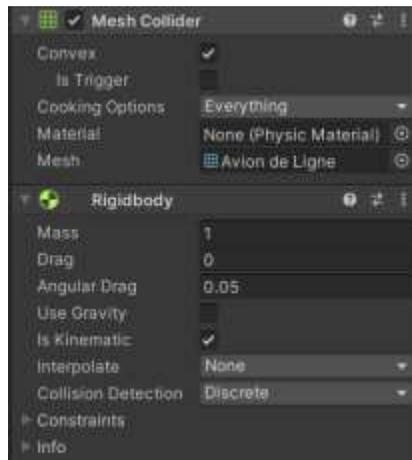


Figura 3.20 Componentes Mesh Collider y Rigidbody

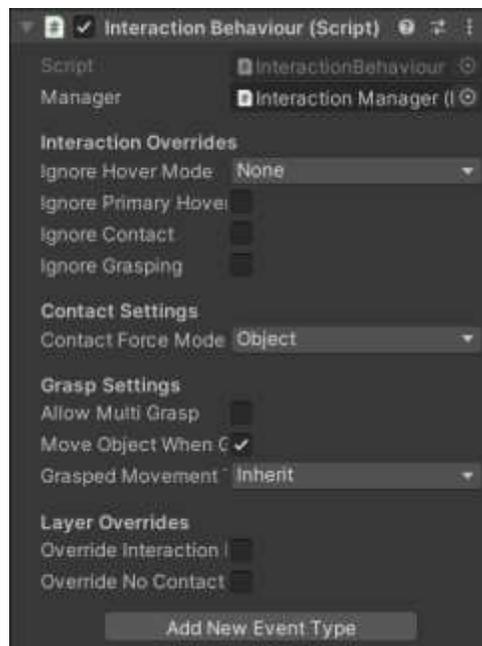


Figura 3.21 Componente Interaction Behaviour

Dentro del componente *Interaction Behaviour* se agregaron dos interfaces, *GraspBegin()* y *GraspEnd()*, además en manager se asignó como controlador el objeto *Interaction Manager* que se arrastró a la escena como se describió en el punto 3.1.7.1.

En `GraspBegin()` se asignaron el audio de la instrucción dos y el nombre del objeto, en este caso el avión y se les configuró para que la instrucción se detuviera, mientras el nombre del avión se reproducía. En `GraspEnd()`, dentro del objeto marcador, es decir, el avión, se asignaron el audio del nombre del objeto y la instrucción tres, de manera que el primero se detuviera y el segundo se reprodujera al soltarse el objeto, también se asignaron los objetos extra del marcador: anillo, árbol, y araña, dispuestos para mostrarse al mismo tiempo. Esto se observa en la Figura 3.22.

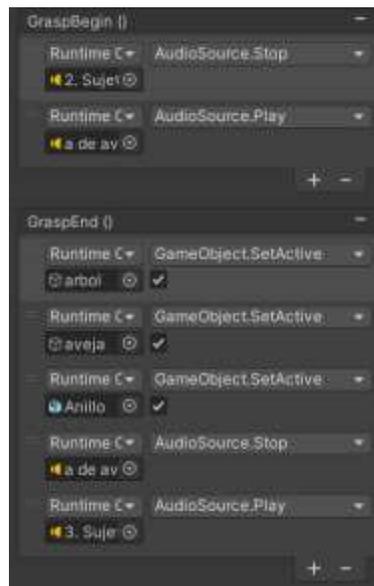


Figura 3.22 Interfaces `GraspBegin` y `GraspEnd`

Dentro del componente `Mesh Collider`, a todos los objetos se les marcó la casilla `Convex`, también, mientras que para el modelo encima del marcador (el avión), se le desmarcó la `Is Trigger`, a los demás objetos se les marcó para la programación del funcionamiento del juego. Además, para el anillo, el árbol y la araña, en `GraspEnd()`, solo se agregó el audio del nombre del objeto, especificando que se detuviera en cuanto se soltara el mismo. Además, tal y como se muestra en la Figura 3.20, a todos los objetos dentro del componente `Rigidbody`, se les marcó la opción `Is Kinematic`, que permite el movimiento de los objetos.

Una vez terminadas las configuraciones de todos los objetos dentro de la letra A, dicho comportamiento se programó en las demás letras.

3.1.7.6 Programación de funcionamiento del nivel

Dentro de la carpeta Códigos dentro de `Assets` se creó un archivo llamado `EspaciadoraSalir.cs`, en ella se escribió el código que se presenta en el Listado 3.2. El código proporcionado verifica de forma constante que el usuario teclee la barra espaciadora, en caso de que se haga se ejecuta una corrutina que redirige el programa a la selección de nivel. También se muestra un panel de carga para indicar al usuario que se está redirigiendo de escena el programa. Este código se agregó como componente al panel llamado `Canvas`, que contiene los elementos que no son tridimensionales: letras, nombres y botón de salida del nivel.

Listado 3.2 Clase `EspaciadoraSalir()`

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5. using UnityEngine.UI;
6.
7. public class EspaciadoraSalir : MonoBehaviour
8. {
9.     public GameObject PanelDeCarga;
10.
11.     void Start () {}
12.
13.     void Update ()
14.     {
15.         if (Input.GetKeyDown(KeyCode.Space)) {
16.             StartCoroutine(CargarNivel(1));
17.         }
18.     }
19.
20.     IEnumerator CargarNivel(int nombreNivel) {
21.         AsyncOperation cargar = SceneManager.LoadSceneAsync(nombreNivel);
22.         PanelDeCarga.SetActive(true);
23.         while (!cargar.isDone) {
24.             yield return null;
25.         }
26.     }
27. }
```

Luego se creó un código para actualizar los valores de la etiqueta que contendrá el puntaje, el algoritmo se puede observar en el Listado 3.3. Este código se asignó al elemento llamado `puntaje`, que es invisible a los ojos del usuario y se encuentra dentro del `Canvas` anteriormente mencionado. El algoritmo recupera el componente

bidimensional en el que esta incrustado (objeto `puntaje`) y cada vez que el método `SumarPuntos` es invocado, suma un punto que ayudará para terminar el nivel, en cuanto se pierde de vista el marcador, dicho puntaje se reinicia para comenzar de ceros.

Listado 3.3 Clase PuntajeAgarrados()

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using TMPro;
5.
6. public class PuntajeAgarrados : MonoBehaviour
7. {
8.     private float puntos;
9.     private TextMeshProUGUI texto;
10.    private void Start(){
11.        texto = GetComponent<TextMeshProUGUI>();
12.        puntos = 0;
13.    }
14.
15.    private void Update(){
16.        texto.text = puntos.ToString("");
17.    }
18.
19.    public void SumarPuntos(){
20.        puntos += 1;
21.    }
22.
23.    public void PerderPuntos(){
24.        puntos = 0;
25.    }
26.
27.    public float getPuntaje(){
28.        return this.puntos;
29.    }
30.}
```

Luego se codificó una clase llamada `ObjetosJuego()`, que se observa en el Listado 3.4. En ella se configuró que cualquier objeto que cuente con esta clase, al colisionar con el modelo que se encuentre encima del marcador, desaparezca e invoque el método `SumarPuntos()`, luego verifica el puntaje total en la etiqueta `puntaje` y si se cumple con los puntos requeridos para finalizar el nivel, en caso de que esto sea así reproduce el audio de finalización. Acto seguido, desactiva todos los modelos en la pantalla y activa el panel de felicitación.

Listado 3.4 Clase ObjetosJuego()

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.SceneManagement;
5. using UnityEngine.UI;
6.
7. public class ObjetosJuego : MonoBehaviour
8. {
9.     //public Transform inicial;
10.    [SerializeField] private PuntajeAgarrados puntaje;
11.    public AudioSource audio;
12.    public AudioSource final;
13.    public GameObject felicitaciones, manos, marcador, erroneo1,
    erroneo2;
14.
15.    private void OnTriggerEnter(Collider objeto){
16.        if(objeto.tag != "Player" & objeto.tag == "Respawn"){
17.            puntaje.SumarPuntos();
18.            gameObject.SetActive(false);
19.            float puntos = puntaje.getPuntaje();
20.            if(puntos != 3){
21.                audio.Play();
22.            }
23.            if(puntos == 3){
24.                if(erroneo1 != null && erroneo2 != null){
25.                    erroneo1.SetActive(false);
26.                    erroneo2.SetActive(false);
27.                }
28.                marcador.SetActive(false);
29.                manos.SetActive(false);
30.                felicitaciones.SetActive(true);
31.                final.Play();
32.            }
33.        }
34.        else{
35.            if(objeto.tag != "Player" & objeto.tag != "Respawn"){
36.
37.                //this.transform.position = inicial.position;
38.            }
39.        }
40.    }
41. }
```

Ese código se incrustó en cada uno de los objetos extra, en el caso de la letra A, la araña, el árbol y el anillo. También se creó un algoritmo alternativo llamado `ObjetosJuego2()`, destinado para aquellas letras cuyos modelos identificables relacionados con las letras se redujeron a dos, esto debido al conflicto existente por la correcta conceptualización en objetos de las palabras. Se observa en el Listado 3.5.

Listado 3.5 Clase ObjetosJuego2()

```
1. public class ObjetosJuego2 : MonoBehaviour
2. {
3.     //public Transform inicial;
4.     [SerializeField] private PuntajeAgarrados puntaje;
5.     public AudioSource audio;
6.     public AudioSource final;
7.     public GameObject felicitaciones, manos, marcador, erroneo1,
   erroneo2, erroneo3;
8.
9.     private void OnTriggerEnter(Collider objeto){
10.         if(objeto.tag != "Player" & objeto.tag == "Respawn"){
11.             puntaje.SumarPuntos();
12.             gameObject.SetActive(false);
13.             float puntos = puntaje.getPuntaje();
14.             if(puntos != 2){
15.                 audio.Play();
16.             }
17.             if(puntos == 2){
18.                 if(erroneo1 != null && erroneo2 != null && erroneo3 !=
   null){
19.                     erroneo1.SetActive(false);
20.                     erroneo2.SetActive(false);
21.                     erroneo3.SetActive(false);
22.                 }
23.                 marcador.SetActive(false);
24.                 manos.SetActive(false);
25.                 felicitaciones.SetActive(true);
26.                 final.Play();
27.             }
28.         }
29.         else{
30.             if(objeto.tag != "Player" & objeto.tag != "Respawn"){
31.
32.                 //this.transform.position = inicial.position;
33.             }
34.         }
35.     }
36. }
```

Después, a cada uno de los objetos en los que se incrustó el código, se le agregaron los componentes correspondientes en las variables públicas que las clases `ObjetosJuego()` y `ObjetosJuego2()` requieren. Esto se visualiza en la Figura 3.23, que utiliza de ejemplo los elementos de la letra A. En el caso de los elementos Erróneo 1 y Erróneo 2, no se les asignaron objetos ya que el nivel 1 no los requiere.



Figura 3.23 Elementos de *ObjetosJuego* dentro de *marcador A*

También, se creó un algoritmo llamado `FinalizarNivel()`, observable en el Listado 3.6, cuya función es verificar que en cuanto se comience a reproducir el audio número cinco de felicitaciones, realice un conteo que redirija el programa al menú de juegos en cuanto el audio finalice.

Listado 3.6 Clase `FinalizarNivel()`

