



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Ingeniería de Software

**Desarrollo de juegos para la mejora de la
pronunciación**

Autor:

Jorge Obispo Tamariz



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Ingeniería de Software

**Desarrollo de juegos para la mejora de la
pronunciación**

Autor:

Jorge Obispo Tamariz

Tutores:

César González Ferreras

Mario Corrales Astorgano

¡La ciencia no se trata de por qué! ¡Se trata de por qué no!

Cave Johnson, Portal 2

Agradecimientos

Para empezar, agradecer a mis padres, por su apoyo constante y por ser mi fuente de inspiración. Gracias por creer en mí y por brindarme las herramientas necesarias para alcanzar mis metas.

A mis tutores, César González Ferreras y Mario Corrales Astorgano, por su guía y consejos a lo largo del proyecto. Sin su apoyo el TFG no habría salido adelante.

Por último, a mis amigos, por su paciencia y comprensión durante estos años. Vuestra compañía ha sido fundamental para superar los desafíos, dándome apoyo en los momentos difíciles.

Resumen

Este Trabajo de Fin de Grado se centra en el desarrollo de actividades educativas adicionales para el juego Pradia, diseñado para ayudar a personas con síndrome de Down a mejorar sus habilidades de habla y comprensión. Utilizando Unity, se han creado actividades interactivas que incorporan tecnologías de reconocimiento de voz. El proyecto también abarca la implementación de estas actividades de manera modular, asegurando su portabilidad.

Abstract

This Final Degree Project focuses on developing additional educational activities for the game Pradia. Designed to help people with Down syndrome improve their speech and comprehension skills. Using Unity, interactive activities incorporating voice recognition technologies have been created. The Project also includes the implementation of these activities in a modular manner, ensuring their portability.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Objetivos	3
1.4. Metodología	3
1.5. Estructura de la memoria	4
2. Estado de la cuestión	5
2.1. Uso de Unity en el desarrollo de actividades educativas	5
2.1.1. Unity	5
2.1.2. Unity en la educación	7
2.2. Elección de actividades	9
2.3. Impacto del reconocimiento de voz en el aprendizaje	12
2.3.1. Mejora de la conciencia fonológica y competencia lectora	13
2.3.2. Impacto de la IA y el reconocimiento de Voz en el aprendizaje	14
2.3.3. Implicaciones y Futuro de la Integración Tecnológica	14
3. Planificación	16
3.1. Planificación Inicial	16
3.2. Riesgos	17
3.3. Entorno tecnológico	19
3.4. Costes	20
3.4.1. Costes de personal	20
3.4.2. Costes Hardware	20

3.4.3.	Costes Software.....	20
3.5.	Desviaciones de la planificación.....	21
4.	Descripción de las iteraciones.....	23
4.1.	Iteración 1.....	23
4.1.1.	Tareas realizadas.....	23
4.2.	Iteración 2.....	24
4.2.1.	Implementación.....	25
4.2.2.	Implementación reconocimiento de voz.....	27
4.3.	iteración 3.....	29
4.4.	Iteración 4.....	33
4.4.1.	Parejas.....	33
4.4.2.	Sílabas Correctas.....	35
4.4.3.	Pares Mínimos.....	36
4.5.	iteración 5.....	37
4.5.1.	Grabación de audios.....	38
4.5.2.	Resultados.....	38
4.6.	iteración 6.....	39
5.	Análisis.....	40
5.1.	Actores del sistema.....	40
5.2.	Requisitos.....	40
5.2.1.	Requisitos Funcionales.....	40
5.2.2.	Requisitos no funcionales.....	41
5.2.3.	Requisitos de información.....	41
5.3.	Modelo de dominio.....	42
5.3.1.	Clases.....	42
5.4.	Diagramas de actividad.....	43

5.4.1.	Nómbrolo.....	43
5.4.2.	Selección correcta.....	44
5.4.3.	Discriminación de fonema.....	45
5.4.4.	Parejas.....	46
5.4.5.	Sílabas correctas.....	47
5.4.6.	Pares mínimos.....	48
6.	Diseño.....	50
6.1.	Estructura del proyecto.....	50
6.2.	Patrones de diseño.....	51
6.2.1.	Singleton.....	51
6.2.2.	Observador.....	51
6.2.3.	Polimorfismo.....	52
6.2.4.	Patrón fachada.....	52
6.2.5.	Patrón Update Method y Game Loop.....	52
6.3.	Diagrama de arquitectura.....	53
6.4.	Diagrama de paquetes.....	54
6.5.	Diagrama de despliegue.....	57
6.6.	Diseño de la Interfaz de usuario.....	58
7.	Tests.....	59
7.1.	Tests Unitarios.....	59
7.2.	Resultados de los tests.....	61
7.2.1.	Reconocedores Edit Tests.....	61
7.2.2.	Reconocedores Play Tests.....	62
7.2.3.	Arrastrar imágenes Edit Tests.....	63
7.2.4.	Arrastrar imágenes Play Tests.....	64
8.	Conclusiones.....	65

8.1. Trabajo futuro	66
A. Manual de Despliegue	69
A.1. Despliegue del desarrollador	69
A.2. Despliegue del cliente	70
B. Manual de Uso	70
B.1. Inicio de la aplicación	70
B.2. Nómbralo	71
C. Contenido del TFG	75
Bibliografía	77

Índice de figuras

Figura 1 Pantalla de inicio de Pradia	1
Figura 2 Flujo de la metodología ágil	4
Figura 3 Logo de Unity	5
Figura 4 Interfaz de Unity	6
Figura 5 Logo de C#	7
Figura 6 Entorno 3d de Langosta	8
Figura 7 Taller presente en una de las tareas	8
Figura 8 Selección correcta en Articulation Teacher	10
Figura 9 Discriminación de fonema en TipTopTalk	11
Figura 10 Parejas en Articulation Station	12
Figura 11 Diagrama de Gantt con la planificación inicial	17
Figura 12 Diagrama de Gantt con la planificación final	22
Figura 13 Actividad "Nómbrale"	25
Figura 14 JSON de palabras	27
Figura 15 Actividad "Selección Correcta"	30
Figura 16 Actividad "Parejas"	30
Figura 17 JSON con palabras para "Discriminación Fonema"	32
Figura 18 Actividad "Parejas" con las imágenes volteadas	33
Figura 19 Algoritmo Fisher-Yates para aleatorizar las imágenes	34
Figura 20 Actividad "Sílabas Correctas"	35
Figura 21 Actividad "Pares Mínimos"	36
Figura 22 JSON palabras para "Pares Mínimos"	37
Figura 23 Script de grabación de audios	38
Figura 24 Modelo de dominio.....	42
Figura 25 Diagrama de actividad de "Nómbrale"	44
Figura 26 Diagrama de actividad de "Selección Correcta"	45
Figura 27 Diagrama de actividad de "Discriminación Fonema"	46
Figura 28 Diagrama de actividad de "Parejas"	47
Figura 29 Diagrama de actividad de "Sílabas Correctas"	48
Figura 30 Diagrama de actividad de "Pares Mínimos"	49
Figura 31 Estructura del proyecto.....	50
Figura 32 Diagrama de arquitectura	54
Figura 33 Estructura de Unity por paquetes	55
Figura 34 Diseño detallado del paquete Scripts.....	55
Figura 35 Diseño detallado del paquete Scenes.....	56
Figura 36 Diseño detallado del paquete Resources	57

Figura 37 Diseño del diagrama de despliegue	57
Figura 38 Configuración de los Edit tests	59
Figura 39 Ejemplo de ejecución de una prueba en el modo Edit	60
Figura 40 Ejemplo de limpieza de una prueba en el modo Edit	60
Figura 41 Configuración de los Play Tests	60
Figura 42 Ejemplo de ejecución de una prueba en el modo Play	61
Figura 43 Ejemplo de limpieza de una prueba en el modo Play	61
Figura 44 Pantalla de Inicio.....	71
Figura 45 Ventana emergente en "Nómbrale"	72
Figura 46 Ventana emergente en "Selección Correcta"	72
Figura 47 Ventana emergente de grabación de palabras	73
Figura 48 Escena de resultados	74
Figura 49 Escena de reproducción de audios	75

Índice de tablas

Tabla 1 Riesgos.....	18
Tabla 2 Plan de acción frente a los riesgos	19
Tabla 3 Tabla de costes	21
Tabla 4 Tecnologías de reconocimiento de voz.....	28
Tabla 5 Test de inicialización de la grabación	62
Tabla 6 Test de detención de la grabación.....	62
Tabla 7 Test reconocedor de palabras clave.....	62
Tabla 8 Test de detección de palabras clave.....	62
Tabla 9 Test de parar el reconocedor	63
Tabla 10 Test comprobar posición al soltar objeto	63
Tabla 11 Test comprobar estado al arrastrar objeto	63
Tabla 12 Test comprobar capas	63
Tabla 13 Test para comprobar los contenedores.....	64
Tabla 14 Test arrastrar y soltar	64
Tabla 15 Test soltar fuera de los contenedores.....	64

Capítulo 1

1. Introducción

1.1. Contexto

Pradia [\[1\]](#) es una aventura gráfica en 2D desarrollada en Flash por Mario Corrales Astorgano, diseñada para mejorar las habilidades de habla y comprensión en personas con síndrome de Down. Este juego se destaca por su enfoque en usuarios con necesidades educativas especiales, proporcionando una herramienta interactiva que facilita el aprendizaje de una manera divertida y atractiva. Sin embargo, con el desuso de Flash, otros motores gráficos como Unity se volvieron más populares y surgió la necesidad de migrar Pradia a una plataforma más moderna y duradera.



Figura 1 Pantalla de inicio de Pradia

En 2022, Gonzalo Fernández González realizó un trabajo de fin de grado (TFG) [\[2\]](#) con el objetivo de migrar algunas de las funcionalidades de Pradia de Flash a Unity. Este proyecto inicial sentó las bases para adaptar el juego a un entorno más actual y con más posibilidades, aprovechando las capacidades avanzadas de Unity para mejorar la experiencia del usuario. Unity no solo proporciona

una plataforma más robusta para el desarrollo de aplicaciones interactivas, sino que también ofrece herramientas potentes para la integración de gráficos, animaciones y funcionalidades interactivas.

Con la intención de añadir diversidad a Pradia, César Gonzalez y Mario Corrales publicaron una oferta de TFG para añadir nuevas actividades centradas en el aprendizaje de los fonemas, desarrolladas completamente en Unity, que se puedan integrar en Pradia para diversificar y enriquecer su jugabilidad. Estas nuevas actividades no solo aumentan la variedad del contenido educativo, sino que también incorporan técnicas de reconocimiento de voz para evaluar el rendimiento de los usuarios en tiempo real.

1.2. Motivación

El desarrollo de aplicaciones educativas requiere una actualización constante para mantenerse al día con los avances tecnológicos. En el caso de Pradia, la migración a Unity es un paso necesario para asegurar su viabilidad y expansión en el futuro. Sin embargo, más allá de simplemente migrar la plataforma, existe una oportunidad significativa para enriquecer el contenido educativo de Pradia mediante la incorporación de nuevas actividades centradas en el aprendizaje de los fonemas.

La motivación para este proyecto surge de la necesidad de proporcionar herramientas educativas más efectivas para personas con Síndrome de Down. La educación tradicional a menudo no logra captar la atención de los usuarios o proporcionar la retroalimentación inmediata necesaria para un aprendizaje efectivo. Incorporar actividades que utilicen reconocimiento de voz permitirá a los usuarios interactuar de manera más natural con la aplicación, haciendo el aprendizaje más dinámico. Más en detalle las motivaciones que llevan a realizar un proyecto en el ámbito de la tecnología como parte de la enseñanza de personas con Síndrome de Down son las siguientes:

- **Inclusión educativa y TIC:** La tecnología ha demostrado ser una herramienta valiosa para facilitar la inclusión educativa y mejorar el aprendizaje de personas con necesidades especiales, incluyendo aquellos con Síndrome de Down. Las herramientas tecnológicas permiten adaptar el contenido educativo a las necesidades individuales de los estudiantes, promoviendo una educación más inclusiva [\[3\]](#).
- **Beneficios de las herramientas tecnológicas:** El uso de tecnologías de la información y la comunicación (TIC) en la educación especial no solo mejora la accesibilidad y la interacción de los estudiantes con el contenido educativo, sino que también fomenta su integración en la comunidad y mejora su calidad de vida [\[4\]](#).
- **Necesidad de innovación:** La constante evolución tecnológica exige la actualización y mejora de las aplicaciones educativas para aprovechar al máximo las nuevas tecnologías.

Por ejemplo, la migración de Pradia a Unity no solo asegurará su viabilidad a futuro, sino que permitirá la integración de nuevas funciones.

1.3. Objetivos

El principal objetivo de este TFG es desarrollar actividades adicionales que se integren con Pradia, enfocadas en el aprendizaje y la práctica de los fonemas. Estas actividades no solo diversificarán la jugabilidad de Pradia, sino que también ofrecerán nuevas maneras de involucrar a los usuarios en su proceso de aprendizaje.

Además de este objetivo principal, también se han planteado una serie de objetivos más específicos:

- Incorporar tecnologías de reconocimiento de voz en estas actividades para una retroalimentación inmediata y precisa sobre la pronunciación de los usuarios.
- Utilización de ficheros de configuración para describir el contenido de las distintas actividades.
- Desarrollo modular de las actividades para facilitar su incorporación al videojuego Pradia.
- Localización de imágenes y archivos de audio con una licencia de uso libre para su incorporación al videojuego.

1.4. Metodología

Para el desarrollo de este proyecto, se ha optado por utilizar una metodología de desarrollo ágil, específicamente el enfoque iterativo e incremental [\[5\]](#). Esta metodología se ha elegido por varias razones:

- Al trabajar en iteraciones cortas, se puede ver el progreso realizado y tomar decisiones en consecuencia.
- Esta metodología permite refinar los requisitos conforme se avanza en el proyecto y se obtienen resultados.
- Debido a que las necesidades del cliente (en este caso los tutores) pueden cambiar durante el desarrollo del proyecto, esta metodología permite ajustar el desarrollo sin alterar el progreso global. Por ejemplo, si durante el desarrollo se identifica una nueva técnica de

reconocimiento de voz que podría mejorar las actividades de fonemas, esta puede ser incorporada en una iteración futura sin grandes dificultades.

- Al final de cada iteración, se obtiene un feedback sobre el trabajo realizado, que es crucial para saber si se está trabajando en consonancia con el cliente y realizar los ajustes necesarios.
- Con esta metodología los riesgos se identifican y se mitigan en cada iteración evitando acumulaciones de errores.

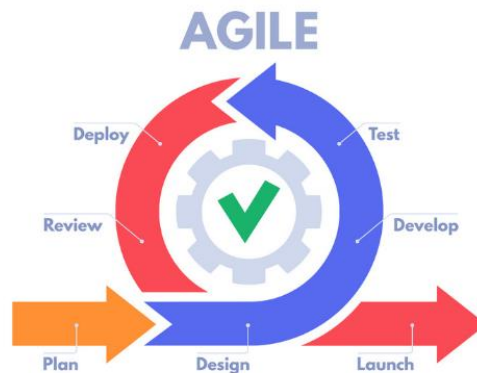


Figura 2 Flujo de la metodología ágil

1.5. Estructura de la memoria

La memoria se ha dividido en los siguientes capítulos:

- En el primer capítulo se presentan las ideas generales del proyecto, como son el contexto, la motivación, los objetivos y la metodología a seguir.
- El segundo capítulo sirve para hacer una radiografía de como se ha llegado al desarrollo de esta aplicación, junto con antecedentes de aplicaciones similares y una explicación de las tecnologías que se han usado.
- En el tercer capítulo se analiza la planificación del proyecto, incluyendo los riesgos que pueden aparecer y los costes estimados.
- En el cuarto capítulo se describen las iteraciones reales realizadas durante el desarrollo del proyecto.
- En el quinto y sexto capítulo se desarrolla el análisis y el diseño respectivamente. En el análisis se incluyen los requisitos, modelo de dominio y diagramas de actividades. En la parte del diseño se tratan los distintos patrones utilizados en la aplicación y se añade el diagrama de despliegue y el diseño de la interfaz.
- El sexto capítulo sirve para describir las pruebas que se han realizado.
- En la parte final de la memoria se incluyen el apéndice con documentaciones de despliegue y uso, y la bibliografía con las referencias que se han usado.

Capítulo 2

2. Estado de la cuestión

2.1. Uso de Unity en el desarrollo de actividades educativas

2.1.1. Unity

Unity es un motor de desarrollo de videojuegos multiplataforma creado por Unity Technologies, lanzado por primera vez en 2005. Es utilizado en el desarrollo de juegos 2D y 3D para diversas plataformas [\[6\]](#).



Figura 3 Logo de Unity

La interfaz de Unity es bastante accesible. Está compuesta de paneles que se pueden posicionar de manera personalizada [\[7\]](#):

- **Scene View:** Permite visualizar y editar los objetos en la escena.
- **Game View:** Muestra cómo se verá el juego en tiempo de ejecución.
- **Hierarchy:** Lista todos los objetos de la escena actual.
- **Project:** muestra todos los assets disponibles en el proyecto.
- **Inspector:** Permite editar las propiedades de los objetos seleccionados

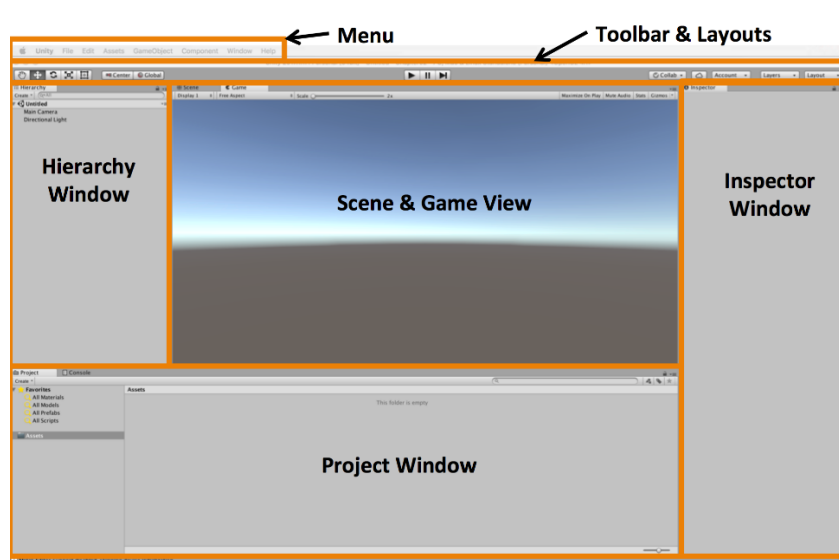


Figura 4 Interfaz de Unity

Unity se basa en un sistema de componentes para construir y gestionar los objetos en el juego [7]. Algunos de los componentes principales que se han usado para el proyecto han sido:

- **Transform:** Define la posición, rotación y escala de un objeto.
- **Renderer:** Controla cómo se representa visualmente un objeto.
- **Collider:** Define los límites físicos de un objeto para las interacciones de colisión.
- **Rigidbody:** Permite que un objeto sea afectado por la física.
- **Audio Source:** Permite reproducir sonidos y música.
- **Script:** Código asociado a un objeto para definir su comportamiento.

Unity tiene una amplia gama de funcionalidades para facilitar el desarrollo de aplicaciones interactivas [7]:

- **Gráficos y Renderizado:** Unity soporta gráficos 2D y 3D, incluyendo iluminación avanzada y efectos visuales.
- **Física:** Motor físico para simular colisiones, gravedad y otros comportamientos físicos [8].
- **Animaciones:** Unity tiene herramientas para crear y controlar animaciones complejas [9].
- **UI:** Sistema robusto para crear interfaces de usuario interactivas [10].
- **Realidad Virtual y Aumentada:** Herramientas para desarrollar experiencias de VR y AR.
- **Servicios en la nube:** Soporte para análisis, monetización y servicios en la nube para gestionar y mejorar los juegos.

Unity utiliza C# como su lenguaje principal de scripting. Este lenguaje tiene varias ventajas como son su facilidad de aprendizaje y que es lenguaje de programación es orientado a objetos, lo que facilita la organización del código y la implementación de conceptos como la encapsulación la

herencia y el polimorfismo. Además, su sistema de gestión automática de memoria simplifica el desarrollo [\[11\]](#).



Figura 5 Logo de C#

También cuenta con una gran comunidad de desarrolladores, lo que facilita el acceso a recursos, tutoriales y soporte técnico. La documentación y los recursos disponibles en la web de Unity y en otras plataformas educativas proporcionan un amplio apoyo para los desarrolladores.

2.1.2. Unity en la educación

Unity se ha consolidado como una plataforma versátil y poderosa para el desarrollo de aplicaciones educativas debido a su capacidad de crear experiencias interactivas y envolventes. A continuación, se presentan varias razones respaldadas por ejemplos específicos de aplicaciones desarrolladas con Unity [\[12\]](#).

- **Creación de experiencias inmersivas:** Unity permite desarrollar aplicaciones de realidad virtual (VR) y aumentada (AR) que pueden transformar el aula en un espacio de aprendizaje inmersivo. Por ejemplo, Zoe es una plataforma de creación VR/AR que permite a los estudiantes construir experiencias desde cero.
- **Simulaciones educativas avanzadas:** Las simulaciones en Unity pueden mejorar la retención y los resultados de aprendizaje. Langosta es un entorno de laboratorio simulado que capacita a los estudiantes en disciplinas STEM a través de simulaciones virtuales. Los estudiantes pueden realizar experimentos de manera segura y económica.



