



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

PROTOTIPO DE VIDEOJUEGO EN DISPOSITIVO MÓVIL PARA FOMENTO DEL TURISMO Y CULTURA DE UNA REGIÓN

Alumno: Álvaro Morales García

Tutor: Prof. D. Juan Roberto Jiménez Pérez

Dpto.: Departamento de Informática

Junio, 2021



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

D. Juan Roberto Jiménez Pérez, tutor del Proyecto Fin de Carrera titulado: Prototipo de videojuego en dispositivo móvil para fomento del turismo y cultura de una región, que presenta Álvaro Morales García, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, junio de 2021

El alumno:

El tutor:

Álvaro Morales García

D. Juan Roberto Jiménez Pérez

AGRADECIMIENTOS

Agradezco enormemente la compañía y apoyo de toda la gente que ha estado a mi lado durante el desarrollo de este proyecto y que sin la cual no habría sido posible este trabajo.

A mis padres, Tomás y Susy, y a mi hermano, Gonzalo, por darme ánimos durante todo el proceso y animarme a hacerlo lo mejor posible.

A mi pareja, Rubén, por aguantarme y darme ánimos cuando lo necesitaba, así como por ser un usuario de pruebas clave para el desarrollo del proyecto.

Por último, a mi tutor, Juan Roberto Jiménez Pérez, por creer desde un principio en este proyecto y a animarme a hacerlo lo mejor posible con la máxima libertad.

GRACIAS A TODOS

ÍNDICE

Índice de tablas	11
Índice de ilustraciones	12
Glosario de términos	15
Capítulo 1: Introducción	17
1.1 Motivación	17
1.2 Objetivo general	19
1.3 Contexto del proyecto	19
Capítulo 2: Características del videojuego	21
2.1 Título del videojuego	21
2.2 Género del videojuego	21
2.3 Clasificación PEGI	21
2.4 Influencia de otros videojuegos	22
2.5 La cámara	25
2.6 Gráficos	26
2.7 Interfaz	26
2.8 Modos de juego	28
2.9 Mecánicas principales	31
2.10 Objetivos	32
2.11 Economía	32
2.12 Criterios de éxito	33
2.13 Criterios de fracaso	33
Capítulo 3: Tecnología y herramientas	34
3.1 Motor de Videojuegos	34
3.2 Plataforma de uso	35

3.3 Herramienta de mapas: MapBox	35
3.4 Otros programas usados	36
Capítulo 4: Proceso de desarrollo de la aplicación	37
4.1 Metodología: Proceso Unificado	37
4.2 Análisis del sistema	38
4.2.1 Definición del proyecto	38
4.2.2 Extracción de requisitos	39
4.2.3 Declaración general del proyecto	43
4.3 Planificación del proyecto	44
4.3.1 Definición de tareas	44
4.3.2 Estimación de tiempos	46
4.3.2 Estimación de recursos y costes	51
4.4 Ejecución del proyecto	55
4.4.1 Iteración 1	56
4.4.2 Iteración 2	59
4.4.3 Iteración 3	62
4.4.4 Iteración 4	67
4.4.5 Iteración 5	79
4.4.6 Iteración 6	88
Capítulo 5: Proyección empresarial y comercialización	98
5.1 Investigación de mercado	98
5.2 Segmentos de mercado	98
5.3 Estrategia de promoción y marketing	99
5.4 Fuentes de ingresos	100
5.5 Expansión del proyecto	101
5.5.1 Ampliación de rutas y puzzles	101
5.5.2 Factores a tener en cuenta	102

5.6 Estimación de costes del proyecto completo	103
5.6.1 Completitud del videojuego	103
5.6.2 Página web	107
5.6.3 Creación del gestor de cupones	110
5.7 Estimación de costes de mantenimiento	113
5.7.1 Videojuego	114
5.7.1 Página Web	116
5.7.3 Gestor de cupones	119
5.7.4 Gasto total de mantenimiento	121
Capítulo 6: Conclusiones	124
6.1 Problemas encontrados	124
6.2 Conclusiones	125
6.3 Trabajo futuro	126
Bibliografía	128
Anexos	133
ANEXO 1: DIAGRAMAS DE DISEÑO FINALES	133
DIAGRAMA DE CASOS DE USO	133
DIAGRAMAS DE SECUENCIA	134
DIAGRAMA FINAL DE CLASES DE DISEÑO	138
ANEXO 2: RESOLUCIÓN DE LOS PUZZLES	139
EL CHICO Y EL ESPECTRO	139
UNA DECLARACIÓN DE AMOR ESCALOFRIANTE	141
UN LAGARTO EXPLOSIVO	142
ANEXO 3: MANUAL DE INSTALACIÓN	143

