



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Vila Romana: videojuego multiplataforma para el  
aprendizaje de vocabulario en lenguas extranjeras

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Espinoza Faubla, Eduardo Andres

Tutor/a: Mollá Vayá, Ramón Pascual

Director/a Experimental: LARCHEN COSTUCHEN, ALEXIA

CURSO ACADÉMICO: 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

## Mnemo Mission

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Eduardo Andrés Espinoza Faubla

**Tutor:** Mollá Vayá, Ramón Pascual

**Director Experimental:** Larchen Costuchen, Alexia

2021-2022



# Resumen

---

Este trabajo describe el proceso de desarrollo del videojuego Mnemo Mission cuyo fin reside en el aprendizaje del léxico de una segunda lengua (L2) o una lengua extranjera (LE) haciendo uso de la técnica mnemónica conocida como el método loci para facilitar el recuerdo de información y de entornos inmersivos gracias al empleo de la realidad virtual.

**Palabras clave:** realidad virtual, aprendizaje de idiomas, método loci, videojuegos, mnemotécnicas.

# Abstract

---

This paper describes the development process of the video game Mnemo Mission, which aims at learning the lexicon of a second or foreign language using the mnemonic technique known as the loci method to facilitate the recall of information and immersive environments through the use of virtual reality.

**Keywords :** virtual reality, language learning, loci method, videogames, mnemonics.

# Índice de contenidos

---

1.	Introducción .....	8
1.1	Motivación.....	9
1.2	Objetivos .....	9
1.3	Estructura .....	10
1.4	Convenciones.....	11
2.	Estado del arte .....	12
2.1	Aplicaciones similares .....	13
2.2	Propuesta.....	16
3.	Análisis del problema.....	17
3.1	Modelo de dominio.....	17
3.2	Características del sistema.....	18
3.3	Análisis de requisitos .....	19
3.4	Análisis de soluciones posibles.....	28
	Motor de videojuego .....	28
	Sistema de gestión de bases de datos .....	31
	Dispositivo de RV.....	32
3.5	Herramientas seleccionadas.....	34
	Motor de videojuego .....	35
	Dispositivo de RV.....	35
	Sistema de gestión de bases de datos .....	35
3.6	Plan de trabajo .....	35
4.	Diseño de la solución .....	37
4.1	Arquitectura del sistema.....	37
4.2	Diseño detallado.....	39
4.3	Tecnología utilizada .....	44
5.	Desarrollo de la solución .....	47
6.	Conclusiones .....	49
6.1	Relación del trabajo desarrollado con los estudios cursados .....	50
7.	Trabajos futuros.....	51
	Referencias.....	52

Apéndices.....	54
A. UNITY.....	54
B. CONEXIÓN CON FIREBASE .....	59
C. SDK GOOGLE CARDBOARD.....	63
D. GAME DOCUMENT DESIGN .....	65
ANEXO .....	74



# Índices de figuras

---

Figura 1. Instantánea del videojuego Mondly VR .....	14
Figura 2. Instantánea del videojuego Noun Town .....	15
Figura 3. Instantánea de la aplicación ImmerseMe .....	15
Figura 4. Modelo de dominio de Mnemo Mission .....	18
Figura 5. Google Cardboard .....	32
Figura 6. Oculus Quest 2 .....	33
Figura 7. HP Reverb G2.....	33
Figura 8. Sony PlayStation VR .....	34
Figura 9. Tablero Kanban empleado. ....	36
Figura 10. Arquitectura por capas del sistema .....	38
Figura 11. Clases identificadas.....	39
Figura 12. Clases en el componente MenuPrincipal .....	40
Figura 13. Clases en el componente Nivel .....	41
Figura 14. Clases en el componente Prueba .....	42
Figura 15. Estructura colección Objetos .....	43
Figura 16. Estructura de la colección Configuraciones .....	43
Figura 17. Estructura de la colección Jugadores .....	44
Figura 18. Recursos disponibles en Snaps .....	45
Figura 19. Interacción con un objeto y muestra del léxico.....	47
Figura 20. Apartado GameObject en el editor de Unity.....	54
Figura 21. Componente Camera en Unity .....	55
Figura 22. Apartado Component en el editor de Unity.....	56
Figura 23. Script C# en Unity.....	56
Figura 24. Agregar proyecto en Firebase .....	59
Figura 25. Registro de aplicación Unity en Firebase.....	60
Figura 26. Importación del SDK de Firebase en Unity .....	61
Figura 27. Script básico para la conexión con Firebase .....	61
Figura 28. Componentes de GameObject Camara para RV .....	63
Figura 29. Instantánea de la escena de ejemplo HelloCardboard .....	64
Figura 30. Diseño inicial nivel 1 - Lexical Office .....	68
Figura 31. Diseño inicial nivel 2 - Lexical Factory.....	69
Figura 32. Pantalla de bienvenida .....	70
Figura 33. Pantalla de inicio de sesión .....	71
Figura 34. Pantalla de registro de nuevo jugador .....	71
Figura 35. Pantalla de selección de configuración de partida.....	71
Figura 36. Pantalla encontrando un objeto.....	72
Figura 37. Pantalla objeto muestra palabra tras interacción .....	72
Figura 38. Pantalla de prueba de traducción .....	72
Figura 39. Pantalla de selección de opción en prueba de traducción .....	73

# Índices de tablas

---

Tabla 1. Comparativa motores de videojuegos .....	30
Tabla 2. Comparativa de sistemas de gestión de bases de datos.....	32
Tabla 3. Comparativa de dispositivos de RV .....	34
Tabla 4. Traducción al húngaro de algunas palabras.....	67
Tabla 5. Ejemplo de prueba ortográfica .....	67
Tabla 6. Ejemplo de prueba de traducción.....	68
Tabla 7. Controles disponibles en Mnemo Mission .....	70



---

# 1. Introducción

---

Aprender una segunda lengua (L2) o una lengua extranjera (LE) es un objetivo compartido por muchos. Pueden ser varios los motivos por los que se quiera estudiar nuevos idiomas como trabajar en el extranjero, viajar por el mundo o conocer otras culturas. Desde los años 50 hasta ahora el enfoque del aprendizaje de lenguas extranjeras ha ido cambiando progresivamente. En un principio predominaba la condición estímulo-respuesta (Muyskens, 1997), un enfoque conductista donde el profesor era la única fuente de conocimiento y los estudiantes, bajo una visión reactiva, aprendían por imitación. Sin embargo, no fue hasta los años 70 y 80 que se produjo un cambio en el paradigma de la enseñanza de lenguas extranjeras. Se produce una transformación hacia una visión cognitiva del aprendizaje donde el lenguaje se adquiere a través del procesamiento de la información, característica inherente a la mente humana. Por lo tanto, el alumno es activo en el proceso de aprendizaje y el profesor actúa como una figura de apoyo o un organizador avanzado.

Desde entonces, se han ido explorando nuevas formas de aprendizaje cuestionando como se están enseñando conceptos del léxico, de la gramática, de la escritura, de la lectura e incluso de la cultura de otros sitios del mundo.

A pesar de ser considerados por muchos “nada más que entretenimiento”, los videojuegos se posicionan como una seria opción para transformar la manera con la que aprendemos (Shaffer, Squire, Halverson, & Gee, 2005). Además de ser una industria que no para de crecer en ingresos (Skyles, 2018), los videojuegos permiten la participación en nuevos mundos en los que los jugadores pueden pensar, comunicarse, actuar e incluso tomar roles inaccesibles fuera de ellos.

La sensación de presencia y de autenticidad determina el grado con el que un jugador transfiere lo que aprende en estos entornos a la vida real (Wang, Petrina, & Feng, 2017; Cheng, Yang, & Andersen, 2017; McLeod, 2007; Krokos, Plaisant, & Varshney, 2019).

La realidad virtual ofrece la posibilidad de sumergirnos en experiencias inmersivas e interactivas que incentivan al aprendizaje autónomo proporcionando espacios de aprendizaje seguros ante errores (Lin & Lan, 2015). Ejemplo de ello son las simulaciones que se llevan a cabo en la educación quirúrgica, donde los estudiantes pueden realizar operaciones sin poner en riesgo a un paciente real (Cheng, Yang, & Andersen, 2017). También lo son cualquier formación que requiera cierta práctica previa como son los simuladores de vuelo, simuladores de entrenamientos policiales o como perciben algunos estudiantes de psicología, simulaciones experimentando sufrir trastornos mentales (McLeod, 2007).

El método loci es una técnica mnemónica cuyo uso es conocido desde la Antigüedad clásica y consiste en formar una estructura mental representada por una serie ordenada de lugares de un ambiente muy conocido por aquel que lo aplica, en la que cada lugar se asocia con uno de los conceptos que se quiere aprender (Yates, 1966; Yates, 1999). La técnica recibe diversos nombres, como "Habitación romana", "Palacio de la memoria" o "Método del viaje". Según Moè & De Beni (2005), la aplicación del método loci implica, en primer lugar, la selección de una serie de lugares distintos a lo largo de un camino conocido; a continuación, la creación de una imagen para cada elemento que se va a recordar o, si se aplica a los pasajes, para cada palabra clave correspondiente a un concepto; y, por último, la colocación de imágenes de los

elementos en los lugares seleccionados y su recuperación mental cuando se recorre el camino en la fase de recuerdo. Como ha podido ser analizado (Huttner & Robra-Bissantz, 2017; Larchen Costuchen, Darling, & Uytman, 2021), en un recorrido de este tipo cualquier objeto simple como una silla, un armario o una mesa puede servir como posiciones loci adecuadas.

Según fuentes recientes (Krokos, Plaisant, & Varshney, 2019; Larchen Costuchen, Mollá Vayá, & Dinkova Dimitrova, 2022), el hecho de organizar la información espacialmente facilita la memorización de información.

Construir un palacio de la memoria generalmente requiere de cierta carga cognitiva, atención y entrenamiento, que pueden ser mitigados proporcionando localizaciones preestablecidas para la vinculación de la información con el espacio (Yang, y otros, 2020).

Tras lo expuesto surge la idea de desarrollar un videojuego de realidad virtual en el que, combinando espacios inmersivos con una técnica para facilitar el recuerdo de información, los jugadores puedan aprender lenguas extranjeras de una forma entretenida, interactiva y efectiva.

## **1.1 Motivación**

---

Los motivos por los que se decide realizar este trabajo surgen de la posibilidad de ofrecer una herramienta software para hacer del aprendizaje de una segunda lengua o lengua extranjera una experiencia más entretenida, inmersiva e interactiva que los métodos tradicionales de enseñanza. Entre ellos destacan:

- Descubrir las diferentes tecnologías que permiten el desarrollo de videojuegos de realidad virtual.
- Conocer las técnicas que facilitan la capacidad de recordar información, como es el método loci.
- Colaborar en la aportación de trabajos relacionados con el aprendizaje de lenguas haciendo uso de videojuegos.

## **1.2 Objetivos**

---

Este trabajo persigue el desarrollo de un videojuego de realidad virtual cuyo fin reside en el aprendizaje de idiomas haciendo uso de una variante de la estrategia mnemotécnica Loci. Se pretende la construcción de lo que se conoce como palacio de la memoria, que en el videojuego representaría un entorno virtual 3D que el jugador irá recorriendo en busca de objetos anacrónicos que, al interactuar con ellos, muestran la palabra a aprender. Finalmente, los conocimientos adquiridos son evaluados a través de una prueba.

- Desarrollar un videojuego de realidad virtual intuitivo y accesible a todo tipo de público.
- Conseguir un producto software a partir del uso de un proceso software con el fin de obtener un producto de calidad y acorde con los estándares de desarrollo.

- Ofrecer un entorno virtual inmersivo e interactivo que incentive al jugador a llevar a cabo un aprendizaje efectivo y entretenido.
- Contar con una amplia variedad de lenguajes y de temáticas.

### 1.3 Estructura

---

La estructura de la memoria está formada por siete capítulos donde se describen todos aquellos aspectos que han sido fundamentales en el proceso de creación del proyecto, un apartado donde se listan las referencias a partir de las cuales se ha extraído conocimiento, un apéndice en el que se amplía la información relativa al contenido mostrado en ciertas secciones y un anexo donde se reflexiona acerca de la relación de este trabajo con los Objetivos de Desarrollo Sostenible. A continuación, se resume brevemente el contenido de cada uno de ellos.

En el apartado “Introducción”, se presentan los motivos y la finalidad que se espera conseguir mediante la realización de este proyecto.

El capítulo “Estado del arte”, muestra la situación del uso de la tecnología en la educación profundizando en el empleo de herramientas inmersivas como es la RV, así como la efectividad de las mnemotécnicas para un mejor recuerdo de la información. También se muestra una descripción de aplicaciones similares con una intención similar al proyecto propuesto a realizar.

En el tercer capítulo llamado “Análisis del problema” se lleva a cabo una descripción de los procesos seguidos para el entendimiento del dominio del problema, se identifican los requisitos que debe tener el sistema acorde a las características y funcionalidades que se quieren ofrecer, se comparan las diferentes herramientas disponibles y cuáles de ellas se han seleccionado en la construcción del proyecto, para finalmente presentar la metodología a seguir para su realización.

La sección número cuatro “Diseño de la solución”, muestra información específica sobre la arquitectura en la que está basada la futura solución, una explicación más detallada de las clases identificadas que formarán el sistema y todas aquellas tecnologías software que han intervenido en la realización de este trabajo.

El capítulo cinco, “Desarrollo de la solución”, describe adversidades y dificultades encontradas mientras se avanzaba en la implementación del sistema y cómo se afrontaron para conseguir la solución final.

En el apartado de “Conclusiones” se analiza la consecución de los objetivos iniciales, se reflexiona sobre aquellos aspectos mejorables y se presenta la relación existente entre el desarrollo de este proyecto con los estudios cursados.

El séptimo y último capítulo llamado “Trabajos futuros”, presenta aspectos que se pueden ampliar para continuar con la implementación de un sistema más completo y avanzado.

Una vez presentados los capítulos relevantes de la memoria, se muestra el listado de “Referencias” que se han empleado para el recabado de información en la que se basa el trabajo. Finalmente, el apartado “Apéndice” complementa con datos extra sobre aspectos mencionados

en algunos de los capítulos de la memoria como son los Game Objects y los distintos componentes empleados en el videojuego en el apartado de “Unity”, el mecanismo que se sigue para poder usar y comunicarse con la base de datos seleccionada en el apartado de “Conexión con Firebase”, las herramientas usadas para hacer posible la experiencia de RV en la sección “SDK Google Cardboard” y por último el apartado “Game Document Design”, un documento de contenidos altamente descriptivos acerca del videojuego Mnemo Mission.

## **1.4 Convenciones**

---

A continuación, se detallan el significado de las convenciones empleadas a lo largo del proyecto.

- RV – Realidad Virtual
- L2 – Segunda Lengua
- LE – Lengua Extranjera
- HMD – Head-Mounted Display
- SDK – herramientas de desarrollo para una plataforma específica
- IU – Interfaz de Usuario

Para el mencionado de referencias a otros trabajos, libros o recursos empleados en la realización del proyecto se ha seguido el formato APA.

---

## 2. Estado del arte

---

Las cinco aplicaciones más populares del aprendizaje de idiomas según el volumen de descargas (Statista, 2022) en el último año han sido Duolingo<sup>1</sup>, Cake<sup>2</sup>, Lingokids<sup>3</sup>, Mondly<sup>4</sup> y Buddy.ai<sup>5</sup>, sin embargo, gran parte de las experiencias del mundo real que hacen del aprendizaje de idiomas una experiencia divertida y entretenida, no están presentes en estos sistemas.

Es por esto por lo que estudiar en el extranjero, se posiciona como una de las opciones más enriquecedoras a la hora de aprender un nuevo idioma, debido a la exposición de la lengua, la asimilación de la cultura y la aplicación de la lengua en contextos reales a la que se enfrenta el estudiante. No obstante, estudiar en el extranjero no es una opción viable para todo el mundo, ya sea por motivos económicos, falta de oportunidades o de tiempo.

Por esto, gracias a la tecnología existente, cabe la posibilidad de contar con herramientas lingüísticas inmersivas que simulan entornos virtuales a los que se puede acceder desde la comodidad del hogar o la escuela, sin requerir de una presencialidad física real.

Los videojuegos se sitúan como una tecnología a partir de la cual conseguir un aprendizaje más ameno. Suelen ser vistos como otra simple forma de entretenimiento obviando el importante papel que puede desarrollar en la educación. Esto se debe a que los jugadores pueden participar en estos nuevos mundos virtuales de muchas maneras, permitiéndoles adoptar roles inaccesibles fuera de estos videojuegos (Shaffer, Squire, Halverson, & Gee, 2005), como ser un soldado combatiendo en la segunda guerra mundial<sup>6</sup> o un cirujano interviniendo a un paciente en una operación<sup>7</sup>.

Los jugadores pueden entender conceptos complejos sin perder la conexión entre las ideas abstractas y los problemas reales que ellos pueden resolver. Además, con los videojuegos surge la posibilidad de que se lleve a cabo un aprendizaje basado en la experimentación, es decir, el jugador puede jugar y probar multitud de opciones sin que esto suponga un riesgo real aparte de las penalizaciones que puede haber dentro del videojuego cuando no se logra un objetivo.

Existen varios estudios en los que se analiza el aprendizaje potenciado por el juego, es decir, aquel aprendizaje que se produce al jugar cualquier juego comercial disponible en el mercado, y el aprendizaje basado en el juego, que aparece con el uso de videojuegos creados específicamente para la enseñanza y el aprendizaje de lenguas (Skyes, 2018). Algunas de las conclusiones que reflejan estas investigaciones incluyen la creación de una comunidad de aprendizaje, la oportunidad de que se produzca un aprendizaje intercultural, el acceso a diversos discursos escritos y hablados y la evidencia de la práctica auténtica de la lengua gracias a las facilidades para los procesos sociocognitivos de aprendizaje y socialización del lenguaje (Skyes, 2018).

---

<sup>1</sup> Duolingo. ¡La forma divertida, efectiva y gratis de aprender un idioma! <https://es.duolingo.com>

<sup>2</sup> Cake. Free English expressions updated every day! <https://mycake.me/>

<sup>3</sup> Lingokids. La Aventura del Aprendizaje Divertido en Inglés. <https://lingokids.com/es/>

<sup>4</sup> Mondly. Aprende idiomas online gratis. <https://es.mondly.com/>

<sup>5</sup> Buddy.ai. El tutor virtual de inglés hablado para niños. <https://buddy.ai/es>

<sup>6</sup> Call of Duty : WWII. Videojuego de disparos en primera persona. <https://www.callofduty.com/es/wwii>

<sup>7</sup> Surgeon Simulator 2. <https://www.surgeonsim.com/>

Es muy probable que la RV pase a formar parte de videojuegos comerciales a medida que la tecnología vaya bajando de precio. El uso de dispositivos de RV de bajo coste conectados a los dispositivos móviles existentes hace posible su aplicación en escuelas con pocos recursos y con un acceso mínimo a otras tecnologías (Skyes, 2018).

La RV puede inducir una poderosa inmersión para el aprendizaje a través de diseños instructivos que imitan la inmersión activa en un lugar utilizando factores sociales y narrativos, así como estímulos sensoriales, de maneras que las pantallas de televisión bidimensionales no pueden (Mills, Courtney, Dede, Dressen, & Gant, 2020). También es importante destacar la sensación de autenticidad y de presencialidad en los entornos virtuales (Wang, Petrina, & Feng, 2017; Cheng, Yang, & Andersen, 2017; Cho, 2018), ya que esto determina el grado con el que el estudiante transfiere lo que aprende en estos mundos digitales a las situaciones del mundo real.

La RV ofrece un espacio para que las personas se muevan y piensen. Ayuda a crear sentido y a externalizar el proceso de razonamiento. Estas propiedades corroboran que la RV puede ser un entorno adecuado para utilizar la mnemotecnica como ayuda para el recuerdo de conocimiento (Yang, y otros, 2020).

Se considera estrategia mnemotécnica a aquel mecanismo utilizado para organizar y/o codificar la información con el único fin de hacerla más memorable (McLeod, 2007). El uso de estas técnicas es muy común en competiciones de memoria, donde los aspirantes crean asociaciones mentales entre ubicaciones físicas y la información a recordar. Esto demuestra el importante papel de la memoria espacial, que se cree que proporciona una estructura para vincular y organizar el conocimiento sobre información no espacial (Reggente, Essoe, Baek, & Rissman, 2020).

Esta asociación visual entre conceptos y espacios es lo que se conoce como el método loci, técnica empleada desde la época clásica por oradores y políticos de la antigua Grecia y Roma, de ahí que la técnica también sea ampliamente conocida como palacios de la memoria (McLeod, 2007).