```

1. public class FinalizarNivel : MonoBehaviour{
2.     public AudioSource audio;
3.     public AudioClip clip;
4.     private float time;
5.     void Start(){
6.         time = 0;
7.     }
8.     void Update()
9.     {
10.         if(audio.isPlaying){
11.             Debug.Log("Llevo: " + time + " de " + clip.length);
12.             time += Time.deltaTime;
13.         }
14.         if(time >= 6){
15.             SceneManager.LoadScene(1);
16.         }
17.     }
18. }

```

Finalmente, debido a que la configuración de la aparición de los modelos extra se hizo a través de las interfaces físicas y no a código, cuando el usuario tocaba el modelo principal, en apariencia se reiniciaba el juego, es decir, que los modelos volvían a aparecer, aunque el puntaje no se reiniciaba. Para solucionar esto se escribió un algoritmo llamado `NoSpawn()`, que se observa en el Listado 3.7.

Listado 3.7 Clase NoSpawn()

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class NoSpawn : MonoBehaviour{
6.     [SerializeField] private PuntajeAgarrados puntaje;
7.     public GameObject uno, dos, tres;
8.     private bool primero, segundo, tercero;
9.     float puntos;
10.
11.     public void Excepciones(){
12.         puntos = puntaje.getPuntaje();
13.         Debug.Log(puntos);
14.         if (puntos > 0){
15.             primero = uno.activeSelf;
16.             segundo = dos.activeSelf;
17.             tercero = tres.activeSelf;
18.         }
19.     }
20.
21.     public void Actividad(){
22.         if (puntos > 0){
23.             uno.SetActive(primero);
24.             dos.SetActive(segundo);
25.             tres.SetActive(tercero);
26.         }
27.     }
28. }
```

Esta clase cuenta con dos métodos: `Excepciones()` y `Actividad()`. `Excepciones()` se encarga de guardar el estado actual de los modelos tridimensionales, si están activos o no, verificando en primer lugar, si el usuario ya acumuló puntaje. En cuanto a `Actividad()`, revisa que efectivamente el usuario ya haya generado puntos y de ser así, en lugar de permitir que se reactiven todos los objetos, toma los valores asignados en `Excepciones()` y vuelve a desactivar aquellos objetos que ya se desactivaron.

3.1.8 Sprint 8 Realización de video de ayuda

En la tabla 3.14 se describen las actividades que se realizaron en el *sprint* número ocho.

Tabla 3.14 Sprint 8.

<i>Sprint</i>	Nombre	Tareas
8	Realización de video de ayuda	Realización del video de ayuda
Propósito	Diseñar el video de ayuda para guía	
Estado	Finalizado	

En el video de ayuda que se realizó se explicaron de forma básica los elementos necesarios para el correcto funcionamiento del programa:

- Los marcadores
- *Leap Motion*[®]
- La plantilla para los marcadores

También se muestra de forma breve el funcionamiento del primer nivel. Esta explicación es principalmente orientada a los maestros o padres de familia que supervisarán a los alumnos durante la ejecución de los niveles. El audio que se reproduce en el video se puede leer en el punto 3.1.5 de este documento de tesis.

3.1.9 *Sprint 9* Diseño del nivel 2

En la tabla 3.15 se describen las actividades que se realizaron en el *sprint* número nueve.

Tabla 3.15 Sprint 9

<i>Sprint</i>	Nombre	Tareas
9	Diseño del nivel 2	Edición de los audios
		Edición de los modelos tridimensionales
		Codificación del nivel
Propósito	Modificar el nivel 1 para el desarrollo del nivel 2	
Estado	Finalizado	

A continuación, se explicarán las tareas realizadas para el desarrollo del nivel dos. Se toma en cuenta que gran parte de las configuraciones realizadas durante el nivel uno, se reutilizaron en el desarrollo del segundo nivel.

Este nivel funciona de la siguiente forma: se presenta un marcador frente a la cámara, que al reconocer la figura muestra un objeto tridimensional encima de la misma, cuando el usuario sujeta dicha figura, de fondo se reproduce el nombre de la misma. Cuando el modelo tridimensional se suelta, cinco objetos más aparecen en escena, tres tienen la misma funcionalidad que el primer modelo, al sujetarlas se reproducen audios con sus respectivos nombres. Los otros dos objetos comienzan con una letra diferente a la del marcador que se está repasando. Una vez que el estudiante sujetó los tres modelos y repitió sus nombres, debe arrastrar cada uno de estos objetos al objeto principal sobre el marcador, ganando un punto por cada uno de estos modelos. En caso de que se equivoque y arrastre hacia el objeto principal uno de los modelos cuyo nombre no comienza con la letra que se está repasando, los objetos visibles se reorganizarán para que el estudiante vuelva a jugar. Cuando el usuario acumula tres puntos, el nivel se termina.

3.1.9.1 Edición de los audios

Los audios de instrucciones utilizados en el nivel uno se modificaron, además de agregarse un sexto audio que se reproduce cada vez que el usuario comete un error, estos audios modificados se observan en la Figura 3.24.

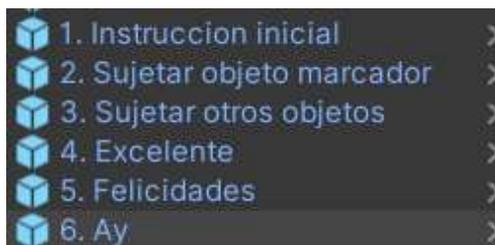


Figura 3.24 Audios instructivos modificados

Durante el transcurso de la edición de los audios se le agregaron efectos y se les editó la velocidad para facilitar su comprensión, tanto para el usuario como para el maestro o tutor que vigilará al estudiante.

3.1.9.2 Edición de los modelos tridimensionales

Lo primero que se hizo, fue asignar de forma totalmente aleatoria objetos extra a cada uno de los marcadores, la intención es que cada uno de estos objetos nuevos pertenecieran a otro marcador para que no fueran objetos desconocidos para el

estudiante y así evitar confundirlo. Estos modelos nuevos cumplirían con la función de proporcionar posibilidades erróneas al estudiante a fin de que empezara a discriminar unos objetos de otros en función del sonido con el que empieza su nombre. Como ejemplo, en el caso de la letra A, se agregaron como objetos un doctor y un oso, tal y como se observa en la Figura 3.25. En otros casos, como por ejemplo la letra F, se agregaron tres modelos erróneos.



Figura 3.25 Marcador A en el nivel dos.

Por otro lado, en las posiciones de cada uno de estos objetos se crearon `GameObjects`, con la finalidad de usarlos como puntos de reparación. Ya que la finalidad del nivel es que los estudiantes aprendan a discriminar los sonidos de las letras los unos de los otros, es necesario que cada vez que se inicie el nivel con la misma letra, los modelos tridimensionales aparezcan en posiciones diferentes a la ocasión anterior para evitar que el niño memorice las posiciones y complete el juego en función de su memoria en concordancia con el patrón del juego. La posición de la jerarquía interior de los marcadores se observa en la Figura 3.26.

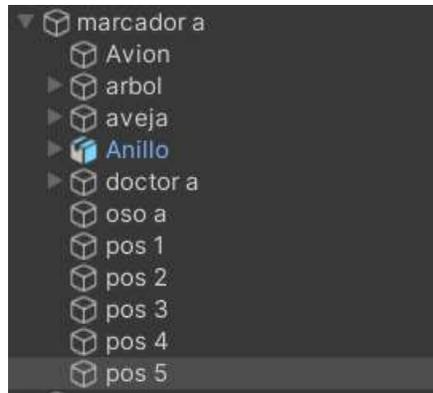


Figura 3.26 Jerarquía del marcador A

3.1.9.3 Codificación del nivel

En primera instancia se creó un algoritmo llamado `ObjetoErroneo()`, que se visualiza en el Listado 3.8.

Listado 3.8 Clase `ObjetoErroneo()`

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class ObjetoErroneo : MonoBehaviour {
6.     private Transform inicial;
7.     public AudioSource audio;
8.     public GameObject respawn;
9.
10.    private void OnTriggerEnter(Collider objeto) {
11.        if(objeto.tag == "Respawn"){
12.            gameObject.SetActive(false);
13.            respawn.GetComponent<Spawn>().Aparecer();
14.            audio.Play();
15.            gameObject.SetActive(true);
16.        }
17.    }
18. }
```

Esta clase verifica que cuando el objeto que haya entrado en contacto con el modelo tridimensional que se está tocando sea el modelo principal del marcador, en el caso de la letra A el avión, todos los objetos extra desaparezcan, se invoque un método llamado `Aparecer()`, se reproduzca el audio de error y los objetos que habían desaparecido reaparezcan en nuevas posiciones aleatorias.

El método `Aparecer()`, es el encargado de asignar a los objetos tridimensionales nuevas posiciones aleatorias. Esta función se encuentra de la clase `Spawn()`, se observa en el Listado 3.9 en la línea 12.

Listado 3.9 Clase `Spawn()`

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using static System.Random;
5. using System;
6.
7. public class Spawn : MonoBehaviour
8. {
9.     public Transform s1, s2, s3, s4, s5;
10.    public GameObject o1, o2, o3, o4, o5;
11.
12.    public void Aparecer(){
13.        int[] numeros = new int[5];
14.        int[] objetos = new int[5];
15.        System.Random num = new System.Random();
16.        int numeroAleatorio;
17.        for (int i = 0; i < numeros.Length; i++){
18.            do{
19.                numeroAleatorio = num.Next(1, 6);
20.            } while (Array.IndexOf(numeros, numeroAleatorio) != -1);
21.            numeros[i] = numeroAleatorio;
22.        }
23.        for (int i = 0; i < objetos.Length; i++){
24.            do{
25.                numeroAleatorio = num.Next(1, 6);
26.            } while (Array.IndexOf(objetos, numeroAleatorio) != -1);
27.            objetos[i] = numeroAleatorio;
28.        }
29.        for (int i = 0; i < objetos.Length; i++){
30.            if(objetos[i] == 1){
31.                if(numeros[i] == 1){
32.                    o1.transform.position = s1.position;
33.                }
34.                if(numeros[i] == 2){
35.                    o1.transform.position = s2.position;
36.                }
37.                if(numeros[i] == 3){
38.                    o1.transform.position = s3.position;
39.                }
40.                if(numeros[i] == 4){
41.                    o1.transform.position = s4.position;
42.                }
43.                if(numeros[i] == 5){
44.                    o1.transform.position = s5.position;
45.                }
46.            }
47.            if(objetos[i] == 2){
48.                if(numeros[i] == 1){
49.                    o2.transform.position = s1.position;
50.                }