Índice de tablas

Tabla 1: Requisitos.....	41
Tabla 2: Declaración general del proyecto	44
Tabla 3: Estimación de tiempo de las tareas	50
Tabla 4: Estimación de tiempo de las iteraciones	50
Tabla 5: Recursos digitales del prototipo	52
Tabla 6: Recursos materiales del prototipo.....	54
Tabla 7: Personal del prototipo	55
Tabla 8: Estimación del coste total del prototipo.....	55
Tabla 9: Estimación de costes de recursos digitales del videojuego	104
Tabla 10: Estimación de costes de recursos materiales del videojuego	105
Tabla 11: Estimación de costes de personal implicado en el videojuego.....	106
Tabla 12: Estimación total de costes del videojuego.....	107
Tabla 13: Estimación de costes de recursos digitales de la página web.....	108
Tabla 14: Estimación de costes de recursos materiales de la página web	109
Tabla 15: Estimación de costes de personal implicado en la página web	109
Tabla 16: Estimación total de costes de la página web	110
Tabla 17: Estimación de costes de recursos digitales del gestor de cupones	111
Tabla 18: Estimación de costes de recursos materiales del gestor de cupones.....	111
Tabla 19: Estimación de costes de personal implicado en el gestor de cupones	112
Tabla 20: Estimación total de costes del gestor de cupones.....	112
Tabla 21: Estimación de costes de recursos digitales del mantenimiento del videojuego	114
Tabla 22: Estimación de costes de recursos materiales del mantenimiento del videojuego	115
Tabla 23: Estimación de costes del personal implicado en el mantenimiento del videojuego	116
Tabla 24: Estimación total de costes del mantenimiento del videojuego.....	116
Tabla 25: Estimación de costes de recursos digitales del mantenimiento de la página web	117
Tabla 26: Estimación de costes de recursos materiales del mantenimiento de la página web	118
Tabla 27: Estimación de costes del personal implicado en el mantenimiento de la página web	118
Tabla 28: Estimación total de costes del mantenimiento de la página web	119
Tabla 29: Estimación de costes de recursos digitales del mantenimiento del gestor de cupones	119
Tabla 30: Estimación de costes de recursos materiales del mantenimiento del gestor de cupones	120
Tabla 31: Estimación de costes del personal implicado en el mantenimiento del gestor de cupones	120
Tabla 32: Estimación total de costes del mantenimiento del gestor de cupones	121
Tabla 33: Estimación de costes de recursos materiales del personal común de mantenimiento	122
Tabla 34: Estimación de costes de costes del personal común de mantenimiento	122
Tabla 35: Estimación total de costes de mantenimiento	123

Índice de ilustraciones

Ilustración 1: Aportación del turismo a la economía española [1]	17
Ilustración 2: Facturación del mercado del videojuego en España [3]	18
Ilustración 3: Diagrama de relación de sistemas	19
Ilustración 4: Clasificación de edad PEGI [6]	22
Ilustración 5: Aviso PEGI "Incluye Compras" [6]	22
Ilustración 6: Imágenes de Pokémon GO [8]	23
Ilustración 7: Imágenes de Harry Potter Wizards Unite [10]	24
Ilustración 8: Pueblo de Profesor Layton [12]	25
Ilustración 9: Puzle de Profesor Layton [12]	25
Ilustración 10: Prototipo de interfaz de mapa	27
Ilustración 11: Prototipo de interfaz de perfil	27
Ilustración 12: Prototipo de interfaz de menú	27
Ilustración 13: Prototipo de interfaz de puzle	27
Ilustración 14: Flowchart	30
Ilustración 15: RBS	42
Ilustración 16: WBS	45
Ilustración 17: Imagen 1 de prototipo de puzle	57
Ilustración 18: Imagen 2 de prototipo de puzle	57
Ilustración 19: Imagen 3 de prototipo de puzle	57
Ilustración 20: Imagen 4 de prototipo de puzle	57
Ilustración 21:: Imagen 5 de prototipo de puzle	57
Ilustración 22: Imagen 6 de prototipo de puzle	57
Ilustración 23: MapBox Studio [31]	60
Ilustración 24: Localización GPS	61
Ilustración 25: Diseño de Localización GPS	61
Ilustración 26: Jerarquía de objetos en Unity	63
Ilustración 27: Elementos del puzle en la escena	64
Ilustración 28: Diseño de los puzles y su controlador	65
Ilustración 29: Diseño de obtención de monedas	68
Ilustración 30: Ventana de pistas	69
Ilustración 31: Diseño del sistema de pistas	70
Ilustración 32: Ventana de confirmación "Conseguir pista"	71
Ilustración 33: Diseño de ubicación de puzles	73
Ilustración 34: Marca de puzle en el mapa	74
Ilustración 35: Ventana 1 de selección de ruta	76
Ilustración 36: Ventana 2 de selección de ruta	76