Figura 6 Entorno 3d de Langosta

- **Educación personalizada y adaptativa:** Unity permite el desarrollo de aplicaciones que se adapten a las necesidades individuales de los estudiantes. Kognito es una empresa de simulación que utiliza juegos de roles con humanos generados por inteligencia artificial para promover cambios positivos en comportamientos relacionados con la salud.
- **Educaverse [13]:** El proyecto Educaverse, desarrollado como una plataforma educativa para la educación diabética, utiliza realidad virtual para crear una experiencia inmersiva y educativa para debutantes diabéticos. Este ejemplo resalta cómo Unity puede ser utilizado para desarrollar aplicaciones educativas innovadoras que combinan tecnología avanzada con objetivos pedagógicos específicos, ofreciendo experiencias de aprendizaje efectivas.
- El proyecto “Diseño y desarrollo de juegos educativos 3D en Unity basados en la Industria 4.0 y en herramientas Lean” [14] muestra cómo Unity puede ser utilizado para crear aplicaciones educativas que simulan entornos industriales. Este TFG desarrolla videojuegos que permiten a los usuarios explorar un modelo de fábrica de Industria 4.0.

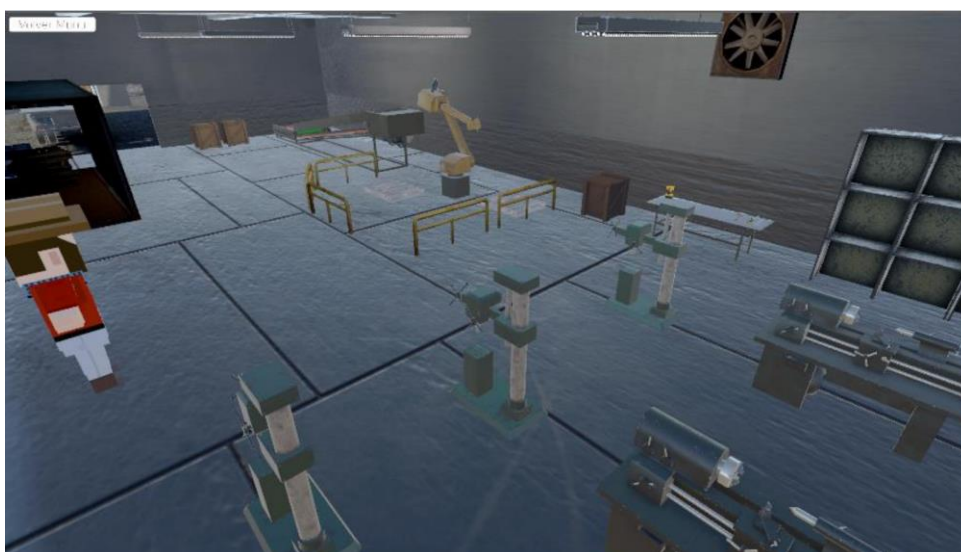


Figura 7 Taller presente en una de las tareas

Estos son solo algunos ejemplos en los que se usa Unity enfocado en ayudar a los usuarios en el aprendizaje de diversas materias. La versatilidad de Unity lo convierte en una herramienta valiosa para desarrollar experiencias educativas similares o más simplificadas, como es el caso de este TFG. La amplia documentación y la comunidad activa proporcionan un sólido respaldo para los desarrolladores que buscan crear aplicaciones educativas innovadoras y personalizadas.

2.2. Elección de actividades

Las actividades que se han desarrollado para Pradia están diseñadas para mejorar diversas habilidades lingüísticas y cognitivas en personas con síndrome de Down, utilizando técnicas de reconocimiento de voz. Estas actividades contribuyen significativamente al desarrollo educativo de los usuarios de varias maneras.

Para empezar, un ámbito en el que estas actividades ayudan a estas personas es con el almacenamiento de la información fonológica de las palabras en la memoria a largo plazo [15]. Una correcta producción del habla hace que estas palabras mejoren la memoria fonológica. Este tipo de memoria hace que se recuerden palabras que se acaban de escuchar y es necesaria para aprender nuevas palabras.

Un requisito importante es el uso de muchas palabras para el desarrollo del vocabulario, haciendo que el sistema fonológico sea más sensible a las diferencias fonémicas entre las palabras.

A la hora de elegir las actividades a utilizar, se han tomado como inspiración diversas aplicaciones que también tratan los fonemas o el reconocimiento de voz aplicado a la pronunciación:

- **Nómbrale:** Para la primera actividad, los usuarios deben identificar y nombrar imágenes para practicar la correcta pronunciación de las palabras. Esta actividad está basada en la aplicación “Articulation Station” [16] y en “Articulation Teacher” [17], que también utilizan ejercicios de repetición para mejorar la pronunciación.
- **Selección Correcta:** Los usuarios deben seleccionar la imagen correcta basándose en un audio o texto, reforzando la asociación entre palabra y representación visual. “Articulation Teacher” [16] ofrece ejercicios de selección similares para mejorar la comprensión.



Figura 8 Selección correcta en Articulation Teacher

- **Discriminación de Fonema:** Los usuarios deben identificar si un fonema específico está presente en palabras mostradas, mejorando la discriminación auditiva. En la aplicación “TipTopTalk” [\[18\]](#) se ayuda a mejorar la discriminación fonémica a través de ejercicios específicos y “Articulation Teacher” [\[17\]](#) también utiliza ejercicios similares, aunque en este proyecto se va a implementar de manera distinta, ya que se pretende arrastrar las imágenes a una posición correcta.

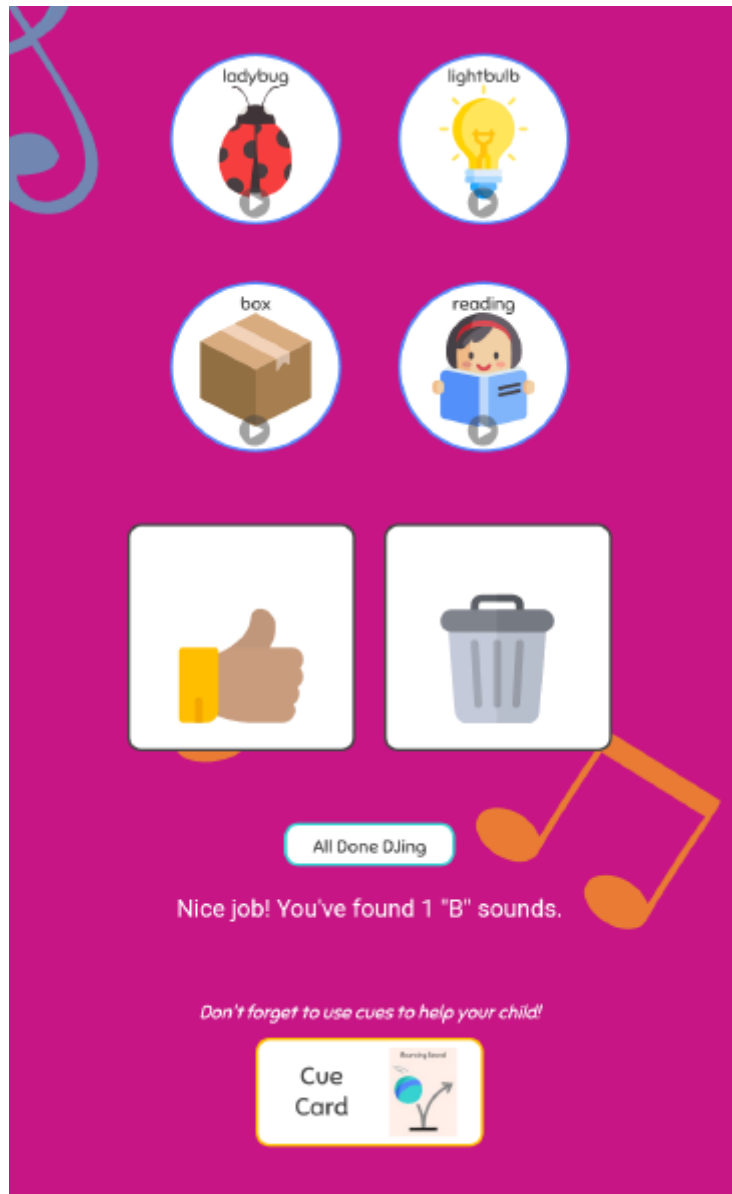


Figura 9 Discriminación de fonema en TipTopTalk

- Parejas:** En la actividad “Parejas” se está usando el principio de asociación palabra referente [\[19\]](#). Con este mecanismo las personas con Síndrome de Down aprenden vocabulario por medio de asociar imágenes con referentes visuales concretos. Durante la actividad, si se presenta una imagen de un “gato”, el usuario deberá recordar dónde está ubicada la otra imagen del “gato” y emparejarlas. Al acertar, el usuario pronuncia la palabra, reforzando así la asociación entre la palabra y la imagen del animal. Este ciclo de emparejamiento y pronunciación crea un fuerte vínculo entre la palabra y su referente visual. La aplicación “Articulation Station” [\[16\]](#) también incluye esta actividad.



Figura 10 Parejas en Articulation Station

- **Sílabas Correctas:** Los usuarios identifican sílabas que contienen un fonema específico, reforzando la combinación y separación de sonidos. Aunque las aplicaciones mencionadas anteriormente han podido servir de inspiración, ya que en parte es una actividad que deriva de la discriminación de fonemas, no está basada en ninguna en concreto.
- **Pares Mínimos:** Las actividades de pares mínimos hacen que los usuarios identifiquen y diferencien entre palabras que solo tienen un fonema distinto, lo que da como resultado efectos positivos demostrados en la mejora de los problemas de habla y en la precisión de las representaciones fonológicas. “TipTopTalk” [18] también trabaja con pares mínimos y se ha usado como referencia para esta actividad.

2.3. Impacto del reconocimiento de voz en el aprendizaje

El reconocimiento de voz se ha convertido en una herramienta emergente con el potencial de transformar la educación y el aprendizaje. Se han escrito estudios que exploran diferentes aspectos de esta tecnología y su impacto en el ámbito educativo.

Antes de describir los resultados, creo necesario explicar algunos conceptos que se van a usar:

- **Conciencia silábica:** Es la habilidad para identificar y manipular las sílabas dentro de las palabras. Incluye tareas como dividir palabras en sílabas.

- **Conciencia fonémica:** Es la habilidad para identificar y manipular los fonemas, que son las unidades de sonido más pequeñas que pueden cambiar el significado de una palabra.
- **Fonemas iniciales y finales:** Aquellos sonidos que se encuentran al principio y al final de las palabras.
- **Palabras regulares:** Son palabras que siguen las reglas fonéticas del idioma, donde cada letra o grupo de letras corresponde a un sonido específico.
- **Palabras irregulares:** Son palabras que no siguen las reglas fonéticas estándar del idioma, por lo que no se pueden leer correctamente solo a partir de sus letras.
- **Pseudopalabras:** Son combinaciones de letras que parecen palabras reales, pero no tienen significado.
- **Ruta fonológica:** Es el proceso de traducción de las sílabas a fonemas, útil para leer palabras nuevas.
- **Ruta léxica:** Es el proceso de leer las palabras como unidades completas. Es útil para leer palabras conocidas.
- **Habilidades de segmentación y análisis fonológico:** Son las habilidades para dividir las palabras en sílabas y fonemas y analizarlas.

2.3.1. Mejora de la conciencia fonológica y competencia lectora

En los “Estudios de Desarrollo del Lenguaje y Educación” [\[19\]](#) de la Universidad de Oviedo se evaluaron los efectos del sistema de reconocimiento de voz (SVR) en la conciencia fonológica y la competencia lectora en niños de 5 años. Se observaron los siguientes resultados:

- **Conciencia Silábica y Fonémica:** Los niños mostraron mejoras significativas en tareas de adición y omisión de sílabas y fonemas después de ser entrenados con SVR. Las mejoras en la conciencia silábica incluyeron la capacidad para añadir y omitir sílabas iniciales y finales, mientras que, en la conciencia fonémica, se observaron mejoras en la adición y omisión de fonemas, así como en la identificación de fonemas finales.
- **Competencia lectora:** Se observaron mejoras notables en la lectura de palabras regulares, irregulares y pseudopalabras. Estas mejoras sugieren que el SRV ayuda a integrar y mejora

las rutas fonológica y léxica, permitiendo un reconocimiento y pronunciación más eficiente de las palabras.

- **Estrategias de aprendizaje:** El proceso de vocalización requerido por el SRV fomenta la autocorrección y una mayor conciencia sobre los componentes fonológicos de las palabras, que desemboca en un aprendizaje más autónomo y eficaz. Los niños primero deben vocalizar claramente lo que quieren escribir, lo que facilita así el desarrollo de habilidades de segmentación y análisis fonológico.

2.3.2. Impacto de la IA y el reconocimiento de Voz en el aprendizaje

En un artículo de AWS [\[20\]](#), se describe cómo el reconocimiento de voz impulsado por IA está creando experiencias de aprendizaje interactivas tanto para niños como para adultos.

- **Aplicaciones en la educación infantil:** Empresas como ROYBI utilizan IA y robótica para crear experiencias de aprendizaje personalizadas para niños, adaptando el contenido educativo según el progreso individual de cada niño. ROYBY evalúa el progreso de aprendizaje y crea informes para los padres, lo que permite un seguimiento detallado del desarrollo educativo [\[20\]](#).
- **Accesibilidad y calidad en la educación de idiomas:** Singsound utiliza IA para proporcionar evaluaciones de habla y escritura en inglés [\[20\]](#).

2.3.3. Implicaciones y Futuro de la Integración Tecnológica

El estudio y las aplicaciones prácticas de SRV e IA en la educación destacan el potencial que tienen estas tecnologías en el ámbito educativo. La integración de SRV no solo mejora habilidades específicas como la lectura y la conciencia fonológica, sino que también promueve estrategias de aprendizaje autónomas y eficaces.

- **Potencial educativo del SRV:** El SRV puede ser una herramienta educativa poderosa para apoyar el desarrollo de habilidades lectoras y fonológicas en las primeras etapas de la educación. La tecnología hace que el aprendizaje sea más dinámico e interactivo, motivando a los estudiantes.
- **Inclusión:** La tecnología SRV es especialmente beneficiosa para estudiantes con dificultades de aprendizaje, ofreciendo un apoyo adicional que se adapta a sus necesidades individuales.

En conclusión, la tecnología de reconocimiento de voz tiene un gran impacto positivo en el aprendizaje, desde la mejora de habilidades específicas hasta la promoción de estrategias de aprendizaje autónomas. Su implementación puede transformar las prácticas educativas y apoyar a los estudiantes en su desarrollo académico, preparándolos mejor para un futuro tecnológico.

Capítulo 3

3. Planificación

3.1. Planificación Inicial

Este TFG se realizará entre el primer y el segundo cuatrimestre del curso 2023-2024. La primera iteración se realizará el 5 de octubre, con la primera reunión con los tutores y la última iteración terminará en junio. Se usarán como días de trabajo de lunes a viernes, aunque los fines de semana estarán disponibles por si surge algún imprevisto.

La división del trabajo del TFG se hará por iteraciones:

- 1ª Iteración (5 semanas, 40 horas (1,5 horas al día)): Esta primera iteración se realizará durante el primer cuatrimestre. Servirá principalmente para entender cómo funciona de manera general el proyecto y para aprender el funcionamiento las herramientas que se van a usar (Unity, distintos speech to text).
- 2ª Iteración (5 semanas, 40 horas (1,5 horas al día)): En la segunda iteración, una vez ya familiarizado con el funcionamiento de las herramientas, se empezará con el desarrollo de la primera actividad (Nómbalo) y la implementación en Unity de las dinámicas de producción speech to text, de manera que una vez implementadas estas funciones en la primera actividad, se pueda replicar su funcionamiento de una manera más sencilla para las siguientes actividades que lo requieran.
- 3ª Iteración (6 semanas, 90 horas (3 horas al día)): A partir de enero sube la carga de trabajo. Una vez adquiridos los conocimientos básicos y aprendido el funcionamiento de creación de una actividad se procederá a realizar la actividad 2 y 3, implementando speech to text en las que sean necesarias.
- 4ª Iteración (5 semanas, 75 horas (3 horas al día)): A partir de la mitad de febrero se estará realizando el TFG a la vez con las prácticas, pero se mantendrá la misma distribución horaria. Esta iteración se usará para terminar el resto de las actividades.
- 5ª Iteración (6 semanas, 90 horas (3 horas al día)): Esta iteración servirá para comprobar el correcto funcionamiento de las actividades e implementar su funcionamiento de manera

secuencial. También se usará esta iteración para añadir un sistema de puntuación al final y para intentar la implementación simultanea del reconocedor de voz con la grabación de audios.

- 6ª Iteración (3 semanas, 45 horas (3 horas al día)): En la última iteración se terminarán de resolver los problemas que hayan quedado en la aplicación, teniendo reuniones más continuas con los tutores para solventar rápidamente los errores que queden. También se usará este tiempo para la realización la memoria.

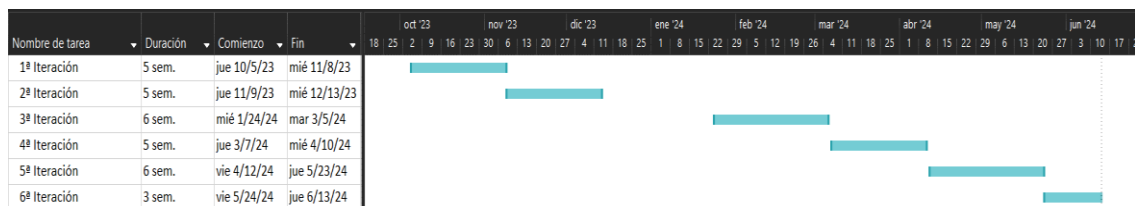


Figura 11 Diagrama de Gantt con la planificación inicial

3.2. Riesgos

En esta parte se hará una descripción de los riesgos que pueden aparecer a lo largo del desarrollo del proyecto y como podrían ser los retrasos que causen. En la tabla aparecen los riesgos, seguido de la probabilidad que tienen de aparecer, una descripción de este y el retraso que pueden causar medido en días útiles (3 horas).

N.º	Riesgos	Probabilidad	Descripción	Retraso estimado
1	Enfermedad	0.05	El estudiante contrae una enfermedad, lo que afecta al desarrollo del TFG.	7
2	Cambios en los requisitos del proyecto	0.5	Los requisitos pueden cambiar a lo largo de la realización del TFG, pudiendo afectar a la planificación y al alcance del proyecto.	12
3	Falta de experiencia técnica	0.2	Dada la poca experiencia con las tecnologías a usar, se podría requerir más tiempo del planificado.	15

4	Cambios en el entorno externo	0.1	Se producen retrasos por cambios en las tecnologías que se van a usar.	15
5	Problemas con el ordenador	0.1	El ordenador que se usa en el proyecto sufre alguna clase de avería.	3
6	Falta de motivación	0.2	Trabajar en un proyecto largo puede llevar a la falta de motivación, causando retrasos.	10
7	Falta de planificación de riesgos	0.2	No planificar de manera adecuada los riesgos puede provocar que aparezcan otros riesgos inesperados que provoquen retrasos.	7

Tabla 1 Riesgos

Esta segunda tabla sirve para explicar para cada riesgo que acciones se podrían tomar previamente a que ocurra para reducir las probabilidades de aparición y por otro lado las acciones que se podrían tomar después de que sucediera el riesgo, para hacer que el gasto de tiempo repercuta lo menos posible en el desarrollo del proyecto.

N.º	Riesgos	Exposición	Plan de Reducción	Acciones de Mitigación
1	Enfermedad	0.5	No exponerse a personas con virus	Usar las horas del fin de semana
2	Cambios en los requisitos del proyecto	7	Revisar regularmente los requisitos	Actualizar los requisitos, intentando cambiar lo menos posible para aumentar lo mínimo las horas
3	Falta de experiencia técnica	0.2	Formarse de manera continua en las tecnologías que se usen	Dedicar horas extra al estudio de las tecnologías
4	Cambios en el entorno externo	1	Estar atento a cómo van evolucionando las tecnologías	Adaptarse a los cambios necesarios en las tecnologías, principalmente dedicando más horas

5	Problemas con el ordenador	1	Limpiar el ordenador con regularidad	Se buscará otro ordenador donde realizar el proyecto
6	Falta de motivación	4	Mantener un horario regular con descansos	Aumentar los descansos entre tareas para ir adquiriendo un horario
7	Falta de planificación de riesgos	2	Actualizar regularmente los riesgos	Usar las horas del fin de semana para paliar los retrasos

Tabla 2 Plan de acción frente a los riesgos

3.3. Entorno tecnológico

En la realización de este TFG se ha empleado un conjunto de herramientas tecnológicas que han facilitado el desarrollo y gestión del proyecto. Estas herramientas abarcan desde el diseño de diagramas y el desarrollo de software, hasta la gestión de versiones y la comunicación. A continuación, se detallan las principales herramientas utilizadas:

- **Astah** [\[21\]](#): Esta herramienta se ha utilizado para la creación de diagramas UML, permitiendo una planificación de la estructura del proyecto.
- **C#** [\[22\]](#): El lenguaje de programación principal empleado en el desarrollo del proyecto. C# ha sido elegido por su integración con los entornos de desarrollo de Unity y Visual Studio, que son los que se han utilizado en este proyecto.
- **Word**: Microsoft Word ha sido la herramienta principal para la redacción y edición de la memoria de TFG.
- **Gitlab**: Se ha empleado Gitlab para la gestión de versiones del código fuente.
- **Sourcetree** [\[23\]](#): Esta interfaz gráfica para Git ha sido utilizada para gestionar de manera más visual y simplificada el repositorio alojado en Gitlab. Sourcetree facilita la realización de commits y la resolución de conflictos de manera eficiente.
- **Teams**: Microsoft Teams ha sido fundamental para la comunicación y colaboración en tiempo real con los tutores del proyecto. Esta herramienta ha permitido realizar reuniones, compartir documentos y coordinar tareas del proyecto.

- **Unity 2021-2-9f1:** El motor de desarrollo de juegos Unity ha sido utilizado para la creación y el desarrollo de la aplicación principal del proyecto.
- **Visual Studio code** [\[24\]](#): Este editor de código ha sido la herramienta principal para la escritura y edición del código fuente. VScode ofrece extensiones y funcionalidades avanzadas que han mejorado significativamente la productividad.