Construir un palacio de la memoria generalmente requiere de cierta carga cognitiva, atención y entrenamiento, además de que cada individuo tiene referencias espaciales personales que no tienen por qué coincidir con el resto de personas.

Este esfuerzo puede ser contrarrestado con palacios de la memoria ya establecidos, es decir, proporcionando una lista de localizaciones en una historia o una escena 3D virtual, donde se alivia la carga al individuo de asociar la información que se quiere recordar con ubicaciones personalmente íntimas (Yang, y otros, 2020).

Algunas investigaciones (Krokos, Plaisant, & Varshney, 2019; Reggente, Essoe, Baek, & Rissman, 2020) estudiaron el uso de palacios de memoria virtuales junto a dispositivos de RV y su relación con la capacidad de recordar la información dada, concluyendo que con la RV se consigue un mejor recuerdo que cuando se utiliza una pantalla de escritorio tradicional con una interacción basada en teclado y ratón.

## **2.1 Aplicaciones similares**

---



A continuación, se mostrarán un conjunto de aplicaciones las cuales tengan como finalidad el aprendizaje de idiomas.

- **Mondly VR**<sup>8</sup>

Este videojuego ofrece el aprendizaje de hasta 30 idiomas diferentes. Cuenta con reconocimiento de voz y una gran variedad de entornos inmersivos propios de la lengua que se desea aprender, donde el jugador practica el idioma mediante conversaciones basadas en situaciones de la vida real, como una charla en un bar, haciendo una reserva en un hotel o haciendo la compra en un supermercado local. Mondly VR se encuentra disponible en la tienda de Steam<sup>9</sup> tan solo para ordenadores con un precio de 10 dólares.



Figura 1. Instantánea del videojuego Mondly VR

- **Noun Town**<sup>10</sup>

Desde minijuegos que incluyen el léxico de multitud de objetos hasta personajes interactivos con los que mantener una conversación, este videojuego ofrece una experiencia de RV para el aprendizaje de 25 lenguas. Cuenta con reconocimiento de voz y con la correcta pronunciación de las palabras. Su fecha de lanzamiento es en 2022 aunque dispone actualmente de una demo. El videojuego estará disponible tan solo para ordenadores.

<sup>8</sup> Mondly VR. Practice languages in virtual reality. <https://www.mondly.com/vr>

<sup>9</sup> Steam. Steam is the ultimate destination for playing, discussing, and creating games. <https://store.steampowered.com/>

<sup>10</sup> Noun Town. The world's largest VR language learning game. <https://noun.town/>



Figura 2. Instantánea del videojuego Noun Town

- **ImmerseMe<sup>11</sup>**

Esta aplicación software presenta el aprendizaje de 12 idiomas mediante videos grabados en 360 grados de situaciones cotidianas, además de contar con las transcripciones y la pronunciación de las interacciones que se llevan a cabo. Esta herramienta es accesible desde cualquier navegador web permitiendo su uso sin la necesidad de un dispositivo de RV, a pesar de ser compatible con un amplio número de ellos. La licencia para acceder a todo el contenido ronda los 300 dólares anuales.

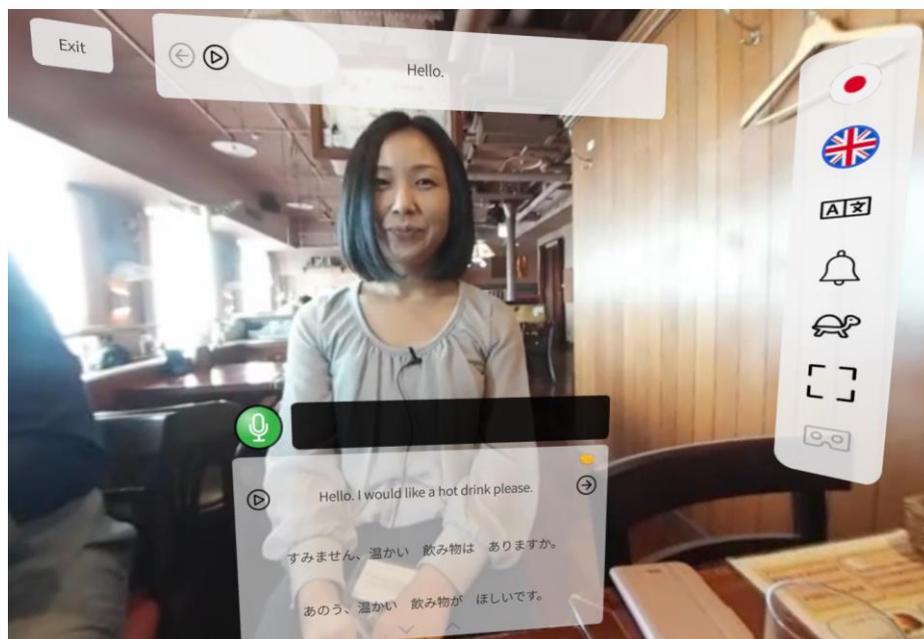


Figura 3. Instantánea de la aplicación ImmerseMe

<sup>11</sup> ImmerseMe. Harnessing the power of virtual reality to immerse users in language learning scenarios. <https://immerseme.co/>

Como se puede observar, es evidente la ausencia de aplicaciones de RV con una intención educativa en el panorama actual, y más en plataformas móviles accesibles a multitud de personas como es Android<sup>12</sup>.

## 2.2 Propuesta

---

Una vez entendidos el potencial uso de los videojuegos en el aprendizaje y como la mnemotecnia puede ayudar a recordar conceptos, se propone el desarrollo de un videojuego de RV en el que los jugadores aprendan el léxico de una segunda lengua (L2) o una lengua extranjera (LE) mediante el método loci.

El videojuego contiene objetivos lúdicos como son el movimiento por los diferentes entornos en busca de las tarjetas escondidas en los objetos y la superación de niveles de dificultad, así como objetivos formativos que incluyen la memorización de léxico en la L2 o LE que se va a aprender.

Se pretende conseguir una herramienta accesible al mayor número de personas por lo que el uso de tecnologías que implementan RV de bajo coste será un aspecto importante por considerar.

Se planea recoger datos cualitativos y cuantitativos para realizar un análisis de los resultados. Los posibles participantes para el estudio piloto son individuos en el rango de edad entre 18 y 60 años.

Una de las principales cuestiones que intenta responder el diseño de este videojuego es si realmente la RV facilita la retención de léxico desconocido en L2 o en LE comparado con alternativas más tradicionales analógicas.

---

<sup>12</sup> Android. La plataforma que amplía los límites de lo posible. [https://www.android.com/intl/es\\_es/](https://www.android.com/intl/es_es/)

---

## 3. Análisis del problema

---

La base principal con la que nace la idea del juego es la de ir descubriendo alrededor de todo el mapa diferentes objetos que al accionarlos muestran una palabra con la traducción tanto en el idioma nativo del jugador como en el idioma que se ha escogido aprender. Posteriormente, se realiza una prueba al jugador con el fin de comprobar que el aprendizaje del léxico de una L2 o LE ha sido exitoso.

Para el estudio estadístico que se pretende hacer, también es importante llevar un registro de ciertos aspectos de las partidas de los jugadores, como son los instantes de tiempo en el que se localiza un objeto, el tiempo empleado para la memorización de las palabras, las palabras que no ha sido capaz de recordar o la puntuación obtenida en una prueba.

### 3.1 Modelo de dominio

---

Una vez entendida la idea principal del videojuego, se puede elaborar un modelo que representa conceptualmente al sistema software que se pretende desarrollar.

Los modelos en la fase análisis son apropiados para entender el dominio del problema y facilitar la comunicación. Permiten compartir una visión y un conocimiento común entre desarrolladores, clientes y cualquier otro interesado. Además, los modelos hacen que la planificación del proyecto sea más efectiva, al proporcionar una perspectiva más apropiada del sistema que se va a desarrollar y permitiendo alcanzar las metas propuestas en base a criterios objetivos (Da Silva, 2015).

A continuación, se muestra el modelo de dominio elaborado a partir de la idea principal, no obstante, mientras se va a construyendo el proyecto puede darse el caso de que conozcamos nueva información acerca del dominio o que no se hayan tenido algunos puntos en consideración, que conlleva una evolución del modelo inicial (Sommerville & Sawyer, 1997).

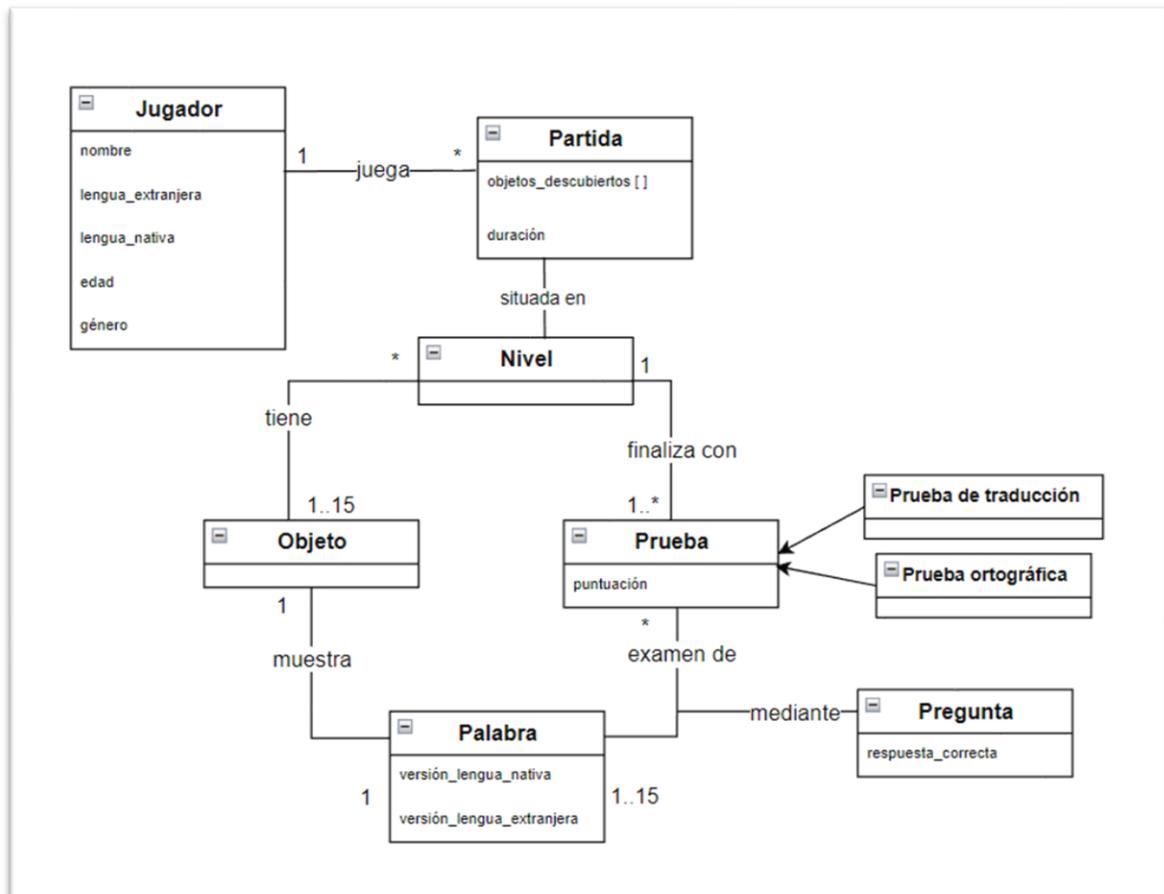


Figura 4. Modelo de dominio de Mnemo Mission

### 3.2 Características del sistema

Conociendo los elementos que van a intervenir en el dominio del proyecto, ahora se describen las principales características detectadas que ofrecerá el videojuego y que posteriormente se verán con más detalle en el análisis de requisitos.

- Gestión de movimiento. Funcionalidades que permiten al jugador moverse por todo el palacio de la memoria.
- Gestión de controles del jugador. Funcionalidades que permiten al jugador interactuar con elementos de la partida.
- Gestión de partida. Funcionalidades referentes al desarrollo de la partida.
- Gestión de prueba del conocimiento. Funcionalidades referentes al desarrollo de una prueba.
- Gestión de preferencias del jugador. Funcionalidades que permiten establecer los ajustes con los que se va a jugar la partida.
- Administración de usuarios. Funcionalidades que permiten el alta, baja o modificación de datos del usuario.

- Gestión de historial de partidas de jugador. Apartado donde se muestran las últimas partidas del jugador con la puntuación, preguntas contestadas incorrectamente, idiomas escogidos, duración.
- Gestión de datos. Interfaz accesible a todo elemento del videojuego que ofrece las funciones CRUD con la base de datos.

### 3.3 Análisis de requisitos

Para la elicitación de requisitos se hace uso de la técnica historias de usuario que se caracteriza por estar focalizada en el usuario y consiste en la descripción del comportamiento esperado del sistema mediante frases simples. Cada historia de usuario representa una funcionalidad o requisito a implementar donde aparece información relativa al actor, la acción que puede llevar a cabo, el motivo que le ha llevado a realizar esa acción y un criterio de aceptación con el que se comprobará el correcto funcionamiento del código desarrollado.

Esta técnica sigue una estructura dada por un tema, que describe el sistema o subsistema al que pertenece el requisito; una epic, que representa aquella característica de alta granularidad a la que pertenece el requisito; una historia de usuario, donde se produce una descripción simple del requisito; y unas tareas que se obtienen de la descomposición de la funcionalidad en unidades de trabajo.

#### MENÚ PRINCIPAL – PREFERENCIAS DEL JUGADOR

<b>PJ 1</b>	Elección de la temática relacionada con las palabras que se irán descubriendo en la partida	
Como jugador quiero elegir la temática relacionada con las palabras que voy a aprender para conocer más vocabulario sobre esa temática.		
<b>Pruebas de aceptación</b>		
<b>PPJ 1.1</b>	<b>Descripción</b>	Se elige la temática de las palabras que se van a aprender
	<b>Precondición</b>	Haber iniciado sesión como jugador
	<b>Pasos</b>	1. El jugador selecciona la opción de elegir temática 2. El jugador selecciona una temática diferente a la actual
	<b>Resultado esperado</b>	Se guarda la nueva temática escogida

<b>PJ 2</b>	Elección de la lengua materna y de la lengua extranjera a aprender	
Como jugador quiero elegir la lengua materna y la lengua extranjera que quiero aprender para que las palabras que aparezcan se adapten a mis preferencias de idioma. No se puede escoger como lengua materna aquel idioma seleccionado como lengua extranjera y viceversa.		
<b>Pruebas de aceptación</b>		
<b>PPJ 2.1</b>	<b>Descripción</b>	El jugador elige la lengua materna y la lengua extranjera
	<b>Precondición</b>	Haber iniciado sesión como jugador
	<b>Pasos</b>	1. El jugador selecciona la opción de elegir lengua materna 2. El jugador selecciona una lengua que aparece en la lista de idiomas



		3. El jugador selecciona la opción de elegir lengua extranjera 4. El jugador selecciona una lengua que aparece en la lista de idiomas
	<b>Resultado esperado</b>	Se guarda la nueva selección de lenguas del jugador

<b>PJ 3</b>	Elección de una configuración previa para iniciar una partida, es decir, la partida tendrá los mismos objetos y sus posiciones dentro del entorno que una partida jugada anteriormente	
Como jugador quiero elegir una configuración previa de los objetos y sus posiciones dentro del entorno para volver a practicar el léxico de una partida determinada.		
<b>Pruebas de aceptación</b>		
<b>PPJ 3.1</b>	<b>Descripción</b>	Se elige una configuración previa y cuando se inicia la partida, el nivel contiene los objetos y sus posiciones guardados en la configuración
	<b>Precondición</b>	Haber generado una configuración mediante el método aleatorio
	<b>Pasos</b>	1. El jugador selecciona una configuración 2. Se inicia la partida con el primer nivel
	<b>Resultado esperado</b>	Los objetos y sus posiciones en el entorno son los que estaban recogidos en la configuración

<b>PJ 4</b>	Elección de una configuración aleatoria para iniciar una partida, es decir, se generarán aleatoriamente los objetos y sus posiciones que aparecerán en la partida	
Como jugador quiero elegir una configuración aleatoria de los objetos y sus posiciones dentro del entorno para practicar nuevo léxico.		
<b>Pruebas de aceptación</b>		
<b>PPJ 4.1</b>	<b>Descripción</b>	Se elige una configuración aleatoria y cuando se inicia la partida, el nivel contiene los objetos y sus posiciones generados aleatoriamente
	<b>Precondición</b>	
	<b>Pasos</b>	1. El jugador selecciona una configuración aleatoria 2. Se inicia la partida con el primer nivel
	<b>Resultado esperado</b>	Los objetos y sus posiciones en el entorno son generados aleatoriamente

### MENÚ PRINCIPAL – HISTORIAL DE PARTIDAS DEL JUGADOR

<b>HPJ 1</b>	Visualización del historial de partidas del jugador
Como jugador quiero ver el historial de partidas anteriores con información de la temática, el lenguaje extranjero y la puntuación total obtenida.	
<b>Pruebas de aceptación</b>	

<b>PHPJ 1.1</b>	<b>Descripción</b>	Se muestra un historial de las partidas del jugador
	<b>Precondición</b>	Haber iniciado sesión como jugador, tener al menos una partida jugada
	<b>Pasos</b>	1. El jugador accede al menú principal con un usuario registrado 2. El jugador comienza una nueva partida 3. El jugador completa la partida y vuelve al menú principal
	<b>Resultado esperado</b>	Se muestra un historial de partidas del jugador con información de la temática, el lenguaje extranjero y la puntuación total obtenida

### MENÚ PRINCIPAL – ADMINISTRACIÓN DE USUARIOS

<b>AU 1</b>	Registro de jugador	
Como jugador quiero introducir mi nombre para registrar una nueva cuenta de jugador		
<b>Pruebas de aceptación</b>		
<b>PAU 1.1</b>	<b>Descripción</b>	Se registra la nueva cuenta del jugador en el sistema
	<b>Precondición</b>	
	<b>Pasos</b>	1. El jugador inicia el juego 2. El jugador introduce un nombre en el campo de registro
	<b>Resultado esperado</b>	Se registra la cuenta del nuevo jugador en el sistema
<b>PAU 2.1</b>	<b>Descripción</b>	Al tratarse de un nuevo jugador, se muestra los menús que permiten la selección de lenguajes y de la temática de las palabras
	<b>Precondición</b>	PAU 1.1
	<b>Pasos</b>	1. El jugador escoge una lengua como materna que aparece en la lista de idiomas 2. El jugador escoge una lengua como extranjera que aparece en la lista de idiomas 3. El jugador escoge la temática de las palabras
	<b>Resultado esperado</b>	Se inicia una nueva partida

<b>AU 2</b>	Inicio de sesión de jugador	
Como jugador quiero seleccionar mi cuenta de jugador para acceder con mis preferencias y ver mi historial de partidas.		
<b>Pruebas de aceptación</b>		
<b>PAU 2.1</b>	<b>Descripción</b>	Se inicia sesión con una cuenta registrada en el sistema
	<b>Precondición</b>	Existe una cuenta de jugador en el sistema
	<b>Pasos</b>	1. El jugador inicia el juego 2. El jugador selecciona una cuenta de jugador
	<b>Resultado esperado</b>	Se accede al menú principal donde aparecen las preferencias y el historial de partidas del jugador

**JUGADOR – GESTIÓN DE MOVIMIENTO JUGADOR**

<b>GMJ 1</b>	Movimiento del jugador por el entorno	
Como jugador necesito moverme por el entorno para explorar los diferentes objetos repartidos por el palacio de la memoria.		
<b>Pruebas de aceptación</b>		
<b>PGMJ 1.1</b>	<b>Descripción</b>	Al accionar el joystick con una posición hacia arriba, el jugador se desplaza hacia delante en la dirección a la que apunta su cabeza
	<b>Precondición</b>	Estar dentro de una partida
	<b>Pasos</b>	1. Accionar el joystick con una posición hacia arriba
	<b>Resultado esperado</b>	El jugador se desplaza hacia delante en la dirección a la que apunta su cabeza
<b>PGMJ 1.2</b>	<b>Descripción</b>	Al accionar el joystick con una posición hacia abajo, el jugador se desplaza hacia atrás en la dirección a la que apunta su cabeza
	<b>Precondición</b>	Estar dentro de una partida
	<b>Pasos</b>	1. Accionar el joystick con una posición hacia abajo
	<b>Resultado esperado</b>	El jugador se desplaza hacia atrás en la dirección a la que apunta su cabeza
<b>PGMJ 1.3</b>	<b>Descripción</b>	Al accionar el joystick con una posición hacia la izquierda, el jugador se desplaza hacia la izquierda de la dirección a la que apunta su cabeza
	<b>Precondición</b>	Estar dentro de una partida
	<b>Pasos</b>	1. Accionar el joystick con una posición hacia la izquierda
	<b>Resultado esperado</b>	El jugador se desplaza hacia la izquierda de la dirección a la que apunta su cabeza
<b>PGMJ 1.4</b>	<b>Descripción</b>	Al accionar el joystick con una posición hacia la derecha, el jugador se desplaza hacia la derecha en la dirección a la que apunta su cabeza
	<b>Precondición</b>	Estar dentro de una partida
	<b>Pasos</b>	1. Accionar el joystick con una posición hacia la derecha
	<b>Resultado esperado</b>	El jugador se desplaza hacia la derecha de la dirección a la que apunta su cabeza

<b>GMJ 2</b>	Relación de los movimientos de la cámara del jugador con los de su cabeza
Como jugador necesito que la dirección a donde apunte mi cabeza coincida con la dirección a donde apunta la cámara en el videojuego para tener una experiencia de realidad virtual inmersiva.	