Ilustración 37 : Ventana 3 de selección de ruta	76
Ilustración 38: Ventana 4 de selección de ruta	76
Ilustración 39: Diseño de selección de ruta.....	77
Ilustración 40: Guía en el mapa.....	79
Ilustración 41: Radio de activación del puzle	80
Ilustración 42: Diseño de la guía.....	81
Ilustración 43: Diseño del control de puntuación del jugador	84
Ilustración 44: Cálculo del decremento de puntuación.....	84
Ilustración 45: Ventana de ranking.....	85
Ilustración 46: Diseño del ranking	86
Ilustración 47: Implementación del cambio de territorio en el ranking.....	87
Ilustración 48: Ventana Tienda de cupones	89
Ilustración 49: Ventana Información de compra.....	89
Ilustración 50: Ventana Mis cupones	90
Ilustración 51: Ventana Información de cupón	90
Ilustración 52: Ventana Código QR de cupón.....	91
Ilustración 53: Diseño del sistema de cupones	92
Ilustración 54: Ventana de perfil	94
Ilustración 55: Notificación de comienzo de ruta.....	95
Ilustración 56: Notificación de continuación de ruta	95
Ilustración 57: Ventana de finalización de ruta.....	95
Ilustración 58: Notificación de selección de ruta ya acabada	95
Ilustración 59: Grafica de la capacidad de movilidad según la edad de la persona [33]	99
Ilustración 60: Situación de los elementos del puzle en la jerarquía de objetos de Unity.....	102
Ilustración 61: Diagrama de casos de uso	133
Ilustración 62: Diagrama de secuencia del caso de uso "Cambiar de ubicación"	134
Ilustración 63: Diagrama de secuencia del caso de uso "Resolver puzle"	134
Ilustración 64: Diagrama de secuencia del caso de uso "Mostrar pista"	135
Ilustración 65: Diagrama de secuencia del caso de uso "Seleccionar ruta"	135
Ilustración 66: Diagrama de secuencia del caso de uso "Mostrar ranking"	136
Ilustración 67: Diagrama de secuencia del caso de uso "Comprar cupón"	136
Ilustración 68: Diagrama de secuencia del caso de uso "Activar cupón"	137
Ilustración 69: Diagrama final de clases de diseño.....	138
Ilustración 70: Enunciado del puzle "El chico y el espectro"	139
Ilustración 71: Elementos del puzle "El chico y el espectro"	139
Ilustración 72: Enunciado del puzle "Una declaración de amor escalofriante"	141
Ilustración 73: Elementos del puzle "Una declaración de amor escalofriante"	141
Ilustración 74: Solución del puzle "Una nota de amor escalofriante"	141

Ilustración 75: Enunciado del puzle "Un lagarto explosivo"	142
Ilustración 76: Elementos del puzle "Un lagarto explosivo"	142
Ilustración 77: Resolución del puzle "Un lagarto explosivo"	142
Ilustración 78: Gestor de archivos del dispositivo	143
Ilustración 79: Instalación de la aplicación	143
Ilustración 80: Permitir permisos	144