```

Listado 3.9 Clase Spawn() (Continuación)

```
51.         if(numeros[i] == 2){
52.             o2.transform.position = s2.position;
53.         }
54.         if(numeros[i] == 3){
55.             o2.transform.position = s3.position;
56.         }
57.         if(numeros[i] == 4){
58.             o2.transform.position = s4.position;
59.         }
60.         if(numeros[i] == 5){
61.             o2.transform.position = s5.position;
62.         }
63.     }
64.     if(objetos[i] == 3){
65.         if(numeros[i] == 1){
66.             o3.transform.position = s1.position;
67.         }
68.         if(numeros[i] == 2){
69.             o3.transform.position = s2.position;
70.         }
71.         if(numeros[i] == 3){
72.             o3.transform.position = s3.position;
73.         }
74.         if(numeros[i] == 4){
75.             o3.transform.position = s4.position;
76.         }
77.         if(numeros[i] == 5){
78.             o3.transform.position = s5.position;
79.         }
80.     }
81.     if(objetos[i] == 4){
82.         if(numeros[i] == 1){
83.             o4.transform.position = s1.position;
84.         }
85.         if(numeros[i] == 2){
86.             o4.transform.position = s2.position;
87.         }
88.         if(numeros[i] == 3){
89.             o4.transform.position = s3.position;
90.         }
91.         if(numeros[i] == 4){
92.             o4.transform.position = s4.position;
93.         }
94.         if(numeros[i] == 5){
95.             o4.transform.position = s5.position;
96.         }
97.     }
98.     if(objetos[i] == 5){
99.         if(numeros[i] == 1){
100.             o5.transform.position = s1.position;
101.         }
102.         if(numeros[i] == 2){
103.             o5.transform.position = s2.position;
104.         }
105.         if(numeros[i] == 3){
106.             o5.transform.position = s3.position;
107.         }
108.         if(numeros[i] == 4){
```

Listado 3.9 Clase Spawn() (Continuación)

```
109.         o5.transform.position = s4.position;
110.     }
111.     if (numeros[i] == 5) {
112.         o5.transform.position = s5.position;
113.     }
114. }
115. }}}
```

La clase `Spawn()` recibe los cinco objetos presentes en la escena (línea 10), así como las cinco posiciones de los objetos de juego (línea 9), que se pusieron en las posiciones específicas alrededor del marcador en la sección 3.1.9.2. Se declararon dos arreglos, uno para almacenar los números aleatorios (línea 13) y otro para almacenar los objetos (línea 14). En dos ciclos se generaron números aleatorios del uno al cinco (líneas 17 y 23), de forma que no se repitan dentro del mismo arreglo, para esto se utilizaron bucles anidados que revisan nuevamente todo el arreglo. En un tercer ciclo se verifica cada número en el arreglo de los objetos y los objetos se movieron a la posición correspondiente de cada uno de los números aleatorios (línea 29).

Este código se incrustó dentro de los marcadores como tal, no dentro de los objetos del marcador. Se le asignaron las posiciones y los objetos del marcador. También `ObjetoErroneo()` se incrustó dentro de los objetos extras que se asignaron en este nivel y se le asignó también el objeto del marcador para especificar los objetos que se reordenaran aleatoriamente.

3.1.10 *Sprint 10* Diseño del nivel 3

En la tabla 3.16 se describen las actividades que se realizaron en el *sprint* número diez.

Tabla 3.16 *Sprint 10*

<i>Sprint</i>	Nombre	Tareas
10	Diseño del nivel 3	Edición de los audios Edición de los modelos tridimensionales
Propósito	Modificar el nivel 2 para el desarrollo del nivel 3	
Estado	Finalizado	

A continuación, se describirán las actividades que se llevaron a cabo durante la realización del nivel tres. Cabe mencionar que la codificación necesaria para dicho nivel es la explicada en los niveles anteriores, siendo la diferencia que se alteraron los modelos de figuras a palabras.

Este nivel funciona de la siguiente forma: se presenta un marcador frente a la cámara, que al reconocer la figura muestra un objeto tridimensional encima de la misma, cuando el usuario sujeta dicha figura, de fondo se reproduce el nombre de la misma. Cuando el modelo tridimensional se suelta, cinco objetos más aparecen en escena, son palabras, tres tienen la misma funcionalidad que el primer modelo, al sujetarlas se reproducen audios con sus respectivos nombres. Las otras dos palabras comienzan con una letra diferente a la del marcador que se está repasando. Una vez que el estudiante sujetó las cinco palabras y repitió sus nombres, debe arrastrar cada una de las palabras al objeto principal sobre el marcador, ganando un punto por cada una de las palabras correctas. En caso de que se equivoque y arrastre hacia el objeto principal uno de los modelos cuyo nombre no comienza con la letra que se está repasando, las palabras se reorganizarán para que el estudiante vuelva a jugar. Cuando el usuario acumula tres puntos, el nivel se termina.

3.1.10.1 Edición de los audios

En primer lugar, se modificaron todos los audios de la escena. Los audios de instrucciones se modificaron del uno al cinco para acomodarlos con respecto al nivel correspondiente. Esos audios se observan en la Figura 3.27.

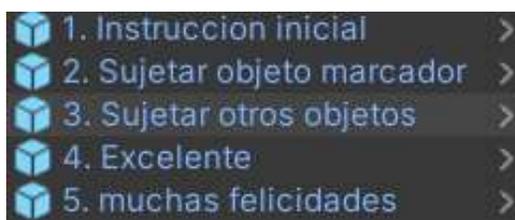


Figura 3.27 Audios de instrucción nivel tres

Luego se modificaron en la escena todos los audios de los objetos tridimensionales, bajo el concepto de que los audios de los niveles anteriores funcionaban con la

nomenclatura “letra de palabra”, por ejemplo “a de avión”, en este nivel los audios de los objetos son solo el nombre del mismo, por ejemplo en lugar de “o de oso” simplemente suena “oso”. Esto se puede observar en la Figura 3.28 usando como ejemplo la letra A.

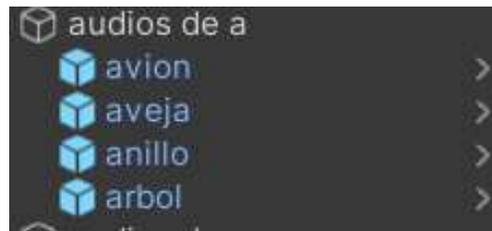


Figura 3.28 Audios de la letra A

3.1.10.2 Edición de los modelos tridimensionales

En los marcadores se procedió a cambiar los modelos extras de cada marcador por palabras en tres dimensiones. Como ejemplo, en el caso del marcador A, los objetos abeja, oso, anillo, doctor y árbol se cambiaron por sus palabras. También, al igual que en el nivel uno, a cada una de estas palabras se les agregaron los componentes `Mesh Collider` y `Interaction Behavior`, esto para agregarles bordes a las palabras y hacerlas manipulables por *Leap Motion*[®], esto se observa en la Figura 3.29.



Figura 3.29 Marcador A en el nivel cuatro

También, se volvieron a asignar los audios para cada palabra y se realizaron las configuraciones necesarias en cada una de las letras. El resultado final fue un nivel parecido en funcionamiento al nivel dos, pero en lugar de objetos tridimensionales se utilizaron palabras.

3.1.11 ***Sprint* 11 Diseño del nivel 4**

En la tabla 3.17 se describen las actividades que se realizaron en el *sprint* número once.

Tabla 3.17 Sprint 11

<i>Sprint</i>	Nombre	Tareas
11	Diseño del nivel 4	Edición de los audios
		Edición de los modelos tridimensionales
		Programación del nivel
Propósito	Desarrollar las funcionalidades del nivel 4	
Estado	Finalizado	

A continuación, se describirán las actividades que se realizaron durante la realización del nivel cuatro. Cabe mencionar que este nivel cuenta con aparentes subniveles, apareciendo en lugar de objetos palabras, de las cuales el usuario tiene que seleccionar las letras que estén relacionadas con el marcador que se está repasando.

Este nivel funciona de la siguiente forma: se presenta un marcador frente a la cámara, que al reconocer la figura muestra un objeto tridimensional encima de la misma, cuando el usuario sujeta dicha figura, de fondo se reproduce el nombre de la misma. Cuando el modelo tridimensional se suelta, aparece una palabra que contiene la letra que se está repasando. El estudiante debe arrastrar a la figura en el marcador las letras correctas. Si el usuario arrastra una letra incorrecta hacia la figura principal, esta reaparecerá en su lugar original. Cuando en una palabra se eliminan todas las letras correctas, automáticamente se acumulará un punto, la palabra se borrará y aparecerá la siguiente palabra con la que se jugará. Cuando el usuario acumula cuatro puntos, el nivel se termina.

3.1.11.1 Edición de los audios

En primera instancia, se editaron los audios de instrucciones para el nivel, específicamente los audios tres, cinco y seis. Luego, se eliminaron los audios extra, aquellos que no pertenecen a los modelos tridimensionales principales (los que aparecen cuando la cámara captura el marcador), tal y como se observa en la Figura 3.30.

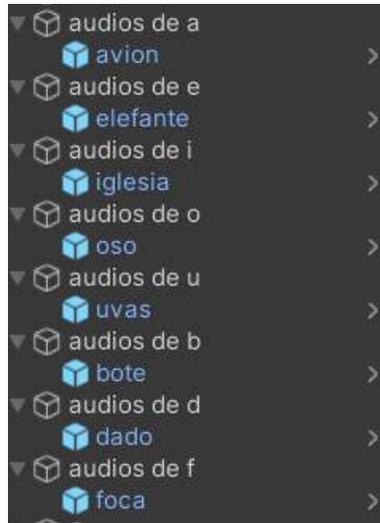


Figura 3.30 Audios del nivel

Esto se hizo así debido a que en este nivel se prueba que los estudiantes sean capaces de reconocer letras, por tanto, es necesario que el estudiante distinga el pictograma sin el sonido.

3.1.11.2 Edición de los modelos tridimensionales

Primero se eliminaron todos y cada uno de los objetos extras. Luego, utilizando Blender, se hicieron modelos tridimensionales de cada una de las letras minúsculas del abecedario, se colocó dentro de la subcarpeta “Letras sueltas”, en la dirección de “Assets/Modelos”. Lo anterior se observa en la Figura 3.31. Esto se hizo ya que las palabras utilizadas en el nivel anterior funcionan como objetos prefabricados haciendo imposible su separación en letras.