**JUGADOR – GESTIÓN DE CONTROLES JUGADOR**

<b>G CJ 1</b>	Interacción del jugador manteniendo la vista sobre un elemento
Como jugador necesito poder mantener la mirada sobre un elemento durante 2,5 segundos para interactuar con dicho elemento, ya sea una opción en la prueba o un objeto en partida. Es sumamente aconsejable contar con un pequeño elemento visual ubicado en el centro de la	

pantalla para ayudar al jugador en la coordinación de la vista con el elemento.		
<b>Pruebas de aceptación</b>		
<b>PGCJ 1.1</b>	<b>Descripción</b>	Al mantener la mirada sobre un objeto en partida durante 2,5 segundos se muestra la palabra a aprender sobre el objeto
	<b>Precondición</b>	Estar en una partida
	<b>Pasos</b>	1. El jugador se encuentra frente a un objeto 2. El jugador centra su mirada con el objeto durante 2,5 segundos
	<b>Resultado esperado</b>	Tras una animación que indica que se está produciendo la interacción, se muestra la palabra a aprender sobre el objeto descubierto
<b>PGCJ 1.2</b>	<b>Descripción</b>	Al mantener la mirada sobre una de las respuestas en la prueba durante 2,5 segundos el jugador está seleccionando su respuesta a la pregunta
	<b>Precondición</b>	Estar en una prueba
	<b>Pasos</b>	1. El jugador centra su mirada con la opción que considera la respuesta correcta a la pregunta
	<b>Resultado esperado</b>	El jugador hace efectiva su selección de la respuesta a la pregunta

<b>G CJ 2</b>	Interacción del jugador mediante el mando controlador	
Como jugador necesito poder recorrer las diferentes opciones de menús mediante el mando para hacer más sencilla la interacción con los menús.		
<b>Pruebas de aceptación</b>		
<b>PGCJ 2.1</b>	<b>Descripción</b>	Con el joystick el jugador puede desplazarse por las diferentes opciones del menú en el que nos encontremos
	<b>Precondición</b>	Estar en un menú
	<b>Pasos</b>	1. El jugador acciona el joystick hacía la dirección de aquello que se quiere seleccionar
	<b>Resultado esperado</b>	Lo seleccionado corresponde con lo esperado
<b>PGCJ 2.2</b>	<b>Descripción</b>	Pulsando el botón X el jugador puede confirmar la selección sobre la que nos encontremos
	<b>Precondición</b>	Estar en un menú
	<b>Pasos</b>	1. El jugador recorre con el joystick las opciones 2. El jugador se detiene en la opción que quiere confirmar 3. El jugador pulsa el botón X
	<b>Resultado esperado</b>	Se confirma la selección de la opción escogida
<b>PGCJ 2.3</b>	<b>Descripción</b>	Pulsando el botón O el jugador puede volver hacia atrás en la navegación entre escenas
	<b>Precondición</b>	Estar en un menú
	<b>Pasos</b>	1. El jugador pulsa el botón O
	<b>Resultado esperado</b>	Se vuelve a la anterior escena que hubiese por pantalla, ya sea otro menú o la partida en el caso de encontrarse en

	pausa
--	-------

### PARTIDA – GESTIÓN DE PARTIDA

<b>GP 1</b>	Registro de la secuencia de objetos descubiertos	
Como jugador necesito un registro de aquellos objetos que voy descubriendo para contar con la ayuda en la prueba del recorrido explorado. *como observación comentar que este requisito viene dado por el método loci ya que estudio vio mejoras cuando se tenía el recorrido presente* De esta manera el jugador puede ayudarse del método loci al contestar preguntas en la prueba.		
<b>Pruebas de aceptación</b>		
<b>PGP 1.1</b>	<b>Descripción</b>	Cada vez que se descubre un objeto, se ha de llevar un registro por orden de aparición
	<b>Precondición</b>	Estar en una partida
	<b>Pasos</b>	1. El jugador interactúa con un objeto
	<b>Resultado esperado</b>	El objeto es registrado en una colección de objetos

<b>GP 2</b>	Visualización del estado de palabras aprendidas	
Como jugador necesito que aparezca en pantalla el listado de palabras que he aprendido a medida que voy descubriendo objetos para conocer el estado de la partida.		
<b>Pruebas de aceptación</b>		
<b>PGP 2.1</b>	<b>Descripción</b>	Cada vez que se descubre un objeto, se muestra por pantalla un listado de las palabras que van apareciendo
	<b>Precondición</b>	Estar en una partida
	<b>Pasos</b>	1. El jugador interactúa con un objeto
	<b>Resultado esperado</b>	Al cabo de unos segundos, se muestra por pantalla un listado de las palabras descubiertas más la nueva añadida

<b>GP 3</b>	Iniciar la prueba del conocimiento	
Como jugador necesito que se inicie la prueba del conocimiento cuando haya descubierto todos los objetos repartidos por el mapa para poder realizar la prueba que valida si el aprendizaje ha sido efectivo.		
<b>Pruebas de aceptación</b>		
<b>PGP 3.1</b>	<b>Descripción</b>	Una vez es descubierto el último objeto, tras 10 segundos, comienza la prueba
	<b>Precondición</b>	$N - 1$ objetos descubiertos, siendo $N$ el total de objetos
	<b>Pasos</b>	1. Descubrir el último objeto posible 2. Esperar 10 segundos
	<b>Resultado esperado</b>	Inicia la prueba del conocimiento

<b>GP 4</b>	Pausa de partida	
Como jugador necesito que se muestre el menú de opciones cuando se pulsa el botón de pausa para escoger la opción que desee.		

<b>Pruebas de aceptación</b>		
<b>PGP 4.1</b>	<b>Descripción</b>	Al pulsar el botón de pausa, se muestra el menú de opciones y la partida queda en pausa
	<b>Precondición</b>	No estar en el menú de opciones
	<b>Pasos</b>	1. Pulsar el botón de pausa
	<b>Resultado esperado</b>	Se muestra el menú de opciones y la partida queda en pausa

<b>GP 5</b>	Reinicio de partida	
Como jugador necesito poder reiniciar la partida para volver a la posición inicial y empezar de nuevo.		
<b>Pruebas de aceptación</b>		
<b>PGP 5.1</b>	<b>Descripción</b>	En el menú de opciones se selecciona la opción de reiniciar partida y la partida vuelve a comenzar
	<b>Precondición</b>	Estar en el menú de opciones
	<b>Pasos</b>	1. Pausar la partida 2. Seleccionar la opción de reiniciar la partida
	<b>Resultado esperado</b>	Se inicia otra vez la partida desde el principio

<b>GP 6</b>	Activar o desactivar el sonido ambiente de la partida	
Como jugador quiero poder activar o desactivar el sonido ambiente de la partida.		
<b>Pruebas de aceptación</b>		
<b>PGP 6.1</b>	<b>Descripción</b>	En el menú de opciones se selecciona la opción de activar/desactivar el sonido y el jugador al volver a la partida escucha o deja de escuchar el sonido ambiente
	<b>Precondición</b>	Estar en el menú de opciones
	<b>Pasos</b>	1. Pausar la partida 2. Seleccionar la opción de activar/desactivar sonido 3. Salir del menú de pausa y volver a la partida
	<b>Resultado esperado</b>	Se escucha o se deja escuchar el sonido ambiente, según lo que se haya seleccionado

<b>GP 7</b>	Abandonar la partida y salir al menú principal	
Como jugador quiero poder abandonar la partida en la que estoy jugando para salir al menú principal.		
<b>Pruebas de aceptación</b>		
<b>PGP 7.1</b>	<b>Descripción</b>	En el menú de opciones se selecciona la opción de abandonar la partida y el jugador vuelve al menú principal
	<b>Precondición</b>	Estar en el menú de opciones
	<b>Pasos</b>	1. Pausar la partida 2. Seleccionar la opción de abandonar la partida
	<b>Resultado esperado</b>	Aparece el menú principal

## PRUEBA – GESTIÓN DE PRUEBA DEL CONOCIMIENTO



<b>GPC 1</b>	Verificación de respuesta del jugador	
Como jugador necesito que se ilumine de verde la respuesta correcta y en el caso de haber seleccionado una opción incorrecta, se ilumine de rojo para saber si la opción que he seleccionado es la respuesta correcta a la pregunta.		
<b>Pruebas de aceptación</b>		
<b>PGPC 1.1</b>	<b>Descripción</b>	El jugador selecciona la respuesta correcta a la pregunta
	<b>Precondición</b>	Estar en una prueba
	<b>Pasos</b>	1. El jugador selecciona la respuesta correcta manteniendo la vista durante 2,5 segundos sobre ella
	<b>Resultado esperado</b>	Se ilumina de verde la opción seleccionada y se muestra la siguiente pregunta
<b>PGPC 1.2</b>	<b>Descripción</b>	El jugador selecciona una respuesta incorrecta a la pregunta
	<b>Precondición</b>	Estar en una prueba
	<b>Pasos</b>	1. El jugador selecciona una respuesta incorrecta manteniendo la vista durante 2,5 segundos sobre ella
	<b>Resultado esperado</b>	Se ilumina de rojo la opción seleccionada y de verde la respuesta que sí es correcta y se muestra la siguiente pregunta

<b>GPC 2</b>	Cálculo de puntuación total	
Como jugador necesito que se lleve un registro de las respuestas a las preguntas para conocer el recuento de respuestas correctas que llevo en la prueba.		
<b>Pruebas de aceptación</b>		
<b>PGPC 2.1</b>	<b>Descripción</b>	El jugador selecciona la respuesta correcta a la pregunta
	<b>Precondición</b>	Estar en una prueba
	<b>Pasos</b>	1. El jugador selecciona la respuesta correcta manteniendo la vista durante 2,5 segundos sobre ella
	<b>Resultado esperado</b>	Se suma una unidad al recuento de respuestas correctas y se muestra la siguiente pregunta
<b>PGPC 2.2</b>	<b>Descripción</b>	El jugador selecciona una respuesta incorrecta a la pregunta
	<b>Precondición</b>	Estar en una prueba
	<b>Pasos</b>	1. El jugador selecciona una respuesta incorrecta manteniendo la vista durante 2,5 segundos sobre ella
	<b>Resultado esperado</b>	No se suma una unidad al recuento de respuestas correctas y se muestra la siguiente pregunta

<b>GPC 3</b>	Registro de respuestas incorrectas	
Como jugador necesito que se lleve un registro de las respuestas incorrectas a las preguntas para conocer aquellas palabras que me han resultado difícil de aprender.		
<b>Pruebas de aceptación</b>		
<b>PGPC 3.1</b>	<b>Descripción</b>	El jugador selecciona una respuesta incorrecta a la pregunta
	<b>Precondición</b>	Estar en una prueba
	<b>Pasos</b>	1. El jugador selecciona una respuesta incorrecta manteniendo la vista durante 2,5 segundos sobre ella
	<b>Resultado</b>	La palabra que aparece en la pregunta es registrada en una

	<b>esperado</b>	colección de respuestas incorrectas y se muestra la siguiente pregunta
--	-----------------	--

<b>GPC 4</b>	Visualización de resultados de la prueba	
Como jugador quiero ver la puntuación total, la duración y aquellas posibles respuestas incorrectas al acabar la prueba para conocer los resultados de la prueba.		
<b>Pruebas de aceptación</b>		
<b>PGPC 4.1</b>	<b>Descripción</b>	Una vez finalizada la prueba se muestran los resultados de la prueba
	<b>Precondición</b>	Haber completado la prueba
	<b>Pasos</b>	1. El jugador contesta a todas las preguntas de la prueba
	<b>Resultado esperado</b>	Se muestra la puntuación total, la duración y las respuestas incorrectas del jugador en la prueba

<b>GPC 5</b>	Reintentar la prueba	
Como jugador necesito poder reintentar la prueba para obtener mejores resultados en la prueba.		
<b>Pruebas de aceptación</b>		
<b>PGPC 5.1</b>	<b>Descripción</b>	Una vez finalizada la prueba, el jugador tiene la opción de volver a contestar a las preguntas
	<b>Precondición</b>	Haber completado la prueba
	<b>Pasos</b>	1. El jugador contesta a todas las preguntas de la prueba 2. El jugador selecciona la opción de reintentar
	<b>Resultado esperado</b>	Se restablecen el tiempo y la puntuación y se muestra nuevamente la primera pregunta de la prueba al jugador

<b>GPC 6</b>	Volver al menú principal	
Como jugador necesito poder abandonar la prueba para regresar al menú principal.		
<b>Pruebas de aceptación</b>		
<b>PGPC 6.1</b>	<b>Descripción</b>	Una vez finalizada la prueba, el jugador tiene la opción de volver al menú principal
	<b>Precondición</b>	Haber completado la prueba
	<b>Pasos</b>	1. El jugador contesta a todas las preguntas de la prueba 2. El jugador selecciona la opción de volver al menú principal
	<b>Resultado esperado</b>	Se muestra el menú principal

<b>GPC 7</b>	Pasar al siguiente nivel	
Como jugador necesito continuar al siguiente nivel para continuar con la partida.		
<b>Pruebas de aceptación</b>		
<b>PGPC 6.1</b>	<b>Descripción</b>	Una vez finalizada la prueba, el jugador tiene la opción de pasar al siguiente nivel
	<b>Precondición</b>	Haber obtenido una puntuación igual o superior al 60% en la prueba

	<b>Pasos</b>	1. El jugador contesta a todas las preguntas de la prueba, acertando como mínimo el 60% de ellas 2. El jugador selecciona la opción de volver al menú principal
	<b>Resultado esperado</b>	Comienza el siguiente nivel

Una vez identificados los requisitos funcionales del sistema, se detallan los requisitos no funcionales del sistema. Los requisitos no funcionales son el conjunto de restricciones y/o propiedades emergentes del sistema software a desarrollar, caracterizándose por no alterar el comportamiento del producto.

- El sistema tratará todos los datos personales siguiendo la Ley Orgánica de Protección de Datos de Carácter Personal.
- Se deben proporcionar mensajes de error que sean informativos y entendibles para los usuarios finales.
- El tiempo de respuesta de cualquiera de las funcionalidades del sistema debe ser menor a 3 segundos (excepto para la pantalla de carga inicial).
- La aplicación se tiene que poder ejecutar en plataformas que ofrezcan soporte a la RV como son principalmente Windows, Android y iOS.
- Se deben soportar los siguientes lenguajes en la interfaz: español e inglés.
- Las transacciones con la base de datos no deben tardar más de 50ms.
- Los jugadores de Mnemo Mission deben ser capaces de utilizar todas las funcionalidades del sistema tras una hora de entrenamiento.
- El sistema debe desarrollarse aplicando patrones y recomendaciones de programación para facilitar el mantenimiento y la evolución del software.
- El sistema debe continuar operativo pese a la aparición de algún fallo. Todas las excepciones deberán ser capturadas y tratadas, conservando así un estado consistente y operativo.

### 3.4 Análisis de soluciones posibles

---

En este apartado se analizarán las diferentes herramientas disponibles que permiten el desarrollo de un videojuego de RV. Para este proyecto se ha identificado la necesidad de tres tecnologías:

- un motor de videojuegos
- un sistema de gestión de bases de datos
- un dispositivo de RV

A continuación, se mostrarán con más detalle aquellas alternativas posibles de cada tecnología, presentando sus principales características, con el fin de escoger la mejor opción acorde con los requisitos establecidos previamente.

#### Motor de videojuego

Un motor de videojuego es una herramienta software que permite el diseño y la creación de videojuegos. Proporcionan funcionalidades como el renderizado de gráficos en 2D o 3D, el control de interacciones mediante teclado y ratón u otro hardware o el acoplamiento de sonidos y animaciones entre otras (Andrade, 2015).

Debido a la naturaleza del proyecto, este análisis mostrará aquellos motores que permitan el renderizado de gráficos en 3D.

- **Unity<sup>13</sup>**

Soporta tanto la creación de videojuegos en 2D como 3D, lo que permite llevar a cabo el desarrollo de cualquier tipo de videojuego. Proporciona la exportación de videojuegos hasta en 15 plataformas diferentes lo que posiciona a Unity como la opción preferida para el desarrollo de videojuegos en la web hasta proyectos para consolas.

Unity cuenta con un editor visual en el que se pueden construir escenas, editar objetos del videojuego añadiéndoles componentes, crear animaciones o modificar la configuración general del videojuego con tan solo unos clics. Como lenguaje de programación para la creación de scripts, se usa principalmente C#.

Cabe destacar que Unity cuenta con el respaldo de una gran comunidad lo que implica el fácil acceso a documentación o tutoriales. Además, dispone de una tienda de recursos donde se pueden descargar todo tipo de componentes.

- **Unreal Engine<sup>14</sup>**

Inicialmente fue diseñado para la creación de videojuegos FPS y no contaba con la amplia variedad de opciones que ofrecían otros motores como Unity. No obstante, ha evolucionado hasta convertirse en un motor de videojuegos potente capaz de dar soporte para la creación de videojuegos de todo tipo.

Para la creación de scripts, Unreal Engine usa C++, una opción que puede resultar muy interesante para la optimización de recursos, pero un tanto compleja para aquellos desarrolladores no tan avanzados. Asimismo, cuenta con una interfaz basada en nodos para la creación de elementos del videojuego que es descrita como una herramienta tan potente que pueden crearse videojuegos enteros tan solo usando esta característica.

Proporciona la posibilidad de exportación a múltiples plataformas y también cuenta con una comunidad y tienda de recursos relevantes. Mientras que el editor de Unreal Engine puede resultar complicado para aquellos que desarrollen un videojuego por primera vez, la opinión general de la comunidad es que con este motor es posible conseguir resultados visuales mucho más impresionantes que otros motores como Unity.

- **CryEngine<sup>15</sup>**

Mediante el pago de una suscripción, CryEngine ofrece un motor de videojuegos para la creación de gráficos en 3D y la publicación en las principales plataformas y consolas.

---

<sup>13</sup> Unity. <https://unity.com/es>

<sup>14</sup> Unreal Engine. <https://www.unrealengine.com/en-US>

<sup>15</sup> CryEngine. <https://www.cryengine.com/>



Comparado con los anteriores motores presentados, CryEngine probablemente sea el más difícil de aprender puesto que no cuenta con una interfaz intuitiva, ni con una comunidad y una tienda de activos tan grande.

No obstante, Cry Engine es conocido por la producción de videojuegos con gráficos y rendimiento de última generación.

- **Torque Game Engine<sup>16</sup>**

Soporta tanto la creación de videojuegos en 2D como 3D y permite la exportación tan solo en Windows, OSX y iOS. Dispone de un lenguaje propio llamado TorqueScript, muy similar a C++.

La principal característica de Torque es que es un proyecto de código abierto contando así con el respaldo de una cantidad de desarrolladores y de soporte considerables.

- **Turbulenz<sup>17</sup>**

Permite la creación de gráficos en 2D y 3D, y usa JavaScript como lenguaje principal para añadir nuevas funcionalidades. Turbulenz está pensado para el desarrollo y exportación de videojuegos tan solo en navegadores web, además de ser de código abierto.

A continuación, se muestra una tabla en la que se recogen las principales características de cada motor de videojuego que será útil para una posterior comparativa y toma de decisión.

Motor	Soporte renderizado	Lenguaje de programación	Precio	Plataformas de exportación
	2D/3D	C#	Gratuito	Windows; OSX; Linux; PlayStation; Xbox; iOS; Android; Nintendo; BlackBerry; Windows Phone; Navegadores web; etc.
	2D/3D	C++	Gratuito	Windows; OSX; PlayStation; Xbox; iOS; Android; Navegadores web.
	3D	C++	9.90€/mes	Windows; OSX; Linux; PlayStation; Xbox; Wii U; iOS; Android.
	2D/3D	TorqueScript	Gratuito	Windows; OSX; iOS.
	2D/3D	JavaScript	Gratuito	Navegadores web.