Glosario de términos

Término	Definición
Android	Sistema operativo para plataformas móviles.
cupón	Cupón de descuento digital para una oferta de una empresa concreta que el jugador puede conseguir en la aplicación.
código QR	Código de barras cuadrado que puede codificar información. Se usará para redireccionar a una dirección web.
escena (Unity)	Entorno visual en el que se pueden observar los elementos creados en el motor de videojuegos de Unity
estilo cartoon	Estilo de dibujo humorístico o infantil
Featured Snippets	Resultados de búsqueda destacados por Google que aparecerán siempre en los primeros puestos de los resultados
gameobject	Objetos de Unity que representan elementos como el suelo, los personajes, etc.
gráficos 2.5D	Combinación de gráficos 2D y 3D limitados a un plano bidimensional.
in-game advertising	Enfoque que busca la inclusión de publicidad dentro del videojuego
juego basado en localización	Videojuego que detecta la ubicación del usuario en el mundo real para plasmarlo en dicho juego.
log	Fichero de texto que graba el registro de actividad de la aplicación como avisos o errores.
login	Es el acceso por parte del usuario a un sistema como el acceso con contraseña a una página web.
metodología de desarrollo	Conjunto de técnicas aplicadas al diseño del proyecto.

micropagos	Pequeños pagos que realiza el usuario a través de la aplicación para conseguir más contenido.
pistas	Ayudas que recibirá el usuario para resolver un puzle.
Posicionamiento web SEO	Técnicas aplicadas a una página web para mejorar su visibilidad en los navegadores web
prefab	Gameobjects de Unity reutilizables e instanciables a través de scripting.
puntos de interés	Elementos de la herramienta de mapas Mapbox que sitúan en el mapa una serie de elementos destacados.
puzle	Enigma o acertijo que debe el usuario resolver en el videojuego.
Requirements Breakdown Structure (RBS)	Estructura de desglose de recursos. Utilizada para planificar y controlar los requisitos del proyecto.
ruta	Recorrido por un territorio con un conjunto de puzles a ser resueltos por el usuario.
scroll	Barra que se muestra en pantalla para deslizar el contenido de esta.
serious game	Videojuego cuyo propósito final va más allá del entretenimiento pudiéndose usar para ámbitos formativos.
servicio de hosting	Servicio de alojamiento de páginas web. Con este se pueden almacenar información, vídeos, imágenes, etc.
Work Breakdown Structure (WBS)	Estructura de desglose de trabajo. Utilizada para planificar y controlar las tareas necesarias para satisfacer los requisitos del proyecto.

Capítulo 1: Introducción

1.1 Motivación

En estos últimos meses el mundo ha cambiado de forma drástica debido a la pandemia y uno de los sectores más hostigados en este tiempo ha sido el sector del turismo. Ahora mismo este sector se encuentra en una de sus mayores crisis junto a otros sectores como el de la hostelería, transporte o comercio.

En España, el turismo es uno de los sectores más importantes dentro de nuestra economía, el sector que más riqueza aporta llegando a un 12,4% del PIB en 2019, siendo vital para el desarrollo de nuestro país [1]. Esto lo podemos ver reflejado en el siguiente gráfico del Instituto Nacional de Estadística [Ilustración 1]:

Aportación del turismo a la economía española. - Año 2019

	Valor	Porcentaje sobre el total
Aportación al PIB (millones de euros)	154.487 	12,4 
Aportación al empleo total (millones de puestos de trabajo)	2,72 	12,9 

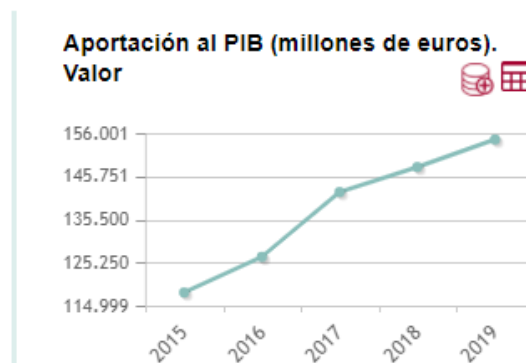


Ilustración 1: Aportación del turismo a la economía española [1]

Esto cambió en 2020, desplomándose hasta el 5,5% del PIB nacional [2].

Actualmente, se espera por parte de todo el país la pronta recuperación del turismo junto a otros sectores debido a la vacunación de la población.