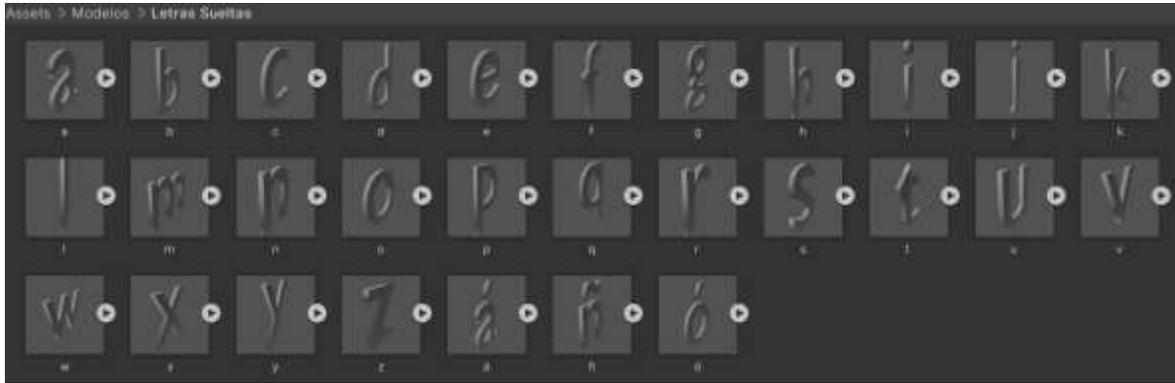


Figura 3.31 Letras tridimensionales

Luego, en los marcadores, tomando como ejemplo la letra A, se realizó lo siguiente: se crearon cuatro `GameObjects`, a los cuales se les asignó el nombre de las palabras que están almacenando. En este ejemplo se usaron las palabras abeja, árbol, avión y anillo, porque son las que se usaron para la letra A a lo largo de los anteriores niveles. Esto se observa en la Figura 3.32.

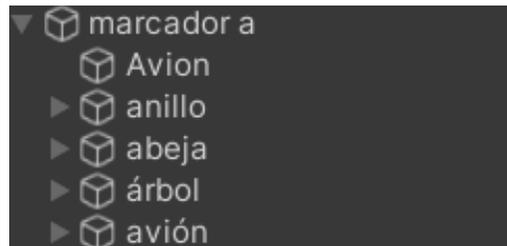


Figura 3.32 Objetos con palabras

Dentro de cada uno de estos objetos se colocaron en orden específico las letras para formar las palabras correspondientes, también se crearon objetos de juego secundario que funcionan de puntos de aparición para las letras incorrectas. Esto se visualiza en la Figura 3.33, que usa de ejemplo solo la palabra abeja.



Figura 3.33 Bloque de palabra abeja

A cada una de estas letras se les agregaron los componentes `Mesh Collider` e `Interaction Behaviour`, configurándolas de forma similar a las palabras del anterior nivel, con la excepción de que al sujetarlas no reproducen ningún sonido. Esta configuración se repitió con todas las palabras de cada uno de los bloques de las letras E, I, O, U, B, D y F. También se duplicó el objeto puntaje, existente dentro del Canvas de la UI, y se le renombró `puntaje temporal`, a fin de utilizarlo como variable global para la configuración del juego.

3.1.11.3 Codificación del nivel

En primer lugar, se creó un *script* llamado `SpawnCuatro()`, llamado así para diferenciarlo de la clase `Spawn()`, que se utiliza en los otros niveles. Este *script* se observa en el Listado 3.10 y sirve como base principal del minijuego de cada uno de los marcadores.

Listado 3.10 Clase `SpawnCuatro()`

```

1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class SpawnCuatro : MonoBehaviour
6. {
7.     private static List<GameObject> objetos = new List<GameObject>();
8.     private static int indiceActual = 0;
9.     public GameObject objeto1, objeto2, objeto3, objeto4;
10.    public GameObject puntaje;
11.
12.    public void Iniciar()
13.    {
14.        CrearListaObjetos();
15.        OrdenarObjetosAleatoriamente();
16.    }

```

Listado 3.10 Clase SpawnCuatro() (Continuación)

```
17.     private void CrearListaObjetos ()
18.     {
19.         objetos.Add(objeto1);
20.         objetos.Add(objeto2);
21.         objetos.Add(objeto3);
22.         objetos.Add(objeto4);
23.     }
24.
25.     private void OrdenarObjetosAleatoriamente(){
26.         int n = objetos.Count;
27.         while (n > 1){
28.             n--;
29.             int randomIndex = Random.Range(0, n + 1);
30.             GameObject temp = objetos[randomIndex];
31.             objetos[randomIndex] = objetos[n];
32.             objetos[n] = temp;
33.         }
34.     }
35.
36.     public GameObject ObtenerObjetoSiguiete(){
37.         if (indiceActual < objetos.Count)
38.         {
39.             GameObject objeto = objetos[indiceActual];
40.             indiceActual++;
41.             return objeto;
42.         }
43.         else
44.         {
45.             // Si ya se han entregado todos los objetos, regresar null
46.             return null;
47.         }
48.     }
49.
50.     public void ObtenerPrimerObjeto(){
51.         GameObject objeto = null;
52.
53.         if(!puntaje.activeSelf){
54.             objeto = objetos[0];
55.             indiceActual++;
56.             objeto.SetActive(true);
57.             Debug.Log("Deberia mostrarte el primer objeto");
58.             Debug.Log("El indice actual es: "+ indiceActual);
59.         }
60.     }
61. }
```

Este código administra la aparición en orden aleatorio de los cuatro bloques de palabras que integran el minijuego de cada marcador. Cuenta con las funciones `Iniciar()`, `CrearListaObjetos()`, `OrdenarObjetosAleatoriamente()`, `ObtenerObjetoSiguiete()`, y `ObtenerPrimerObjeto()`. La función `Iniciar()` solo se encarga de llamar a las funciones `CrearListaObjetos()` y `OrdenarObjetosAleatoriamente()`. `CrearListaObjetos()` toma los cuatro

bloques de las palabras del marcador seleccionado y los agrega a una lista dinámica de objetos. `OrdenarObjetosAleatoriamente()` utiliza un algoritmo llamado Fisher-Yates para organizar de forma aleatoria los objetos de la lista anteriormente mencionada. `ObtenerObjetoSiguiete()` obtiene el siguiente objeto de la lista objetos dependiendo del orden aleatorio que se ha determinado, utiliza una variable global para determinar la posición de la lista en la que se encuentra el minijuego. `ObtenerPrimerObjeto()` obtiene el primer objeto de la lista y lo activa, solo funciona si el objeto puntaje temporal no está activo, esto para determinar si el minijuego se ha iniciado ya y para de esta forma evitar que al tocar el objeto tridimensional principal se reinicie la dinámica. Este código se incrustó en los objetos marcadores de cada una de las letras. Después, se creó un *script* llamado `SiguientePalabra()`, que se incrustó en los modelos principales de cada marcador. Este Script se visualiza en el Listado 3.11.

Listado 3.11 Clase `SiguientePalabra()`

```
1. public class SiguientePalabra : MonoBehaviour
2. {
3.     public GameObject controlador;
4.     public void Siguiente(){
5.
6.         GameObject siguienteObjeto =
           controlador.GetComponent<SpawnCuatro>().ObtenerObjetoSiguiete();
7.
8.         if (siguienteObjeto != null) {
9.
10.            siguienteObjeto.SetActive(true);
11.        }else{
12.
13.            Debug.Log("No hay más objetos disponibles");
14.        }
15.    }
16. }
```

Este código utiliza el método siguiente para conseguir y activar el objeto posterior de la lista de palabras que se definió en la clase `SpawnCuatro()`, para ello utiliza el marcador en el que esta incrustada la clase en cuestión.

Luego, se creó un *script* llamado `Puntos()`, que se incrustó como componente dentro de todos los bloques de palabras. El algoritmo se visualiza en el Listado 3.12.

Listado 3.12 Clase Puntos()

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class Puntos : MonoBehaviour
6. {
7.     [SerializeField] private float puntos;
8.     [SerializeField] private PuntajeTemporal puntajeT;
9.     [SerializeField] private PuntajeAgarrados puntaje;
10.    public AudioSource exito, final;
11.    public GameObject objMarcador, felicitaciones, manos, palabras;
12.
13.    public void AgregarSecundario() {
14.        puntajeT.SumarPuntos();
15.        float p = puntajeT.getPuntaje();
16.
17.        if(p == puntos){
18.            puntaje.SumarPuntos();
19.            objMarcador.GetComponent<SiguientePalabra>().Siguiente();
20.            puntajeT.PerderPuntos();
21.            palabras.SetActive(false);
22.        }
23.
24.        if(puntaje.getPuntaje() != 4){
25.            exito.Play();
26.        }
27.        else{
28.            objMarcador.SetActive(false);
29.            manos.SetActive(false);
30.            felicitaciones.SetActive(true);
31.            final.Play();
32.        }
33.    }
34. }
```

Este código es el alma del minijuego, cuando se llama al método `AgregarSecundario()`, se suman puntos en la variable que almacena los puntos temporales. Cuando el jugador completa una palabra arrastrando todas las letras correctas, se verifica que estos puntos temporales hayan alcanzado los puntos necesarios para completar la palabra y pasar a la siguiente, terminando el minijuego. Una vez que se han acumulado cuatro puntos globales, el nivel se termina y se muestra el mensaje de felicitación. Para esto existe una variable llamada `puntos` que el programador asigna de manera manual, de forma que cada puntaje necesario para completar el minijuego y pasar a la siguiente palabra, dependerá de cada palabra en sí. El siguiente código desarrollado se llama `LetrasJuego()`, se incrustó en cada una de las letras correctas del juego, independientemente de si cuentan con acento o no. Este código se observa en el Listado 3.13.

Listado 3.13 Clase LetrasJuego()

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class LetrasJuego : MonoBehaviour
6. {
7.     [SerializeField] private Puntos puntaje;
8.
9.     private void OnTriggerEnter(Collider objeto){
10.
11.         if(objeto.tag != "Player" & objeto.tag == "Respawn"){
12.
13.             puntaje.AgregarSecundario();
14.             gameObject.SetActive(false);
15.         }
16.     }
17. }
```

Esta clase cuenta con un método *trigger* que verifica que el objeto entre en contacto con el modelo principal del marcador, en ese caso invoca el método `AgregarSecundario()` descrito en el Listado 3.12 y desactiva la letra. Finalmente, el código `LetraErronea()` se incrustó en cada uno de las letras erróneas, utilizándose además, audios de error cada vez que entra en contacto con el modelo tridimensional del marcador. El código se visualiza en el Listado 3.14.