Tabla 1. Comparativa motores de videojuegos

<sup>16</sup> Torque Game Engine. <https://torque3d.org/>

<sup>17</sup> Turbulenz. <http://biz.turbulenz.com>

## Sistema de gestión de bases de datos

Existen ciertos datos de los cuales se pretende tener registro, sobre todo los referentes a las partidas de los jugadores, donde aspectos como el tiempo de la partida o la puntuación de las pruebas serán analizados posteriormente. Por ello, se requiere de un SGBD en el que aparte de información estática como son las traducciones de los objetos, también se recojan la información relativa a las partidas de los jugadores para su posterior estudio estadístico.

- **MySQL<sup>18</sup>**

De código abierto y bajo un esquema relacional, MySQL cuenta con una amplia popularidad posicionándose como uno de los SGBD más empleados globalmente. Es por ello que entre sus ventajas destaquen el apoyo de una gran comunidad y la abundante documentación sobre como instalar y configurar una base de datos. Dispone de una arquitectura cliente-servidor que permite la comunicación con múltiples aplicaciones, además de la posibilidad de control y gestión de usuarios que permite definir roles y privilegios.

- **MongoDB<sup>19</sup>**

Se caracteriza por seguir un esquema NoSQL basado en documentos en lugar de tablas y claves foráneas como se hace en las bases de datos relacionales. Los documentos donde se guarda la información siguen una estructura del estilo JSON con un esquema dinámico, lo que permite una integración de los datos más sencilla y veloz en determinados sistemas.

MongoDB Atlas se ofrece como una plataforma global en la nube, dando a los desarrolladores la libertad de ejecutar sus aplicaciones en cualquier lugar con la flexibilidad de mover los datos a través de nubes privadas y públicas a medida que evolucionan los requisitos. No obstante, también es posible usar MongoDB gratuitamente instalándose localmente en un servidor propio.

- **Firebase Firestore<sup>20</sup>**

Firebase es una plataforma en la nube perteneciente al conjunto de servicios de Google Cloud Platform, que cuenta con múltiples herramientas para la gestión de proyectos, como son Firebase Analytics, donde se muestra una visión del uso de la aplicación por parte de los usuarios, Firebase Auth, que tiene la función de facilitar la creación de sistemas de autenticación, o Firebase Firestore, servicio de almacenamiento de datos en la nube.

Al igual que MongoDB, Firebase Firestore sigue un esquema NoSQL basado en documentos agrupados en colecciones. La característica principal de Firebase es que permite la creación de proyectos sin necesidad de un servidor propio, es decir, todas las herramientas de desarrollo (SDK) necesarias para su uso se incluyen dentro de la misma aplicación. Además, en la página de Firebase se encuentra documentación específica para la conexión con Unity.

A continuación, tabla comparativa

---

<sup>18</sup> MySQL. <https://www.mysql.com/>

<sup>19</sup> MongoDB. <https://www.mongodb.com/es>

<sup>20</sup> Firebase Firestore. <https://firebase.google.com/docs/firestore?hl=es-419>



Base de datos	Esquema	Servicio en la nube	Precio
 MySQL	Relacional	No. Servidor local	Gratuito
 mongoDB	Documentos	Si. MongoDB Atlas	Según el uso / Gratuito hasta 512MB
 Firebase	Documentos	Si. Firebase Firestore	Según el uso / Gratuito hasta 1GiB

Tabla 2. Comparativa de sistemas de gestión de bases de datos

## Dispositivo de RV

Esta tecnología también conocida como Head-Mounted Display (HMD) nos permite sumergirnos en mundos virtuales. El grado de inmersión viene determinado por el resto de las tecnologías que incluya el HMD como son sensores de posición, que permiten establecer un espacio en el mundo real donde los movimientos del jugador se verán igualmente reflejados en el videojuego, o los mandos controladores que permiten interactuar con elementos del videojuego.

- **Google Cardboard<sup>21</sup>**

Las Google Cardboard probablemente sean las gafas de RV que permitan con más facilidad el acceso a la tecnología de RV ya que sus materiales de construcción son baratos y las instrucciones para construirlas se encuentran públicamente en línea. Existen kits que incluyen todo lo necesario para montarlas por menos de 6 dólares.



Figura 5. Google Cardboard

Hechas de cartón (Figura 5), cuentan con unos lentes de distancia focal que requieren la pantalla de un smartphone externo para poder acceder al contenido de RV. A pesar de ser un dispositivo ideal para la iniciación en RV, no disponen de controladores externos ni de ningún tipo de tecnología extra, por lo que tan solo garantiza el acceso al apartado visual.

<sup>21</sup> Google Cardboard. <https://arvr.google.com/cardboard/>

- **Oculus Quest 2<sup>22</sup>**

Disponibles desde finales de 2020, estas HMD se posicionan como una de las alternativas más usadas para iniciarse en la RV. Las Oculus Quest 2 son portátiles, por lo que no dependen de ningún otro dispositivo ya sea el PC o un smartphone, aunque dispone de un cable de enlace con el que conectarse al PC y acceder a más experiencias de RV.



*Figura 6. Oculus Quest 2*

Con un precio de entrada de alrededor 300 dólares, este dispositivo cuenta con una pantalla integrada de alta resolución, más de 1080p por ojo, y una alta tasa de refresco, de 90 a 120 Hz, lo que evita la sensación de mareo. Cuenta también con mandos controladores propios y con una cámara integrada que monitoriza el entorno y alerta al jugador en el caso de acercarse a cualquier obstáculo.

- **HP Reverb G2<sup>23</sup>**



*Figura 7. HP Reverb G2*

Este dispositivo cuenta con una de las pantallas integradas de más alta calidad del mercado con una resolución de 2160 por 2160 en cada ojo, además de una tasa de refresco de 90 Hz proporcionando una imagen sobradamente nítida. Cuenta con dos mandos controladores con detección de movimiento y varios botones y gatillos para la interacción. A diferencia de las Oculus Quest 2 y las Google Cardboard, este dispositivo de RV necesita estar permanentemente conectado por cable al ordenador para su uso.<sup>2</sup> Con un precio de entrada entre los 600€ y 700€ se posicionan como una opción de bastante calidad, aunque menos accesible.

---

<sup>22</sup> Oculus Quest 2. <https://store.facebook.com/es/es/quest/products/quest-2/>

<sup>23</sup> HP Reverb G2. <https://www.hp.com/es-es/vr/reverb-g2-vr-headset.html>

- **Sony PlayStation VR<sup>24</sup>**



*Figura 8. Sony PlayStation VR*

Desde 2016 en el mercado, estas gafas son la apuesta de Sony para las experiencias de RV en sus consolas, en un principio para PlayStation 4 y actualmente también compatibles con la quinta generación de dicha consola. Cuentan con una pantalla OLED de 5,7 pulgadas y una resolución de 960 por 1080 en cada ojo y una tasa de refresco máxima de 120 Hz. Necesitan estar conectadas a la consola para su uso y también de dos dispositivos adicionales, una PlayStation Camera para capturar el ambiente que rodea al jugador, así como de un mando controlador, ya sea un DualShock 4 o un PlayStation Move.

A continuación, tabla comparativa:

HMD	Conectividad	Resolución	Controles	Plataforma Software	Precio
Google Cardboard	Inalámbrico	Según dispositivo acoplado	No incluidos	Android, iOS	5€ – 10€
Oculus Quest 2	Inalámbrico	1,832 x 1,920 (por ojo)	Oculus Touch	Oculus, Steam VR	300€-500€
HP Reverb G2	Conectado	2,160 x 2,160 (por ojo)	Motion Controllers	Steam VR	600€
Sony Playstation VR	Conectado	1,080 x 960 (por ojo)	DualShock 4, PlayStation Move	PlayStation	300€

*Tabla 3. Comparativa de dispositivos de RV*

### 3.5 Herramientas seleccionadas

Con este trabajo se pretende ofrecer una herramienta accesible al mayor número de usuarios posible para el aprendizaje del léxico de una L2 o LE, aunque posteriormente se podría ampliar para el aprendizaje de la gramática en L2 o LE. Asimismo, no hace falta contar con la tecnología más puntera del mercado para acceder a experiencias inmersivas de RV, ni con una base de datos demasiado compleja cuando el proyecto no lo requiere.

<sup>24</sup> Sony PlayStation VR. <https://www.playstation.com/es-es/ps-vr/>

## **Motor de videojuego**

Al ser la primera vez que personalmente desarrollo un videojuego, se ha optado por un motor de videojuegos que principalmente disponga de documentación sobre su uso, sea gratuito y permita la exportación a plataformas móviles.

Como puede observarse en la Tabla 1. Comparativa motores de videojuegos aquellos motores que exportan a iOS y Android son Unity y Unreal Engine, además de ser los dos gratuitos. Ambos cuentan con características similares, sin embargo, la abundante documentación y ayuda existente, así como el uso de un lenguaje (C#) que ya había empleado en el pasado, han sido los factores determinantes para la selección de Unity para la creación del videojuego.

## **Dispositivo de RV**

A partir de la Tabla 3. Comparativa de dispositivos de RV se concluye el empleo de las Google Cardboard o cualquier equivalente de plástico, principalmente con la intención de reducir el coste de acceso a este tipo de tecnología y a productos educativos. A diferencia del resto de dispositivos comparados, las Google Cardboard no cuenta con una pantalla integrada lo que implica la necesidad de un dispositivo smartphone a partir del cual mostrar los gráficos del videojuego. Al no contar con mandos controladores, también se requiere de uno externo, preferiblemente con tecnología bluetooth, para proporcionar al jugador los mecanismos de interacción con el videojuego.

## **Sistema de gestión de bases de datos**

Finalmente, se propone el uso de Firebase Firestore para la persistencia de los datos del proyecto ya que cuenta con documentación específica para su integración con Unity y al ser un servicio en la nube, se puede acceder a la información guardada desde cualquier lugar, en cualquier momento, evitando la necesidad de montar un servidor propio para la aplicación. Además, al ser un proyecto de pequeñas dimensiones permite usar el servicio de Firebase gratuitamente con el plan Spark<sup>25</sup> que, aunque esté limitado, proporciona muchas de las funcionalidades de Firebase y en específico los de bases de datos que son los que se usarán en el proyecto.

### **3.6 Plan de trabajo**

---

Se va a llevar a cabo una metodología TDD (Test-driven Development) para el desarrollo software además de un tablero Kanban en el que se irán apuntando las diferentes pruebas a realizar, así como otras tareas relacionadas con la investigación, mejoras, problemas a arreglar. Con esto conseguimos una visión del camino a seguir hasta llegar a la solución final.

La metodología TDD (Beck, 2003) es un proceso de desarrollo software que se basa en 4 pasos. Primero se redactan las pruebas con las que se verifican la validez de las funcionalidades añadidas. Posteriormente se procede a un desarrollo ágil del código con el objetivo de conseguir pasar las pruebas. En este punto, si el código desarrollado no pasa las pruebas previamente

---

<sup>25</sup> Planes de precios de Firebase. <https://firebase.google.com/pricing?hl=es-419>



redactadas se modifica hasta que lo haga. Una vez comprobado que el código desarrollado pasa las pruebas se procede a la refactorización de este. Estos serían los pasos que repetir en el momento de añadir una nueva funcionalidad o requisito que ha de cumplir el videojuego.

Con Kanban se obtiene un método visual para gestionar y procesar las tareas que intervienen en la realización del proyecto, limitar el trabajo en curso y maximizar la eficiencia (Anderson, 2010). Es un proceso para mejorar constantemente el flujo y la calidad del trabajo.



Figura 9. Tablero Kanban empleado.

Como se puede observar en la Figura 9 se ha empleado un tablero Kanban que cuenta con 3 estados posibles de una tarea: por hacer, en progreso, completado. Una tarea puede ser de tipo:

- Característica: cuando se añade una nueva funcionalidad software.
- Mejora: cuando se detecta que una funcionalidad requiere de una mejora.
- Tarea: cualquier otro tipo de tarea relacionada con el proyecto ya sea investigación sobre un tema, revisión de documentos, coordinación con el tutor, etc.

Las tareas también cuentan con otras propiedades como la fecha límite para su finalización, el responsable asignado a la realización de la tarea, así como la prioridad de esta.

---

## 4. Diseño de la solución

---

El diseño software es el proceso que define la arquitectura, componentes, interfaces y otras funcionalidades del sistema acorde a un conjunto de requisitos establecidos (Arias & Durango, 2016).

Con el diseño se obtiene una perspectiva del funcionamiento esperado del sistema una vez desarrollado, ya que permite contrastar el sistema propuesto con los objetivos iniciales, consiguiendo así un aumento de las probabilidades de éxito del proyecto. También, permite la evaluación de diferentes posibles soluciones, puesto que tener una visión clara del dominio del problema proporciona la obtención de soluciones óptimas, libres de complejidades añadidas en fases iniciales del ciclo de vida de un sistema software.

Siguiendo el estándar ISO 12207 (IEEE/EIA Guide for Information Technology - Software Life Cycle Processes - Implementation Considerations, 1998), se muestra el diseño de la solución bajo dos niveles de detalle: la arquitectura del sistema, donde se presentan los subsistemas o componentes en los que se dividirá la solución, y el diseño detallado, donde se describe específicamente cada componente en cuestión.

### 4.1 Arquitectura del sistema

---

La arquitectura de un sistema software es la abstracción a más alto nivel de los subsistemas y componentes que lo forman y cómo se relacionan entre ellos. Así se consigue una perspectiva general de los elementos que intervienen en la solución favoreciendo la comprensión, comunicación y evaluación de los interesados del proyecto.

Entre los diferentes patrones arquitectónicos destaca la arquitectura por capas, también conocida como arquitectura de N niveles, por su amplio uso y popularidad (Richards, 2015). Este patrón se caracteriza por una separación de responsabilidades entre los componentes, así cada capa tiene un rol específico dentro de la arquitectura, y por su facilidad de desarrollo, ya que no es muy complejo de implementar.

Sin embargo, existen algunas desventajas en cuanto a su uso. Este patrón no es indicado para aquellos entornos que cambian constantemente, ya que a pesar de que el cambio puede ser aislado a través de la característica de separación de responsabilidades, sigue siendo engorroso y requiere mucho tiempo hacer cambios debido a la naturaleza monolítica de la mayoría de las implementaciones, así como el estrecho acoplamiento de los componentes.

Esto también provoca una baja escalabilidad y si se cuentan con varios niveles, puede haber problemas en el rendimiento debido a la ineficiencia de pasar por múltiples capas para responder a una petición. Al no contar con experiencia en el desarrollo de videojuegos, se decide optar por este patrón debido a su fácil y rápida implementación.

En este proyecto se cuenta con una arquitectura formada por 3 niveles (véase Figura 10) : la capa de presentación, la capa de lógica de negocio y la capa de datos.

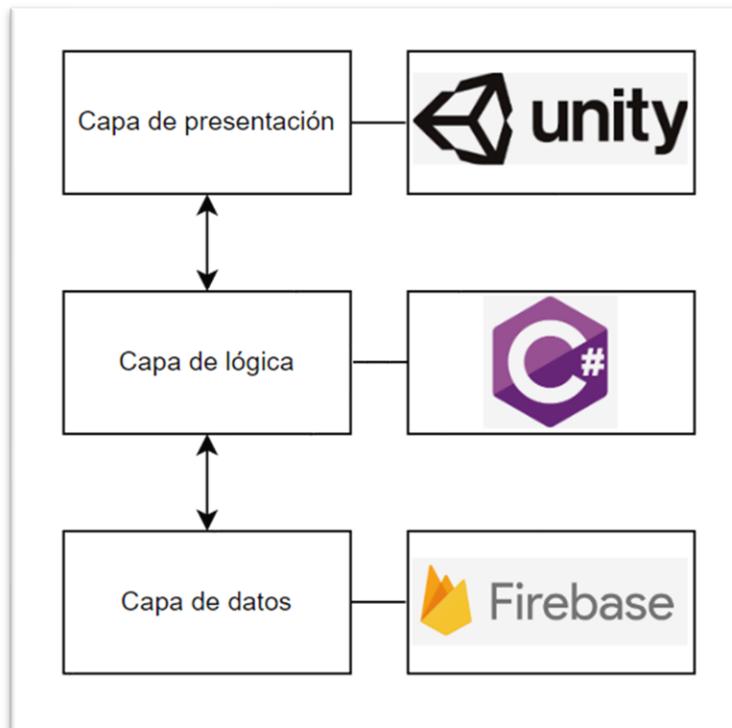


Figura 10. Arquitectura por capas del sistema

La capa de presentación está formada por aquellos componentes que se encargan de mostrar la información por pantalla en un formato específico. Al ser un videojuego, la capa de presentación la componen aquellas escenas y elementos del juego con los que el jugador interactúa. La entidad básica de Unity se conoce como `GameObject`<sup>26</sup>. Una escena en Unity es el contenedor de los `GameObjects` que la componen. Cada `GameObject` cuenta con componentes los cuales definen sus propiedades como la posición y rotación dentro de la escena o el comportamiento gráfico. El lector puede consultar el apéndice en la página 54 donde se ha ampliado la información acerca de dichos elementos.

Unity cuenta con una gran variedad de componentes ya definidos que se pueden añadir a los `GameObjects`, no obstante, es posible definir más lógica añadiendo componentes de script personalizados. Estos scripts acoplados son los que se encargan de gestionar aquellas funcionalidades de la capa de lógica de negocio.

Finalmente, son estos scripts los que establecen la comunicación con la capa de datos y obtienen la información que se requiere en capas superiores. Para conectar el videojuego con Firebase, servicio alojado en la nube, es necesario la instalación en el proyecto de su SDK, conjunto de herramientas que nos permiten llevar a cabo la conexión con la base de datos y realizar peticiones, evitando la necesidad de un middleware. En el apéndice situado en la página 59 es posible consultar más información al respecto.

<sup>26</sup> Unity Game Object. <https://docs.unity3d.com/es/2021.1/Manual/class-GameObject.html>

## 4.2 Diseño detallado

A continuación, se proporciona una descripción más detallada de cada capa empezando por la de mayor nivel, la de presentación, siguiendo con aquellos scripts que manejan su comportamiento y finalizando con la información que se requiere en la capa de datos.

Como se ha descrito en la arquitectura del sistema, la capa de presentación está compuesta por escenas y objetos de Unity. El videojuego requiere esencialmente de tres tipos de escena:

- un menú principal, donde el jugador podrá iniciar sesión e introducir los datos necesarios para el posterior análisis, como el nombre y podrá indicar la configuración del nivel que desea.
- una escena de nivel, entorno representativo del palacio de la memoria donde se muestra el espacio 3D creado para que el jugador lo vaya recorriendo en busca de los objetos escondidos. Por cada nivel, se creará una escena de este tipo ya que se requieren nuevos entornos, nuevos elementos en general que componen el nivel.
- una escena para la prueba del conocimiento, entorno donde el jugador pone a prueba el léxico aprendido.

Para la creación de interfaces de usuario, Unity cuenta con componentes específicos para este fin como Canvas, Panel, Button, Image o Text, entre otros<sup>27</sup>. Son estos componentes los que formarán el menú principal y la escena de prueba, ya que básicamente en estas escenas se requiere la presentación de elementos de texto y botones. En el apéndice localizado en la página 57 se ha ampliado la información relativa a dichos componentes.

En cuanto a las escenas de nivel, también serán necesarios ciertos componentes de interfaz de usuario para mostrar ciertas indicaciones al jugador, pero, sobre todo serán necesarios aquellos componentes para la construcción del palacio de la memoria como son paredes, techos, puertas, ventanas; así como los objetos 3D que el jugador tiene que ir descubriendo para el aprendizaje del léxico, como puede ser una manzana, una bicicleta o un tiburón.