3.4. Costes

3.4.1. Costes de personal

Aunque en este caso no hay ningún coste asociado al personal del proyecto, se va a hacer una simulación de los costes reales que tendría un proyecto de estas características con un desarrollador, dado que en un escenario realista serían unos costes muy importantes para tener en cuenta.

En España el salario de un programador novel suele oscilar entre 12 y 15 euros por hora [\[25\]](#). Para esta estimación, tomaremos de ese intervalo un valor bajo de 12.5 por hora, debido a la inexperiencia laboral del desarrollador. Ese dinero multiplicado por 300 horas daría un total de 3750 euros dedicados al salario bruto del desarrollador.

3.4.2. Costes Hardware

Para el desarrollo del proyecto, se utiliza un portátil personal y accesorios necesarios. Se estima que el coste han sido unos 900 euros en el portátil y unos 200 euros en periféricos (ratón, teclado, monitor externo, auriculares y micrófono).

3.4.3. Costes Software

Las herramientas Software que se han utilizado no han supuesto ningún coste, considerando que los programas usados han tenido una licencia gratuita, como es el caso de VSCode, Sourcetree, Gitlab y Unity, y para otros programas como Office 365 y Windows 11 se ha usado la licencia de la UVa.

Categoría	Descripción	Coste
Costes de personal	Salario estimado para un desarrollador novel, basado en 300 horas de trabajo.	3752 euros
Costes Hardware	Inversión en un portátil personal y periféricos necesarios para el desarrollo.	1100 euros
Costes Software	Herramientas software con licencias gratuitas y otras proporcionadas por la universidad	0 euros

Tabla 3 Tabla de costes

3.5. Desviaciones de la planificación

Iteración 2

Planificación inicial:

- Objetivos: Desarrollo de la primera actividad “Nómbrale” y la implementación inicial de funciones de speech-to-text.
- Estimación: 40 horas.

Planificación final:

- Horas realizadas: 50 horas.
- Se realizaron búsquedas de imágenes y audios que correspondiesen al tono de la aplicación, utilizando recursos de ARASAAC. La implementación de la carpeta “Resources” y el archivo JSON para la gestión de recursos tomó más tiempo de lo previsto debido a que la documentación acerca de este tema en Internet es poco precisa y no se adapta a casos específicos.

Iteración 4

Planificación Inicial:

- Objetivos: Desarrollo de la cuarta actividad “Parejas” y mejoras generales.
- Estimación: 75 horas

Planificación Final:

- Horas realizadas: 90 horas
- Detalles: Esta iteración fue afectada por la adición del nuevo requisito, el sistema de intentos, en las actividades “Discriminación de Fonema” y “Pares Mínimos”.

Iteración 6

Planificación Inicial:

- Objetivos: Redacción de la memoria del proyecto y corrección de errores finales
- Estimación: 45 horas

Planificación Final:

- Horas realizadas: 65 horas
- Detalles: La redacción de la memoria tomó más tiempo del previsto debido a cambios finales en el diseño de la aplicación y a otros capítulos que han costado más de lo esperado en escribir.

Las desviaciones en la planificación inicial del proyecto reflejan la necesidad de adaptarse a imprevistos y cambios en los requisitos iniciales. A pesar de las desviaciones en el tiempo y las horas de trabajo, la previsión de los riesgos y la flexibilidad en la planificación han permitido completar el proyecto con éxito. Los conocimientos extraídos de este proceso se podrán usar en futuros desarrollos y proyectos.

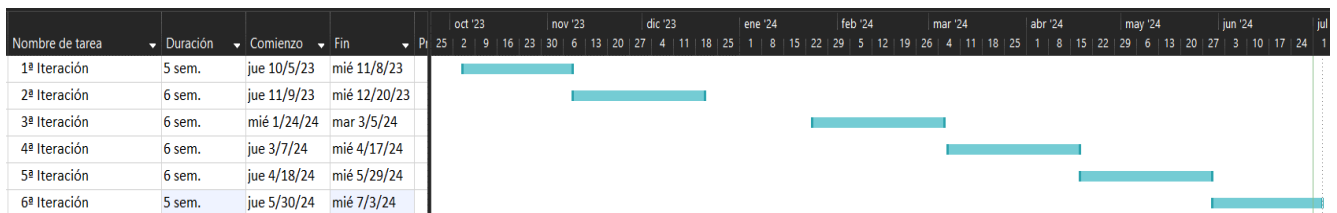


Figura 12 Diagrama de Gantt con la planificación final

Capítulo 4

4. Descripción de las iteraciones

El desarrollo del proyecto se ha llevado a cabo en varias iteraciones, cada una de ellas con objetivos específicos como se ha explicado en el apartado de planificación. Estas iteraciones han permitido un enfoque incremental e integrando funcionalidades y corrigiendo errores a lo largo del proceso. A continuación, se detalla el trabajo realizado en cada iteración.

4.1. Iteración 1

Esta primera iteración, que abarca desde el 5 de octubre de 2024, marca el inicio del proyecto con una serie de actividades fundamentales para sentar las bases del desarrollo del TFG. La iteración comienza con una reunión inicial con los tutores del proyecto, en la cual se presenta de manera general el TFG y se revisa el trabajo previo realizado por Gonzalo Fernández. Gonzalo desarrolló una migración parcial del juego educativo “Pradia” a Unity, y su trabajo sirve como referencia y punto de partida para este proyecto.

El objetivo principal de esta iteración ha sido que sirviera para familiarizarme con las herramientas y tecnologías que se van a utilizar a lo largo del proyecto. Esto incluye el aprendizaje de Unity y la revisión y comprensión del TFG previo de Gonzalo.

4.1.1. Tareas realizadas

Lo primero ha sido configurar el entorno de desarrollo en el equipo personal, asegurado que todas las herramientas necesarias (Unity, Visual Studio, etc.) estén correctamente instaladas y configuradas. Para la versión de Unity se usará la misma versión que en el TFG de Pradia (Unity 2021-2-9f1).

Durante esta etapa, se ha dedicado tiempo a explorar y entender la interfaz de Unity. Esto incluye la familiarización con el editor de Unity, la configuración de escenas, el uso de GameObjects, componentes y prefabs. También se desarrollan pequeños proyectos de prueba para practicar y aplicar los conceptos aprendidos.

Por otro lado, se ha revisado detalladamente el TFG de Gonzalo. Se han examinado las funcionalidades ya implementadas en este proyecto, poniendo especial atención a como se ha realizado la implementación de la dinámica de reconocimiento de voz (speech-to-text), aunque revisando otros aspectos, como la gestión del inventario o como almacenar los datos de la partida.

4.2. Iteración 2

En esta iteración, se ha avanzado significativamente en el desarrollo del proyecto mediante la implementación de la primera actividad llamada "Nómbralo". Este módulo se centra en que los usuarios nombren palabras utilizando su voz, lo cual requiere una integración efectiva de tecnologías de reconocimiento de voz (speech-to-text) dentro del entorno de Unity.

Para empezar, se han identificado los elementos visuales y funcionales necesarios para esta actividad. Esto incluye 2 botones, uno para el reconocedor de voz y otro para pasar a la siguiente palabra; el campo de texto, la imagen del objeto a nombrar y cualquier otro elemento interactivo.



Figura 13 Actividad "Nómbalo"

Se han buscado imágenes y audios que correspondiesen al tono de la aplicación. Para ello se ha optado por usar las proporcionadas por el catálogo ARASAAC [\[26\]](#).

ARASAAC ofrece una gran cantidad de audios e imágenes que son fáciles de entender y tiene un formato de dibujo que se adapta bien a la estética de la aplicación. La elección de ARASAAC se basó en su accesibilidad y a la alta calidad de sus recursos.

Con estos elementos se la interfaz de usuario. Esta interfaz se revisa y ajusta hasta que cumple con los requisitos de usabilidad y accesibilidad. Para crearla se han usado elementos de Unity como son GameObjects, para los botones, imágenes y textos, un Event System para manejar la interacción del usuario con los elementos de la interfaz (clics de botones, movimientos del ratón...) y se ha añadido un Canvas a la escena. El Canvas es el contenedor principal para todos los elementos de la interfaz de usuario en Unity. Todos los GameObjects se colocan dentro del Canvas, permitiendo controlar su disposición en la escena.

4.2.1. Implementación

Una vez que se ha completado la realización de la interfaz se procede a desarrollar los scripts que manejarán la lógica de la actividad. Para empezar de una manera simple, se han dividido en un script que gestiona la lógica de los botones y otro que controla los cambios en la escena.

Para gestionar el almacenamiento y acceso a las palabras, imágenes y audios utilizados en las actividades, se necesitaba una solución que permitiera cargar estos recursos de manera eficiente durante la ejecución del juego. Unity proporciona una solución práctica para este tipo de necesidades mediante el uso de la carpeta “Resources” [\[27\]](#). Esta carpeta es especial en Unity, ya que permite al desarrollador almacenar contenido que puede ser cargado dinámicamente en tiempo de ejecución, proporcionando una gran flexibilidad en la gestión de recursos.

La organización de la carpeta “Resources” se ha dividido en subcarpetas específicas para cada fonema, en las que se han almacenado las imágenes y los audios proporcionados por ARASAAC. Esta organización facilita la localización y carga de los recursos asociados a cada fonema de manera ordenada en tiempo de ejecución. En ellas se han almacenado las imágenes y audios relacionados con el fonema correspondiente.

Por otro lado, para mejorar aún más la eficiencia en el acceso a los recursos, se ha creado un archivo JSON dentro de la carpeta “Resources”. Este archivo actúa como un índice o mapa de los recursos disponibles, conteniendo las rutas de acceso a las imágenes y audios de cada palabra. Durante la ejecución del juego, los scripts de Unity pueden leer este archivo JSON para obtener las rutas de los recursos necesarios y cargarlos dinámicamente. Esto se realiza utilizando la función “Resources.Load” de Unity, que permite cargar recursos en tiempo de ejecución basándose en las rutas especificadas en el JSON.

Esta organización permite añadir nuevos fonemas y sus recursos de manera sencilla, sin necesidad de realizar cambios significativos en el código y hace que la ejecución sea más eficiente al tener un archivo JSON que actúa como índice.


```

{
  "opciones": [
    {
      "fonema": "p",
      "palabras": [
        {
          "nombre": "lupa",
          "textura": "Fonemas/p/Imagenes/lupa",
          "audio": "Fonemas/p/Audios/lupa",
          "silabas": [
            {
              "silaba": "lu",
              "esFonema": false
            },
            {
              "silaba": "pa",
              "esFonema": true
            }
          ]
        },
        {
          "nombre": "pino",
          "textura": "Fonemas/p/Imagenes/pino",
          "audio": "Fonemas/p/Audios/pino",
          "silabas": [
            {
              "silaba": "pi",
              "esFonema": true
            },
            {
              "silaba": "no",
              "esFonema": false
            }
          ]
        }
      ]
    }
  ]
}

```

Figura 14 JSON de palabras

4.2.2. Implementación reconocimiento de voz

El reconocimiento de voz a texto se puede realizar de dos maneras, por un lado, se puede usar una solución basada en la nube, que se ejecutaría en unos servidores externos y enviarían la solución al juego y por otro lado se podría implementar una solución dentro de la aplicación que se ejecutaría en local. La mayor diferencia entre las 2 es que la primera no requeriría de una computación local y la segunda sí, pero al ser un servicio externo suele venir con algún tipo de coste asociado. Para elegir la mejor herramienta con la que realizar la dinámica de producción se han analizado una serie de herramientas para ver cuál es la que tiene mejores características:

Tecnología	Ventajas	Desventajas	Nube/Local
Google Cloud [28]	Integración por medio de una API	Prueba gratis y luego de pago. 60 minutos al mes	Nube
Mozilla DeepSpeech [29]	Código abierto	Integración con Unity compleja	Local
CMU Sphinx (PocketSphinx) [30]	Código abierto	Menos precisa	Local
Google Web Speech API (Chrome)	Código abierto	Solo funciona en algunos navegadores	
Vosk	Código abierto	Muy complejo. Requiere mucho almacenamiento	Local
Wit.ai (meta) [31]	Código abierto, con opción local	Poco precisa	Local

Tabla 4 Tecnologías de reconocimiento de voz

Además de estas tecnologías se han hecho pruebas locales con proyectos de código abierto de otras herramientas que podrían ser útiles:

- **Whisper** [\[32\]](#): Se han realizado pruebas con Whisper, que es una red neuronal de código abierto desarrollada por OpenAI, conocida por su uso intensivo de RAM y CPU. Para probar Whisper se ha usado un proyecto de código abierto que lo integra en Unity y en el que se puede probar su rendimiento con distintos modelos, que dependen del idioma que se quiera reconocer y de la precisión que se quiera tener en el reconocimiento, teniendo en cuenta que cuanto más precisión se quiera tener, más recursos pedirá el sistema.
- **Vosk** [\[33\]](#): Vosk proporciona una solución de reconocimiento de voz offline para Unity, permitiendo el procesamiento de audio directamente en el dispositivo del usuario sin necesidad de conexión a internet. El proyecto que he probado utiliza modelos pre entrenados de Vosk, que capturan audio mediante scripts y generan transcripciones en tiempo real.

Después de probar estas tecnologías de reconocimiento de voz, se ha optado por usar el reconocedor de voz que viene integrado en Unity por diversos motivos. Para empezar los reconocedores locales que tienen integración con Unity, a fecha de la realización de este TFG, todavía les falta algo de perfeccionamiento, debido a que con oraciones largas sí que parecen tener un mejor rendimiento, aunque en ocasiones cometen algún error puntual. Para el reconocimiento de palabras individuales suelen cometer muchos errores, haciendo muy difícil tomarlo como una

opción viable para una aplicación que se centra en reconocer solo palabras. Por contra, los reconocedores basados en la nube, aunque tienen un muy buen rendimiento incluso para palabras, tienen el inconveniente de que son herramientas de pago que solo ofrecen una prueba gratuita de sus servicios y otros como Vosk.ai están en fases muy tempranas de su desarrollo.

Por estos motivos se ha decidido utilizar el reconocedor de voz que viene integrado en Unity, ya que ofrece una precisión muy alta, con una integración bastante sencilla y sin ningún coste asociado.

Una vez decidido como se va a realizar esta dinámica, se ha procedido a implementarla en Unity. Este reconocimiento integrado ofrece al desarrollador 2 maneras de hacerlo. Por un lado, ofrece un reconocedor de palabras clave, con el que solo se detectará el audio cuando contenga la palabra que se está buscando. Por otro lado, ofrece un reconocedor de dictado, con el que se reconoce todo lo que está diciendo el usuario y se procesa en tiempo real. Para este proyecto se han implementado las 2 maneras en scripts separados, que se conectan al script controlador de la escena cuando se reconoce el audio y se realizan las acciones pertinentes. Antes de compilar la aplicación el desarrollador puede elegir en el inspector de la escena que reconocedor quiere utilizar, dependiendo de que acciones quiera realizar.

4.3. iteración 3

En esta tercera iteración, con los conocimientos básicos ya adquiridos y la experiencia ganada en la creación de la primera actividad durante las iteraciones anteriores, el enfoque se centra en el desarrollo de dos nuevas actividades: “Selección Correcta” y “Discriminación de Fonema”.

La actividad “Selección Correcta” presenta al usuario seis imágenes junto con un texto y un audio, donde el objetivo es hacer clic en la imagen que corresponde correctamente al audio y texto proporcionado.



Figura 15 Actividad "Selección Correcta"

Esta actividad es más sencilla de realizar debido a que no incorpora la dinámica de producción, solo la de comprensión y se centra en seleccionar la imagen correcta. Las imágenes que aparecen se eligen de manera aleatoria del JSON que contiene las palabras, con la dirección de sus imágenes y audios.

La tercera actividad, "Discriminación de Fonema", se presentan cuatro imágenes con dos cajas, una verde y otra roja. El objetivo del usuario es arrastrar cada imagen al cuadro verde si el fonema de la imagen coincide con el fonema dado, o al cuadrado rojo si no coincide.

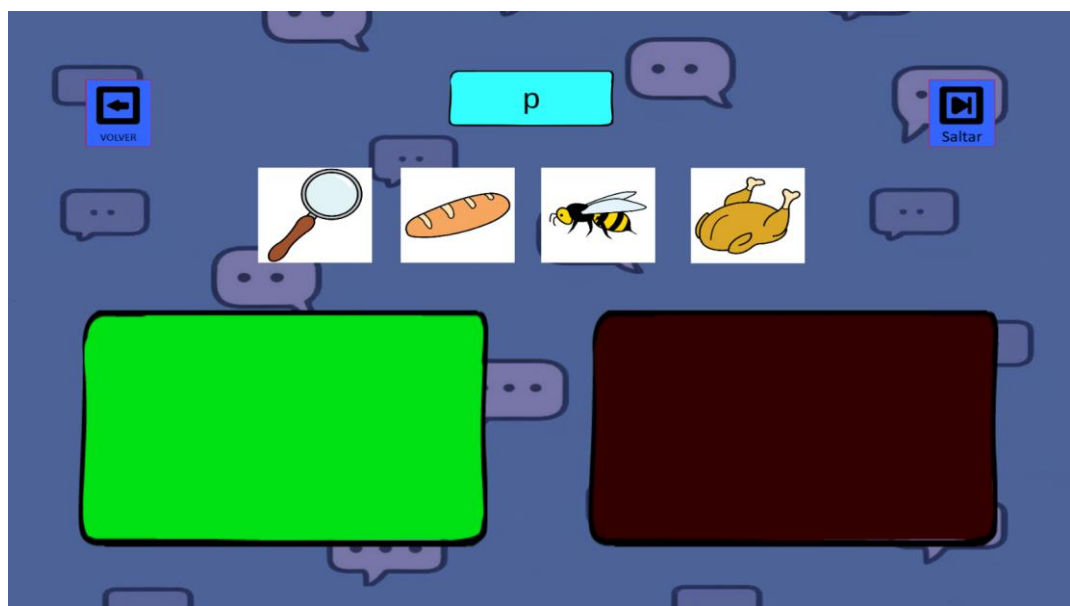


Figura 16 Actividad "Parejas"

Para implementar esta actividad, se ha trabajado en la programación de la funcionalidad de arrastrar y soltar las imágenes. Para que se pudiesen arrastrar las imágenes por la escena ha sido necesaria la utilización de dos componentes que se han añadido a los objetos de Unity para permitir su movimiento:

- **Box Collider 2D** [\[34\]](#): Este componente se utiliza para definir los límites de las imágenes y los huecos. Permite detectar colisiones y determinar cuándo una imagen ha sido colocada en un hueco específico. Un Box Collider 2D es un tipo de colisionador que utiliza una forma rectangular para determinar el área de colisión de un objeto.
- **Rigidbody 2D** [\[35\]](#): Este componente se emplea para que las imágenes puedan ser arrastradas por el usuario. Un Rigidbody 2D añade propiedades físicas a los GameObjects, como gravedad y movimiento controlado, lo que permite que los objetos respondan a las fuerzas físicas y a las interacciones del usuario.

Para la implementación del movimiento de las imágenes, además de esos dos componentes, se ha desarrollado un script que permite detectar cuando una imagen está siendo arrastrada. Una vez que el usuario suelta la imagen, el script utiliza el Box Collider 2D para detectar si la imagen ha sido colocada en uno de los huecos. Si la imagen colisiona con el hueco verde o rojo, el script determina si la colocación es correcta o incorrecta basándose en el fonema asociado a la imagen.

Si todas las imágenes se colocan correctamente, el juego proporciona feedback visual por medio de un objeto prefabricado que avisa al usuario de que ha acertado todas las posiciones de las imágenes.

A la hora de gestionar los movimientos de las imágenes hay que destacar la complejidad de la gestión de las capas para que la imagen que se está arrastrando quede siempre visible por encima del resto. Este desafío se ha abordado modificando el campo “sortingOrder” de los componentes “SpriteRenderer” de los objetos.

Detalles de la implementación:

- **SortingOrder de SpriteRenderer** [\[36\]](#): El campo “sortingOrder” en el componente “SpriteRenderer” de Unity determina el orden de dibujo de los sprites. Ajustar este campo permite controlar qué imágenes se dibujan por encima de otras en la escena.
- **Gestión Dinámica**: Durante la interacción del usuario, se actualiza dinámicamente el “sortingOrder” de la imagen que se está arrastrando, asegurando que esta imagen siempre permanezca en la parte superior de la pila de renderizado.

Actualmente, esta solución solo se ha aplicado a las imágenes. Si las imágenes tuvieran fondos adicionales, cada capa del fondo también necesitaría ser gestionada, lo que aumentaría significativamente la complejidad de la lógica interna. La manipulación de múltiples capas requeriría ajustes adicionales en el "sortingOrder" de cada componente, aumentando la posibilidad de errores y complicaciones en la gestión del orden de renderizado.

Para futuras actualizaciones de esta actividad se podría cambiar la lógica de la gestión de capas para usar el Canvas, ya que este puede renderizar los elementos en diferentes capas de una manera más controlada.

En esta actividad también se ha considerado el posible problema al seleccionar palabras aleatorias del archivo JSON. Podría ocurrir que una palabra perteneciente a otro fonema, utilizada como una palabra que no contiene el fonema objetivo, en realidad sí lo contenga. Para evitar esta situación, se ha decidido crear un archivo JSON adicional que contiene grupos de palabras predefinidos. Estos grupos están seleccionados para asegurar que no haya coincidencias entre las palabras utilizadas.

```
{
  "agrupaciones": [
    {
      "fonema": "p",
      "palabrasPorFonema": {
        "palabras1": [
          "lupa",
          "pollo",
          "pan"
        ],
        "palabras2": [
          "abeja"
        ]
      }
    },
    {
      "fonema": "p",
      "palabrasPorFonema": {
        "palabras1": [
          "pato",
          "puño"
        ],
        "palabras2": [
          "burro",
          "abeja"
        ]
      }
    }
  ],
}
```

Figura 17 JSON con palabras para "Discriminación Fonema"

4.4. Iteración 4

En esta iteración se desarrollan las tres actividades restantes “Parejas”, “Sílabas Correctas” y “Pares Mínimos”.