Listado 3.14 Clase LetraErronea()

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. public class LetraErronea : MonoBehaviour
6. {
7.     public Transform inicial;
8.     public AudioSource error;
9.     private void OnTriggerEnter(Collider objeto){
10.         if(objeto.tag == "Respawn"){
11.             gameObject.SetActive(false);
12.             gameObject.transform.position = inicial.position;
13.             error.Play();
14.             gameObject.SetActive(true);
15.         }
16.     }
17. }
```

El *script* verifica que cuando el objeto entre en contacto con el modelo tridimensional principal, la letra se desactive, regrese a su posición inicial, se reproduzca el audio de error y se vuelva a activar el objeto.

Todos estos códigos y comportamientos se reprodujeron en cada uno de los marcadores, con sus respectivas palabras y letras. Esto da por concluido el desarrollo del programa.

Capítulo 4. Resultados

En este capítulo se describe la implementación del proyecto y los resultados obtenidos. Por tanto, se describe un caso de estudio enfocado a la educación en nivel preescolar. El principal propósito del programa es determinar si una herramienta de esta naturaleza facilita el aprendizaje de las letras. Debido a la edad de los usuarios del programa (niños de preescolar), los resultados se describirán a modo de observación.

4.1 *Sprint* 12 Aplicación a un caso de estudio

En la Tabla 4.1 se muestran las actividades que se realizaron en el *sprint* número doce.

Tabla 4.1 *Sprint* 12

<i>Sprint</i>	Nombre	Tareas
12	Aplicación de un caso de estudio	Buscar una institución para realizar las pruebas y validar el sistema. Validar el funcionamiento de los niveles desarrollados analizando su funcionamiento en un entorno real con usuarios de nivel preescolar, así como su utilidad para el aprendizaje de las letras.
Propósito	Aplicación del sistema para analizar su funcionamiento y utilidad en los procesos de aprendizaje de las habilidades de lectoescritura.	
Estado	Finalizado	

A continuación, se describirán las actividades realizadas durante el *sprint* doce, al igual que los resultados obtenidos.

4.1.1 Caso de estudio: Aprendizaje de las letras utilizando realidad aumentada con interacción a través de gestos manuales

Como resultado de los *sprints* anteriores, se desarrolló un videojuego a través del editor Unity3D, programado en lenguaje C#. El objetivo del programa es auxiliar en los procesos de aprendizaje de las letras en nivel preescolar. Da un giro al método tradicional de enseñanza que utiliza tarjetas relámpago y procura que los estudiantes enlacen los sonidos de las letras, con objetos físicos con las formas de

las mismas letras. Para el funcionamiento del programa se usan dos cámaras: una cámara Web y *Leap Motion*[®], este último rastrea el movimiento de las manos.

En cuanto se inicia el programa, en primer lugar, se muestra el menú principal con las opciones para ir al menú de selección de niveles, al video de ayuda y a los créditos de desarrollo, así como el botón que finaliza el programa. Este menú se observa en la Figura 4.1.



Figura 4.1 Menú Principal

Al seleccionar el botón de ayuda, se redirige a una escena que de inmediato reproduce el video instructivo que está dirigido al tutor o maestro que apoyará al niño en el uso del programa. También, cuenta con un botón que redirige al menú principal y en cuanto se termina el video, el videojuego regresa a dicho menú. Parte de este video se observa en la Figura 4.2.



Figura 4.2 Recuadros del video de ayuda

Los créditos de desarrollo son un simple agradecimiento al Conacyt y al Instituto Tecnológico de Orizaba. Esta pantalla se observa en la Figura 4.3.



Figura 4.3 Créditos del programa

Cuando en el menú principal, se selecciona el botón jugar, se redirige al menú de selección de niveles, que se presenta en la Figura 4.4. Los cuadros parecidos a los marcadores o tarjetas relámpago son botones que redirigen el programa a cada uno de los niveles solicitados.



Figura 4.4 Menú de niveles

Cuando se selecciona cualquiera de los niveles, se tarda dos o tres minutos en cargar la escena, luego, con ayuda de los audios de instrucción, se guía al alumno a través del nivel para completarlo. En la Figura 4.5 se observa la captura del nivel uno durante su ejecución. Una vez que se completa el nivel, el videojuego se redirige al menú de niveles.



Figura 4.5 Funcionamiento del nivel uno

4.1.2 Caso de estudio: Jardín de Niños Covadonga

En esta sección se presenta el caso de estudio en la institución seleccionada.

4.1.2.1 Planteamiento del caso de estudio

Para la implementación del proyecto desarrollado se acudió al Jardín de Niños Covadonga ubicada en Orizaba, Veracruz. La estancia se realizó a cargo de la Lic. Marisol Porras Ramírez, directora de la institución, quien permitió probar la aplicación en alumnos de los tres años escolares para analizar el funcionamiento del programa y su utilidad en la enseñanza de las letras.

4.1.2.2 Características del caso de estudio

Las pruebas del proyecto contaron con la participación de 10 alumnos en total, pertenecientes a los tres años del jardín de niños. La distribución de los alumnos es la siguiente:

- 10 alumnos:
 - 2 alumnos de primer año.
 - 4 alumnos de segundo año.
 - 4 alumnos de tercer año.

En el caso de los alumnos de primer año, solo se cuenta con 4 niños inscritos en total, motivo por el cual solo se seleccionó a dos niños, mientras que en primero y

en segundo se seleccionaron a cuatro. En los tres casos, se seleccionaron en parejas a los niños que muestran un rezago en el aprendizaje de la lectura, así como a los niños con mayor aprovechamiento.

La tecnología que se utilizó para la aplicación del caso de estudio es la siguiente:

- Computadora ASUS TUF Gaming F15
- Procesador Intel Core i5 11400H a 2.70Ghz
- 8 GB de memoria RAM
- Windows 11 de 64 bits
- *Leap Motion Controller*[®]
- Webcam Logitech HD 1080p
- Unity3D
- Vuforia Engine

En la Figura 4.6 se muestra un alumno durante una de las pruebas en el caso de estudio, utilizando *Leap Motion*[®]. La foto se tomó con la autorización de la Lic. Marisol Porras Ramírez.



Figura 4.6 Estudiante realizando pruebas

4.1.2.3 Descripción de las pruebas del caso de estudio

Las pruebas que se realizaron estuvieron enfocadas al correcto funcionamiento del programa, su utilidad para favorecer la memorización de las letras y sus sonidos, así como los sentimientos generales de los estudiantes al utilizar el videojuego. En general, las tres tecnologías más importantes que se utilizan para el funcionamiento del programa son las siguientes:

Leap Motion Controller[®], que cuenta con su *plugin* para uso en línea. Es una cámara infrarroja que rastrea los movimientos de las manos, muestra un esqueleto con falanges para la manipulación e interacción con objetos en un entorno virtual. Se conecta directamente a la computadora.

Vuforia Engine, es un servicio en línea que almacena imágenes a modo de biblioteca para usar como marcadores. En la aplicación que se utilice accede a la cámara que tenga disponible el dispositivo y si durante la captura de imágenes, reconoce algún objeto similar a uno de los marcadores en su biblioteca, permite el despliegue de información. Actualmente solo tiene soporte para dispositivos móviles, aunque para su uso en computadoras es posible probarlo en motores de videojuegos.

Unity3D, es un motor de videojuegos. Permite la integración de los *plugins* de Vuforia y *Leap Motion*[®]. Debido a la necesidad de usar un dispositivo de computadora para usar *Leap Motion*[®], pero ya que Vuforia Engine dio de baja su servicio para computadoras, el programa se tuvo que probar directamente en el motor de videojuegos.

Debido a que la forma de evaluar a estudiantes de preescolar está basada más bien, en la observación de los maestros y en el encargo de actividades, las pruebas para verificar la funcionalidad del programa se basaron en la observación y preguntas de control.

También, cabe mencionar, que el aprovechamiento educativo en esta escuela, al ser un jardín de niños privado, está adelantada a la mayoría de las escuelas públicas

de la zona, por tanto, a pesar de que originalmente se planeó que el programa se pusiera a prueba en niños de tercer año, se terminó usando para los tres niveles de educación.

La implementación del caso de estudio se realizó en la biblioteca de la escuela, con pocas distracciones y ruido.

4.1.2.4 Resultados obtenidos del caso de estudio

Se comenzó con los alumnos de primer grado, en los cuales se probaron los primeros dos niveles. Con ayuda de los niños de segundo grado se probaron los niveles dos y tres, mientras que los alumnos de tercer año auxiliaron en las pruebas de los niveles tres y cuatro. En todos los casos, previo al inicio del juego se realizaron preguntas de control a los niños, a todos los estudiantes se les mostró el marcador de las letras que iban a repasar, para luego cuestionarles:

1. ¿Cómo se llama esto? (Señalando al objeto en el marcador, se solicita una vocalización del nombre).
2. ¿Con qué letra empieza? (Se solicita vocalización de la letra).
3. ¿Cómo se ve esa letra? (Se le muestran diversos marcadores de las letras, esperando que seleccione una).

El siguiente paso fue ejecutar el programa. Al inicio de la prueba, se le solicitó al estudiante, señalándose por medio de *Leap Motion*[®], que dijera los nombres de los objetos que conociera. En la mayoría de los casos los objetos fueron fácilmente reconocidos por los estudiantes, aunque se observó cierta dificultad de *Leap Motion*[®] para identificar las formas de las manos debido a dos factores: el tamaño de los niños y la ejecución del programa directamente en el editor.

Posteriormente, durante la ejecución del juego, se le fue guiando a los niños para completar el nivel, a la vez que se les recordaba el repetir los nombres de los objetos y las palabras que sujetaron a través de *Leap Motion*[®]. En la Figura 4.7 se observa a los niños utilizando el videojuego.



Figura 4. 7 Estudiantes usando el programa

Una vez terminado el repaso de las dos letras correspondientes por cada alumno, se les realizaron cada vez preguntas de retroalimentación. Dichas preguntas incluían un repaso por los cuestionamientos que se hicieron previo a la ejecución del programa:

1. ¿Cómo se llama esto? (Señalando al objeto en el marcador, se solicita una vocalización del nombre).
2. ¿Con qué letra empieza? (Se solicita vocalización de la letra).
3. ¿Cómo se ve esa letra? (Se le muestran diversos marcadores de las letras, esperando que seleccione una).
4. ¿Te gusto el juego?