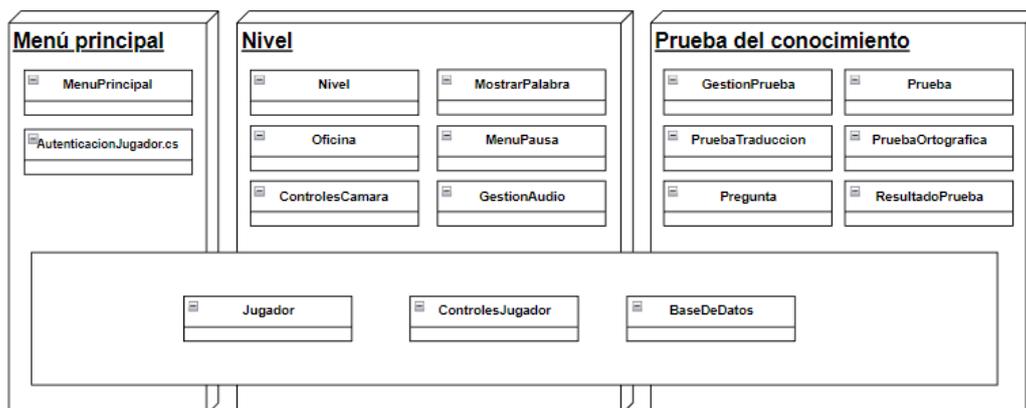


Figura 11. Clases identificadas

<sup>27</sup> Unity UI. Componentes para la creación de interfaces de usuario. <https://learn.unity.com/tutorial/ui-components>

Como se puede observar en la Figura 11, se han detectado tres clases que van a ser usadas en todas las escenas del videojuego. La clase Jugador, recoge todos los atributos identificados del usuario que jugará a Mnemo Mission. La clase ControlesJugador agrupa todos aquellos controles necesarios para poder interactuar con el videojuego. Unity ofrece un paquete llamado InputSystem<sup>28</sup> que permite la creación de acciones del jugador pudiendo asociar diferentes mecanismos de entrada, ya sea mediante un teclado/ratón, un mando o la interacción táctil. Esto resulta realmente útil al querer utilizar los controles en otras clases, ya que al trabajar directamente con las acciones no es necesario saber la manera con la que el jugador interactúa con el videojuego.

La clase BaseDeDatos, es la encargada de establecer la conexión con la capa de datos y a partir de aquí realizar consultas, para ello se aplicará el patrón Singleton, de instancia única con el fin de tener un único punto de referencia para el acceso a la capa de datos.

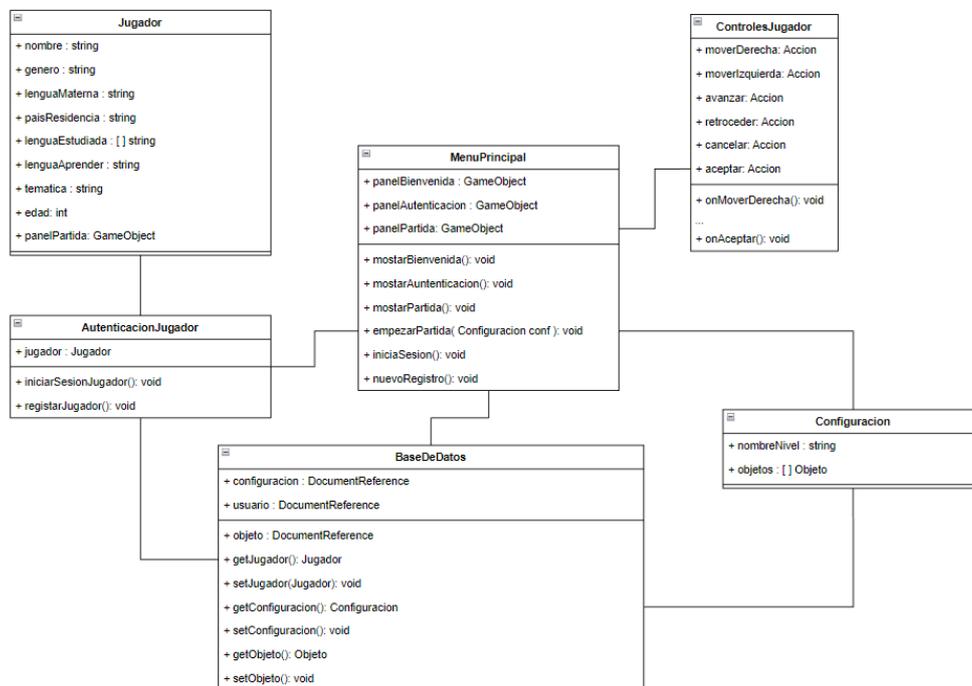


Figura 12. Clases en el componente MenuPrincipal

En la escena de menú principal, como se puede observar en la Figura 12, intervienen hasta 6 clases. La clase MenuPrincipal es la encargada de gestionar las funcionalidades que se pueden realizar en esta escena. Principalmente en esta escena se contará con tres paneles, uno para mostrar una bienvenida al jugador, uno para autenticar al jugador, ya sea iniciando sesión o registrándose por primera vez y otro en el que el jugador puede escoger la configuración con la que iniciar la partida.

Por ello, esta clase necesita tener las referencias de dichos paneles para mostrar u ocultarlos según la navegación por la interfaz. La clase AutenticacionJugador, instanciará la clase Jugador, ofreciendo los métodos para iniciar sesión o registrarse por primera vez. A su vez la clase

<sup>28</sup> Input System.

<https://docs.unity3d.com/Packages/com.unity.inputsystem@1.0/manual/QuickStartGuide.html>

BaseDeDatos, se encargará de guardar la información relativa a los jugadores, así como de obtener las configuraciones de partidas anteriores y de almacenar las traducciones de los objetos. Estas se recogerán en la clase Configuración donde se guarda el nombre del nivel, así como los objetos y sus posiciones dentro del entorno.

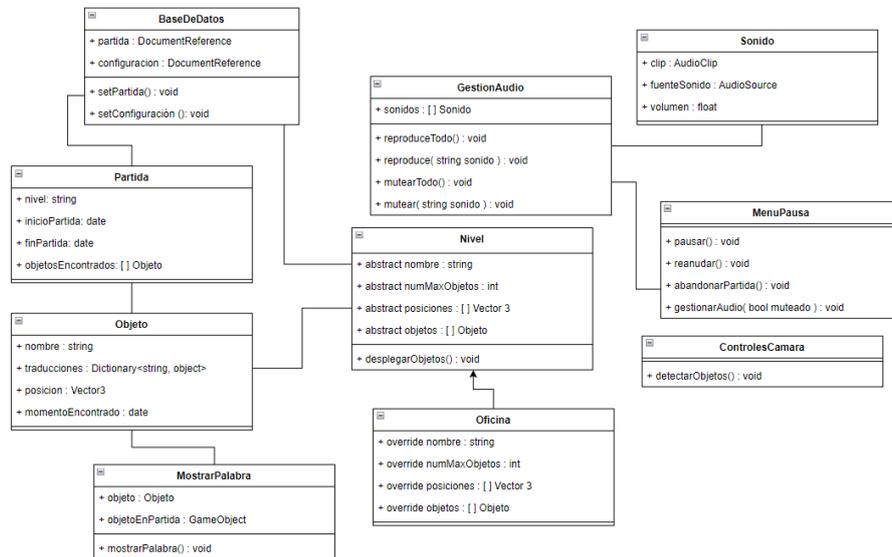


Figura 13. Clases en el componente Nivel

Una vez iniciada la partida, se pasará a la siguiente escena correspondiente al primer nivel donde intervienen las siguientes clases. La clase Nivel se comportará como una clase abstracta la cual preparará los objetos del léxico que se va a aprender en base a la configuración escogida en el menú principal. De cada nivel que se implemente se necesita conocer los objetos que incluirá, las posibles localizaciones donde pueden ir colocados dichos objetos dentro de la escena, así como el nombre del nivel y el número máximo de objetos que puede haber en ese nivel. Estos atributos son abstractos y se deberán implementar en las clases que hereden de Nivel.

Al inicio de la ejecución de la partida, la clase Nivel seleccionará y colocará los objetos según la configuración, aleatoria o previa, seleccionada en el menú principal. Así, la clase Oficina extenderá la clase Nivel, implementando los atributos abstractos mencionados anteriormente.

La clase Nivel contará con un registro de la partida que se hace en la clase Partida la cual recoge datos sobre la hora en la que se inicia y se finaliza la partida, así como los objetos que se han encontrado. La clase Objeto representa a los objetos que el jugador irá descubriendo, recogiendo aspectos como el nombre en sí del objeto y sus traducciones en diferentes idiomas. También tendrá atributos sobre la posición en la que se encuentra dentro del nivel y del momento en el que fueron encontrados.

La clase ControlesCamara, es un componente que va acoplado a la cámara de realidad virtual de la escena que muestra las imágenes del videojuego y se encarga de detectar aquellos elementos con los que el jugador cruza la vista para poder interactuar con ellos. Los elementos con los que el jugador puede interactuar fijando la vista sobre ellos son los objetos que se encuentran por el



nivel. Estos objetos a su vez contarán con la clase *MostrarPalabra* que se encarga de obtener las traducciones del objeto en cuestión y a partir de estas, crear un componente gráfico para mostrarlo en el videojuego.

La clase *GestionAudio* gestiona el audio que se reproduce en la partida que básicamente serán sonidos ambiente representativos del entorno en el que esté basado el nivel, por ejemplo, en el caso de una oficina se reproducen sonidos como el tecleo de un teclado o un teléfono sonando.

La clase *MenuPausa*, proporciona aquellas opciones que se muestran cuando el jugador decide pausar la partida, entre ellas se encuentran la opción de abandonar partida, la opción de deshabilitar o habilitar el sonido ambiente y la opción de regreso a la partida.

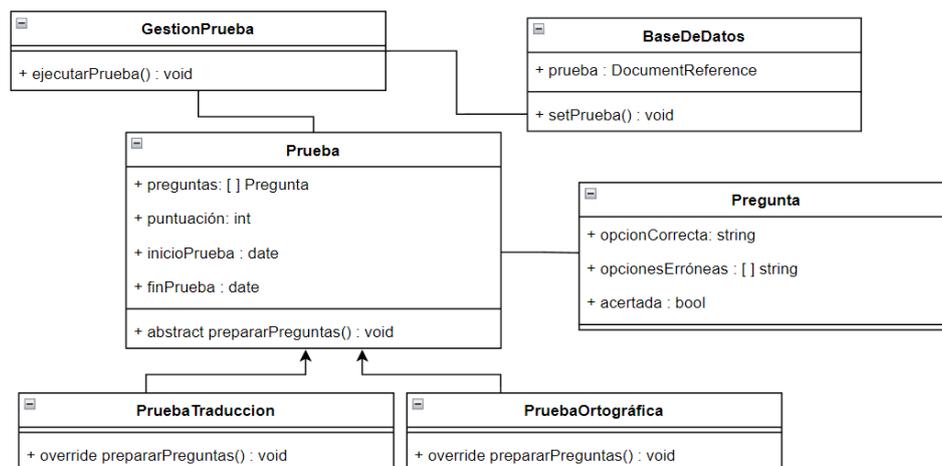


Figura 14. Clases en el componente Prueba

Una vez encontrados todos los objetos del nivel, es posible acceder a la prueba del conocimiento adquirido en la partida. En esta escena el jugador pone a prueba las palabras aprendidas respondiendo una serie de preguntas. En los requisitos se identificaron dos tipos de prueba que se recogen en la clase *PruebaTraduccion* y *PruebaOrtográfica*. En la primera, se preguntará sobre la traducción de una palabra y el jugador tendrá cinco opciones para escoger la correcta. En la segunda, se mostrará la palabra sin una letra y el jugador deberá escoger entre cinco opciones la letra que falta. Estas clases heredan de la clase *Prueba*, que almacena las preguntas que se harán, la puntuación que obtiene el jugador, y la hora de inicio y finalización de la prueba. Esta clase proporciona un método abstracto que se encargará de preparar las preguntas según la implementación que hagan las clases hijas correspondientes al tipo de prueba. La clase *Pregunta* representa como su propio nombre indica la pregunta que debe responder el jugador, teniendo atributos como la opción correcta, las opciones erróneas y si se ha acertado o no dicha cuestión.

En esta escena también se encuentra la clase *GestionPrueba* encargada de funciones como la selección del tipo de prueba a realizar, mostrar los resultados una vez finalizada la prueba y avanzar al siguiente nivel o dar la opción al jugador de repetir la partida nuevamente.

Finalmente, se procede a la descripción de la estructura a seguir para la persistencia de los datos requeridos para el funcionamiento del sistema. Al realizar la conexión con una base de datos

NoSQL, es decir que no sigue un esquema relacional, la información es guardada en colecciones que contienen documentos, que a su vez pueden contener otras colecciones. Esto permite el guardado de información siguiendo el formato de un objeto JSON, una colección de pares clave-valor.

La colección Objetos como se muestra en la Figura 15 contendrá documentos cuya estructura contiene el nombre de dicho objeto y una lista de pares clave-valor que guardan las distintas traducciones en diferentes idiomas, siendo la clave el idioma y el valor la traducción en dicho idioma.

```
{
  "objetos" : [
    {
      "nombre" : "Plátano",
      "traducciones" : [
        {"francés" : "Banane"},
        {"euskera" : "Platanoa"},
        {"polaco" : "Banan"},
        {"ucraniano" : "банан"}
      ]
    },
    {
      "nombre" : "Paraguas",
      "traducciones" : [
        {"francés" : "Parapluie"},
        {"euskera" : "Aterkia"},
        {"polaco" : "parasol"},
        {"ucraniano" : "парасольку"}
      ]
    }
  ]
}
```

Figura 15. Estructura colección Objetos

La colección Configuraciones contendrá documentos que siguen una estructura como la que se muestra en la Figura 16 donde se registra la fecha en la que se genera la configuración y una colección que contiene los objetos con sus posiciones dentro de cada nivel.

```
"configuraciones" : [
  {
    "fechaGenerado" : "08-09-2022 14:20:00 UTC",
    "data" : [
      {
        "nivel" : "Lexical Office",
        "objetos" : [
          { "nombre" : "Plátano", "ejeX" : 1,"ejeY" : 10,"ejeZ" : 12 },
          { "nombre" : "Paraguas", "ejeX" : 35,"ejeY" : 10,"ejeZ" : 90 }
        ]
      },
      {
        "nivel" : "Lexical Factory",
        "objetos" : [
          { "nombre" : "Gafas de sol", "ejeX" : 111,"ejeY" : 103,"ejeZ" : 120 },
          { "nombre" : "Telescopio", "ejeX" : 325,"ejeY" : 10,"ejeZ" : 1 }
        ]
      }
    ]
  }
]
```

Figura 16. Estructura de la colección Configuraciones



La colección Jugadores recoge la información necesaria mencionada anteriormente como es el nombre, género, edad, lengua materna, etc. Además, contiene el registro de partidas jugadas, almacenando las fechas de inicio y fin, el nombre del nivel y los objetos descubiertos anotando su posición y la fecha en la que se encontraron, así como las pruebas realizadas registrando la puntuación obtenida y las preguntas que tuvo que afrontar el jugador, anotando el título de la pregunta, la opción correcta, la opción seleccionada y si se acertó dicha pregunta.

```

"jugadores" : [
  {
    "nombre" : "edu", "género" : "hombre", "edad" : 22,
    "lenguaMaterna" : "español", "paisResidencia" : "españa",
    "...": "...",
    "partidas" : [
      {
        "fechaInicio" : "09-08-2022 13:12:11 UTC", "fechaFin" : "09-08-2022 13:45:21 UTC",
        "nivel" : "Lexical Office",
        "objetosDescubiertos" : [
          {
            "nombre" : "Plátano", "ejeX" : "10", "ejeY" : "2", "ejeZ" : "1",
            "fechaDescubrimiento" : "09-08-2022 13:13:15 UTC"
          },
          {
            "nombre" : "Paraguas", "ejeX" : "10", "ejeY" : "22", "ejeZ" : "1",
            "fechaDescubrimiento" : "09-08-2022 13:14:15 UTC"
          },
          {"...": "..."}
        ]
      },
      {
        "fechaInicio" : "09-08-2022 14:12:10 UTC", "fechaFin" : "09-08-2022 14:18:53 UTC",
        "puntuación" : 4,
        "preguntas" : [
          {
            "titulo" : "Telescopio", "correcta" : "távcsõ",
            "escogida" : "távcsõ", "acertada" : true
          },
          {
            "titulo" : "Casco", "correcta" : "sisak",
            "escogida" : "mellkas", "acertada" : false
          },
          {"...": "..."}
        ]
      }
    ]
  }
]

```

Figura 17. Estructura de la colección Jugadores

### 4.3 Tecnología utilizada

El videojuego requiere de la carga de modelos 3D de aquellos objetos cuyo léxico se pretende aprender. Como se tiene una nula experiencia modelando elementos en 3D, se obtienen modelos ya creados de los objetos desde páginas web que ofrecen estos recursos, algunos de pago, otros gratuitos. Los objetos del videojuego se han extraído de sitios web como Poly Haven<sup>29</sup>, donde aparte de modelos 3D también cuenta con un catálogo de texturas e imágenes de alto rango dinámico todos ellos descargables gratuitamente; TurboSquid<sup>30</sup>, que cuenta con una amplia variedad de modelos gratuitos, aunque principalmente la mayoría sean de pago; Blenderkit<sup>31</sup>,

<sup>29</sup> Poly Haven. <https://polyhaven.com/models>

<sup>30</sup> TurboSquid. <https://www.turbosquid.com/>

<sup>31</sup> Blenderkit. <https://www.blenderkit.com/>

que al igual que Poly Haven contiene tanto modelos 3D como texturas, brochas e imágenes de alto rango dinámico. Para la visualización y ligeros retoques de estos modelos, se ha empleado la aplicación software Blender<sup>32</sup> con la que es posible modelar, renderizar, iluminar y animar gráficos en 3D.

Una de las principales características de Unity es la cantidad de recursos disponibles para facilitar la creación de videojuegos. Esto se encuentra en la tienda de recursos de Unity llamada Asset Store, donde podemos descargar desde plugins hasta modelos 3D, como es el caso de Snaps que cuenta con recursos libres de pago para la construcción de mapas. Entre ellos se encuentran los recursos gratuitos Snaps Office<sup>33</sup> y Snaps Sci-Fi / Industrial<sup>34</sup>, que proporcionan modelos como paredes, ventanas, puertas, es decir elementos útiles para la creación de entornos.

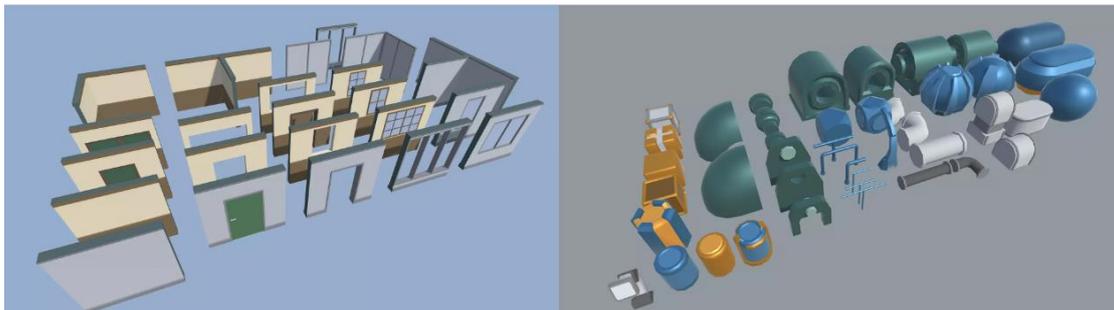


Figura 18. Recursos disponibles en Snaps

Para la implementación de la RV se ha hecho uso del SDK para Unity que proporciona Google Cardboard y que ofrece el soporte a características como el seguimiento del movimiento de la cabeza y la renderización estereoscópica de imágenes, además de una escena de ejemplo donde se pueden probar todas estas características. Pueden consultarse más detalles acerca de las herramientas disponibles en el apéndice situado en la página 63.

En cuanto a la redacción de los componentes de script en C#, se ha hecho uso del editor gratuito de código fuente desarrollado por Microsoft, Visual Studio Code<sup>35</sup>. Disponible para Windows, macOS y Linux, proporciona soporte para una larga cantidad de lenguajes de programación, atajos de teclado intuitivos, multitud de extensiones que aporta la comunidad y es ampliamente personalizable.

Con el fin de llevar un control de las versiones del código que se iba desarrollando, se ha hecho uso de la herramienta Git<sup>36</sup>, que permite la creación de repositorios donde se van guardando las diferentes modificaciones que se realizan sobre el código y recursos del proyecto.

Tras haberla manejado en la asignatura de Gestión de Proyectos de tercero, se ha usado la aplicación Trello<sup>37</sup> con la finalidad de seguir lo descrito en el apartado Plan de trabajo. En Trello se ha llevado un registro del estado de las tareas, es decir, nuevas funcionalidades del

<sup>32</sup> Blender. <https://www.blender.org/>

<sup>33</sup> Snaps Prototype Office. <https://assetstore.unity.com/packages/3d/environments/snaps-prototype-office-137490>

<sup>34</sup> Snaps Prototype Sci-Fi / Industrial. <https://assetstore.unity.com/packages/3d/environments/sci-fi/snaps-prototype-sci-fi-industrial-136759>

<sup>35</sup> Visual Studio Code. <https://code.visualstudio.com/>

<sup>36</sup> Git - Free and open source distributed version control system. <https://git-scm.com/>

<sup>37</sup> Trello. <https://trello.com/es>

videojuego que se iban desarrollando, contando con tres columnas que representan los diferentes estados de una tarea: por hacer, en progreso y completado.