4.4.1. Parejas

En la actividad “Parejas” se presentan 8 imágenes que inicialmente se muestran durante 2 segundos para que el usuario las memorice. Luego, las imágenes se voltean, y el usuario debe recordar y acertar las posiciones de las parejas coincidentes. Con este juego de memoria se pretende incorporar un componente de reconocimiento al tener que recordar las posiciones.

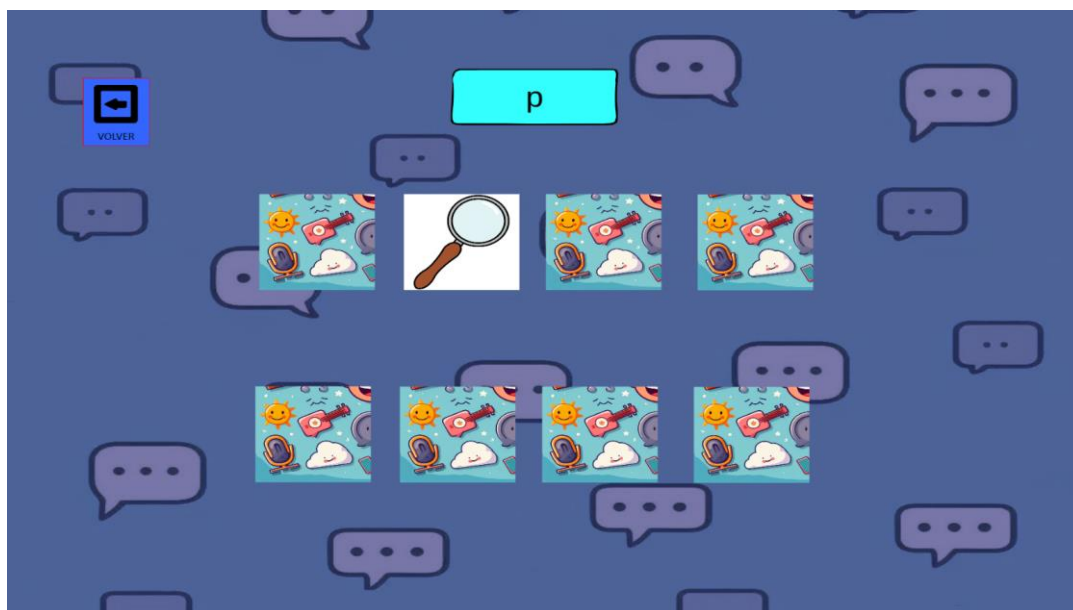


Figura 18 Actividad "Parejas" con las imágenes volteadas

Una vez que el usuario acierta una pareja de imágenes, se abre una ventana emergente en la que se le solicita pronunciar correctamente la palabra asociada con las imágenes emparejadas. Esta dinámica de producción de palabras aprovecha los recursos y funcionalidades desarrolladas en la primera actividad, donde se implementó la funcionalidad de reconocimiento de voz y verificación de pronunciación. Esto permite una reutilización eficiente del código y de los recursos.

En la ventana emergente también hay un botón de altavoz que permite al usuario escuchar la pronunciación correcta de la palabra asociada con la imagen, aunque cada vez que el usuario selecciona una imagen durante el juego, se reproduce automáticamente el audio con el nombre del dibujo.

En la parte de la lógica, primero se implementó un sistema para aleatorizar la posición de las imágenes en la escena. Esto se logra utilizando una lista que contiene las posiciones posibles para las imágenes. La lista se baraja utilizando el algoritmo Fisher-Yates, asegurando que cada partida tenga una disposición distinta de las imágenes.

```
//Aleatoriza las posiciones de las imagenes
public List<int> GenerateRandomNumbers(int x)
{
    // Crea una lista para almacenar los números
    List<int> numbers = new();

    // Agrega números del 0 al x a la lista
    for (int i = 0; i < x; i++)
    {
        if (i >= x / 2)
        {
            numbers.Add(i - x / 2);
        }
        else
        {
            numbers.Add(i);
        }
    }

    System.Random rng = new();
    // Ordena aleatoriamente los números utilizando el algoritmo de Fisher-Yates
    int n = numbers.Count;
    while (n > 1)
    {
        n--;
        int k = rng.Next(n + 1);
        (numbers[n], numbers[k]) = (numbers[k], numbers[n]);
    }

    return numbers;
}
```

Figura 19 Algoritmo Fisher-Yates para aleatorizar las imágenes

Para permitir al usuario memorizar las imágenes antes de que se volteen, se añadió un retraso de 2 segundos. Esto se implementó utilizando la función “Invoke” de Unity, que permite ejecutar una función después de un tiempo determinado. Cuando el usuario selecciona dos imágenes, se verifica si los sprites coinciden. Si no coinciden, se voltea nuevamente a la posición inicial.

Por otro lado, se han reutilizado funciones usadas en las anteriores actividades. Se ha utilizado la misma lógica de los botones, aunque se ha adaptado a las necesidades de la escena. Esto incluye botones para girar las imágenes y para la pronunciación y escucha de palabras. También se ha integrado la lógica de producción y comprensión de palabras, ya implementada en actividades anteriores.

4.4.2. Sílabas Correctas

La actividad “Sílabas Correctas” está diseñada para ayudar a los usuarios a identificar y seleccionar las sílabas que contienen un fonema específico. La interfaz de esta actividad presenta una imagen acompañada de su audio y su nombre. Debajo de la imagen, hay varios recuadros, cada uno referido a una sílaba de la palabra. Los usuarios pueden marcar y desmarcar estos recuadros para seleccionar las sílabas que contienen el fonema deseado.

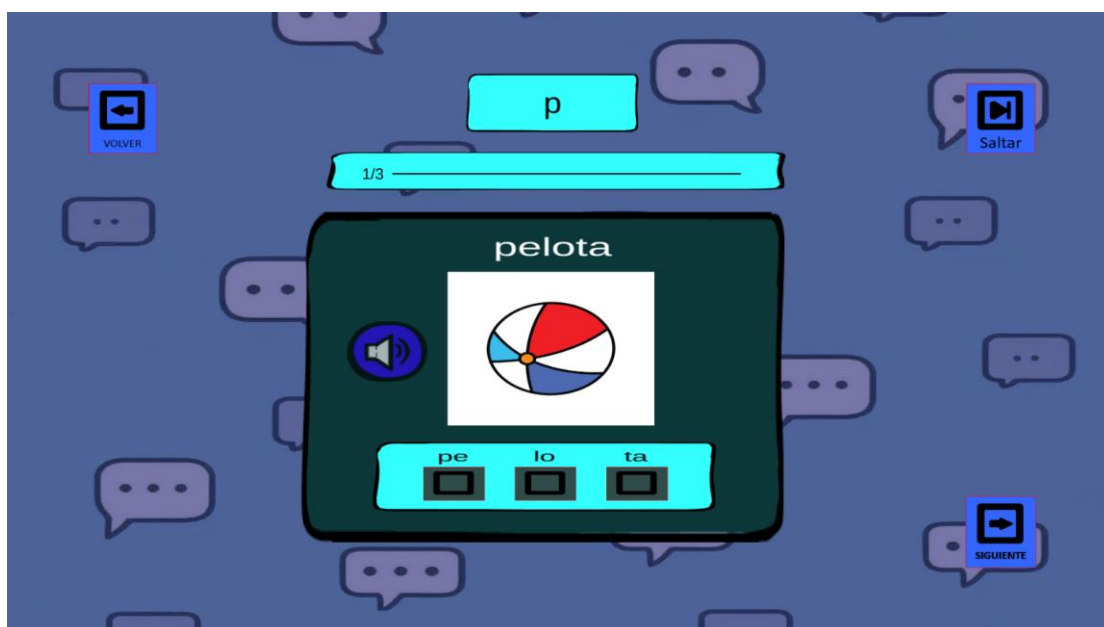


Figura 20 Actividad "Sílabas Correctas"

Para implementar esta actividad, ha sido necesario modificar el archivo JSON que contiene la lista de todas las palabras. Se ha añadido una lista con las sílabas de cada palabra y una indicación de si cada sílaba contiene o no el fonema objetivo.

La lógica de esta actividad es relativamente sencilla en comparación con la actividad “Parejas”. Principalmente, se centró en la implementación de los botones y la gestión de sus estados (marcado y desmarcado). Además, se ha añadido otro botón para comprobar la selección.

La actividad “Sílabas Correctas” consta de una imagen, con su audio y su nombre y debajo unos botones de tipo Toggle para cada sílaba. Estos “Toggle Button” se pueden marcar para seleccionar las sílabas que contienen el fonema deseado.

4.4.3. Pares Mínimos

La última actividad desarrollada se denomina “Pares Mínimos”. En esta actividad, se presentan dos imágenes, cada una con un botón de tipo Toggle asociado. El usuario debe seleccionar la imagen que contiene el fonema indicado. Los pares de imágenes presentadas son pares mínimos [\[38\]](#), lo que significa que las palabras representadas por las imágenes tienen solo una pequeña diferencia en el sonido, destacando el fonema específico que se quiere practicar.

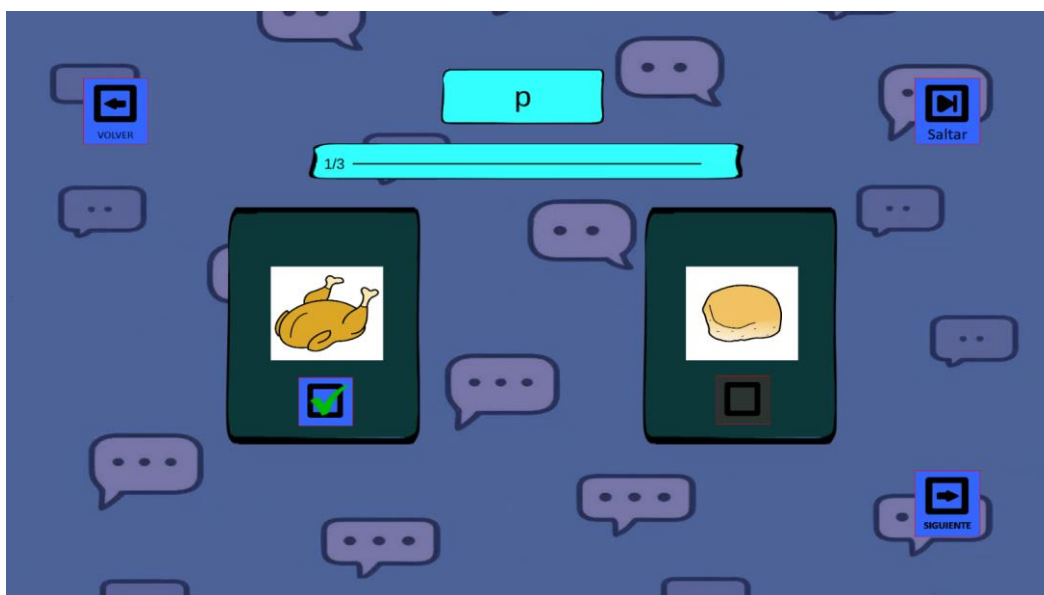


Figura 21 Actividad "Pares Mínimos"

Los botones Toggle están diseñados para permitir solo una selección única. Una vez que el usuario selecciona un botón, el juego directamente desvela si es la correcta o no. Cuando el usuario selecciona la imagen correcta, el botón Toggle asociado cambia su sprite para indicar una selección correcta. Si la selección es incorrecta, se muestra un sprite diferente que indica el error.

Se ha creado un nuevo archivo JSON para almacenar los pares mínimos. Este archivo contiene listas de dos palabras para cada par, junto con el fonema y cuál es la que contiene el fonema.

El script de control de la actividad carga los pares mínimos del archivo JSON y presenta uno al azar en cada ronda. Cuando el usuario hace una selección, el script compara la elección con la palabra correcta indicada en el JSON.

```
{
  "listaParesMinimos": [
    {
      "fonema": "p",
      "paresMinimos": {
        "correcta": "pollo",
        "incorrecta": "bollo"
      }
    },
    {
      "fonema": "p",
      "paresMinimos": {
        "correcta": "pino",
        "incorrecta": "chino"
      }
    }
  ],
}
```

Figura 22 JSON palabras para "Pares Mínimos"

4.5. iteración 5

Una vez completadas todas las actividades individuales, el siguiente paso es implementarlas de manera secuencial. Para lograr esto, se ha decidido dividir el juego en partidas y almacenar la información de cada partida en un objeto Singleton [\[37\]](#). Este enfoque permite que los datos relevantes persistan durante el transcurso del juego.

El objeto Singleton incluye un identificador único para cada partida, formado por la fecha y hora de creación, y un conjunto de listas con las palabras de cada actividad. Estas listas se preparan antes de comenzar el juego para asegurar que las palabras no se repitan entre actividades y se crean mediante un script asociado a la escena previa a la primera actividad, que accede a los archivos JSON que contienen las palabras y las asigna a las listas correspondientes. Una vez iniciada cada actividad, esta aleatoriza su lista específica de palabras y coloca las imágenes y los audios en las posiciones necesarias, para que no se repitan entre actividades.

Además del modo secuencial, se ha implementado un selector de actividades que permite a los usuarios probar las actividades de manera independiente. Esto es útil para realizar pruebas específicas o para que los usuarios practiquen actividades individuales. También se ha añadido un selector de fonemas que permite elegir el fonema con el que se realizarán las actividades.

4.5.1. Grabación de audios

Se han realizado modificaciones en la dinámica de comprensión para proporcionar herramientas adicionales para la evaluación de la pronunciación del usuario.

Se ha incluido la opción de grabar los audios del usuario durante las actividades. Esto permite que la pronunciación del usuario pueda ser evaluada no solo en tiempo real, sino también después de cada partida. Esta característica es especialmente útil para que los usuarios puedan revidar sus progresos y para que los instructores puedan proporcionar retroalimentación.

Para implementar este sistema, se ha reutilizado y adaptado el script "SavWav" del TFG de Gonzalo Fernández. Este script permite almacenar el resultado de una grabación en formato .wav.

```
void IniciarGrabacion()
{
    recordedClip = Microphone.Start(null, false, 10, 44100);
}

void DetenerGrabacion()
{
    Microphone.End(null);
    SavWav.Save(recordedClip);
}
```

Figura 23 Script de grabación de audios

Se han realizado pruebas para asegurarse de que la grabación de audio y el uso del reconocedor de voz de Unity pueden funcionar simultáneamente sin conflictos. Los resultados han sido positivos, permitiendo la integración de ambas funcionalidades de manera fluida.

4.5.2. Resultados

A la hora de gestionar el desempeño del usuario, se ha implementado un sistema de resultados que se presenta al final de cada partida. Este sistema está diseñado para proporcionar una visión clara del rendimiento del usuario.

Para implementarlo lo primero que se ha hecho es crear una opción para saltar las imágenes. Durante cada escena, se ha añadido un botón que permite al usuario saltar las imágenes que no consiga acertar. Este botón añade las imágenes actuales a una lista dentro del objeto Singleton que almacena la información de la partida, y luego avanza a la siguiente escena, o muestra la siguiente palabra de la actividad. Esta funcionalidad asegura que el usuario no se quede atascado en una escena y pueda continuar con la partida.

En la escena de resultados, al final de la partida, se presentan todas las palabras que el usuario ha saltado. Esta presentación permite al usuario revisar los elementos que no ha conseguido acertar, proporcionando una oportunidad para repasar los fallos. A esta escena se le ha añadido otra que permite revisar todos los audios que se han grabado durante la partida, con la opción creada anteriormente de grabar los audios.

4.6. iteración 6

En esta última iteración, se ha enfocado en la creación de la memoria del proyecto, la implementación de cambios finales en la aplicación y la corrección de errores.

Se ha dedicado gran parte del tiempo de esta iteración a redactar la memoria del TFG, documentando cada aspecto del desarrollo del proyecto. Durante el proyecto ya se han ido creando algunas partes, como son la planificación y las tareas principales de las iteraciones, y en este tiempo se ha dedicado a convertir esos apuntes tomados en la memoria.

El principal cambio que se ha realizado en esta iteración en la lógica del proyecto ha sido la implementación de un sistema de intentos en las actividades “Discriminación de Fonema” y “Pares Mínimos”. Inicialmente, los audios y nombres de las palabras no se muestran. Si el usuario falla una vez, los nombres de las palabras se revelan. Si falla dos veces, aparecen los botones con los que el usuario puede reproducir los audios. Este sistema proporciona retroalimentación gradual al usuario, haciendo que puedan avanzar en el desarrollo de estas actividades sin frustrarse.

Además, se han añadido dos escenas. La primera ha sido para la portada, con el título del juego y la segunda con los créditos, en la que se reconoce a los tutores y se añade una referencia a ARASAAC.

En la parte de la refactorización del código, se han realizado cambios en el código para mejorar su claridad y comprensibilidad y se ha cambiado la estructura de carpetas de los scripts para hacerla más intuitiva y fácil de navegar. Esta nueva organización se ha alineado a la utilizada por Gonzalo en su TFG, haciendo más fácil la mantenibilidad del proyecto.

Capítulo 5

5. Análisis

En este apartado se realizará un análisis detallado de los requisitos funcionales y no funcionales de la aplicación con la intención de transformar las necesidades del cliente en especificaciones técnicas.

5.1. Actores del sistema

El proyecto solo tiene el actor “Usuario”, ya que es el que se encarga de interactuar con la interfaz de la aplicación. Si se continuase con el desarrollo, se podría añadir otro actor “Profesor” o “Revisor”, que podría tener un acceso personalizado a las grabaciones del juego, para así poder revisarlas.

5.2. Requisitos

En este apartado, se presentan los distintos requisitos obtenidos después de realizar un análisis de los objetivos que tiene la aplicación. Se enumeran los requisitos funcionales, no funcionales y de información, asegurando que cumplan las necesidades de los usuarios.

5.2.1. Requisitos Funcionales

Los requisitos funcionales son especificaciones detalladas que describen lo que debe de hacer una aplicación. Para esta aplicación se han destacado los siguientes:

- El sistema deberá permitir al usuario elegir un fonema
- El sistema deberá permitir al usuario grabar palabras
- El sistema deberá permitir al usuario reproducir audios

- El sistema deberá permitir al usuario usar el reconocedor de voz para comprobar si su pronunciación es correcta.
- El sistema deberá permitir al usuario elegir una actividad en específico.
- El sistema deberá permitir al usuario jugar a una lista aleatoria de actividades.
- El sistema deberá permitir al usuario ver unos resultados al final de las actividades.
- El sistema deberá permitir al usuario cambiar de actividad una vez se haya terminado la actividad actual.
- El sistema deberá permitir al usuario realizar la dinámica de pares mínimos.
- El sistema deberá permitir al usuario realizar la actividad “Nómbralo”.
- El sistema deberá permitir al usuario realizar la actividad “Selección Correcta”.
- El sistema deberá permitir al usuario realizar la dinámica de discriminación de fonemas.
- El sistema deberá permitir al usuario realizar la actividad “Sílabas Correctas”.
- El sistema deberá permitir al usuario realizar la actividad “Parejas”.
- El sistema deberá permitir saltar las palabras que no se acierten.

5.2.2. Requisitos no funcionales

Los requisitos no funcionales son especificaciones que describen cualidades del sistema. Son los siguientes:

- El sistema deberá tener una interfaz adaptada a las especificidades de las personas con síndrome de Down, asegurando que sea accesible y fácil de entender.
- El sistema deberá poder reproducir las distintas actividades de manera fluida y sin retrasos.
- El sistema deberá ser compatible con una amplia variedad de ordenadores.
- El sistema deberá permitir al usuario comprobar que su grabación se pronuncia correctamente.
- El proyecto debe desarrollarse en Unity versión 2021.2.9f1.
- El sistema deberá utilizar ficheros de configuración en formatos estándar como JSON para la configuración y WAV para los archivos de audio
- El sistema deberá utilizar un reconocedor de voz o local o en la nube.

5.2.3. Requisitos de información

Los requisitos de información detallan como deben de ser los datos del sistema. Son los indicados:

- El sistema deberá incluir una variedad de palabras para cada actividad del juego.
- El sistema deberá incluir una variedad de fonemas para poder realizar el juego con cada uno

- El sistema guardará las grabaciones de partidas anteriores.

5.3. Modelo de dominio

Para representar las distintas clases del proyecto que se han obtenido de los requisitos, incluyendo su información y las relaciones que hay entre ellas, se ha realizado el modelo de dominio.