5. ¿Se te hizo difícil en algún momento?
6. Menciona que te gustó del juego.
7. ¿Volverías a jugarlo?

Cabe mencionar que, debido a la falta de tiempo, pues programas educativos de esta naturaleza requieren al menos un periodo de prueba de seis meses, con cada alumno se repasaron dos letras únicamente, además, entre prueba y prueba se tuvo un tiempo de espera de al menos dos días para comprobar el aprendizaje a largo plazo. Por la premura del tiempo se realizaron solamente cuatro iteraciones con cada alumno por grupo.

En la Tabla 4.2 se presentan las observaciones realizadas durante el proceso de cuestionamientos previos, así como posteriores a la ejecución del juego en las rondas realizadas para probar el programa y su utilidad para el aprendizaje de los niños.

Tabla 4.2 Rondas de pruebas

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego						
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?
Primera	1°	Alumno 1 (Adelantado)	E	Elefante	L	Identificación errónea	Elefante	E	Identificación rápida y correcta	Sí, mucho	No.	Pude hacer puntos.	Sí.
			I	Iglesia	I	Identificación correcta y rápida.	Iglesia	I	Identificación rápida y correcta				
		Alumno 2 (Rezagado)	O	Desconoce	Desconoce	Identificación errónea	Oso	O	Identificación errónea	Sí, mucho	No.	Pude mover las cosas.	Sí.
			U	Uva	Desconoce	Identificación errónea	Uva	U	Identificación correcta				
	2°	Alumno 3 (Adelantado)	A	Avión	A	Identificación Correcta	Avión	A	Identificación rápida	Sí, mucho	No.	Me gustan los sonidos que hace.	Sí.
			E	Elefante	E	Identificación correcta y rápida.	Elefante	E	Identificación rápida y correcta				
		Alumno 4 (Adelantado)	I	Iglesia	Desconoce	Identificación errónea	Iglesia	I	Identificación correcta	Sí.	No.	Me gustan los objetos.	Sí.
			O	Oso	Desconoce	Identificación correcta	Oso	O	Identificación correcta				
		Alumno 5 (Rezagado)	U	Uvas	U	Identificación rápida y correcta	Uvas	U	Identificación correcta	Sí	Agarrar uno de los objetos de la letra A	Le gustaron los marcadores	No.
			A	Avión	A	Identificación rápida y correcta	Avión	A	Identificación correcta				
	Alumno 6 (Rezagado)	E	Elefante	E	Identificación correcta	Elefante	E	Identificación correcta	Sí	Sí, al nombrar las cosas	No se.	Sí.	
		I	Casa	I	Identificación correcta	Iglesia	I	Identificación correcta					

Tabla 4.3 Rondas de pruebas (Continuación)

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego							
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?	
Primera	3°	Alumno 7 (Adelantado)	O	Oso	O	Identificación rápida y correcta	Oso	O	Identificación rápida y correcta	Sí, mucho	No	Pude agarrar las palabras	Sí.	
			U	Uvas	U	Identificación rápida y correcta	Uvas	U	Identificación rápida y correcta					
		Alumno 8 (Adelantado)	A	Avión	A	Identificación rápida y correcta	Avión	A	Identificación rápida y correcta	Sí	No	Pude agarrar las palabras.	Sí.	
			E	Elefante	E	Identificación correcta y rápida	Elefante	E	Identificación correcta y rápida					
		Alumno 9 (Rezagado)	I	Iglesia	I	Identificación lenta	Iglesia	I	Identificación correcta	Sí, mucho	No	los puntos, agarrar las cosas, ganar	Sí.	
			O	Oso	O	Identificación correcta	Oso	O	Identificación rápida					
	Alumno 10 (Rezagado)	U	Uvas	O	Identificación incorrecta	U	U	Identificación rápida	Sí mucho	No	Agarrar las palabras.	Sí		
		A	Avión	Desconoce	Identificación correcta	A	A	Identificación rápida						
	Segunda	1°	Alumno 1 (Adelantado)	E	Elefante	L	Identificación correcta	Elefante	E	Identificación correcta.	Sí, mucho.	No.	Hacer puntos.	Sí.
				I	Iglesia	I	Identificación correcta.	Iglesia	I	Identificación correcta.				
Alumno 2 (Rezagado)			O	Oso	Desconoce	Identificación correcta.	Oso	Desconoce	Identificación correcta	Sí	Sí	Mover las manos.	Sí.	
			U	Uva	Desconoce	Desconoce	Oso	U	Identificación correcta					

Tabla 4.4 Rondas de pruebas (Continuación)

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego						
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?
Segunda	2°	Alumno 3 (Adelantado)	A	Avión	A	Identificación rápida y correcta	Avión	A	Identificación rápida y correcta	Sí, mucho	No, nada	Que pude lanzar los objetos con las manos	Sí.
			E	Elefante	E	Identificación rápida y correcta	Elefante	E	Identificación rápida y correcta				
		Alumno 4 (Adelantado)	I	Iglesia	I	Identificación correcta	Iglesia	I	Identificación rápida y correcta	Sí, mucho	No, nada	Me gusta que hablan	Sí.
			O	Oso	U	Identificación correcta	Oso	O	Identificación rápida y correcta				
		Alumno 5 (Rezagado)	U	Uvas	U	Identificación rápida y correcta	Uvas	U	Identificación correcta	Sí	Agarrar uno de los objetos de la letra A	Le gustaron los marcadores	No.
			A	Avión	A	Identificación rápida y correcta	Avión	A	Identificación correcta				
		Alumno 6 (Rezagado)	E	Elefante	E	Identificación correcta	Elefante	E	Identificación correcta	Sí	Si, no sabe porque.	No se.	Sí.
			I	Iglesia	E	Identificación incorrecta	Iglesia	I	Identificación correcta				
	3°	Alumno 7 (Adelantado)	O	Oso	O	Identificación correcta	Oso	O	Identificación correcta	Sí, muchísimo	No.	Que puedo mover palabras	Sí, por favor.
			U	Uvas	U	Identificación correcta	Uvas	U	Identificación correcte				
		Alumno 8 (Adelantado)	A	Avión	A	Identificación correcta	Avión	A	Identificación correcta	Sí	No	Que tengo que colocar las palabras bien	Sí.
			E	Elefante	E	Identificación correcta	Elefante	E	Identificación correcta				

Tabla 4.5 Rondas de pruebas (Continuación)

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego						
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?
Segunda	3°	Alumno 9 (Rezagado)	I	Iglesia	I	Identificación correcta	Iglesia	I	Identificación correcta	Sí, mucho	Sí, mi mano desaparecía	Agarrar las palabras	Sí
			O	Oso	O	Identificación correcta	Oso	O	Identificación correcta				
		Alumno 10 (Rezagado)	U	Uvas	U	Identificación correcta	Uvas	U	Identificación correcta	Sí, mucho	No, nada	Jugar con las palabras	Sí
			A	Avión	A	Identificación correcta	Avión	A	Identificación correcta y rápida				
Tercera	1°	Alumno 1 (Adelantado)	E	Elefante	L	Identificación correcta y rápida.	Elefante	L	Identificación correcta y rápida.	Sí.	No.	Ganar.	Sí.
			I	Iglesia	I	Identificación correcta y rápida.	Iglesia	I	Identificación correcta y rápida.				
		Alumno 2 (Rezagado)	O	Oso	Desconoce	Identificación errónea	Oso	Desconoce	Identificación errónea	Sí.	Sí.	No me gusto repetir.	No.
			U	Uva	Desconoce	Identificación errónea	Uva	Desconoce	Identificación errónea				
	2°	Alumno 3 (Adelantado)	A	Avión	A	Identificación correcta	Avión	A	Identificación correcta	Sí	No	Las palabras se mueven	Sí
			E	Elefante	E	Identificación correcta	Elefante	E	Identificación correcta				
		Alumno 4 (Adelantado)	I	Iglesia	I	Identificación correcta	Iglesia	I	Identificación correcta	Sí	No	Que habla cuando agarro las cosas	Si
			O	Oso	O	Identificación correcta	Oso	Oso	Identificación correcta				

Tabla 4.6 Rondas de pruebas (Continuación)

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego						
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?
Tercera	2°	Alumno 5 (Rezagado)	U	Uvas	U	Identificación rápida y correcta	Uvas	U	Identificación rápida y correcta	Sí, mucho	no	Los objetos que hay que agarrar	Sí, por favor
			A	Avión	A	Identificación rápida y correcta	Avión	A	Identificación rápida y correcta				
		Alumno 6 (Rezagado)	E	Elefante	E	Identificación correcta	Elefante	E	Identificación correcta	Sí, mucho	No	Los objetos están bonitos	Sí.
			I	Iglesia	I	Identificación incorrecta	Iglesia	I	Identificación incorrecta				
	3°	Alumno 7 (Adelantado)	O	Oso	O	Identificación correcta	Oso	O	Identificación correcta	Sí, mucho	Agarrar bien las palabras	Me gusta mover las palabras al objeto	Sí
			U	Uvas	U	Identificación correcta	Uvas	U	Identificación correcta				
		Alumno 8 (Adelantado)	A	Avión	A	Identificación correcta y rápida	Avión	A	Identificación correcta y rápida	Sí, mucho	Nada	Me gusta mucho agarrar las palabras con mis manos	Sí
			E	Elefante	E	Identificación correcta y rápida	Elefante	E	Identificación correcta y rápida				
		Alumno 9 (Rezagado)	I	Iglesia	I	Identificación rápida	Iglesia	I	Identificación rápida	Sí mucho	No	Me gusta arrastrar palabras y ganar puntos	Sí
			O	Oso	O	Identificación rápida	Oso	O	Identificación rápida				
		Alumno 10 (Rezagado)	U	Uvas	U	Identificación rápida	Uvas	U	Identificación rápida	Sí	No	Me gusta jugar	Sí
			A	Avión	A	Identificación rápida	Avión	A	Identificación rápida				

Tabla 4.7 Rondas de pruebas (Continuación)

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego						
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?
Cuarta	1°	Alumno 1 (Adelantado)	E	Elefante	L	Identificación correcta y rápida	Elefante	E	Identificación correcta y rápida.	Sí, mucho	No.	Que tengo que arrastrar las cosas.	Sí.
			I	Iglesia	I	Identificación correcta y rápida	Iglesia	I	Identificación correcta y rápida				
		Alumno 2 (Rezagado)	O	Oso	P	Identificación lenta y correcta	Oso	O	Identificación correcta	Sí.	Sí, tengo que pronunciar.	Ya no me gustó	No.
			U	Uvas	O	Identificación incorrecta	Uvas	U	Identificación correcta				
	2°	Alumno 3 (Adelantado)	A	Avión	A	Identificación rápida y correcta	Avión	A	Identificación rápida	Sí	No	Que puedo mover todas las cosas	Sí
			E	Elefante		Identificación rápida y correcta	Elefante	E	Identificación rápida				
		Alumno 4 (Adelantado)	I	Iglesia	I	Identificación rápida y correcta	Iglesia	I	Identificación rápida y correcta	Sí	No	Que habla	Sí
			O			Identificación rápida			Identificación rápida				
		Alumno 5 (Rezagado)	U	Uvas	U	Identificación rápida	Uvas	U	Identificación rápida	Sí, mucho	no	Los objetos que hay que agarrar	Sí.
			A	Avión	A	Identificación rápida	Avión	A	Identificación rápida				
	Alumno 6 (Rezagado)	E	Elefante	E	Identificación correcta	Elefante	E	Identificación correcta	Sí, mucho	No	Los objetos me gustan	Sí.	
		I	Iglesia	I	Identificación incorrecta	Iglesia	I	Identificación correcta					

Tabla 4.8 Rondas de pruebas (Continuación)

Ronda	Grado	Nombre Alumno	Previo al juego				Posterior al juego							
			Letras	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Cómo se llama esto?	¿Con qué letra empieza?	¿Cómo se ve esa letra?	¿Te gustó el juego?	¿Se te hizo difícil en algún momento?	Menciona que te gustó del juego.	¿Volverías a jugarlo?	
Cuarta	3°	Alumno 7 (Adelantado)	O	Oso	O	Identificación correcta	Oso	O	Identificación correcta	Sí, mucho	Agarrar bien las palabras	Me gusta mover las palabras al objeto	Sí	
			U	Uvas	U	Identificación correcta	Uvas	U	Identificación correcta					
		Alumno 8 (Adelantado)	A	Avión	A	Identificación correcta y rápida	Avión	A	Identificación correcta y rápida	Sí, mucho	Nada	Me gusta agarrar las palabras con mis manos	Sí	