Finalmente, se ha empleado la aplicación software gratuita diagrams.net<sup>38</sup> para la creación de figuras y diagramas que se han incluido a lo largo de este trabajo.

---

<sup>38</sup> Diagrams.net. <https://www.diagrams.net/>

---

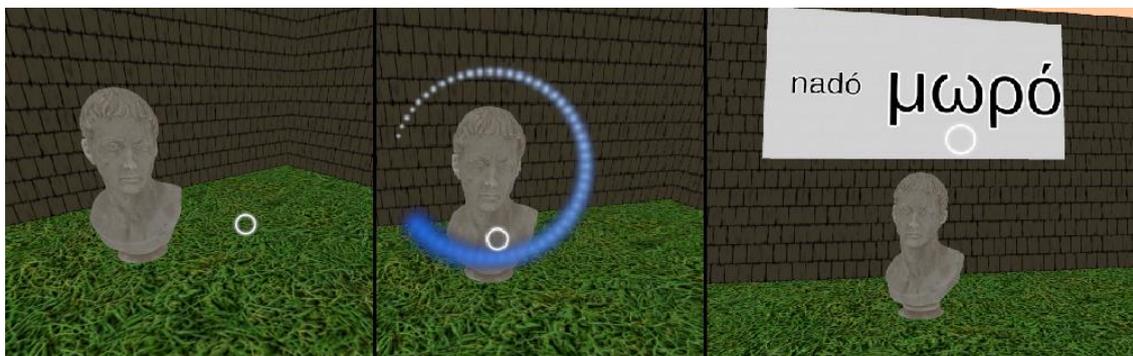
## 5. Desarrollo de la solución

---

Este proyecto continúa con el trabajo realizado el año pasado (Larchen Costuchen, Mollá Vayá, & Dinkova Dimitrova, 2022) añadiendo la RV y el guardado de datos en la nube tal y como se menciona en el proyecto dentro del apartado de Trabajos futuros. En un principio, la idea fue recuperar el ambiente y los elementos del juego original, incluyendo el mapa y todos aquellos elementos que conformaban las escenas, adaptándolos con las modificaciones necesarias para que se mostrasen mediante RV. No obstante, una de las primeras limitaciones encontradas fue el no poder extraer dicho entorno debido a problemas de incompatibilidades en las versiones del editor de Unity. El proyecto el cual se basa este trabajo fue creado con una versión de Unity anterior a la necesaria para el uso del SDK de Google Cardboard, que requiere la 2020.3.36f1 o posterior, detallándose con más información en el apéndice situado en la página 63. Debido a este problema, se optó por la creación de nuevos entornos (una oficina, una fábrica) así como del uso de nuevos objetos para hacer posible la inclusión de la RV en este videojuego.

Otro elemento diferenciador con el anterior proyecto es la manera con la que se interactúa con los objetos también es diferente, ya que ahora para mostrar la traducción del objeto, el jugador debe centrar la vista sobre él durante unos 3 segundos. Se ha añadido una animación de partículas la cual indica el progreso de dicha animación. Si el jugador retira la vista del objeto antes de que termine el tiempo indicado previamente, la animación de partículas se cancela y el léxico no se llega a mostrar.

Para facilitar esta interacción, se ha incorporado una pequeña retícula cuya figura va acoplada al objeto cámara de la escena y se sitúa justo en el centro de la imagen. De esta manera el jugador conoce en todo momento cuando se está centrando la mirada en el objeto en cuestión.



*Figura 19. Interacción con un objeto y muestra del léxico*

Con el objetivo de conseguir una experiencia lo más inmersiva posible y aumentar el realismo del videojuego se han añadido elementos auditivos como son los sonidos diegéticos, es decir, cualquier sonido que emana de la escena; puede incluir todo, desde las voces de los personajes, los diálogos, un monólogo interno e incluso los sonidos de los objetos. Los principales escenarios creados pueden contener sonidos como el tecleo de una computadora, un teléfono recibiendo una llamada, una fotocopiadora escaneando un documento en el caso de una oficina, o engranajes y máquinas en funcionamiento en el caso de la fábrica. En todo caso, estos sonidos pueden activarse o desactivarse desde el menú de pausa según la comodidad del jugador.

También se han añadido elementos visuales como humo saliendo de alguna máquina o polvo en ciertas zonas.

Siguiendo con la idea de anterior, al ser un videojuego en RV se propuso que la ejecución entera del videojuego fuese bajo RV. Desde el menú principal, como se identifican en los requisitos funcionales del sistema, se recogen datos del jugador que requieren de una entrada de texto, como es el nombre o las posibles lenguas extranjeras que hubiera estudiado previamente. Con las herramientas seleccionadas es posible interactuar con el sistema mediante los movimientos de la cabeza y el mando de un videojuego. Entonces, para que el jugador pudiese añadir texto se planteó la opción de mostrar un teclado por pantalla y que el jugador pudiese seleccionar los caracteres necesarios con dichas herramientas, ya sea mirando fijamente aquella tecla usando el seguimiento del movimiento de la cabeza o el joystick del mando. Sin embargo, se encontraron ciertas dificultades para la implementación de dicha solución por lo que se optó por la presentación del menú principal en un formato 2D, así el jugador puede insertar texto mediante el propio teclado del smartphone. Cuando el jugador esté listo para iniciar la partida se le muestra un recordatorio de que a partir de ese momento deberá usar el dispositivo de RV y tanto los niveles como las pruebas empezarán a mostrarse en RV en un formato 3D.

Otro aspecto distinto al proyecto que se tiene de base es la asociación directa de los objetos por descubrir con las palabras que se aprenden, es decir, si en un nivel te encuentras una sandía, el léxico que se muestra al interactuar con el objeto corresponde con la traducción de dicha fruta en la lengua extranjera que se quiere aprender. Esto ha supuesto una limitación en cuanto al número de objetos cuyo significado en otro idioma se quiere aprender debido al uso de modelos en tres dimensiones disponibles gratuitamente en páginas web, al tener una experiencia nula creando dichos modelos de objetos reales. Por lo que finalmente se han empleado un total de 22 objetos de distinta índole, desde alimentos hasta objetos cotidianos como un paraguas o unas gafas de sol.

---

## 6. Conclusiones

---

Tras varias reuniones con mis tutores, multitud de documentación consultada en páginas web y unas cuantas semanas implementando la idea, se ha completado el desarrollo de Mnemo Mission, un videojuego con el que es posible aprender léxico de un idioma extranjero aplicando la regla mnemónica conocida como el método loci en un entorno de RV. Una vez llegados a este punto es posible analizar el cumplimiento de los objetivos iniciales con los que se empezó el proyecto desde una perspectiva mucho más amplia en cuanto a todo lo que envuelve al trabajo en sí, debido al aumento de la información que personalmente he tenido acerca del uso de la tecnología en el aprendizaje, los mecanismos usados para el recuerdo de conocimiento y a los retos a los que uno tiene que enfrentarse para desarrollar un videojuego.

El primero de los objetivos propuestos consistía en desarrollar un videojuego de RV intuitivo y accesible a todo tipo de público. Con la intención de cumplir lo anterior y de acercar la tecnología de RV al mayor número de personas posible, se optó siempre por aquellas soluciones cuyo coste no sea un impedimento para su empleo. Al usar la combinación de la plataforma Android, sistema operativo en móviles ampliamente usado en todo el mundo, así como un dispositivo de RV de bajo coste como pueden ser las Google Cardboard o cualquier alternativa similar de otros materiales, se puede concluir que se ha conseguido alcanzar dicho objetivo con un alto grado de cumplimiento. Sin embargo, el calificativo de intuitivo podría haber sido validado al completo mediante pruebas de aceptación con usuarios reales.

El siguiente objetivo marcado fue el de conseguir un producto software a partir del uso de un proceso software con el fin de obtener un producto de calidad y acorde con los estándares de desarrollo. Al tratarse de un proyecto que es desarrollado por una sola persona sumando a esto las limitaciones temporales que conlleva su realización, a la vista de los resultados obtenidos considero apropiado afirmar que dicho objetivo se ha alcanzado, por motivos como el uso de técnicas para la elicitación de los requisitos del sistema; el diseño previo de la arquitectura, de las clases y sus relaciones y del esquema que se sigue para la persistencia de los datos; todo esto siguiendo la metodología TDD. Pese a la verificación de los requisitos y de las características con los interesados, es decir los tutores de este proyecto, como se ha comentado previamente, hubiera sido beneficioso contar con la validación de la herramienta con usuarios reales.

El objetivo inicial de ofrecer un entorno virtual inmersivo e interactivo que incentive al jugador a llevar a cabo un aprendizaje efectivo y entretenido, podría considerarse que se ha cumplido con un grado medio teniendo en cuenta la cantidad de niveles que finalmente se han incluido, tan solo dos, lo que puede suponer un incentivo poco apetecible para el jugador y la posibilidad de haber ofrecido más mecanismos de pruebas con una naturaleza más lúdica. Además, para conseguir un aprendizaje efectivo del lenguaje se pueden incluir también conceptos relacionados con la gramática, la pronunciación de palabras y aspectos de la cultura.

El último de los objetivos establecidos es el de contar con una amplia variedad de lenguajes y de temáticas, que puede darse como no alcanzado al haber contado con una limitada colección de objetos en tres dimensiones en la solución final debido a la decisión de hacer corresponder los objetos con las palabras que se aprenden. No obstante, tal y como está estructurado el juego, en futuras aportaciones podría ampliarse el conjunto de modelos obteniendo su traducción en muchos más idiomas.

Mi interés personal por el mundo de los videojuegos viene de varios años atrás ya que mi infancia y juventud ha estado siempre acompañada de grandes historias y personajes memorables los cuales recuerdo con gran afecto. Es por lo que cuando tuve la oportunidad de realizar este proyecto, sentí una motivación y un interés adicional por aprender cómo se podía llevar a cabo un juego por mi cuenta, al no haber tenido ninguna experiencia previa desarrollando aplicaciones software de este tipo. Una vez acabado el videojuego puedo decir firmemente que cuento con una experiencia más avanzada en cuanto a los modelos en tres dimensiones, los mecanismos que rigen el funcionamiento de un motor de videojuegos y en especial Unity, el empleo de herramientas externas para hacer uso de un servicio como ha sido con los SDK y, sobre todo, que con una buena planificación y un buen diseño se soluciona gran parte del problema a resolver.

## **6.1 Relación del trabajo desarrollado con los estudios cursados**

---

En este proyecto he tenido la oportunidad de poner en práctica muchos de los aspectos vistos en el grado, y en especial aquellos relacionados con la especialidad de Ingeniería Software.

Para el desarrollo del videojuego empleándose una metodología TDD combinada con un tablero Kanban, se han seguido directrices propias de la gestión de proyectos.

Llevando a cabo una investigación sobre el contexto del problema, un análisis sobre como abordarlo, la creación del diseño de una posible solución y la verificación con los interesados del proyecto de lo que se va creando, se siguen procesos relacionados con lo aprendido acerca del ciclo de vida del desarrollo software con la intención de conseguir un producto software de calidad.

En cuanto al desarrollo en sí de la solución, se han aplicado aspectos vinculados al mantenimiento y evolución del software, llevando un registro de las versiones del código desarrollado y en gran medida aquellos relacionados con la creación de interfaces de usuario y el diseño de software.

---

## 7. Trabajos futuros

---

Con este trabajo se ha conseguido obtener una herramienta innovadora y entretenida para el aprendizaje de lenguas extranjeras debido al uso de la mnemotecnia y de una tecnología con mucho potencial dentro de la educación como es la RV. No obstante, sobre esta base construida todavía hay lugar para nuevos desarrollos y una ampliación en las funcionalidades y características que puede ofrecer este sistema que debido a limitaciones temporales no han podido llevarse a cabo en este mismo proyecto.

- Exportación a más plataformas. El juego finalmente ha sido desarrollado para Android por lo que una adaptación del videojuego para más plataformas como iOS, Windows o consolas resultaría beneficiosa para llegar a más gente que quiera usar la herramienta.
- Ampliación del número de niveles. Sería interesante ofrecer nuevos entornos donde aplicar el método loci dando la posibilidad de escoger aquellos entornos en los que el jugador se encuentre más cómodo para el aprendizaje de nuevo léxico.
- Ampliación del léxico que se va a aprender. Esto supone contar con más modelos en tres dimensiones que representan este nuevo léxico a aprender ofreciendo categorías o temáticas del léxico que se está estudiando.
- Ampliación del número de lenguas extranjeras disponibles para aprender. Es evidente que cuantas más lenguas se puedan aprender con la herramienta, más libertad tendrá el jugador de escoger la lengua extranjera que desea aprender.
- Agregar otros mecanismos de prueba. Contar con más maneras de evaluar el léxico que se aprende potencia el aprendizaje de los conceptos desde otras perspectivas, como pueden ser pruebas auditivas o pruebas de pronunciación.
- Incluir métodos para el aprendizaje de conceptos de gramática o de cultura. Al usar RV es posible la simulación de experiencias mucho más inmersivas, contando por ejemplo con situaciones del día a día.
- Añadir componentes sociales para poder aprender lenguas con otras personas. Dar la posibilidad de conectar con otras personas en el videojuego puede motivar al jugador a sentirse parte de una comunidad donde es más llevadero el aprendizaje de nuevas lenguas.
- Ofrecer una historia mejor dirigida y bajo un mismo hilo conductor. De esta manera el usuario se ve motivado a seguir descubriendo la trama del videojuego.

---

# Referencias

---

- Andrade, A. (2015). *Game engines: A survey*. *EAI Endorsed Trans. Serious Games*, 2(6), e8.
- Arias, Á., & Durango, A. (2016). *Ingeniería y Arquitectura del Software: 2ª Edición*. IT campus Academy.
- Beck, K. (2003). *Test-driven development: by example*. Addison-Wesley Professional.
- Cheng, A., Yang, L., & Andersen, E. (2017). *Teaching language and culture with a virtual reality game*. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 541-549).
- Cho, Y. (2018). *How spatial presence in VR affects memory retention and motivation on second language learning: a comparison of desktop and immersive VR-based learning* (Doctoral dissertation, Syracuse University).
- Da Silva, A. R. (2015). *Model-driven engineering: A survey supported by the unified conceptual model*. *Computer Languages, Systems & Structures*, 43, 139-155.
- Emmerick, K., & Bockholt, M. (2016). *Serious games evaluation: processes, models, and concepts*. In *Entertainment Computing and Serious Games* (pp. 265-283). Springer, Cham.
- IEEE/EIA Guide for Information Technology - Software Life Cycle Processes - Implementation Considerations*. (1998). IEEE/EIA 12207.2-1997.
- Krokos, E., Plaisant, C., & Varshney, A. (2019). *Virtual memory palaces: immersion aids recall*. *Virtual reality*, 23(1), 1-15.
- Larchen Costuchen, A., Mollá Vayá, R. P., & Dinkova Dimitrova, D. (2022). *Roman Palace: A Videogame for Foreign-Language Vocabulary Retention*. *International Journal of Emerging Technologies in Learning*, 17(5), 87-102. <https://doi.org/10.3991/ijet.v17i05.2>.
- Lin, T. J., & Lan, Y. J. (2015). *Language learning in virtual reality environments: Past, present, and future*. *Journal of Educational Technology & Society*, 18(4), 486-497.
- McLeod, V. J. (2007). *The Application of Virtual Reality to Foreign Vocabulary Learning Making Use of the Loci Method*.
- Mills, N., Courtney, M., Dede, C., Dressen, A., & Gant, R. (2020). *Culture and vision in virtual reality narratives*. *Foreign Language Annals*, 53(4), 733-760.
- Muyskens, J. A. (1997). *New Ways of Learning and Teaching: Focus on Technology and Foreign Language Education. Issues in Language Program Direction: A Series of Annual Volumes*. Heinle & Heinle Publishers, 20 Park Plaza, Boston, MA 02116.

- Reggente, N., Essoe, J. K., Baek, H. Y., & Rissman, J. (2020). *The method of loci in virtual reality: explicit binding of objects to spatial contexts enhances subsequent memory recall*. *Journal of Cognitive Enhancement*, 4(1), 12-30.
- Richards, M. (2015). *Software architecture patterns (Vol. 4, p. 1005)*. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Incorporated.
- Shaffer, D. W., Squire, K. R., Halverson, R., & Gee, J. P. (2005). *Video games and the future of learning*. *Phi delta kappan*, 87(2), 105-111.
- Skyes, J. M. (2018). *Digital games and language teaching and learning*. *Foreign Language Annals*, 51(1), 219-224.
- Sommerville, I., & Sawyer, P. (1997). *Requirements engineering : a good practice guide*. John Wiley & Sons.
- Statista. (2022). *Leading language learning apps worldwide in February 2022 by downloads*. Retrieved from <https://www.statista.com/statistics/1239522/top-language-learning-apps-downloads/>.
- Wang, Y. F., Petrina, S., & Feng, F. (2017). *VILLAGE—V irtual I mmersive L anguage L earning and G aming E nvironment: Immersion and presence*. *British Journal of Educational Technology*, 48(2), 431-450.
- Yang, F., Qian, J., Novotny, J., Badre, D., Jackson, C. D., & Laidlaw, D. H. (2020). *A virtual reality memory palace variant aids knowledge retrieval from scholarly articles*. *IEEE Transactions on Visualization and Computer Graphics*, 27(12), 4359-4373.
- Yates, F. A. (1966). *The art of Memory*. London: Routledge & Kegan Paul.
- Yates, F. A. (1999). *The Art of Memory*. New York: Routledge.

---

# Apéndices

---

## A. UNITY

---

### 1. GameObject y componentes

Los GameObjects son la entidad fundamental que Unity tiene para representar cualquier objeto dentro de una escena, desde personajes y objetos coleccionables hasta luces, cámaras y efectos especiales. Un GameObject puede contener a otro o varios GameObjects permitiendo así la construcción de estructuras organizadas y por capas de objetos. Unity ofrece una amplia variedad de objetos preconstruidos Figura 20 desde objetos que describen animaciones y efectos hasta elementos básicos 3D como un cubo o una esfera.

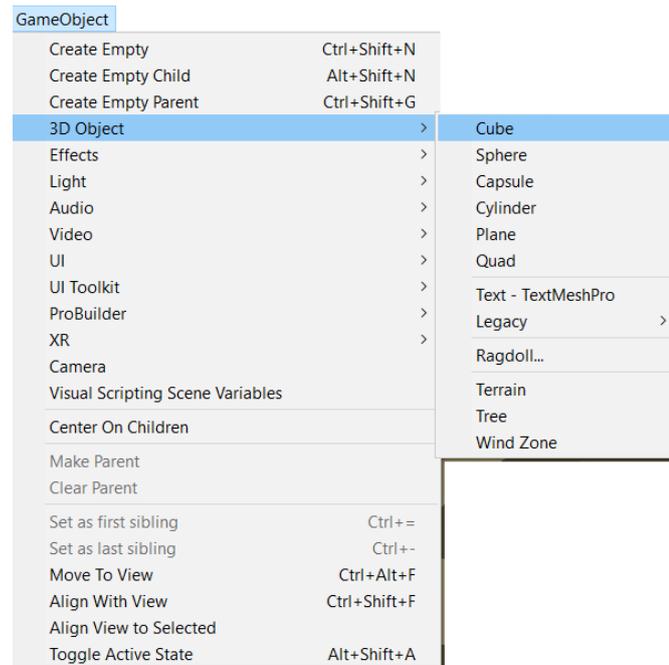


Figura 20. Apartado GameObject en el editor de Unity

Sin embargo, un GameObject no puede hacer nada por sí mismo; necesita de propiedades antes de que pueda convertirse en un personaje, una animación o un entorno. Esto se consigue con el acoplamiento de componentes los cuales describen la funcionalidad del GameObject en cuestión. Por ejemplo, un objeto Cámara el cual se encarga del renderizado de la imagen, es creado al adjuntar un componente Camera Figura 21 a un GameObject.