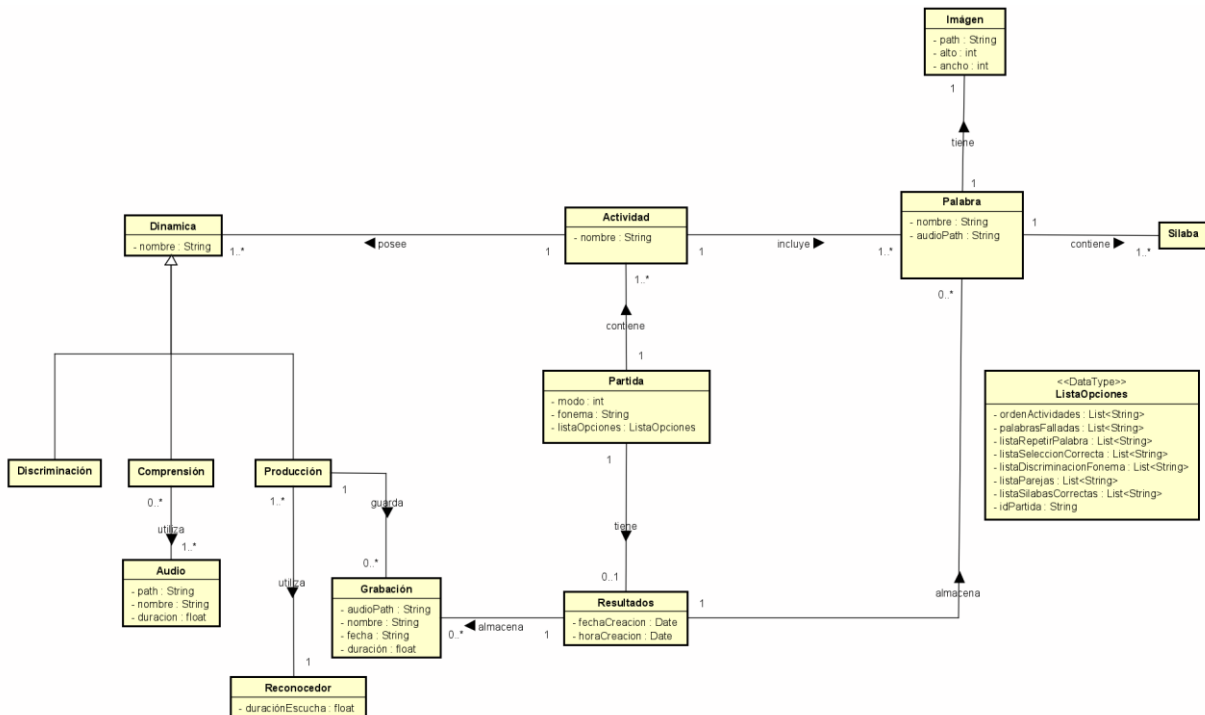


Figura 24 Modelo de dominio

5.3.1. Clases

- **Dinámica:** representa los diferentes tipos de actividades dentro del juego
- **Actividad:** contiene las características de cada actividad del juego
- **Partida:** representa una sesión de juego completa
- **ListaOpciones:** Es un datatype que almacena los datos de la partida
- **Palabra:** Representa una palabra usada en las actividades
- **Sílaba:** Representa una sílaba dentro de una palabra
- **Audio:** Representa los archivos de audio que se usan en las actividades
- **Grabación:** Almacena las grabaciones de las sesiones de juego
- **Resultados:** Almacena los resultados de una partida
- **Reconocedor:** Utilizado para el reconocimiento de voz en actividades

- **Imagen:** Representa imágenes usadas en las actividades.

5.4. Diagramas de actividad

Se han creado los diagramas de actividad de cada una de las actividades que forman el juego.

5.4.1. Nómbralo

La primera actividad cuenta con la dinámica de producción, la de comprensión, saltar a la siguiente palabra y saltar la imagen si no se sabe resolver. El audio solo deja de grabar una vez se acierta la palabra. Cuando no quedan más palabras se avanza a la siguiente escena.

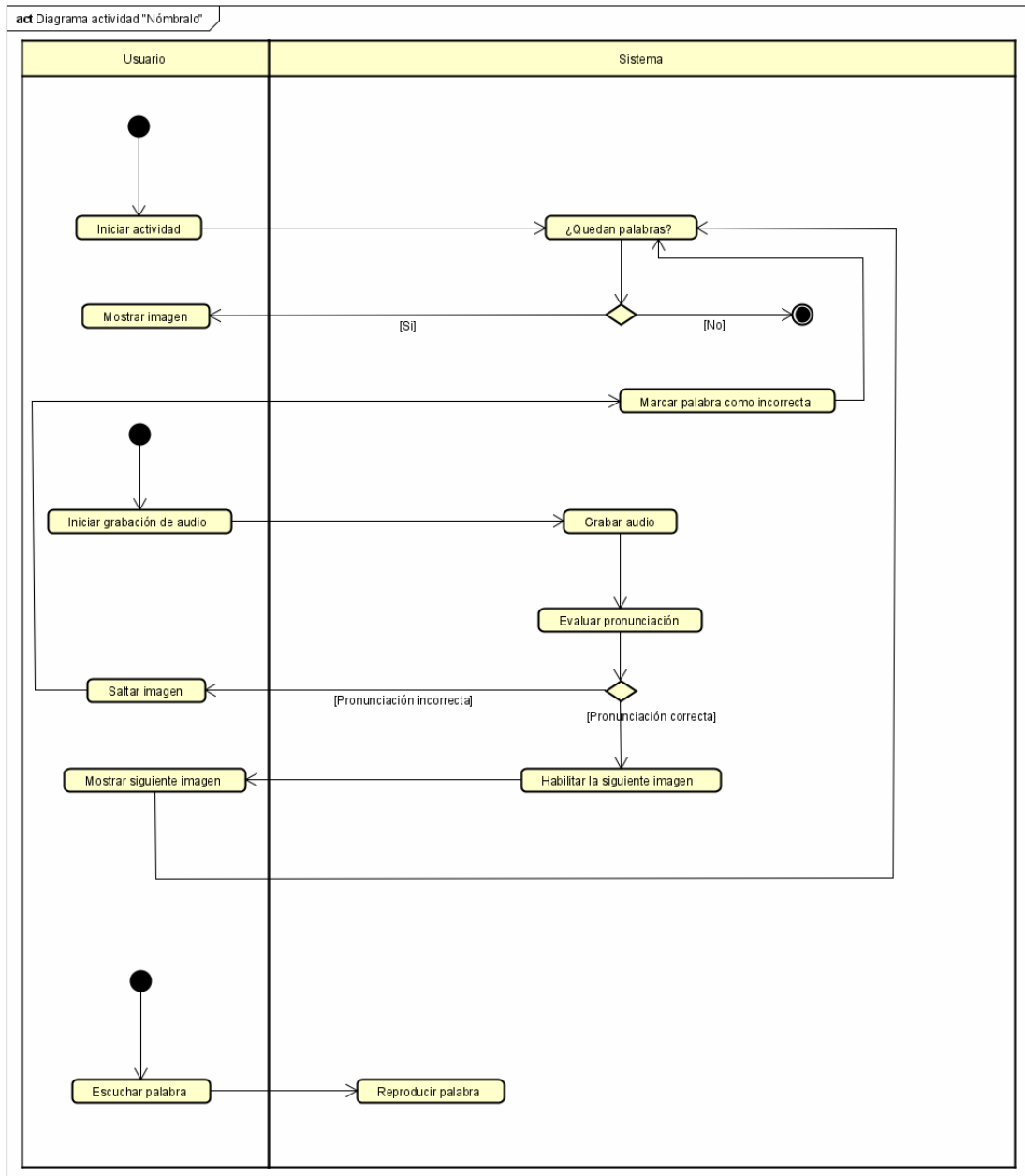


Figura 25 Diagrama de actividad de "Nómbrale"

5.4.2. Selección correcta

La segunda actividad cuenta con las imágenes y la dinámica de comprensión para reproducir sus audios. Si se pulsa la imagen correcta se cambian de posición y hay que buscar la siguiente imagen correcta. También se puede saltar la actividad.

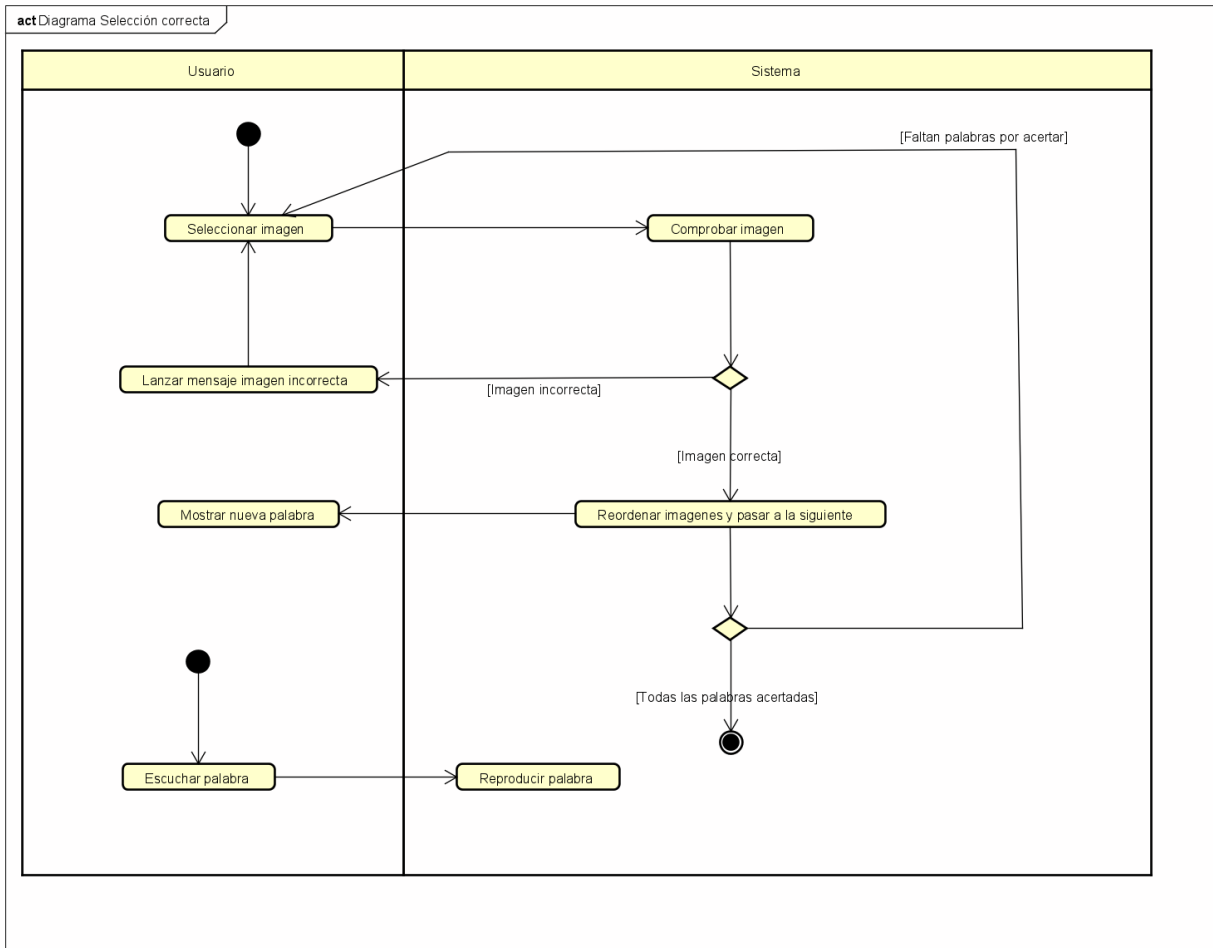


Figura 26 Diagrama de actividad de "Selección Correcta"

5.4.3. Discriminación de fonema

La tercera actividad cuenta con 4 imágenes que el usuario puede arrastrar a su posición correcta. Dependiendo de donde se coloquen suena un sonido de acierto o de fallo. Como las anteriores actividades, esta también se puede saltar.

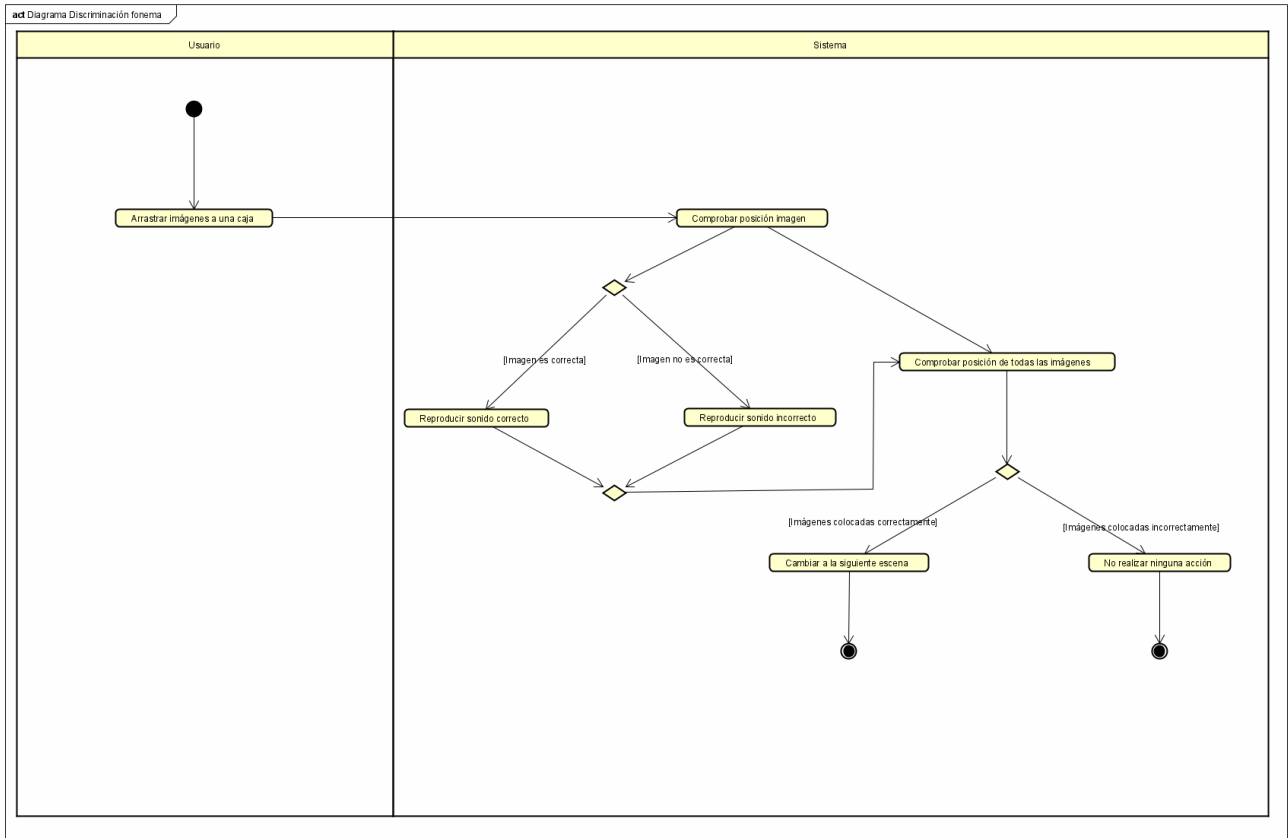


Figura 27 Diagrama de actividad de "Discriminación Fonema"

5.4.4. Parejas

En la cuarta actividad el usuario selecciona imágenes iguales para después reproducir su nombre. Incluye como en las anteriores la opción de saltar la grabación del nombre de las imágenes.

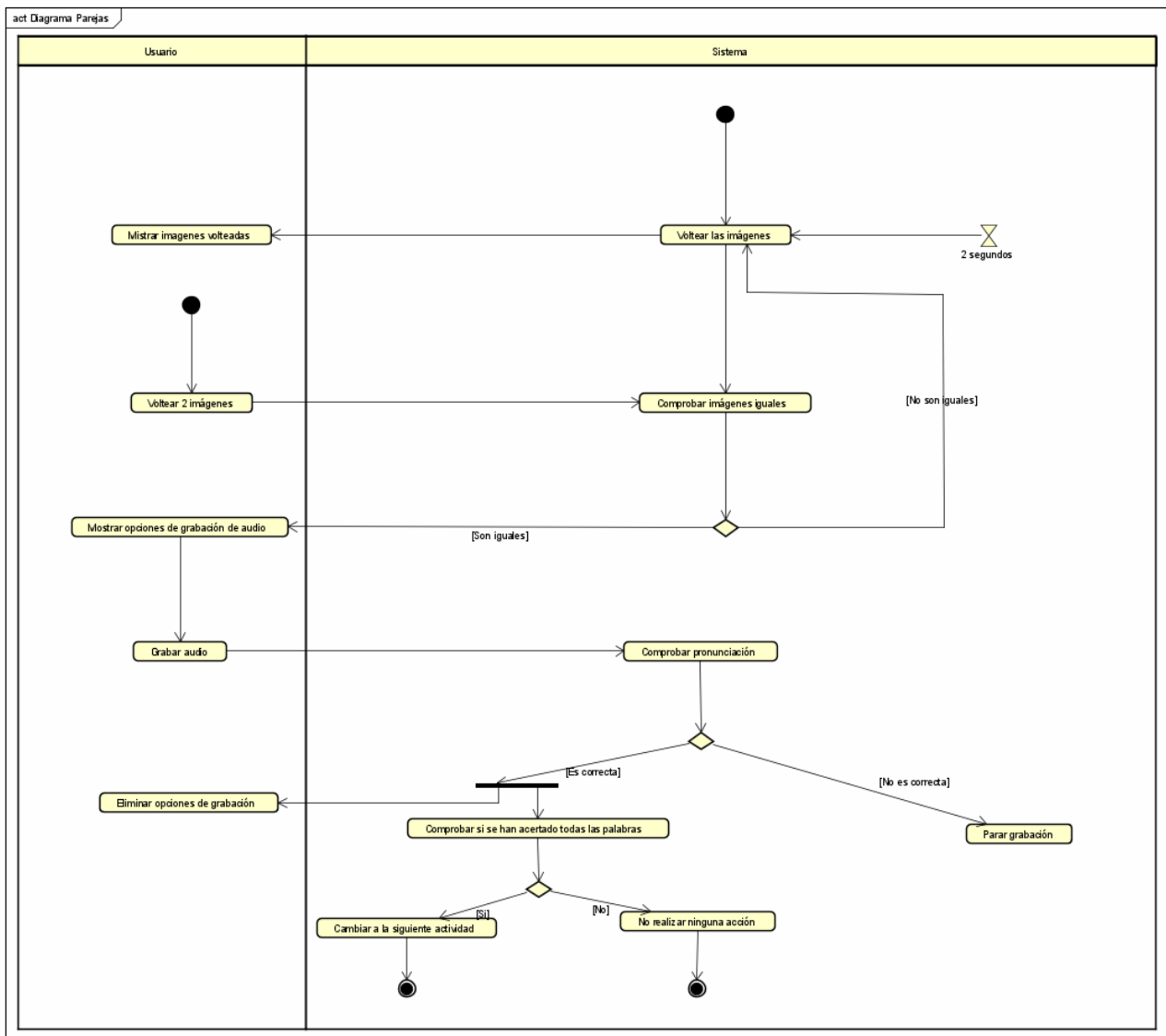


Figura 28 Diagrama de actividad de "Parejas"

5.4.5. Sílabas correctas

Para la quinta actividad el usuario tiene que seleccionar las sílabas correctas. Si al realizar la comprobación el usuario acierta, pasará a la siguiente imagen hasta que ya no queden más imágenes.

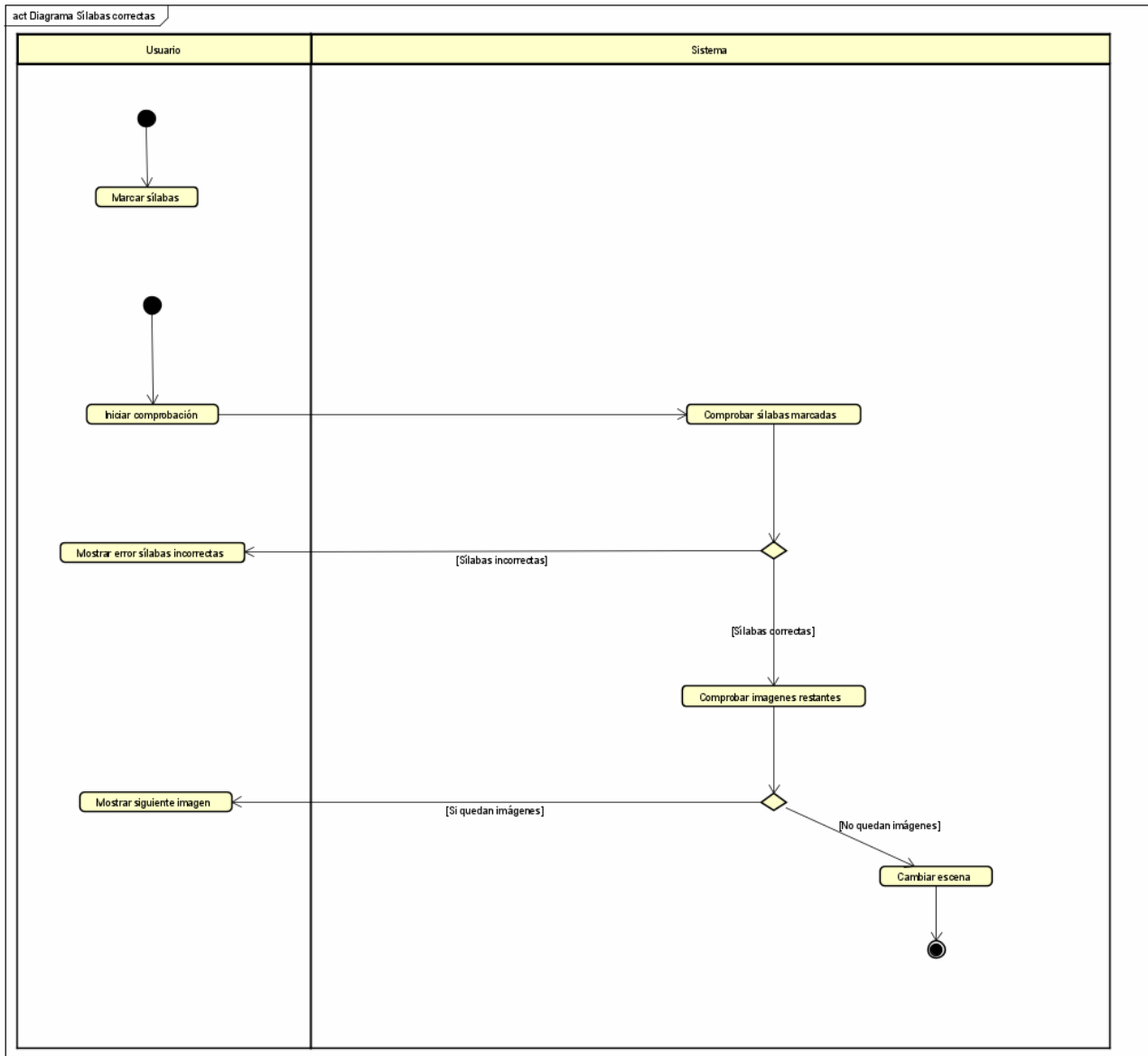


Figura 29 Diagrama de actividad de "Sílabas Correctas"

5.4.6. Pares mínimos

En esta última actividad el usuario tiene que seleccionar una de las dos imágenes disponibles y el juego marcará como incorrecta si se ha fallado o pasará a la siguiente palabra, si se ha acertado. Si no se conoce la respuesta también se puede saltar la prueba.