			E	Elefante	E	Identificación correcta y rápida	Elefante	E	Identificación correcta y rápida					
		Alumno 9 (Rezagado)	I	Iglesia	I	Identificación rápida	Iglesia	I	Identificación rápida	Sí mucho	No	Me gusta ganar puntos	Sí	
			O	Oso	O	Identificación rápida	Oso	O	Identificación rápida					
		Alumno 10 (Rezagado)	U	Uvas	U	Identificación rápida	Uvas	U	Identificación rápida	Sí	No	Me gusta jugar	Sí	
			A	Avión	A	Identificación rápida	Avión	A	Identificación rápida					

Las observaciones generales realizadas a cada alumno son las siguientes:

- Alumno 1: Se repasaron las letras "e" e "i". Este alumno, que mostraba un nivel adelantado en el aprendizaje de las letras, pudo mejorar aún más con cada prueba. Debido a su facilidad para completar el programa, realizó las pruebas con suma rapidez y calma.
- Alumno 2: Se repasaron las letras "o" y "u". Este alumno demostró mejoras progresivas en el aprendizaje de las letras. El hecho de ganar puntos al arrastrar los objetos lo motivó en las primeras tres iteraciones a participar activamente en las tareas. Cabe mencionar que se detectó confusión al pedirle que vocalizara e indicara las letras, al realizársele pruebas con otras, se demostró que cuando una tercera persona vocaliza las letras, no tiene problema en identificarlas, lo cual parece indicar que, es posible que este alumno requiera una atención adicional y estrategias específicas para abordar su problema de vocalización.
- Alumno 3: Se repasaron las letras "a" y "e". Siendo el alumno más adelantado en general, se le permitió jugar en todos los niveles del programa. Demostró una adaptabilidad sorprendente al enfrentar desafíos más complejos y mostró un crecimiento continuo en su comprensión de las letras.
- Alumno 4: Se repasaron las letras "i" y "o". Al igual que el alumno 1, este estudiante adelantado mostró un entusiasmo notable al tocar los objetos y las palabras en el programa. Debido al interés tan enfático que demostraba en escuchar las indicaciones proporcionadas en los audios del programa, parece indicar que es una persona más auditiva.
- Alumno 5: Se repasaron las letras "u" y "a". El alumno 5 demostró tener un pequeño problema de concentración derivado de un ligero rezago en la capacidad para vocalizar palabras, a pesar de esto, dicho problema se fue aminorando a medida que se realizaban las pruebas. El hecho de ganar puntos y ver su progreso lo motivó a prestar más atención y a esforzarse aún más en las actividades.

- Alumno 6: Se repasaron las letras "e" e "i". Un caso similar al del alumno número 5, a pesar de que en las primeras tres iteraciones, parecía no haber un aprendizaje sólido a largo plazo, en la cuarta iteración hubo una facilidad y rapidez sorprendentes en la prueba.
- Alumno 7: Se repasaron las letras "o" y "u". Al ser un alumno adelantado, este estudiante tuvo la oportunidad de jugar con otras letras más allá de las evaluadas. Esta flexibilidad en la utilización de programa le permitió seguir disfrutando del programa a largo plazo.
- Alumno 8: Se repasaron las letras "a" y "e". Este estudiante adelantado se mostró muy entusiasmado al interactuar con los objetos y las palabras del programa. Disfrutó mucho de las actividades y demostró una gran destreza en el aprendizaje de las letras. Fuera del alumno 2, fue el estudiante que sorteó los niveles con mayor facilidad y rapidez.
- Alumno 9: Se repasaron las letras "i" y "o". A medida que participaba en las pruebas, este alumno rezagado mostró una mejora gradual en su aprendizaje de las letras. El programa, al brindarle la oportunidad de ganar puntos y progresar, generó un mayor interés y motivación en su participación activa. El alumno se mostro especialmente emocionado con el hecho de que estaba ayudando a probar un videojuego, según él mismo, es muy fanático de videojuegos como Roblox® y Minecraft®, lo cual puede haber ayudado en su desarrollo e interés.
- Alumno 10: Se repasaron las letras "u" y "a". Además de experimentar mejoras en el aprendizaje de las letras a lo largo de las pruebas, se observó una disminución en los problemas de concentración de este alumno rezagado. La interacción con los objetos y las palabras en el programa captó su atención y ayudó a desarrollar su enfoque y concentración.

En resumen, los resultados de las pruebas en los diez niños mostraron mejoras en el aprendizaje de las letras. La mayoría de los alumnos, tanto los adelantados como los rezagados, demostraron avances significativos a lo largo de las iteraciones.

Los alumnos adelantados (1, 3, 4, 7 y 8) se destacaron por su facilidad para completar el programa, lo que les permitió explorar y jugar con otras letras adicionales. Además, mostraron un gran entusiasmo al tocar los objetos y las palabras, lo que indica una preferencia por el aprendizaje auditivo o táctil.

Por otro lado, los alumnos rezagados (2, 5, 6, 9 y 10) presentaron mejoras progresivas en el aprendizaje de las letras a medida que participaban en las pruebas. Ganar puntos y ver su progreso los motivó a prestar más atención y a esforzarse en las actividades. Además, se observó una disminución en los problemas de concentración de algunos de ellos.

En un punto aparte, es importante resaltar que, aplicado a largo plazo, el programa creado tiene el potencial de nivelar el aprendizaje de la lectoescritura entre niños adelantados y rezagados. Los resultados obtenidos indican que el programa fue efectivo en mejorar las habilidades de los alumnos y que, con un enfoque personalizado y estratégico, se pueden abordar las dificultades específicas de cada niño.

Capítulo 5. Conclusiones y recomendaciones

En este capítulo, se presentan las conclusiones obtenidas tras la implementación del caso de estudio, así como las recomendaciones para el proyecto.

5.1 Conclusiones

Los resultados obtenidos a través de las preguntas de control previas y posteriores al uso del programa destacan varios aspectos importantes. En primer lugar, se observó que nueve de los diez niños participantes mostraron un interés pleno y sostenido en el uso del programa a lo largo del tiempo. Esto sugiere que la herramienta desarrollada captó su atención y generó motivación en su participación activa.

Además, se evidenció que la herramienta fue especialmente útil para aquellos niños con déficit de atención diagnosticada. El programa proporcionó una experiencia interactiva y estimulante que ayudó a mantener su enfoque y compromiso durante las actividades de aprendizaje. Con respecto a esto, la utilización de una interfaz humano-máquina se reveló como un aspecto fundamental del programa. Todos los niños manifestaron su gusto por la capacidad de agarrar los objetos y las palabras, y de poder arrastrarlos literalmente, lo que les brindó una experiencia de juego única. Este enfoque táctil y manipulativo resultó altamente motivador y enriquecedor para su participación.

Los niños que presentaban problemas en el lenguaje, pronunciación de palabras y otras dificultades demostraron mayores dificultades en el uso de la aplicación. Esto sugiere que el programa puede ser especialmente beneficioso para detectar y posiblemente desarrollar estrategias en estas áreas específicas de aprendizaje.

El programa también demostró su capacidad para ayudar a los niños rezagados en el aprendizaje de las letras a discriminar y reconocer las palabras con mayor facilidad. Esto implica que el programa puede contribuir a nivelar las habilidades de lectoescritura entre los niños con diferentes niveles de rendimiento académico. Además, se observó que el programa auxilió en el aprendizaje de nuevas palabras, ampliando así el vocabulario de los niños participantes. Esto indica que el programa

puede tener un impacto positivo en el desarrollo del lenguaje y la comprensión verbal.

Finalmente, los niños demostraron una pronunciada avidez por seguir jugando y participando en las actividades del programa. El hecho de poder tomar los objetos y las palabras para jugar con ellas fue especialmente atractivo y motivador para los niños.

En conclusión, el programa desarrollado mostró resultados prometedores en términos de interés, motivación y mejora en el aprendizaje de las letras. La herramienta resultó especialmente beneficiosa para los niños con déficit de atención, y su enfoque táctil y manipulativo fue altamente atractivo para todos los participantes. El programa demostró su capacidad para abordar dificultades específicas en el lenguaje y mejorar el reconocimiento y comprensión de las palabras. En conjunto, estos hallazgos respaldan la conclusión de que el programa tiene el potencial de ser una herramienta efectiva y enriquecedora para el aprendizaje de la lectoescritura en niños con diferentes niveles de habilidad.

5.2 Recomendaciones

Las recomendaciones específicas son las siguiente:

- **Expansión de las letras:** Se recomienda la expansión del programa para incluir todas las letras del abecedario. Esto permitirá a los niños tener una exposición más completa y amplia a los diferentes sonidos y formas de las letras, fortaleciendo su aprendizaje en general. Además, se sugiere diseñar actividades específicas para cada letra, brindando ejercicios interactivos que promuevan la identificación y asociación de cada letra con su respectivo sonido.
- **Módulo de enseñanza de sílabas:** Se sugiere la incorporación de un módulo adicional enfocado en la enseñanza de sílabas. Este módulo puede ayudar a los niños a comprender cómo se forman las palabras a través de la combinación de diferentes sílabas, lo que les dará una base sólida para su desarrollo lingüístico. Este módulo puede incluir actividades interactivas que

permitan a los niños combinar sílabas y formar palabras, fortaleciendo así su comprensión fonética y su habilidad para construir palabras.

- Investigación de interfaces humano-máquina: Es importante llevar a cabo una investigación exhaustiva sobre interfaces humano-máquina que sean capaces de detectar manos más pequeñas. Dado que se ha identificado que el programa actual presenta dificultades para detectar las manos de los niños más pequeños, se recomienda explorar opciones tecnológicas que permitan una detección precisa y adecuada en este grupo de usuarios. Esto garantizará que todos los niños, independientemente de su tamaño, puedan interactuar de manera efectiva con el programa.
- Pruebas con un grupo más amplio de niños: Se sugiere ampliar el grupo de participantes en las pruebas del programa. Involucrar a más niños con diferentes niveles de habilidad y antecedentes académicos permitirá obtener una perspectiva más amplia y representativa de los posibles usuarios. Esto proporcionará datos más sólidos sobre la efectividad del programa en diferentes contextos y permitirá realizar análisis comparativos más rigurosos.