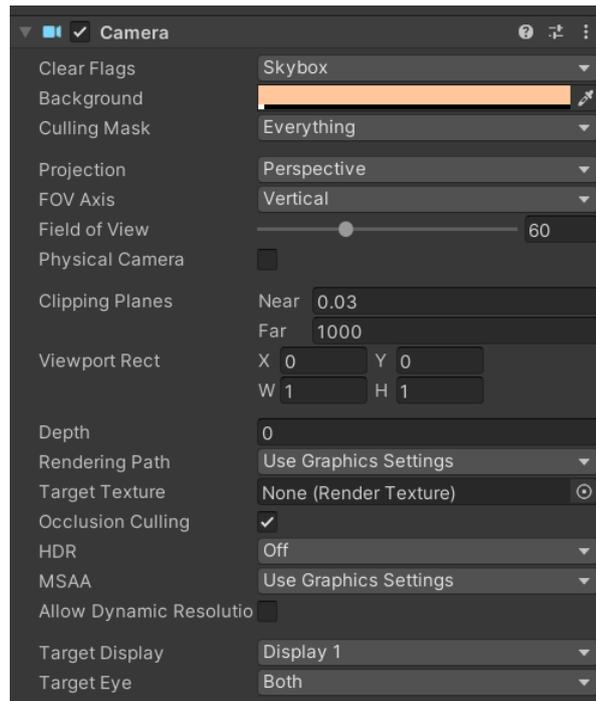


Figura 21. Componente Camera en Unity

Al igual que los GameObject, existen varios componentes ya creados por Unity Figura 22 los cuales nos permiten dotar de funcionalidad adicional a los objetos de una escena. Así, un objeto puede contar con varios componentes acoplados que definen su comportamiento. Existe un componente inherente en todo GameObject llamado Transform el cual define la posición, la rotación y la escala de este dentro de la escena del juego. Otro componente como puede ser Light hace que el objeto que lo lleve acoplado ilumine cualquier cosa cerca suya; el componente Audio Source, para la reproducción de sonidos o el componente Particle System, con el que poder generar efectos visuales.

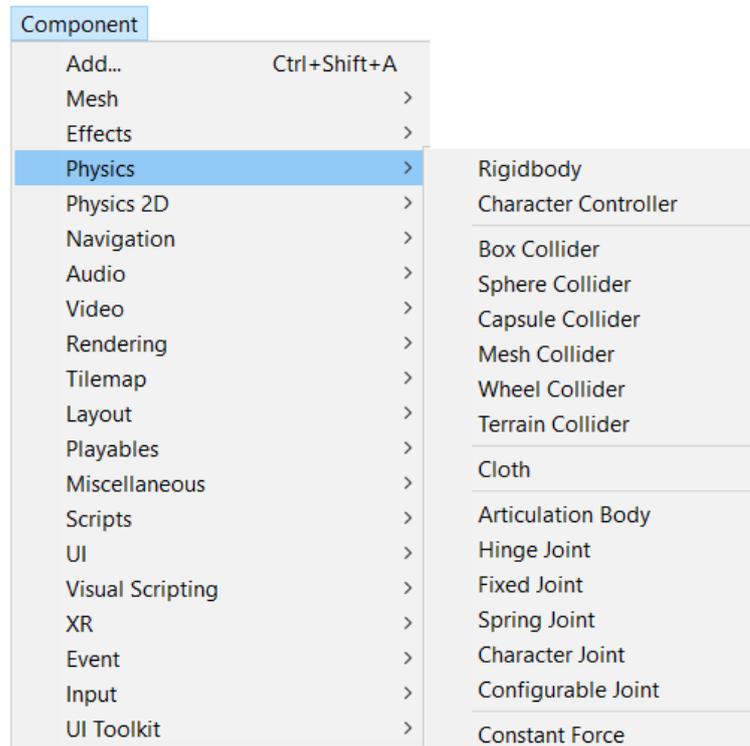


Figura 22. Apartado Component en el editor de Unity

En todo caso, si no se encuentra un componente que se adapte a las necesidades del desarrollo es posible crearlo desde cero, gracias a los componentes de Script. En Unity se usa el lenguaje de programación creado por Microsoft para plataformas móviles y de escritorio, C#. Todo script acoplado a un objeto deriva de la clase MonoBehaviour, encargada de las conexiones con Unity para el funcionamiento interno entre scripts y objetos. Al implementar esta clase, podemos sobrescribir métodos como Start(), que ejecutará todo código en su interior en el momento en el que se crea una instancia del objeto; o el método Update() que se llama en cada fotograma de la ejecución del videojuego.

```
using UnityEngine;
using System.Collections;

public class MainPlayer : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

Figura 23. Script C# en Unity

## 2. Unity UI

Existen ciertos componentes en Unity que ya vienen implementados de serie para cumplir con su cometido como es el caso de los componentes gráficos. A continuación, se describirán los principales componentes usados en este proyecto para la presentación de la interfaz de usuario (IU), como ha sido necesario en menús, mostrando las traducciones de las palabras en los diferentes niveles y presentando las preguntas en las pruebas del léxico aprendido.

En Unity cualquier componente gráfico que quiera mostrarse en la IU lo hace mediante el uso del GameObject Canvas o lienzo, el cual dentro de la escena sería el nodo raíz que contiene el resto de los componentes gráficos. Incluso, si al crear un componente gráfico y en la escena no existe ningún objeto de tipo Canvas, Unity lo crea implícitamente para poder mostrar dicho componente. Los objetos de tipo Canvas pueden configurarse para distintos modos de visualización:

- superposición, cuando se colocan los elementos de la interfaz de usuario enfrente de la pantalla renderizada sobre la escena y el lienzo se autoajusta al cambio de resolución o relación de aspecto.
- cámara, similar a la superposición, pero el lienzo se coloca a una distancia determinada enfrente de una cámara específica, los elementos de la IU son renderizados por dicha cámara lo que supone que los ajustes de esta afectan a la apariencia de la IU.
- espacial, cuando la visualización del lienzo viene dada por la localización 3D en la que esté situada, es decir, como cuando se quieren una IU que forme parte del mundo 3D.

Un GameObject que contenga el componente gráfico Text permite la presentación de texto en la IU pudiendo ajustar la posición y el área donde se insertará el contenido. También se pueden modificar aspectos como el tipo de letra, el estilo y el tamaño de la fuente, la alineación del texto o el desbordamiento horizontal y vertical dentro del área que contiene el texto. Otro componente gráfico fundamental para el proyecto ha sido Image, el cual nos permite mostrar instantáneas en la IU. Para conseguir esto, una vez se tenga la imagen en los recursos del proyecto, es necesario convertirla al tipo de textura Sprite y posteriormente podremos modificar parámetros como el color de fondo de la imagen o el tipo de material que puede ser:

- simple, se escala la totalidad del Sprite por igual.
- sliced, las esquinas permanecen del mismo tamaño, las partes superior e inferior se estiran horizontalmente, los lados del Sprite se estiran verticalmente, y el centro del Sprite se estira horizontal y verticalmente para ajustarse al tamaño del Sprite.
- tiled, en este modo el Sprite permanece del mismo tamaño y no se escala, de esta manera la parte superior e inferior del Sprite se repiten horizontalmente, los lados se repiten verticalmente, y el centro del Sprite se repite en forma de mosaico para ajustarse al tamaño del Sprite.



- filled, se rellena el Sprite a partir de un origen dado en una dirección determinada, siguiendo el método y la cantidad definida.

Ahora, se muestran los componentes interactivos que se han hecho uso en la IU del videojuego. La interacción con estos elementos se hace mediante un ratón, un teclado, eventos táctiles o un gamepad y suelen aparecer en combinación con componentes gráficos con el fin de indicar su función.

Uno de ellos es el componente Button, el cual tiene asociado el UnityEvent On Click el cual permite desencadenar una acción cuando el usuario hace click sobre el botón. El otro elemento interactivo usado ha sido el Input Field, que permite al usuario introducir cadenas de caracteres que se registrarán en la propiedad Text cuyo valor puede ser recuperado una vez se haya acabado la introducción del texto.

## B. CONEXIÓN CON FIREBASE

---

### 1. Creación de proyecto en Firebase

Firebase es un servicio en la nube perteneciente a Google Cloud Platform que ayuda a los desarrolladores a la construcción y el crecimiento de aplicaciones y juegos. Desde la página de Firebase Console se maneja todo lo relacionado con los proyectos y donde es posible registrar aplicaciones y juegos para navegador, Android o Apple. Un proyecto en Firebase se describe como el contenedor de todas las aplicaciones y todos los recursos y servicios necesarios para el proyecto. En un proyecto de Firebase puede haber una o varias aplicaciones registradas, un ejemplo de ello serían las distintas versiones de una aplicación para Android o iOS, o la versión gratuita y de pago de una aplicación.

Como se ha comentado anteriormente, para la creación de un proyecto en Firebase se accede a la página de Firebase Console donde podremos añadir un nuevo proyecto y designar el nombre del proyecto.

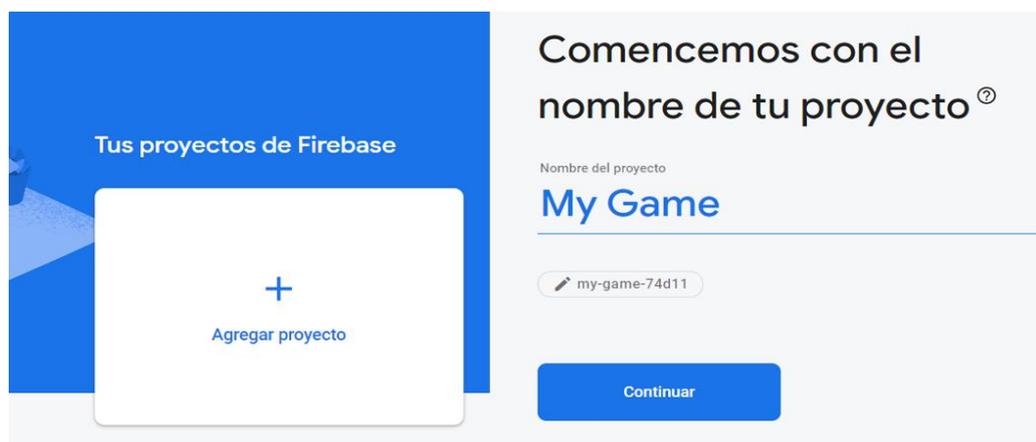


Figura 24. Agregar proyecto en Firebase

### 2. Instalación del SDK para Unity

Una vez creado el proyecto en Firebase, el siguiente paso consiste en registrar una aplicación, en nuestro caso al ser un videojuego, se escoge la opción para registrar aplicaciones de Unity.

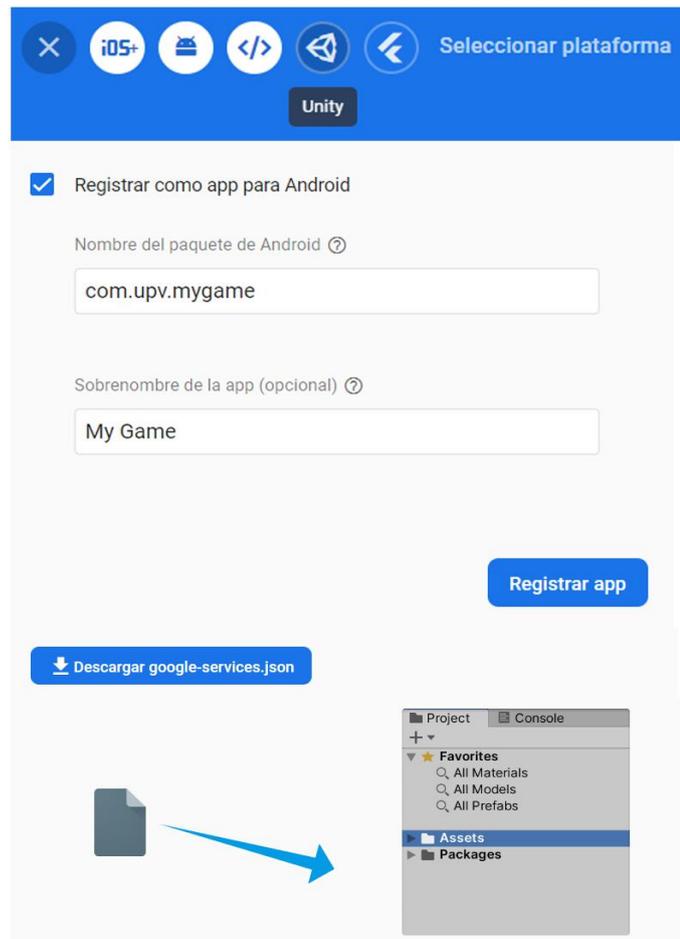


Figura 25. Registro de aplicación Unity en Firebase

Como se puede observar en la Figura 25, los primeros pasos del registro consisten en la introducción del nombre del paquete y de la aplicación que se instalarán los usuarios. Posteriormente, Firebase genera el archivo `google-services.json` que debe ser añadido en el directorio de recursos de nuestro proyecto en Unity. Este archivo sirve para asociar el proyecto Firebase con el videojuego en Unity e incluye parámetros que requieren Firebase y los servicios de Google para comunicarse con la API del servidor de Firebase.

Continuando con el último paso, se procede a descargar el SDK de Firebase para Unity el cual contiene un paquete Unity por cada servicio que se quiera añadir a la aplicación. Como se muestra en la Figura 26, se importa el paquete del servicio Firestore para el guardado de información en la nube.

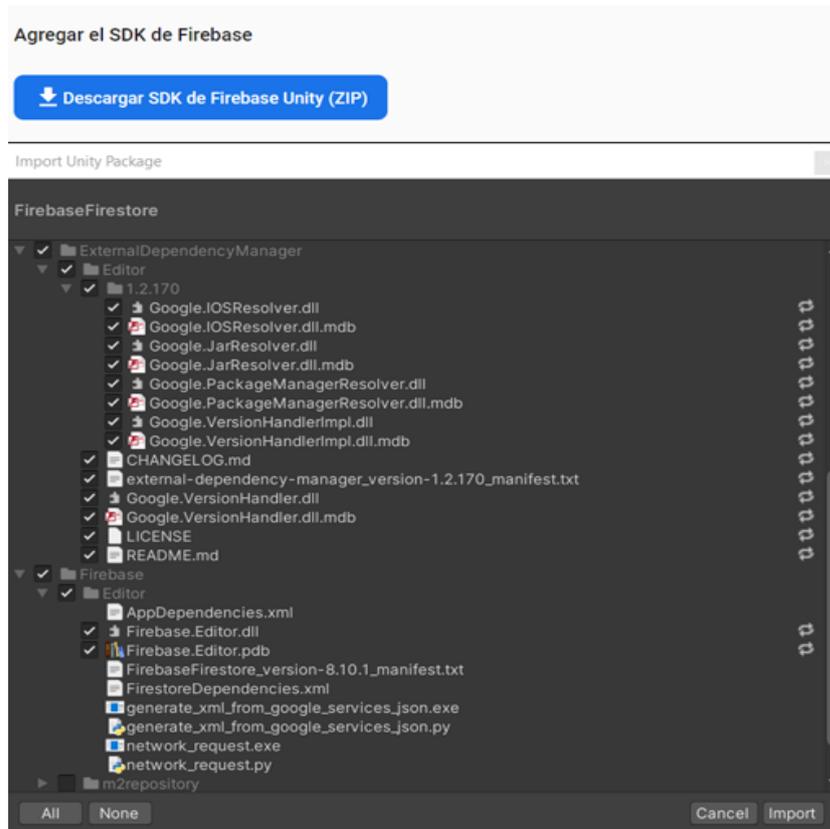


Figura 26. Importación del SDK de Firebase en Unity

### 3. Conexión vía Script

Al haber instalado el SDK de Firebase en nuestro proyecto de Unity, podemos conectarnos con el servicio en la nube de una manera muy sencilla.

```

using Firebase;
using Firebase.Firestore;
using Firebase.Extensions;
using Firebase.Analytics;

public class Database
{
    private FirebaseFirestore db;
    public Database(){
        db = FirebaseFirestore.DefaultInstance;
    }