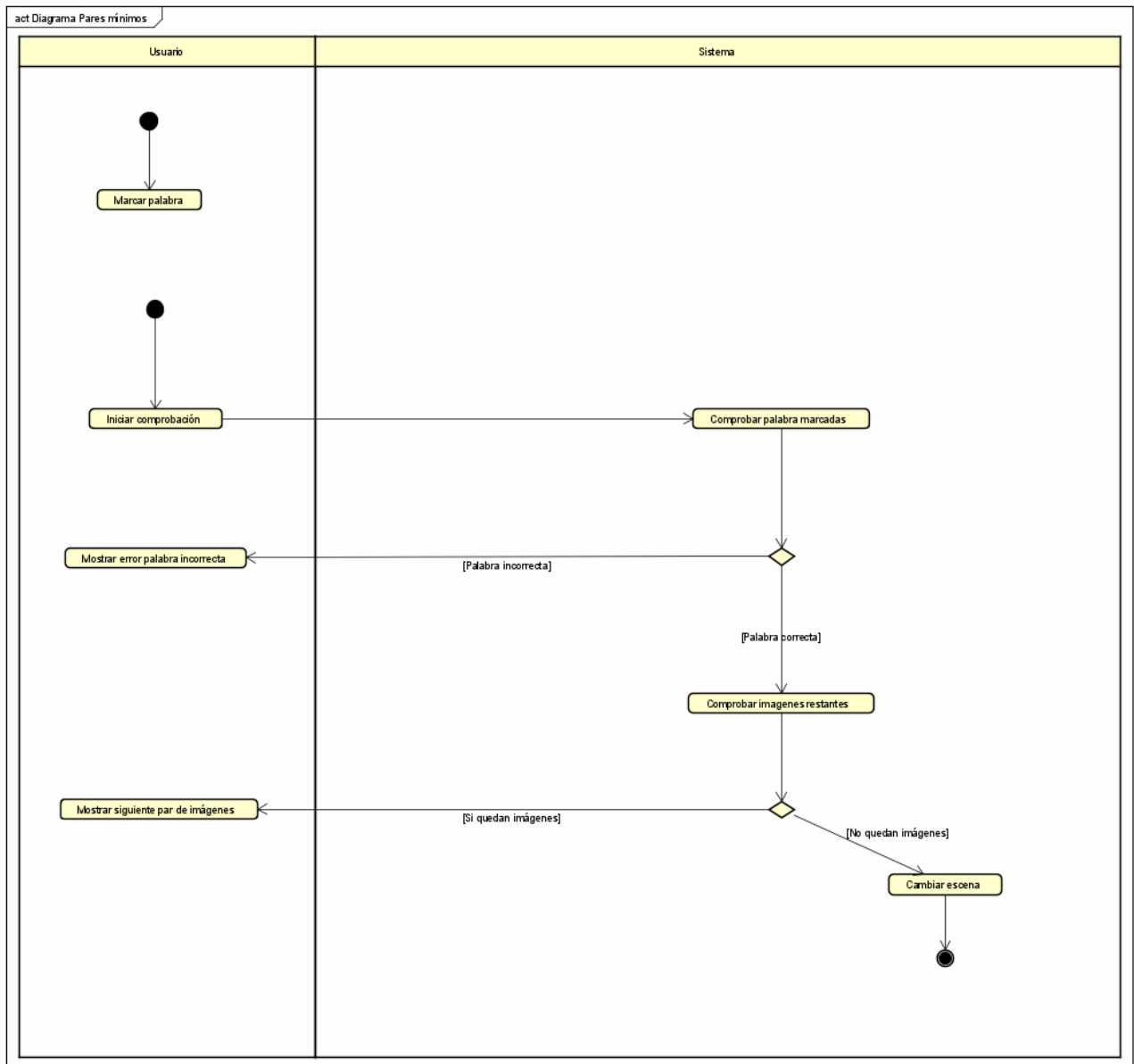


Figura 30 Diagrama de actividad de "Pares Mínimos"

Capítulo 6

Capítulo 6

6. Diseño

6.1. Estructura del proyecto

En la siguiente figura se puede observar cómo se ha estructurado el proyecto.

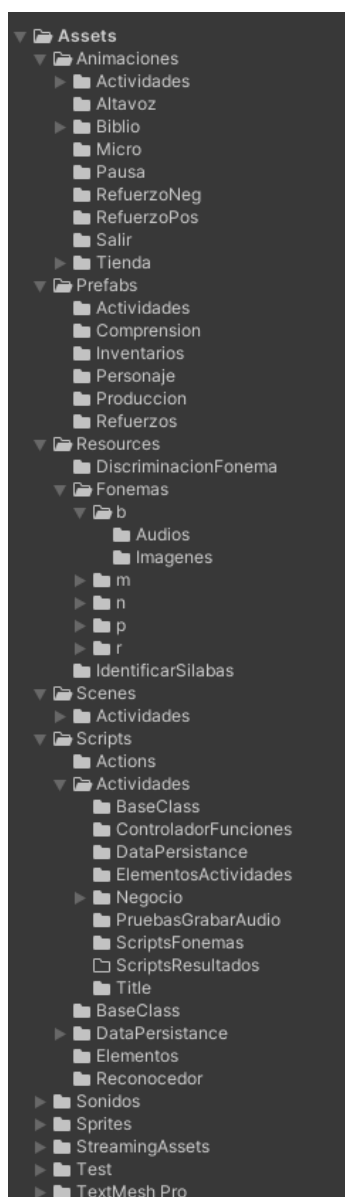


Figura 31 Estructura del proyecto

- **Carpeta de prefabs:** La carpeta prefabs almacena los objetos prefabricados de Unity para utilizarlos en varias actividades.
- **Carpeta de animaciones:** Esta carpeta almacena los elementos que se usan a la hora de crear las animaciones, como son el animator controller y los animator clip.
- **Carpeta escenas:** En esta carpeta se almacenan las distintas escenas que forman el juego.
- **Carpeta resources:** La carpeta Resources se usa para almacenar los elementos que aparecen en tiempo de ejecución.
- **Carpeta scripts:** Esta carpeta almacena los diferentes scripts que se usan para el funcionamiento de la lógica del proyecto. Su subdivisión se explica en el apartado de arquitectura.

6.2. Patrones de diseño

Los patrones son soluciones a un problema general que pueden adaptarse a problemas concretos. En este apartado voy a describir los patrones de diseño que he usado a la hora de programar la aplicación.

6.2.1. Singleton

Con el patrón Singleton [\[38\]](#) nos cercioramos de que solo existe una instancia de una clase. En este proyecto se ha usado para almacenar y gestionar la información de cada partida, como el identificador único, las listas de palabras para cada actividad y los resultados de estas. De esta manera solo hay una instancia de este objeto, pudiendo acceder a ella desde cualquier parte del juego.

6.2.2. Observador

El patrón observador [\[39\]](#) establece una relación de dependencia que, al cambiar el estado de un objeto, se notifica ese cambio a todos los objetos dependientes.

Su implementación en Unity cuenta con dos partes. Primero, la acción que avisa del cambio que se ha producido, que en el código se expresa declarando un evento e invocándolo:

- `public static event Action PalabraAcertada;`

- `PalabraAcertada?.Invoke();`

Posteriormente hay que suscribirse en el objeto dependiente:

- `ReconocedorKeywords.PalabraAcertada += PalabraDetectada;`

En este caso “PalabraDetectada” es un método que se activa cuando el audio detectado es correcto.

En el desarrollo del proyecto, este patrón se ha utilizado en varias áreas clave:

- **Dinámica de Producción:** Se lanza un aviso al detectar la palabra correcta, lo que permite reaccionar inmediatamente al evento.
- **Actividad “Discriminación de Fonema”:** Al pulsar los objetos, se avisa para actualizar las capas del resto, asegurando la correcta visibilidad y orden de los elementos. Al soltar los objetos, se comprueba si están en la posición correcta, facilitando la evaluación de la acción del usuario.

6.2.3. Polimorfismo

El uso de prefabs en Unity permite aplicar polimorfismo al definir objetos base que pueden ser instanciados. Los prefabs actúan como plantillas que se pueden heredar y modificar, facilitando la reutilización y la gestión de objetos en el juego.

En el proyecto se han usado prefabs a modo de ventanas emergentes para confirmar los cambios de escenas. También se ha usado en la actividad “Parejas” para generar una ventana emergente con la que grabar los audios.

6.2.4. Patrón fachada

El patrón fachado [\[40\]](#) unifica una interfaz para un conjunto de interfaces de un subsistema. En este proyecto se ha reutilizado su uso del TFG de Gonzalo Fernández dado que la parte del almacén de grabaciones se ha implementado de la misma manera.

6.2.5. Patrón Update Method y Game Loop

En Unity, el patrón Update Method [\[41\]](#) es fundamental para actualizar el estado del juego y responder a las interacciones del usuario en cada frame. Este patrón asegura que las acciones y eventos se gestionen de manera continua y fluida. Se ha usado en muchas partes del desarrollo, desde los reconocedores de voz para ir comprobando si tiene que seguir grabando, hasta en “Discriminación de Fonema” para comprobar de manera continua si se está arrastrando algún objeto.

El patrón Game Loop [\[42\]](#) está muy relacionado y es muy importante para mantener el flujo del juego, ya que procesa la entrada del usuario y actualiza y renderiza el juego. El método Update se integra dentro del bucle de juego, permitiendo que se actualicen todas las partes del juego y garantizando que el juego responda a las acciones del usuario en tiempo real.

6.3. Diagrama de arquitectura

En la estructura de la arquitectura se ha decidido usar el patrón Programación por Capas igual que Gonzalo en el TFG de migración de Pradia, para mantener una coherencia entre los dos proyectos. El código se ha organizado en tres capas:

- Capa de Presentación: Esta capa es responsable de la interacción con el usuario. Incluye la interfaz, con botones, menús y otros elementos visuales.
- Capa de Lógica de Negocio: Esta capa gestiona la entrada recibida de la capa de presentación, aplicar las reglas de negocio y devolver los resultados adecuados.
- Capa de Acceso a Datos: Esta capa maneja la comunicación con las fuentes de datos.

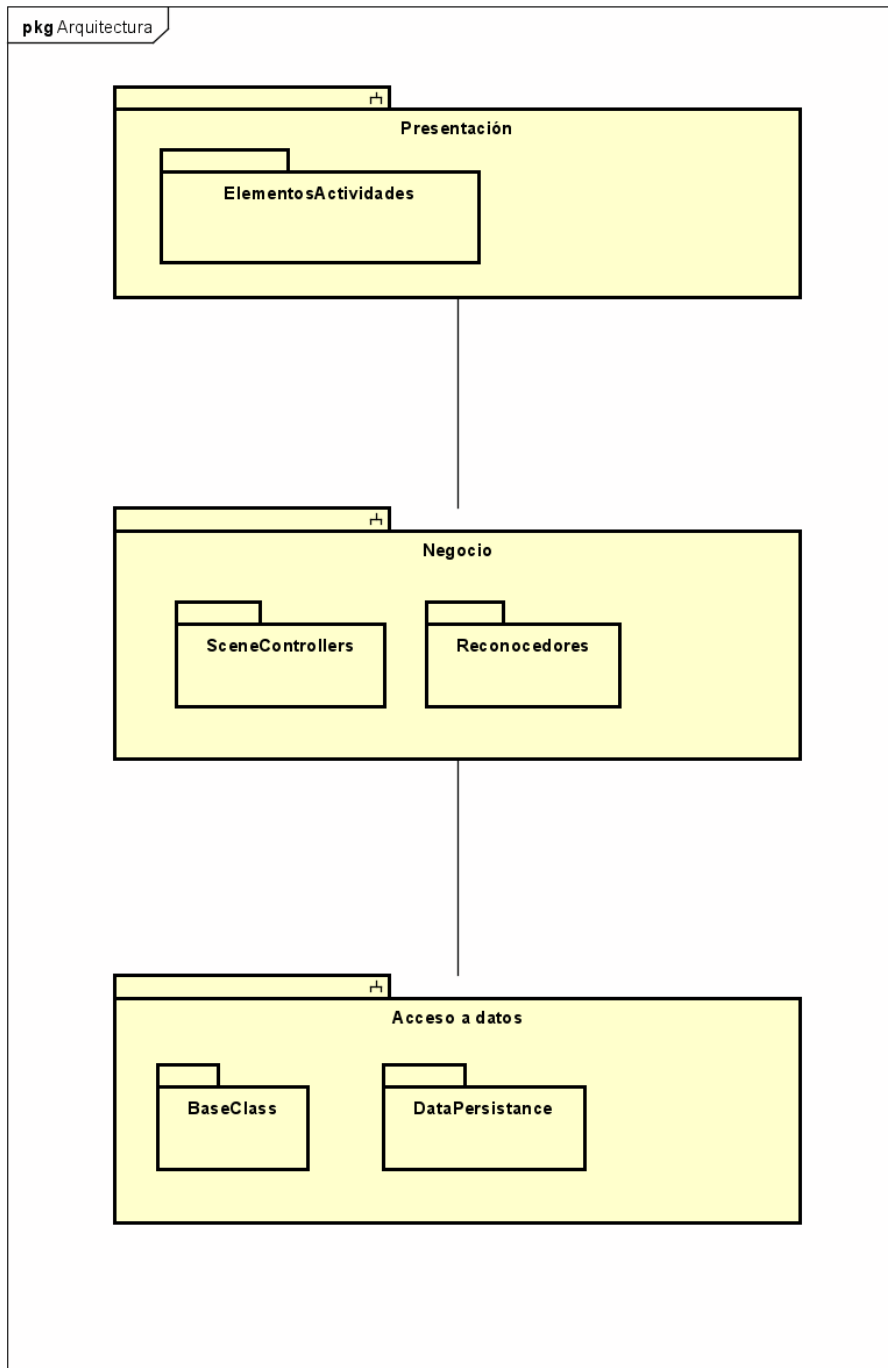


Figura 32 Diagrama de arquitectura

6.4. Diagrama de paquetes

En este apartado se muestra como ha quedado la estructura de paquetes del proyecto. La estructura general del proyecto se mantiene de la misma manera que el TFG de Gonzalo:

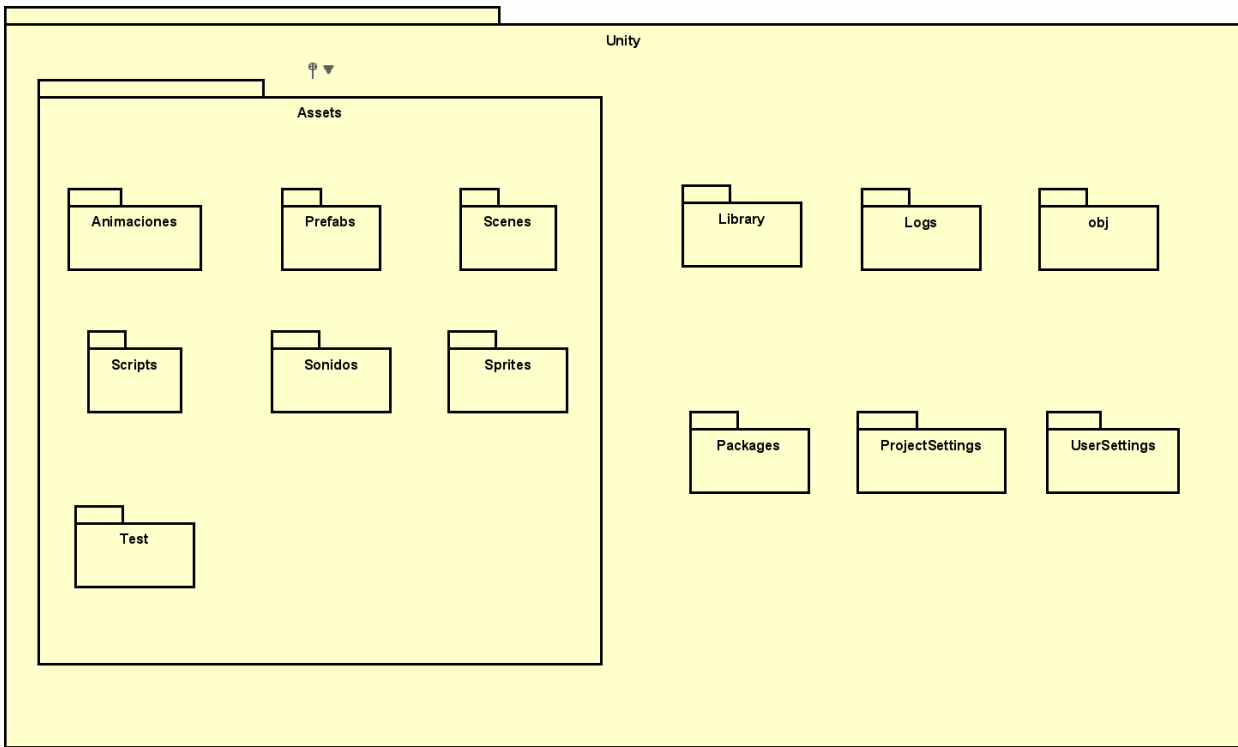


Figura 33 Estructura de Unity por paquetes

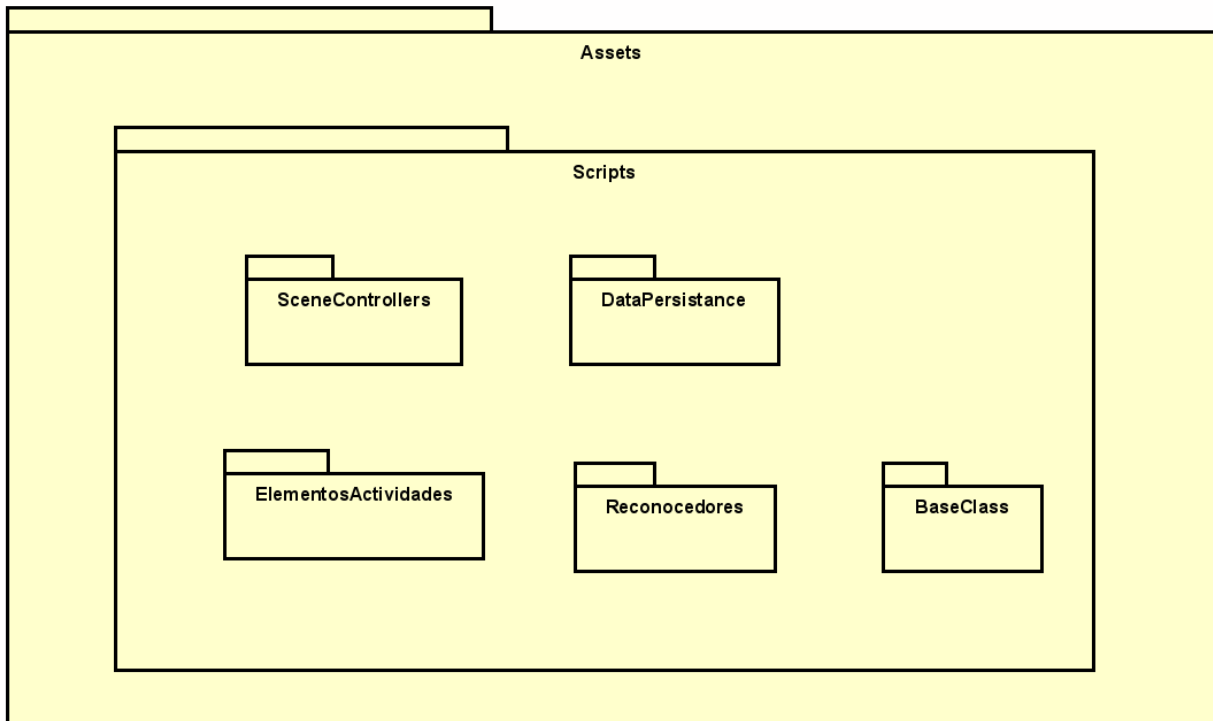


Figura 34 Diseño detallado del paquete Scripts

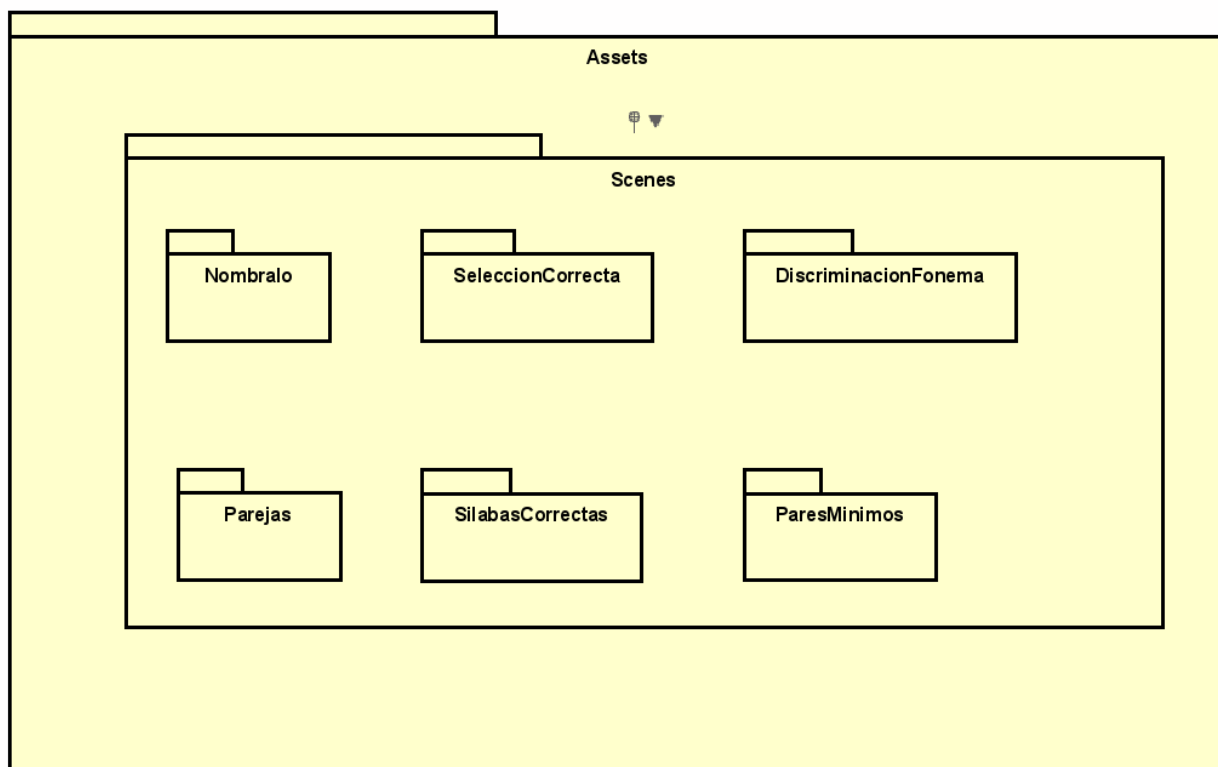


Figura 35 Diseño detallado del paquete Scenes

En la carpeta Resources se han almacenado todas las imágenes y los audios para acceder a ellas en tiempo de ejecución y los archivos JSON que almacenan los objetos de las palabras.