- Implementación a largo plazo: Se recomienda evaluar la posibilidad de implementar el programa a largo plazo en entornos educativos. Para ello, se sugiere realizar un seguimiento de los niños participantes en el estudio para medir su progreso a lo largo del tiempo y determinar si el programa tiene un impacto sostenido en el aprendizaje de la lectoescritura. Esto permitirá evaluar la viabilidad de utilizar el programa como una herramienta complementaria en la enseñanza regular y explorar la posibilidad de colaborar con escuelas o instituciones educativas para su implementación a mayor escala.

Al considerar estas recomendaciones, se espera mejorar y expandir el programa, brindando a los niños una experiencia de aprendizaje más completa y efectiva en el desarrollo de sus habilidades de lectoescritura. Estas recomendaciones buscan fortalecer la base teórica y práctica del proyecto, así como proporcionar elementos para su posible implementación y escalabilidad en el ámbito educativo.

Anexos y/o apéndices

Productos académicos



Raúl de Jesús Sánchez Martínez, María Antonieta Abud Figueroa,
Ulises Juárez Martínez, Hilarión Muñoz Contreras, Lisbeth Rodríguez Mazahua

Arquitectura de software para utilizar realidad aumentada con control
gestual de manos en apoyo al aprendizaje de la lectoescritura

9JCyTA, noviembre 2022

Estado: Presentado.



Raúl de Jesús Sánchez Martínez, María Antonieta Abud Figueroa,
Ulises Juárez Martínez, Hilarión Muñoz Contreras, Lisbeth Rodríguez Mazahua

Sistema de Realidad Aumentada con control gestual de manos en
apoyo al aprendizaje de la lectoescritura

CICOS, septiembre 2023

Estado: Presentado.

Bibliografía

- [1] Z. Bassfar, “Educational Software Product: Investigating the Need for a New Software Methodology,” *2020 International Conference on Computing and Information Technology, ICCIT 2020*, pp. 77–80, 2020, doi: 10.1109/ICCIT-144147971.2020.9213755.
- [2] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, “Exploring the intersection between software industry and Software Engineering education - A systematic mapping of Software Engineering Trends,” *Journal of Systems and Software*, vol. 172, p. 110736, 2021, doi: 10.1016/j.jss.2020.110736.
- [3] S. Aukstakalnis, *Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR (Usability)*, 1st ed. Addison-Wesley Professional, 2016.
- [4] M. Sun, X. Wu, Z. Fan, and L. Dong, “Augmented reality based educational design for children,” *International Journal of Emerging Technologies in Learning*, vol. 14, no. 3, pp. 51–60, 2019, doi: 10.3991/ijet.v14i03.9757.
- [5] V. Marín-Díaz, J. Cabero-Almenara, and O. M. Gallego-Pérez, “Motivación y realidad aumentada: Alumnos como consumidores y productores de objetos de aprendizaje. Motivation and augmented reality: Students as consumers and producers of learning objects,” *Aula Abierta*, vol. 47, no. 3, p. 337, 2018, doi: 10.17811/aula_abierta.47.3.2018.337.
- [6] R. A. Roberto, J. P. Lima, R. C. Mota, and V. Teichrieb, “Authoring tools for augmented reality: An analysis and classification of content design tools,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9748, pp. 237–248, 2016, doi: 10.1007/978-3-319-40406-6_22.
- [7] T. Hewett *et al.*, *ACM SIGCHI Curricula for Human-Computer Interaction*. 1992. doi: 10.1145/2594128.
- [8] H. Q. Le and J. I. Kim, “An augmented reality application with hand gestures for learning 3D geometry,” *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, pp. 34–41, 2017, doi: 10.1109/BIGCOMP.2017.7881712.
- [9] A. Menache, “Motion Capture Primer,” *Understanding Motion Capture for Computer Animation*, pp. 1–46, 2011, doi: 10.1016/b978-0-12-381496-8.00001-9.
- [10] T. B. Moeslund, A. Hilton, and V. Krüger, “A survey of advances in vision-based human motion capture and analysis,” *Computer Vision and Image*

Understanding, vol. 104, no. 2-3 SPEC. ISS., pp. 90–126, 2006, doi: 10.1016/j.cviu.2006.08.002.

- [11] M. Oudah, A. Al-Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques,” *J Imaging*, vol. 6, no. 8, 2020, doi: 10.3390/JIMAGING6080073.
- [12] A. Efklides, “The role of metacognitive experiences in the learning process,” *Psicothema*, vol. 21, no. 1, pp. 76–82, 2009.
- [13] D. A. Kolb, “Management and the Learning Process,” *Calif Manage Rev*, vol. 18, no. 3, pp. 21–31, 1976, doi: 10.2307/41164649.
- [14] C. Perfetti, N. Landi, and J. Oakhill, “The Acquisition of Reading Comprehension Skill Introduction: Simple Ideas about Reading Comprehension,” *The Science of Reading: A Handbook*, pp. 227–247, 2005.
- [15] T. K. Shackelford and V. A. Weekes-Shackelford, Eds., “Learning Disorder,” in *Encyclopedia of Evolutionary Psychological Science*, Cham: Springer International Publishing, 2021, p. 4518. doi: 10.1007/978-3-319-19650-3_302831.
- [16] K. F. Miller, “Children’s Early Understanding of Writing and Language: The Impact of Characters and Alphabetic Orthographies,” in *Chinese Children’s Reading Acquisition: Theoretical and Pedagogical Issues*, L. Wenling, J. S. Gaffney, and J. L. Packard, Eds., Boston, MA: Springer US, 2002, pp. 17–29. doi: 10.1007/978-1-4615-0859-5_2.
- [17] M. Saada-Robert, K. Balslev, and K. Mazurczak, “Emergent Writing in Kindergarten and the Emergence of the Alphabetic Principle,” in *Effective Learning and Teaching of Writing: A Handbook of Writing in Education*, G. Rijlaarsdam, H. van den Bergh, and M. Couzijn, Eds., Dordrecht: Springer Netherlands, 2005, pp. 17–30. doi: 10.1007/978-1-4020-2739-0_2.
- [18] R. Guzmán Rosquete, “MÉTODOS DE LECTURA Y ACCESO AL LÉXICO,” Jan. 1997.
- [19] J. Varela, “Aprender a leer: Programa de Lectura,” Guadalajara, Jalisco, 2008. [Online]. Available: www.conducta.org
- [20] D. Salinas, C. De Moraes, and M. Schwabe, “Programa Para La Evaluación Internacional De Alumnos (Pisa) Pisa 2018 - Resultados - Nota País México,” *Ocde*, vol. I–III, pp. 1–12, 2018, [Online]. Available: https://www.oecd.org/pisa/publications/PISA2018_CN_MEX_Spanish.pdf
- [21] S. Schutera *et al.*, “On the potential of augmented reality for mathematics teaching with the application cleARmaths,” *Educ Sci (Basel)*, vol. 11, no. 8, 2021, doi: 10.3390/educsci11080368.

- [22] A. Karambakhsh, A. Kamel, B. Sheng, P. Li, P. Yang, and D. D. Feng, "Deep gesture interaction for augmented anatomy learning," *Int J Inf Manage*, vol. 45, pp. 328–336, 2019, doi: 10.1016/j.ijinfomgt.2018.03.004.
- [23] Y. et al. Sun, "Manipulating IoT Devices in Augmented Reality Environment MagicHand : A Deep Learning Approach towards Manipulating IoT Devices in Augmented Reality Environment," pp. 0–9, 2019.
- [24] P. Bifulco, F. Narducci, R. Vertucci, P. Ambruosi, M. Cesarelli, and M. Romano, "Telemedicine supported by Augmented Reality: An interactive guide for untrained people in performing an ECG test," *Biomed Eng Online*, vol. 13, no. 1, pp. 1–16, 2014, doi: 10.1186/1475-925X-13-153.
- [25] I. K. A. A. Putra and I. G. N. A. C. Putra, "Development of Augmented Reality Application for Canang Education Using Marker-Based Tracking Method," *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 9, no. 3, p. 365, 2021, doi: 10.24843/jlk.2021.v09.i03.p07.
- [26] M. Sarosa, A. Chalim, S. Suhari, Z. Sari, and H. B. Hakim, "Developing augmented reality based application for character education using unity with Vuforia SDK," *J Phys Conf Ser*, vol. 1375, no. 1, 2019, doi: 10.1088/1742-6596/1375/1/012035.
- [27] I. Nugraha, A. R. Suminar, D. W. Octaviana, M. T. Hidayat, and A. Ismail, "The application of augmented reality in learning English phonetics," *J Phys Conf Ser*, vol. 1402, no. 7, 2019, doi: 10.1088/1742-6596/1402/7/077024.
- [28] Mursyidah, Indrawati, and Ramadhona, "Augmented Reality Application for Introduction Traditional House and Objects of Aceh," *IOP Conf Ser Mater Sci Eng*, vol. 536, no. 1, 2019, doi: 10.1088/1757-899X/536/1/012152.
- [29] V. Lin, G. Z. Liu, and N. S. Chen, "The effects of an augmented-reality ubiquitous writing application: a comparative pilot project for enhancing EFL writing instruction," *Comput Assist Lang Learn*, vol. 0, no. 0, pp. 1–42, 2020, doi: 10.1080/09588221.2020.1770291.
- [30] N. SyafiqahSafiee and A. Wanis Ismail, "AR Home Deco: Virtual Object Manipulation Technique Using Hand Gesture in Augmented Reality," pp. 1–6, 2018.
- [31] B. Allagui, "Writing a Descriptive Paragraph Using an Augmented Reality Application: An Evaluation of Students' Performance and Attitudes," *Technology, Knowledge and Learning*, vol. 26, no. 3, pp. 687–710, 2021, doi: 10.1007/s10758-019-09429-2.
- [32] S. Yamada and N. P. Chandrasiri, "Evaluation of Hand Gesture Annotation in Remote Collaboration Using Augmented Reality," pp. 727–728, 2018.

- [33] et al. Bagner, William, "A tour of the C# language."
<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [34] D. K. Ponce, "*Análisis Comparativo De Los Entornos De Desarrollo Integrados (Ide): Eclipse, Netbeans Y Jdeveloper Para El Desarrollo De Aplicaciones Java Enterprise Edition Proyecto.*," 2019.
- [35] Microsoft, "Visual Studio Code." <https://visualstudio.microsoft.com/es/>
 (accessed Feb. 27, 2022).
- [36] H. ; Shamsee, N.; Klebenov, D.; Fayed, *CCNA Data Center DCICT 640-916: Official Cert Guide. Cisco.* 2015.
- [37] Vuforia, "Vuforia Engine Developer Library." <https://library.vuforia.com/>
 (accessed Feb. 27, 2022).
- [38] AgileAlliance, "Scrum."
[https://www.agilealliance.org/glossary/scrum/#q=~\(infinite~false~filters~\(post~Type~\(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video\)~tags~\(~'scrum\)\)~searchTerm~'~sort~false~sortDirection~'asc~page~1](https://www.agilealliance.org/glossary/scrum/#q=~(infinite~false~filters~(post~Type~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'scrum))~searchTerm~'~sort~false~sortDirection~'asc~page~1) (accessed Feb. 27, 2022).
- [39] SCRUM organization, "What is SCRUM?,"
<https://www.scrum.org/resources/what-scrum-module>.
- [40] J. Atherton and G. Wang, "Chunity: Integrated Audiovisual Programming in Unity," *New Interfaces for Musical Expression*, p. Virginia, US, 2018,
 [Online]. Available:
http://www.nime.org/proceedings/2018/nime2018_paper0024.pdf
- [41] J. Haas, "A History of the Unity Game Engine An Interactive Qualifying Project Submitted to the Faculty of WORCESTER POLYTECHNIC INSTITUTE in partial fulfillment of the requirements for graduation," p. 44, 2014, [Online]. Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/unrestricted/Haas_IQP_Final.pdf
- [42] UltraLeap, "Leap Motion® Controller."
<https://www.ultraLeap.com/product/leap-motion-controller/#overview>