    void setUserGame(){
        DocumentReference userGames = db.Collection("games").Document();
        Dictionary<string, object> game = new Dictionary<string, object>
        {
            { "startTime", DateTime.Now.ToString("dddd, dd MMMM yyyy HH:mm:ss") },
            { "finishTime", null },
            { "world", "Lexical Office" },
            { "objectsDiscovered", null},
        };
        userGames.SetAsync(game);
    }
}

```

Figura 27. Script básico para la conexión con Firebase

Como se puede apreciar en la Figura 27, se deben declarar aquellas librerías de los servicios que van a ser usadas. En este caso se hace uso especialmente de `Firestore`, aunque también es necesaria la importación del resto de librerías para el correcto funcionamiento de `Firestore`. En el constructor de la clase se obtiene la instancia de `Firestore` que se comunica con el servicio y en el método `setUserGame()` se muestra el ejemplo de una escritura en la colección `games` del objeto `game`.

Gracias a la documentación específica para Unity<sup>39</sup>, resulta muy sencillo la vinculación con el proyecto en `Firestore` y como se ha podido apreciar, en pocas líneas de código se puede establecer la conexión con la base de datos para la lectura y escritura de información.

---

<sup>39</sup> Documentación `Firestore`. <https://firebase.google.com/docs/firestore/#unity>

## C. SDK GOOGLE CARDBOARD

### 1. Recursos para Unity

Google Cardboard ofrece una serie de herramientas de código abierto para la construcción de plataformas de RV mediante el uso de un smartphone, el cual puede mostrar escenas en 3D con renderizado estereoscópico además de seguir y reaccionar a los movimientos de la cabeza.

Siguiendo con la documentación<sup>40</sup> disponible, es necesaria la instalación de la versión 2020.3.36f1 o una posterior de Unity para el correcto funcionamiento de las herramientas. También es necesaria la modificación de ciertos apartados de la configuración del proyecto como es la habilitación del acceso a internet, el establecimiento de la versión mínima de Android para ejecutarlo o la API de gráficos que se va a usar, en este caso OpenGL.

Una vez importado en Unity el paquete proporcionado, podremos encontrar entre los recursos del proyecto los elementos necesarios para el desarrollo de entornos en RV. En el SDK se incluyen elementos fundamentales para la RV como es un GameObject el cual contiene todos los componentes Figura 28 de cámara necesarios para la representación de imágenes en RV, así como los scripts para el seguimiento de la cámara con los movimientos de la cabeza del jugador.

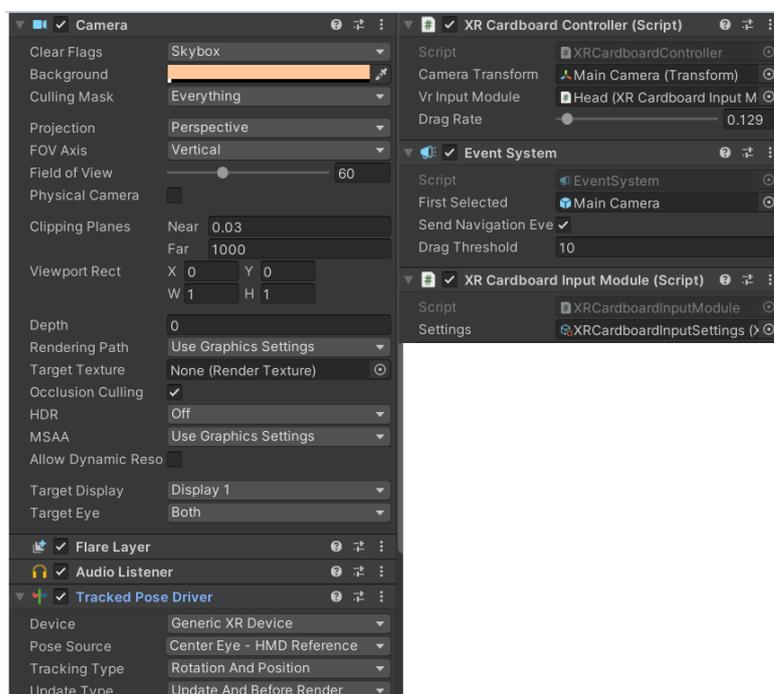
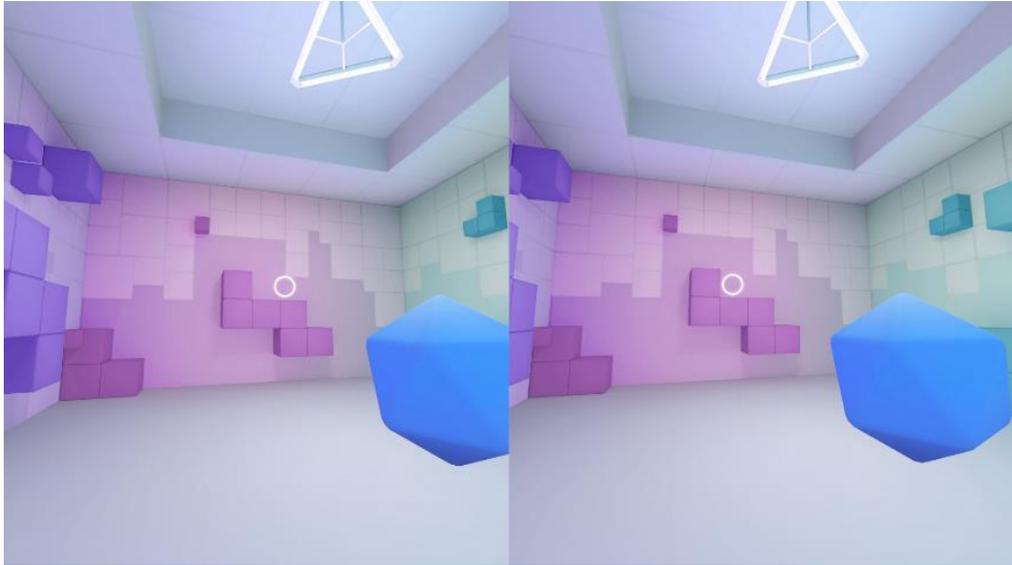


Figura 28. Componentes de GameObject Camara para RV

<sup>40</sup> Documentación Google Cardboard. <https://developers.google.com/cardboard/develop/unity/quickstart>

También se incluye una escena de ejemplo Figura 29, en la que se emplean elementos como el objeto cámara anteriormente comentado y una pequeña sala donde podemos probar el entorno de RV.



*Figura 29. Instantánea de la escena de ejemplo HelloCardboard*

## D. GAME DOCUMENT DESIGN

---

### 1. Análisis del juego

Mnemo Mission es un videojuego de realidad virtual que pretende cumplir con el objetivo didáctico de ofrecer una herramienta útil en el aprendizaje del léxico de una lengua extranjera. Al tener lugar en un entorno inmersivo el jugador tiene la percepción de encontrarse físicamente presente en un mundo no físico. Estas simulaciones fuera del mundo real ofrecen experiencias interactivas y entretenidas.

Debido a su diseño y jugabilidad, Mnemo Mission se categoriza como un videojuego de realidad virtual dentro del género walking simulator (juego de exploración) ya que la mecánica principal se basa en el recorrido del entorno en busca de diferentes objetos. Cabe destacar también la presencia de aspectos referentes a una aventura gráfica al tener que superar pruebas con el fin de pasar a niveles superiores de dificultad.

**Género.** Realidad virtual, walking simulator / juego de exploración, aventura gráfica.

**Misión.** Se formulan dos tipos de objetivos, por una parte, los didácticos que incluyen la memorización del léxico perteneciente a los objetos repartidos por el entorno para su posterior puesta en práctica en las pruebas del nivel. Por otra parte, se encuentran los objetivos lúdicos propios de un videojuego, en este caso predomina la experiencia de realidad virtual a la hora de recorrer el entorno en busca de objetos anacrónicos y superar los niveles para descubrir las diferentes salas que componen Mnemo Mission.

**Público.** Siguiendo las recomendaciones de la mayoría de los fabricantes en cuanto a la edad mínima para el uso de dispositivos de realidad virtual, el videojuego va destinado a un público mayor de 12 años.

**Plataformas.** Android, iOS.

**Hardware requerido.** Un smartphone, un dispositivo de realidad virtual, un mando de videojuegos.

**Música.** Sonidos diegéticos.

### 2. Historia y mecánica del juego

Mnemo Mission comienza con un menú principal dando la bienvenida al jugador para posteriormente mostrar las opciones de registro o inicio de sesión, en el primer caso el jugador debe introducir datos como su alias, una contraseña, su edad, su género, su lengua materna, la nacionalidad, el país de residencia y si es el caso, las lenguas extranjeras que ya ha estudiado. Si el jugador ya ha iniciado sesión, introduciendo alias y contraseña previamente, este panel no se mostrará en futuros accesos al juego. Posteriormente, se presenta el panel previo al inicio de la partida, en el que se configurarán aspectos como la lengua extranjera y la temática del léxico a aprender,

así como si se desea una configuración previa de una partida anterior o se desea generar una nueva aleatoria. Estos paneles al requerir de la introducción de cadenas de texto se mostrarán en formato 2D al encontrar limitaciones a la hora de usar el mando de videojuegos y las gafas de realidad virtual para esta función.

Una vez se abandona el menú principal y se comienza la partida el jugador debe colocarse el dispositivo de realidad virtual y hacer uso del mando de videojuegos. El resto de las escenas proporcionan una perspectiva en primera persona simulando la presencia en las diferentes salas de Mnemo Mission. Cada sala corresponde con un nivel los cuales, al superarlos, aumenta el grado de dificultad necesario para pasar de un nivel a otro. Se plantean los siguientes niveles:

- Nivel 0 - Lexical Island. En este nivel se pretende la familiarización del jugador con la mecánica del juego y con los controles que permiten interactuar con los objetos y desplazarse por el entorno. En este nivel de entrada, la extensión del mapa será reducida y son 3 el número de objetos escondidos por el mapa. Una vez encontrados, se habilita la opción de acceso a la prueba de nivel que será de tipo visual y ortográfica. Independientemente de la puntuación obtenida, el jugador puede avanzar al siguiente nivel o volver a repetirlo.
- Nivel 1 – Lexical Office. Este nivel contiene entre 5-8 objetos escondidos en un mapa ambientado en una oficina. Cuando se encuentren la mitad de los objetos, se lanzará una prueba de tipo visual ortográfica que una vez superada, el jugador puede continuar con la partida buscando los objetos ocultos. Finalmente, una vez descubiertos la totalidad de los objetos se procede a realizar una prueba visual de traducción, en la que según la puntuación se podrá avanzar al siguiente nivel.
- Nivel 2 - Lexical Factory. Ambientado en una fábrica moderna, este nivel contiene entre 9-15 objetos escondidos. Se procede a mostrar 2 pruebas visuales ortográficas correspondientes al descubrimiento del primer y segundo tercio de la totalidad de los objetos ocultos. Finalmente, una vez descubiertos todos los objetos se realiza una prueba visual de traducción con un límite de tiempo de 5 minutos.

Las pruebas visuales ortográficas al tener que acertar tan solo una letra, en el caso de que se escoja una opción errónea, esta pregunta se añadirá a una cola para volver a mostrarla posteriormente hasta que se acierte. Para superar una prueba visual de traducción se deben acertar un 60% por ciento de las preguntas de la prueba. Por cada pregunta acertada, se suma 1 al valor de la puntuación del nivel. La puntuación final de la partida será la suma de las puntuaciones conseguidas en los diferentes niveles.

A continuación, se muestra un ejemplo del funcionamiento de las pruebas.

Suponiendo que se quiere aprender el léxico relacionado con comida en húngaro, al encontrar por el mapa una manzana, una sandía y una calabaza, el jugador aprende el siguiente léxico:

<b>Lengua materna (L1) - Español</b>	<b>Lengua extranjera (LE) – Húngaro</b>
--------------------------------------	---

manzana	alma
sandía	görögdinnye
calabaza	tök

Tabla 4. Traducción al húngaro de algunas palabras.

Seguidamente, las preguntas de la prueba visual ortográfica serían las siguientes:

<b>Completa la palabra con la letra que falta</b>	
al_a	<ul style="list-style-type: none"> <li>▪ s</li> <li>▪ r</li> <li>▪ m</li> <li>▪ e</li> <li>▪ t</li> </ul>
görö_dinnye	<ul style="list-style-type: none"> <li>▪ g</li> <li>▪ q</li> <li>▪ m</li> <li>▪ e</li> <li>▪ a</li> </ul>
_ök	<ul style="list-style-type: none"> <li>▪ x</li> <li>▪ p</li> <li>▪ n</li> <li>▪ m</li> <li>▪ t</li> </ul>

Tabla 5. Ejemplo de prueba ortográfica

Siguiendo con los mismos objetos, ahora se muestran las preguntas que componen a una prueba visual de traducción:

<b>Escoge la traducción correcta de la palabra</b>	
manzana	<ul style="list-style-type: none"> <li>▪ ösana</li> <li>▪ panna</li> <li>▪ nsana</li> <li>▪ alma</li> <li>▪ tana</li> </ul>

sandía	<ul style="list-style-type: none"> <li>▪ görögdinnye</li> <li>▪ fohagyma</li> <li>▪ sisak</li> <li>▪ esernyö</li> <li>▪ tafia</li> </ul>
calabaza	<ul style="list-style-type: none"> <li>▪ tör</li> <li>▪ söte</li> <li>▪ nole</li> <li>▪ tök</li> <li>▪ rol</li> </ul>

Tabla 6. Ejemplo de prueba de traducción.

### 3. Diseño de los niveles

Los niveles de Mnemo Mission serán detallados gráficamente a continuación.

#### Nivel 1 – Lexical Office

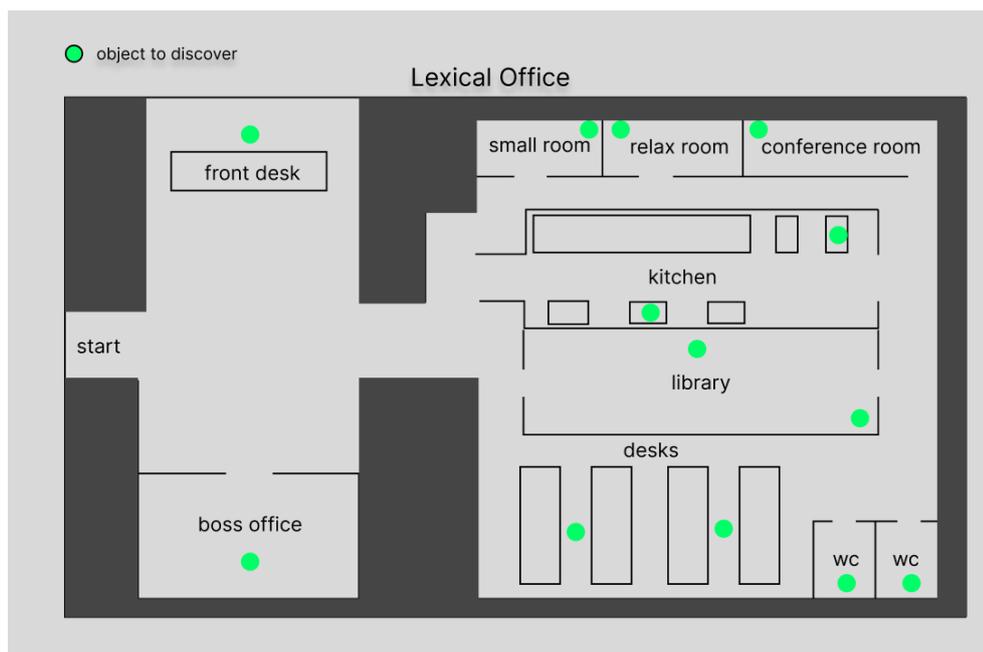


Figura 30. Diseño inicial nivel 1 - Lexical Office

La ambientación de este nivel se inspira en lo que se podría encontrar en una oficina, contando con salas como la recepción, una biblioteca o una sala de reuniones entre otras. En este nivel hay doce posibles localizaciones donde los objetos pueden estar escondidos. La posición de los objetos vendrá determinada por la configuración seleccionada al inicio de la partida. Si se escoge una configuración aleatoria los cinco u ocho objetos que puede haber en este nivel serán repartidos en las posiciones marcadas en verde en la Figura 30. En este nivel, el jugador obtendrá la máxima puntuación posible entre las dos pruebas que se presentan, teniendo que conseguir un

60% de preguntas acertadas en la prueba de traducción para poder avanzar al siguiente nivel.

## Nivel 2 – Lexical Factory

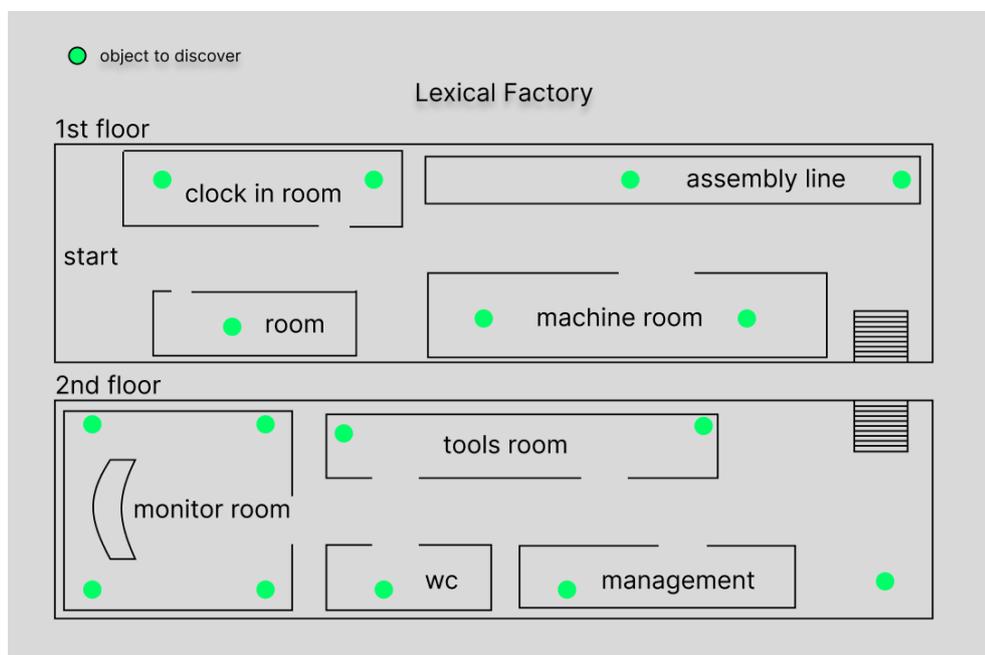


Figura 31. Diseño inicial nivel 2 - Lexical Factory

Con este segundo nivel el jugador se adentra en una fábrica donde se produce un aumento de la dificultad al aumentar también el tamaño del entorno contando con dos plantas por explorar. En el nivel existen dieciséis posiciones donde pueden encontrarse colocados los objetos. La prueba de traducción asociada a este nivel tendrá una limitación de tiempo de cinco minutos, que una vez acabados, se procederá al registro de la puntuación de la prueba hasta el momento.

## 4. Controles

Al ser un videojuego en realidad virtual, el jugador puede visualizar el entorno que le rodea moviendo la cabeza. No obstante, es necesario de ciertos controles para desplazarse por los niveles y realizar ciertas acciones. Por ello, se presentan los controles que el jugador tiene a su disposición para interactuar con el videojuego. Se toma como referencia los botones correspondientes al mando DualShock de PS4.

Entrada	Acción
Joystick hacia arriba	Desplazarse hacia delante
Joystick hacia abajo	Desplazarse hacia atrás
Joystick hacia la izquierda	Desplazarse hacia la izquierda

Joystick hacia la derecha	Desplazarse hacia la derecha
X	Seleccionar
O	Atrás
Start	Pausar partida

Tabla 7. Controles disponibles en Mnemo Mission

## 5. Interfaz de usuario

A continuación, se muestran los prototipos que engloban las principales características mencionadas anteriormente pertenecientes al sistema.



Figura 32. Pantalla de bienvenida

← INICIAR SESIÓN

NOMBRE  
EDUARDO

CONTRASEÑA  
\*\*\*\*\*

COMENZAR

Figura 33. Pantalla de inicio de sesión

← NUEVO REGISTRO

NOMBRE JOE	GÉNERO MASCULINO	LENGUA MATERNA INGLÉS
CONTRASEÑA *****	PAÍS DE RESIDENCIA ESPAÑA	LENGUA A APRENDER HÚNGARO
EDAD 34	LENGUAS ESTUDIADAS ALEMÁN	TEMÁTICA DEPORTES

COMENZAR

Figura 34. Pantalla de registro de nuevo jugador

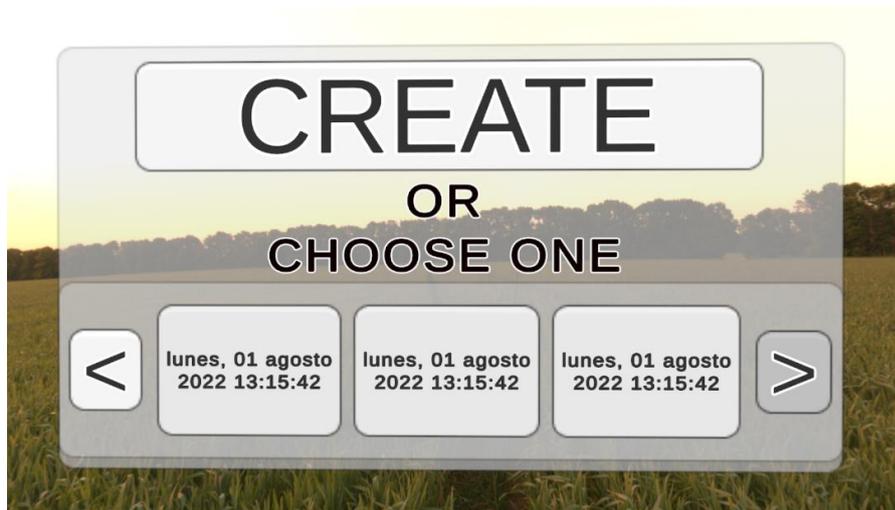


Figura 35. Pantalla de selección de configuración de partida

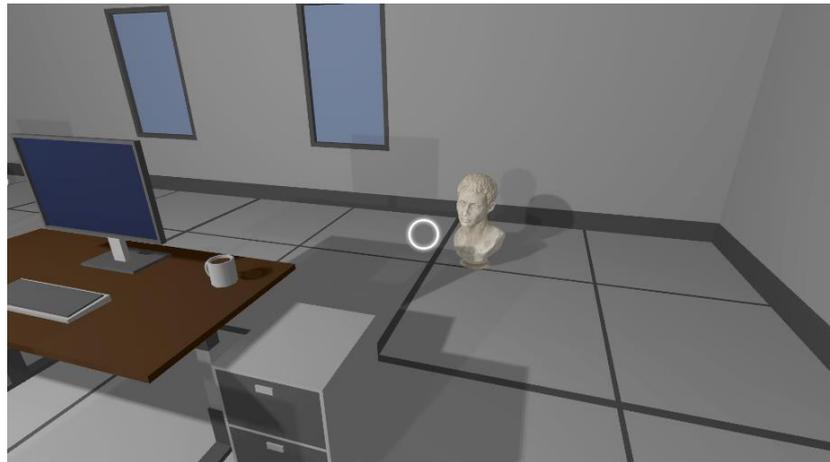


Figura 36. Pantalla encontrando un objeto



Figura 37. Pantalla objeto muestra palabra tras interacción



Figura 38. Pantalla de prueba de traducción



Figura 39. Pantalla de selección de opción en prueba de traducción

## ANEXO

### OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

<b>Objetivos de Desarrollo Sostenibles</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No Procede</b>
ODS 1. <b>Fin de la pobreza.</b>				<b>X</b>
ODS 2. <b>Hambre cero.</b>				<b>X</b>
ODS 3. <b>Salud y bienestar.</b>				<b>X</b>
ODS 4. <b>Educación de calidad.</b>	<b>X</b>			
ODS 5. <b>Igualdad de género.</b>				<b>X</b>
ODS 6. <b>Agua limpia y saneamiento.</b>				<b>X</b>
ODS 7. <b>Energía asequible y no contaminante.</b>				<b>X</b>
ODS 8. <b>Trabajo decente y crecimiento económico.</b>				<b>X</b>
ODS 9. <b>Industria, innovación e infraestructuras.</b>				<b>X</b>
ODS 10. <b>Reducción de las desigualdades.</b>	<b>X</b>			
ODS 11. <b>Ciudades y comunidades sostenibles.</b>				<b>X</b>
ODS 12. <b>Producción y consumo responsables.</b>				<b>X</b>
ODS 13. <b>Acción por el clima.</b>				<b>X</b>
ODS 14. <b>Vida submarina.</b>				<b>X</b>
ODS 15. <b>Vida de ecosistemas terrestres.</b>				<b>X</b>
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>				<b>X</b>
ODS 17. <b>Alianzas para lograr objetivos.</b>				<b>X</b>

A continuación, se detalla el grado de relación de los anteriores Objetivos de Desarrollo Sostenible (ODS) mencionados con el trabajo realizado:

- Educación de Calidad. Con este trabajo se pretende ofrecer una herramienta de soporte al aprendizaje de lenguas extranjeras que además introduce métodos para facilitar el recuerdo de información y el uso de una tecnología innovadora como es la realidad virtual. De esta manera se promueve un aprendizaje eficaz y entretenido mediante experiencias inmersivas, que personalmente pienso que guarda una estrecha relación con el cumplimiento de este objetivo.

- Reducción de las desigualdades. Al haber planteado el proyecto como una herramienta accesible al mayor número de personas posible, haciendo uso de tecnologías de realidad virtual de bajo coste y publicando una solución destinada a dispositivos móviles ampliamente utilizados alrededor del mundo, pienso que se está ofreciendo la oportunidad de acceso a una herramienta útil para el desarrollo y avance de aquellos individuos que lo necesiten y no puedan acceder a otras alternativas. Es por este motivo que considero oportuno afirmar que este trabajo promueve una reducción de las desigualdades.