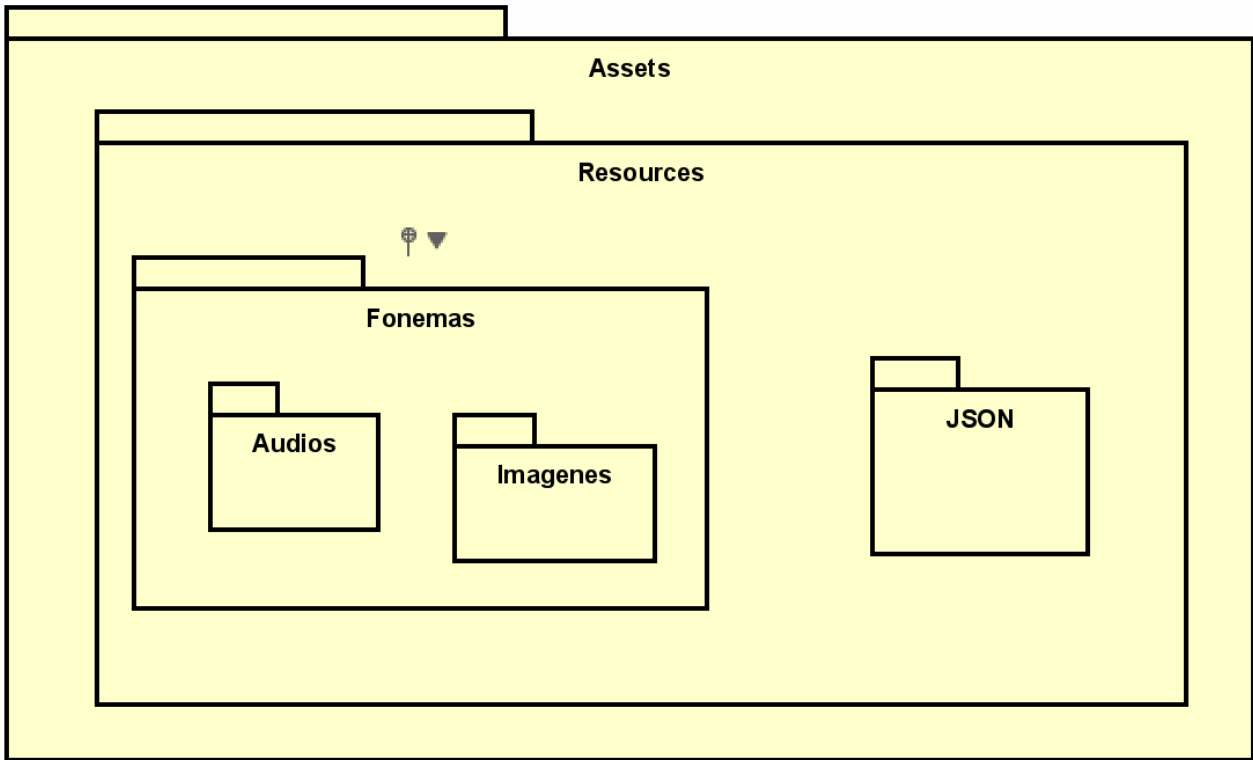


Figura 36 Diseño detallado del paquete Resources

6.5. Diagrama de despliegue

En este apartado se describe el diagrama de despliegue de la aplicación, detallando como se estructuran y relacionan los diferentes componentes del sistema en el entorno de ejecución. La aplicación se ejecuta en un PC y utiliza el reconocedor de voz de Unity sin necesidad de servidores ni servicios externos.

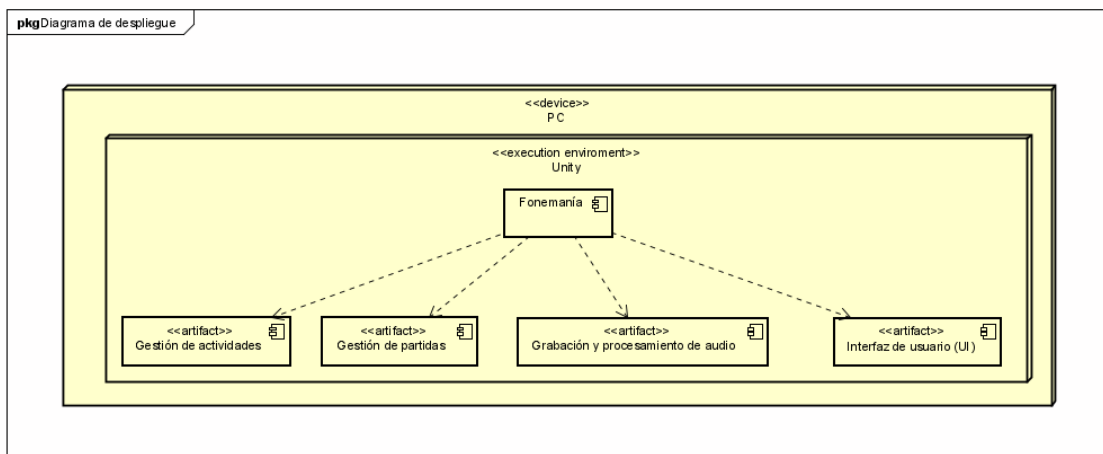


Figura 37 Diseño del diagrama de despliegue

El sistema consta solo de un componente principal, el dispositivo del usuario. La aplicación principal desarrollada en Unity se ejecuta en el PC del usuario. Esta aplicación contiene todos los elementos necesarios para la funcionalidad del juego.

Los principales componentes del código son los siguientes:

- **Gestión de Actividades:** Código que maneja la lógica de las actividades del juego.
- **Gestión de Partidas:** Código que utiliza un objeto Singleton para almacenar información de la partida actual.
- **Grabación y Procesamiento de audio:** Código que utiliza el reconocedor de voz de Unity para procesar el audio del usuario.
- **Interfaz de Usuario:** Código que gestiona la interfaz gráfica del usuario, incluyendo botones, menús y elementos visuales.

6.6. Diseño de la Interfaz de usuario

La interfaz de usuario se define como el medio a través del cual los usuarios interactúan con la aplicación. Un diseño de UI efectivo es fácil de usar y accesible, asegurando que los usuarios puedan navegar y utilizar la aplicación sin dificultad. En el caso de las aplicaciones de juegos educativos, como la desarrollada en este proyecto, la interfaz debe ser especialmente clara y atractiva para mantener el interés de los usuarios y facilitar el aprendizaje.

La interfaz de usuario abarca todos los elementos visuales y de interacción que el usuario utiliza para comunicarse con la aplicación, lo que incluye botones, menús, sliders, iconos...

Los botones en la interfaz son grandes y visibles, con símbolos intuitivos que indican claramente su función. Por ejemplo, un botón con un símbolo de altavoz se utiliza para reproducir un audio, mientras que un botón con una flecha se utiliza para avanzar o retroceder. Esto facilita su uso especialmente a los usuarios más jóvenes o con habilidades de lectura limitadas.

En algunas escenas, se han incorporado sliders que muestran el progreso del usuario a través de las imágenes de una actividad. Estas sliders permiten al usuario ver fácilmente en qué punto se encuentra y cuántas imágenes o pasos quedan por completar. Se colocan en una posición destacada en la pantalla para que los usuarios puedan ver su progreso de un vistazo.

Capítulo 7

7. Tests

7.1. Tests Unitarios

Los tests son procesos esenciales en el desarrollo de software, que sirven para verificar si el código realiza sus funcionalidades correctamente y que además cumple con los requisitos que se han especificado previamente. En Unity existen principalmente dos tipos de tests: los tests en modo Edit y los tests en modo Play.

- **Tests en Modo Edit:** Estos tests se ejecutan sin necesidad de iniciar el juego. Sirven para verificar la lógica del código y las funciones que no dependen de la ejecución del juego. Son útiles para probar métodos y clases aisladas. Están formados por varias partes:
 - **Configuración:** Preparar el entorno de prueba, incluyendo la instancia de los objetos y los datos necesarios.

```
[SetUp]
public void Setup()
{
    gameObject = new GameObject();
    reconocedorKeywords = gameObject.AddComponent<ReconocedorKeywords>();
}
```

Figura 38 Configuración de los Edit tests

- **Ejecución y verificación:** Ejecutar la función o método de prueba y comprobar el resultado obtenido con el esperado usando aserciones.

```

[Test]
public void IniciarGrabacion_IniciaCorrectamente()
{
    reconocedorKeywords.IniciarGrabacion();
    Assert.IsTrue(Microphone.IsRecording(null));
}

```

Figura 39 Ejemplo de ejecución de una prueba en el modo Edit

- **Limpieza:** Restablecer el entorno para evitar efectos secundarios en otros tests.

```

[TearDown]
public void TearDown()
{
    Object.DestroyImmediate(gameObject);
}

```

Figura 40 Ejemplo de limpieza de una prueba en el modo Edit

- **Tests en Modo Play:** Los tests en modo Play se ejecutan mientras el juego está en marcha. Estos tests verifican el comportamiento del juego en tiempo de ejecución, incluyendo la interacción entre objetos y la respuesta a eventos en tiempo real. Son muy importantes para asegurarse de que el juego funciona como se espera durante la ejecución.

- **Configuración:** Iniciar la escena de juego y preparar los elementos necesarios.

```

[SetUp]
public void SetUp()
{
    gameObject = new GameObject();
    reconocedorDictation = gameObject.AddComponent<ReconocedorDictation>();
}

```

Figura 41 Configuración de los Play Tests

- **Ejecución y verificación:** Simular las interacciones y eventos del usuario en tiempo real y observar y comparar el comportamiento del juego con el esperado

```

[UnityTest]
public IEnumerator GrabarPalabraClave_DetectaPalabraCorrectamente()
{
    reconocedorDictation.GrabarPalabraClave("test");
    yield return new WaitForSeconds(1);
    reconocedorDictation.DictationRecognizer_DictationResult("test", ConfidenceLevel.Medium);
    Assert.Pass();
}

```

Figura 42 Ejemplo de ejecución de una prueba en el modo Play

- **Limpeza:** Restablecer el entorno para evitar efectos secundarios en otros tests.

```

[TearDown]
public void TearDown()
{
    Object.DestroyImmediate(gameObject);
}

```

Figura 43 Ejemplo de limpieza de una prueba en el modo Play

En este proyecto, se han realizado tests específicos en las dinámicas de comprensión y producción y en la parte que contiene la lógica de arrastrar los objetos en la actividad “Discriminación de Fonemas”. Se han seleccionado estos apartados por los siguientes motivos:

- **Dinámicas de Comprensión y Producción:** Los tests aseguran que las funciones de reconocimiento de voz y evaluación de pronunciación se ejecutan correctamente, proporcionando retroalimentación precisa.
- **Lógica de arrastrar objetos en “Discriminación de Fonema”:** La actividad “Discriminación de Fonema” incluye una mecánica de arrastrar y soltar objetos que es muy importante para su funcionamiento. Realizar tests en este script garantiza que los objetos se arrastren y posicionen correctamente, y que las interacciones se manejen correctamente.

7.2. Resultados de los tests

7.2.1. Reconocedores Edit Tests

Nombre de la prueba	IniciarGrabacion_IniciaCorrectamente
Descripción	Verifica que IniciarGrabacion comienza la grabación de audio correctamente.
Entrada	Llamar a IniciarGrabacion
Resultado Esperado	Microphone.IsRecording(null) es true
Resultado	Correcto

Tabla 5 Test de inicialización de la grabación

Nombre de la prueba	DetenerGrabacion_DetieneCorrectamente
Descripción	Verifica que DetenerGrabacion detiene la grabación de audio correctamente.
Entrada	Llamar a IniciarGrabacion, luego DetenerGrabacion
Resultado Esperado	Microphone.IsRecording(null) es false
Resultado	Correcto

Tabla 6 Test de detención de la grabación

Nombre de la prueba	GrabarPalabraClave_IniciaReconocedor
Descripción	Verifica que GrabarPalabraClave inicia el reconocedor de palabras clave.
Entrada	Llamar a GrabarPalabraClave("test")
Resultado Esperado	keywordToDetect es "test"
Resultado	Correcto

Tabla 7 Test reconocedor de palabras clave

7.2.2. Reconocedores Play Tests

Nombre de la prueba	GrabarPalabraClave_DetectaPalabraCorrectamente
Descripción	Verifica que el reconocedor detecta la palabra clave correctamente.
Entrada	Llamar a GrabarPalabraClave("test"), simular KeywordRecognized("test")
Resultado Esperado	La palabra clave es detectada
Resultado	Correcto

Tabla 8 Test de detección de palabras clave

Nombre de la prueba	PararReconocedor_DetieneReconocedorCorrectamente
Descripción	Verifica que PararReconocedor detiene el reconocedor correctamente.
Entrada	Llamar a GrabarPalabraClave("test"), luego PararReconocedor
Resultado Esperado	El reconocedor está detenido (IsRunning es false)
Resultado	Correcto

Tabla 9 Test de parar el reconocedor

7.2.3. Arrastrar imágenes Edit Tests

Nombre de la prueba	SetInOriginalPosition_ResetsPosition
Descripción	Verifica que SetInOriginalPosition restablece la posición original del objeto.
Entrada	Llamar a SetInOriginalPosition
Resultado Esperado	La posición del objeto es igual a originalPosition
Resultado	Correcto

Tabla 10 Test comprobar posición al soltar objeto

Nombre de la prueba	SetEstado_SetsEstadoCorrectly
Descripción	Verifica que SetEstado y GetEstado funcionan correctamente.
Entrada	Llamar a SetEstado y luego a GetEstado
Resultado Esperado	GetEstado devuelve el valor pasado a SetEstado
Resultado	Correcto

Tabla 11 Test comprobar estado al arrastrar objeto

Nombre de la prueba	SetLayer_SetsLayerCorrectly
Descripción	Verifica que SetLayer y GetLayer funcionan correctamente.
Entrada	Llamar a SetLayer y luego a GetLayer
Resultado Esperado	GetLayer devuelve el valor pasado a SetLayer
Resultado	Correcto

Tabla 12 Test comprobar capas

Nombre de la prueba	SetContenedor_SetsContenedorCorrectly
---------------------	---------------------------------------

Descripción	Verifica que SetContenedor y GetContenedor funcionan correctamente.
Entrada	Llamar a SetContenedor y luego a GetContenedor
Resultado Esperado	GetContenedor devuelve el valor pasado a SetContenedor
Resultado	Correcto

Tabla 13 Test para comprobar los contenedores

7.2.4. Arrastrar imágenes Play Tests

Nombre de la prueba	DragAndDrop_Success
Descripción	Verifica que el objeto se pueda arrastrar y soltar correctamente en un área de contención.
Entrada	Simular eventos de arrastrar y soltar en área válida
Resultado Esperado	El objeto está en el área de contención correcta
Resultado	Correcto

Tabla 14 Test arrastrar y soltar

Nombre de la prueba	DragAndDrop_Fail
Descripción	Verifica que el objeto regrese a su posición original si se suelta fuera de un área de contención.
Entrada	Simular eventos de arrastrar y soltar fuera de área válida
Resultado Esperado	El objeto regresa a su originalPosition
Resultado	Correcto

Tabla 15 Test soltar fuera de los contenedores

Capítulo 8

8. Conclusiones

El desarrollo de este Trabajo de Fin de Grado ha permitido crear nuevas actividades educativas enmarcadas en el desarrollo de Pradia y centradas en el aprendizaje de los fonemas. Estas actividades no solo han servido para añadir diversidad a la jugabilidad de Pradia, sino que también han introducido nuevos métodos para involucrar a los usuarios en el proceso de aprendizaje, proporcionando una herramienta para personas con síndrome de Down.

Desde el inicio, uno de los objetivos fundamentales fue la incorporación de tecnologías de reconocimiento de voz en las actividades. Esta tecnología se ha conseguido implementar exitosamente gracias a las posibilidades que ofrece Unity, primeramente, por la opción de aplicarla directamente usando las capacidades propias del motor y también por la cantidad de bibliografía en forma de webs instructivas y foros en los que solucionar los problemas relacionados con la codificación de estas tecnologías en el entorno de Unity.

La utilización de ficheros de configuración para describir el contenido de las actividades ha sido esencial para mantener la escalabilidad del proyecto. Este enfoque ha permitido que las actividades sean fácilmente ampliables con nuevas palabras, imágenes y audios, pudiendo introducirlas tanto en los fonemas existentes como en nuevos fonemas, gracias también a estos archivos de configuración. En este apartado, quiero destacar el uso de la carpeta “Resources” de Unity, que ha permitido realizar todo esto de una manera dinámica y en tiempo de ejecución.

Por otro lado, se han podido desarrollar las actividades en formato individual y en formato conjunto. De esta manera las actividades se pueden extraer para implementarlas de manera externa y a la vez se pueden jugar todas de manera seguidas, para tener una experiencia completa de la aplicación.

La localización de imágenes y archivos de audio con una licencia de uso libre ha sido una consideración importante para garantizar el cumplimiento de las normativas de propiedad intelectual y para mantener la accesibilidad económica del proyecto. Al utilizar los recursos libres

de ARASAAC, se ha asegurado que el juego pueda ofrecer contenido de calidad sin incurrir en costos adicionales, lo que es especialmente relevante en el ámbito educativo.

En cuanto a Unity, este TFG me ha permitido comprender a fondo el funcionamiento del motor y cómo llevar a cabo un proyecto completo en esta plataforma. He adquirido conocimientos prácticos sobre el diseño y desarrollo de actividades interactivas, utilizando herramientas diversas como la creación de animaciones y el motor de físicas integrado. Este aprendizaje ha sido enriquecedor no solo desde un punto de vista técnico, sino también en la gestión de los desafíos que surgen durante el desarrollo de un proyecto, aplicando conceptos teóricos a situaciones prácticas.

Personalmente, esta experiencia me ha permitido mejorar mis habilidades de programación y comprender mejor el proceso de desarrollo de software. El uso de C# y la integración de tecnologías como el reconocimiento de voz han sido desafíos que me han ayudado a crecer como desarrollador. Además, trabajar en un proyecto con un impacto potencialmente positivo en la educación de personas con síndrome de Down ha sido una experiencia gratificante y motivadora.

8.1. Trabajo futuro

Debido a que el TFG es un proyecto que se realiza en un marco de tiempo limitado se presentan muchas vertientes por las que mejorar el trabajo realizado o incluso ampliar ciertas partes:

- **Expansión de contenidos y actividades:** Se podrían desarrollar actividades adicionales que aborden otras áreas del lenguaje, como la comprensión de frases completas o el uso de tiempos verbales. Se podría ampliar el vocabulario, con la integración de más palabras. Esta parte no sería excesivamente complicada, ya que solo implicaría actualizar los ficheros JSON necesarios y añadir las imágenes y los audios.
- **Personalización y adaptabilidad:** Otro aspecto que se podría incluir son perfiles de usuario, dividiendo a los usuarios entre jugadores y revisores. Los revisores podrían tener acceso a los resultados de la aplicación para poder realizar una revisión más exhaustiva y precisa de las habilidades de los usuarios.
- **Tecnología de reconocimiento de voz:** Dado que el reconocimiento de voz es una tecnología que avanza continuamente, se podrían implementar nuevos modelos más

avanzados y personalizados o incluso con la explosión de los modelos de lenguaje (LLM) [\[43\]](#), se podrían añadir estos modelos para aumentar la profundidad de los resultados tomados por la aplicación.

- **Soporte multiplataforma:** Gracias a las posibilidades que ofrece Unity, se podría adaptar la aplicación a dispositivos móviles, tabletas u otros sistemas operativos.

Apéndices

Apéndice A

A. Manual de Despliegue

A.1. Despliegue del desarrollador

Este manual de despliegue proporciona instrucciones detalladas para clonar, configurar y ejecutar el proyecto de Unity. Sigue estos pasos para asegurar que el proyecto se despliegue correctamente.

A.1.1. Clonación del proyecto

- Instalar Git: Asegúrate de tener Git instalado en tu sistema. Puedes descargarlo e instalarlo desde [Git \(git-scm.com\)](https://git-scm.com)
- Clonar el repositorio: Abre una terminal y navega al directorio donde deseas clonar el proyecto. Una vez en el directorio ejecuta el siguiente comando:
 - `git clone -b master https://gitlab.inf.uva.es/jorobis/tfg_jorgeobispotamariz.git`

A.1.2. Configuración en Unity Hub

- Instalar Unity Hub: Descarga e instala Unity Hub desde <https://unity.com/es>.
- Añadir el proyecto a Unity Hub: Abre Unity Hub y pulsa el botón “Add”. Navega hasta el directorio donde clonaste el proyecto y haz clic en “SelectFolder”.
- Seleccionar la versión de Unity: Asegúrate de tener instalada la versión de Unity requerida por el proyecto (**Unity 2021-2-9f1**).
- Una vez instalada la versión correcta y añadido el proyecto ya se puede abrir clicando en el nombre del proyecto.

A.1.3. Configuración del Proyecto en Unity

- Configurar Build Settings: Abre Unity y navega a ‘File -> Build Settings’. Asegúrate que la plataforma de destino esté seleccionada (en este caso PC).
- Asegúrate que todas las escenas necesarias están añadidas en la sección “Scenes in Build”.

A.1.4. Compilación del Proyecto

- En la ventana de Build Settings, selecciona ‘Build’ para compilar el proyecto. Elige un directorio de destino donde se guardará el ejecutable del proyecto.
- Una vez esté compilado, navega al directorio y comprueba que el proyecto funciona correctamente.

A.2. Despliegue del cliente

La aplicación está pensada para ejecutarse en un sistema Windows, Mac o Linux. Para ejecutarla primero hay que seguir la parte de clonación del proyecto definida anteriormente. Una vez clonado simplemente hay que buscar el ejecutable.

Apéndice B

B. Manual de Uso

Este manual está diseñado para guiar a los usuarios a través de las diferentes actividades disponibles en la aplicación.

B.1. Inicio de la aplicación

En la pantalla de inicio, pulsa el nombre del juego para acceder al menú principal. En el menú principal aparecen varias opciones. En los extremos aparece el botón para salir de la aplicación, el botón para ver los créditos y el botón para elegir el fonema con el que se jugará. En el centro están el botón inicio, que da acceso a las actividades de manera continua, y el botón actividades, que permite elegir entre todas ellas.

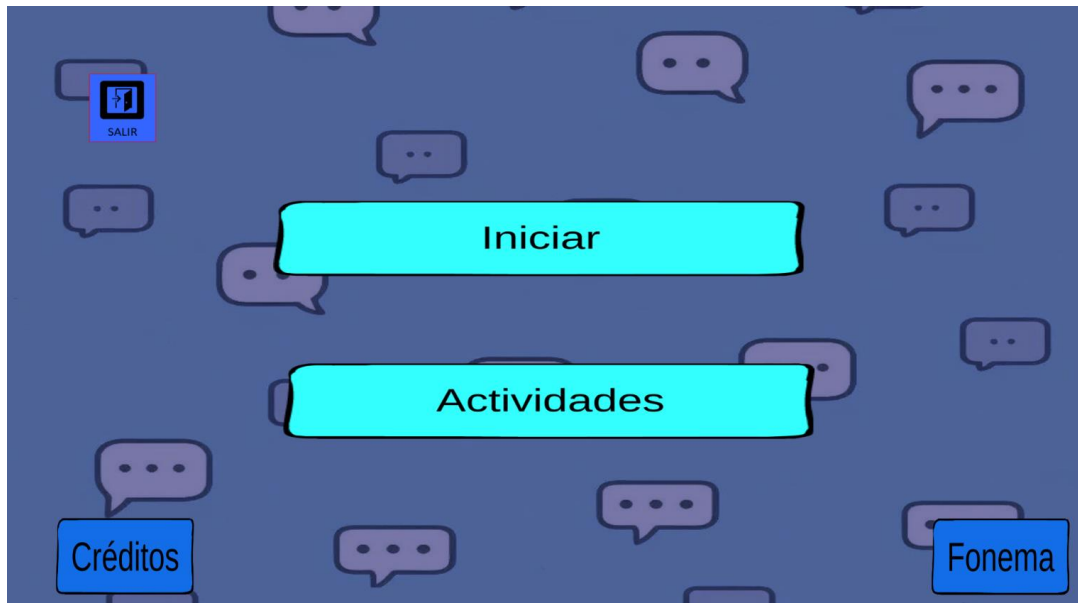


Figura 44 Pantalla de Inicio

Aunque las actividades ya han sido explicadas en el apartado 4 voy a explicar cómo sería el funcionamiento de una partida, pasando por las actividades de manera secuencial.

B.2. Nómbralo

En esta actividad, se presenta una imagen y el usuario debe nombrarla correctamente. Para escuchar el nombre de la imagen pulsar el altavoz y para grabarla, el micrófono. Una vez se pronuncia correctamente aparecerá un mensaje avisando de que se ha acertado. Entonces se procederá a pulsar el botón siguiente, que mostrará la siguiente imagen, volviendo a tener que repetir el proceso.

Además de esto, todas las actividades tienen un botón para volver al menú, en la parte superior izquierda y otro para saltar la imagen en la parte superior izquierda.

Una vez que se han acertado o saltado todas las palabras aparecerá una ventana pop up para saltar a la siguiente actividad.

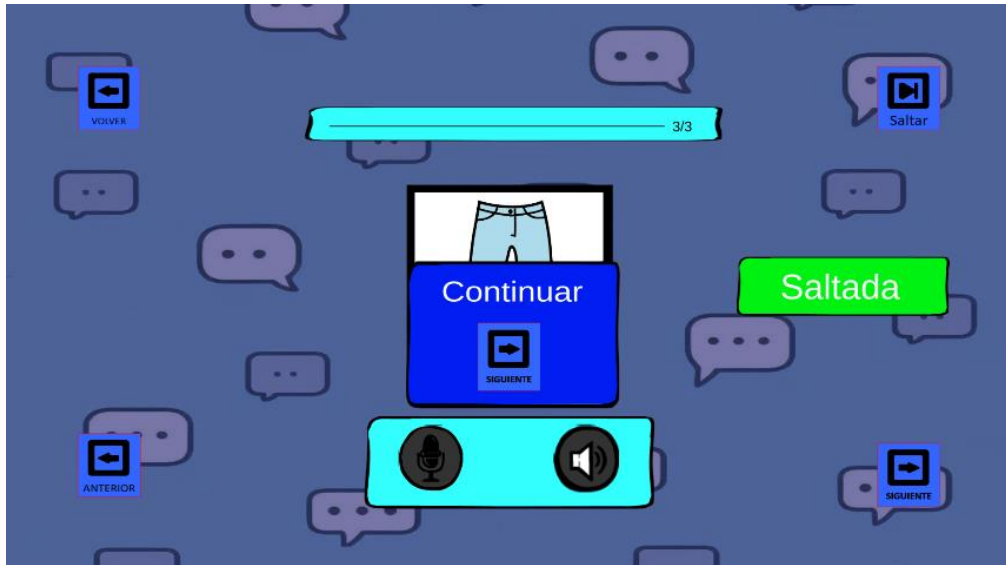


Figura 45 Ventana emergente en "Nómbalo"

B.3. Selección Correcta

En esta actividad hay que seleccionar la imagen correcta basándose en un audio que salta al pulsar el altavoz o el propio texto con el nombre de la palabra. El juego se repite con las seis palabras o hasta que se pulse el botón de saltar.

De la misma manera que en la anterior actividad, aparece una ventana para continuar a la siguiente actividad.



Figura 46 Ventana emergente en "Selección Correcta"

B.4. Discriminación del fonema

Consiste en identificar si un fonema específico está presente en las palabras mostradas. Para ello hay que arrastrar las palabras al recuadro correspondiente. La aplicación dará una retroalimentación en forma de sonido para avisar si se está o no acertando con la colocación de las imágenes.

Para cambiar de actividad a las siguientes, funciona de la misma manera que en las actividades anteriores.

B.5. Parejas

Una vez en esta escena, hay que emparejar las imágenes que representan palabras iguales. Primero hay que esperar dos segundos a que las imágenes se oculten, memorizándolas en este tiempo. Después hay que ir las pulsando hasta que se pulsen dos iguales. En ese momento hay que pulsar el micrófono para pronunciar correctamente la palabra, pudiendo escucharla al pulsarla el altavoz y saltarla si no se sabe pronunciar correctamente.

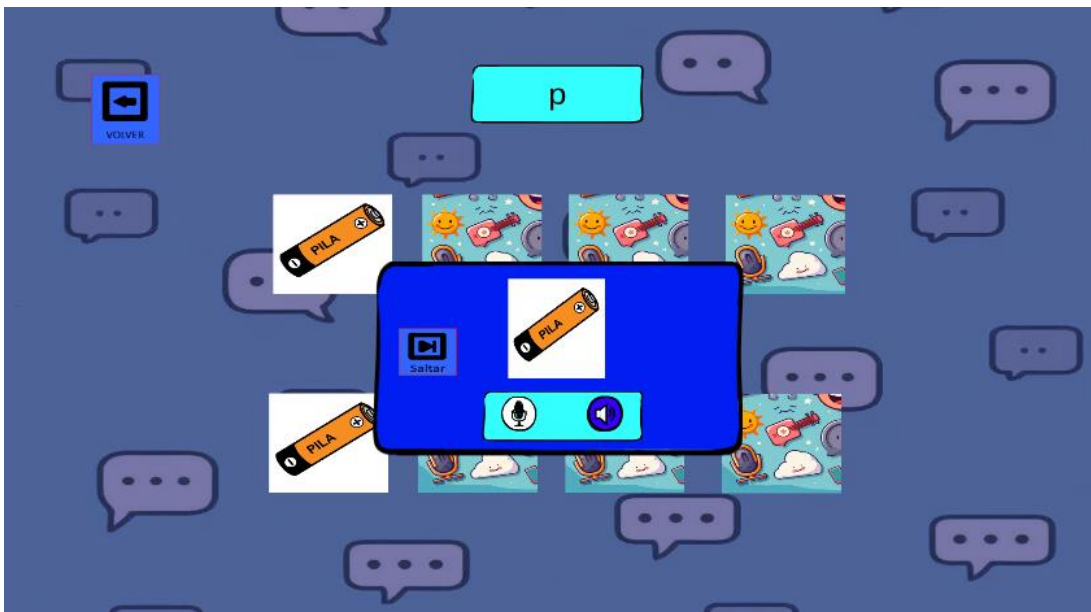


Figura 47 Ventana emergente de grabación de palabras

Igual que en las anteriores actividades, al terminar con todas las imágenes, aparecerá una ventana emergente para pasar a la siguiente actividad.

B.6. Sílabas Correctas

En esta actividad hay que pulsar las sílabas que tengan el fonema indicado en la parte superior de la escena y una vez con la selección hecha hay que pulsar el botón siguiente para comprobarla. Si se pulsan incorrectamente saldrá un mensaje avisando de que se ha fallado en la selección y si se pulsan correctamente, aparecerá la siguiente imagen. En esta escena también se pueden saltar las palabras que no se sepan y al acabar saldrá la misma ventana emergente.

B.7. Pares Mínimos

Hay que comparar la pronunciación de palabras similares que difieren en solo un fonema. Se presentan dos palabras similares y hay que seleccionar la palabra que contiene el fonema indicado. La aplicación proporcionará retroalimentación sobre la elección

B.8. Resultados

Una vez que no queden actividades disponibles aparecerá una escena con las palabras que se han saltado, con un botón para volver y otro para revisar los audios que se han grabado.

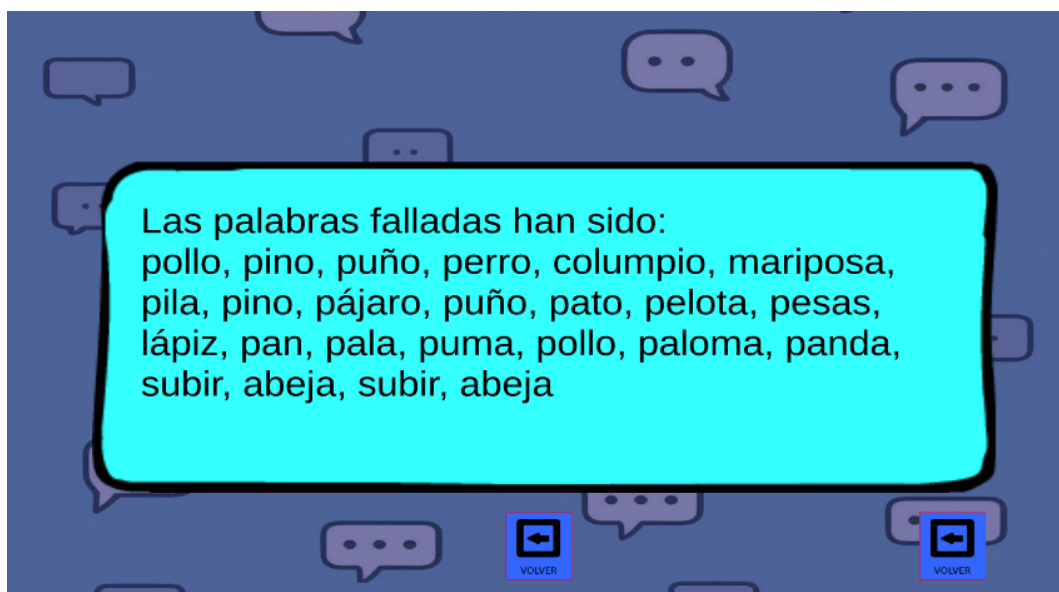


Figura 48 Escena de resultados

Los audios aparecerán separados por palabras en forma de listas, por si una misma palabra ha tenido varias grabaciones.



Figura 49 Escena de reproducción de audios

Apéndice C

C. Contenido del TFG

Además de la memoria, el repositorio de GitLab contiene el código del proyecto y la aplicación en formato ejecutable. El repositorio se encuentra en el siguiente enlace:

- https://gitlab.inf.uva.es/jorobis/tfg_jorgeobispotamariz/-/tree/master?ref_type=heads

Bibliografía

- [1] “Evaluating the impact of an autonomous playing mode in a learning game to train oral skills of users with Down syndrome. David Escudero-Mancebo, Mario Corrales-Astorgano, Valentín Cardeñoso-Payo, César González-Ferreras. IEEE Access. 2021
- [2] “Migración de juego educativo a Unity3D. Gonzalo Fernández González. Trabajo Fin de Grado, Universidad de Valladolid. 2022 <https://uvadoc.uva.es/handle/10324/57271>”
- [3] Jessica Dominguez, Teresa Guarda, Washington Torres, Samuel Bustos, Freddy Villao, “Herramientas Tecnológicas Contribuyendo a la Inclusión de Personas con Necesidades Especiales”, 2019, Revista Ibérica de Sistemas y Tecnologías de la Información. [Online]. Available: [Herramientas Tecnológicas Contribuyendo a la Inclusión de Personas con Necesidades Especiales - ProQuest](#)
- [4] María Eulalia Briones Ponce, Joselyn Johanna Córdova Cedeño, Ángela Monserrate Franco Segovia, "Integración de estudiantes con síndrome de Down de primaria en el uso de herramientas tecnológicas", 2021, Universidad Técnica de Manabí. [Online]. Available: <https://publicacionescd.uleam.edu.ec/index.php/sapientiae/article/view/193/346>
- [5] Proyectos Ágiles, "Desarrollo iterativo e incremental", 2023. [Online]. Available: <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- [6] Wikipedia, "Unity (motor de videojuego)", 2023. [Online]. Available: [https://es.wikipedia.org/wiki/Unity_\(motor_de_videojuego\)#:~:Unity%20es%20un%20motor%20de,Windows%2C%20Mac%20OS%2C%20Linux](https://es.wikipedia.org/wiki/Unity_(motor_de_videojuego)#:~:Unity%20es%20un%20motor%20de,Windows%2C%20Mac%20OS%2C%20Linux)
- [7] Unity Technologies, "Manual de Unity", 2015. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/UnityManual.html>
- [8] Unity Technologies, "Sección de Física", 2015. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/PhysicsSection.html>
- [9] Unity Technologies, "Sección de Animación", 2015. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/AnimationSection.html>
- [10] Unity Technologies, "Sistema de UI", 2015. [Online]. Available: <https://docs.unity3d.com/es/530/Manual/UISystem.html>
- [11] Tokio School, "Lenguaje Unity", 2023. [Online]. Available: <https://www.tokioschool.com/noticias/lenguaje-unity/>
- [12] Unity Technologies, "EdTech Solutions", 2023. [Online]. Available: <https://unity.com/es/solutions/edtech>
- [13] Jhon Steeven Cabanilla Alvarado, “EDUCAVERSE: plataforma para la educación diabética a través de una experiencia compatible con el metaverso,” Universidad de Valladolid, 2023. [Online]. Available: <https://uvadoc.uva.es/handle/10324/62846>.

- [14] Rodrigo Díez Grarcía, "Diseño y desarrollo de juegos educativos 3D en Unity basados en la Industria 4.0 y en herramientas Lean," Universidad de Valladolid, 2022. [Online]. Available: <https://uvadoc.uva.es/handle/10324/59930>.
- [15] CEUR-WS, "Integration of Technologies in Education", 2021. [Online]. Available: <https://ceur-ws.org/Vol-3082/paper3.pdf>
- [16] Little Bee Speech, "Articulation Station," [Online]. Available: https://littlebeespeech.com/articulation_station.php.
- [17] Articulation Teacher, "Articulation Teacher," [Online]. Available: <https://www.articulationteacher.com/>.
- [18] Tip Top Talk, "Tip Top Talk App," Google Play, [Online]. Available: <https://play.google.com/store/apps/details?id=uva.eca.simm.tiptoptalk&hl=es> 419.
- [19] AEAL, "Aula Abierta", 2021. [Online]. Available: <https://aeal.eu/wp-content/uploads/2021/02/Libro-completo-Aula-abierta.pdf>
- [20] AWS, "El reconocimiento de voz impulsado por IA está creando experiencias de aprendizaje interactivas para niños y adultos", 2023. [Online]. Available: <https://aws.amazon.com/es/blogs/aws-spanish/el-reconocimiento-de-voz-impulsado-por-ia-esta-creando-experiencias-de-aprendizaje-interactivas-para-ninos-y-adultos/>
- [21] Astah, "Astah Professional", 2023. [Online]. Available: <https://astah.net/>
- [22] Microsoft, ".NET y C#", 2023. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/csharp/>
- [23] SourceTree, "SourceTree", 2023. [Online]. Available: <https://www.sourcetreeapp.com/>
- [24] Visual Studio Code, "Visual Studio Code", 2023. [Online]. Available: <https://code.visualstudio.com/>
- [25] Talent.com, "Salario para Programador en España", 2024. [Online]. Available: <https://es.talent.com/salary?job=programador#:~:El%20salario%20programador%20promedio%20en,hasta%20E%2%82%AC%2039.874%20al%20a%C3%B1o>.
- [26] ARASAAC, "Pictogramas color," [Online]. Available: http://old.arasaac.org/pictogramas_color.php?busqueda=basico&tipo_palabra=99&TXTIocate=1&letra=lapiz&filtrado=1&orden=desc&Submit=Buscar.
- [27] Unity Technologies, "Resources," [Online]. Available: <https://docs.unity3d.com/ScriptReference/Resources.html>.
- [28] Google Cloud, "Speech-to-Text," [Online]. Available: https://cloud.google.com/speech-to-text/?hl=es&utm_source=google&utm_medium=cpc&utm_campaign=emea-es-all-es-dr-bkws-all-all-trial-e-gcp-1707574&utm_content=text-ad-none-any-DEV_c-CRE_593880918158-ADGP_Hybrid+%7C+BKWS+-+EXA+%7C+Txt+-+AI+And+Machine+Learning+-+Speech+to+Text+-+v1-KWID_43700053288209417-kwd-21425535976-userloc_20298&utm_term=KW_google%20speech%20to%20text-NET_g-PLAC_&&gad_source=1&gclid=EAlaIQobChMIraq_o8blhgMV-RoGAB03FALnEAAAYASAAEgLhkfd_BwE&gclsrc=aw.ds.
- [29] Mozilla, "DeepSpeech," [Online]. Available: <https://github.com/mozilla/DeepSpeech>.

- [30] CMUSphinx, “CMUSphinx Open Source Speech Recognition Toolkit,” [Online]. Available: <https://cmusphinx.github.io/>.
- [31] Wit.ai, “Wit.ai Natural Language Platform,” [Online]. Available: <https://wit.ai/>.
- [32] Macoron, “Whisper Unity,” GitHub, [Online]. Available: <https://github.com/Macoron/whisper.unity>.
- [33] Babilinski, “Vosk STT Unity,” GitHub, [Online]. Available: <https://github.com/Babilinski/vosk-stt-unity>.
- [34] Unity Technologies, “BoxCollider2D,” [Online]. Available: <https://docs.unity3d.com/es/2019.4/Manual/class-BoxCollider2D.html>.
- [35] Unity Technologies, “Rigidbody2D,” [Online]. Available: <https://docs.unity3d.com/es/2019.4/Manual/class-Rigidbody2D.html>.
- [36] Unity Technologies, “Renderer.sortingOrder,” [Online]. Available: <https://docs.unity3d.com/ScriptReference/Renderer-sortingOrder.html>.
- [37] J. Llisterri, “El inventario de fonemas consonánticos del español peninsular,” [Online]. Available: https://joaquimllisterri.cat/phonetics/fon_esp/fonologia_espanol_sistema_respuestas.html.
- [38] Refactoring Guru, “Singleton Pattern,” [Online]. Available: <https://refactoring.guru/es/design-patterns/singleton>.
- [39] Refactoring Guru, “Observer Pattern,” [Online]. Available: <https://refactoring.guru/es/design-patterns/observer>.
- [40] Refactoring Guru, “Facade Pattern,” [Online]. Available: <https://refactoring.guru/es/design-patterns/facade>.
- [41] R. Nystrom, “Update Method,” Game Programming Patterns, [Online]. Available: <https://gameprogrammingpatterns.com/update-method.html>.
- [42] R. Nystrom, “Game Loop,” Game Programming Patterns, [Online]. Available: <https://gameprogrammingpatterns.com/game-loop.html>.
- [43] Cloudflare, “What is a Large Language Model?,” [Online]. Available: <https://www.cloudflare.com/es-es/learning/ai/what-is-large-language-model/>.