Teniendo esto en mente, las preguntas que nos hacemos ahora son: ¿Cómo podemos favorecer esa recuperación? ¿Cómo podemos salvar el turismo y ayudar a que todo vuelva a esa “normalidad” tan ansiada?

El sector del videojuego, al contrario que el del turismo, ha crecido debido a la situación actual [Ilustración 2] y cabe pensar que este pueda ser un gran apoyo para la recuperación de otros sectores.

EL MERCADO ESPAÑOL DEL VIDEOJUEGO

FACTURACIÓN TOTAL

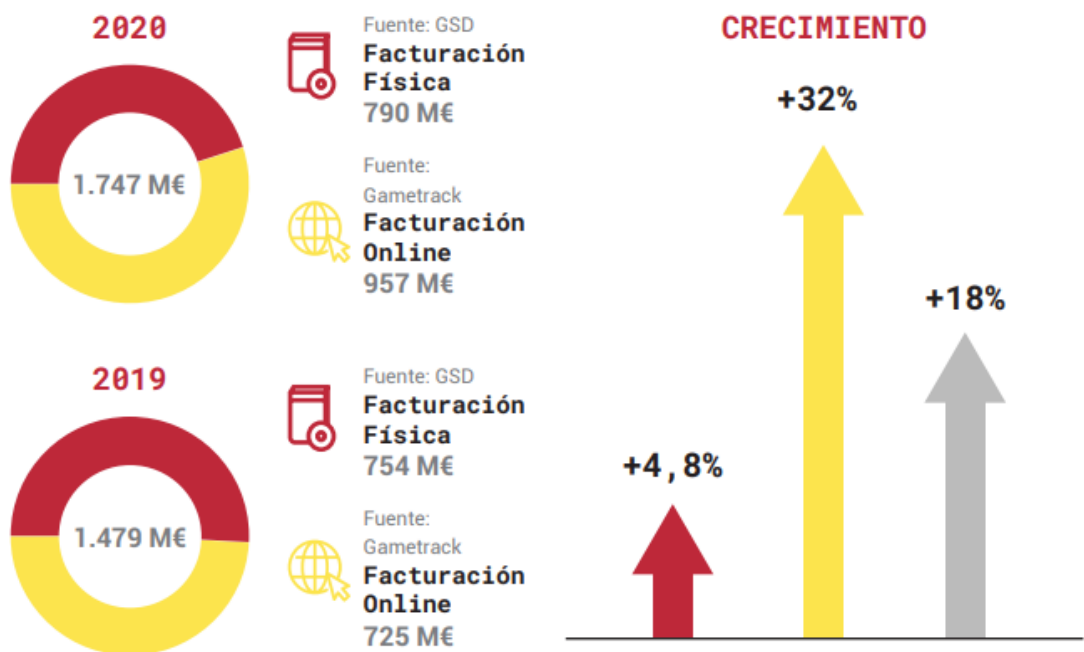


Ilustración 2: Facturación del mercado del videojuego en España.[3]

Partiendo de esta premisa, este proyecto pretende favorecer la futura recuperación de dichos sectores, sobre todo el turismo, con el desarrollo de un videojuego, un “serious game” para plataformas móviles, donde se promueva el turismo dentro de una región, siendo aplicable a nivel de ciudad, comunidad, país o, incluso, todo el mundo.

1.2 Objetivo general

En este Trabajo de Fin de Grado titulado “Prototipo de videojuego en dispositivo móvil para fomento del turismo y cultura de una región” se describe el proceso de desarrollo de un prototipo de “serious game”.

El objetivo principal de este proyecto será la creación de un videojuego para dispositivos móviles basado en la localización que guiará al usuario a través de las calles de un territorio, dando a conocer lugares emblemáticos de este mientras resuelve enigmas y problemas lógicos que tendrán relación con estos lugares y su historia.

Gracias a esta aplicación se incentivará el turismo atrayendo a usuarios de la aplicación a dicha región y contribuyendo a un crecimiento económico de la misma.

1.3 Contexto del proyecto

Este proyecto de fin de grado en realidad formaría parte de un proyecto aún más grande, donde sería necesario el desarrollo de varios sistemas para total completitud [Ilustración 3].

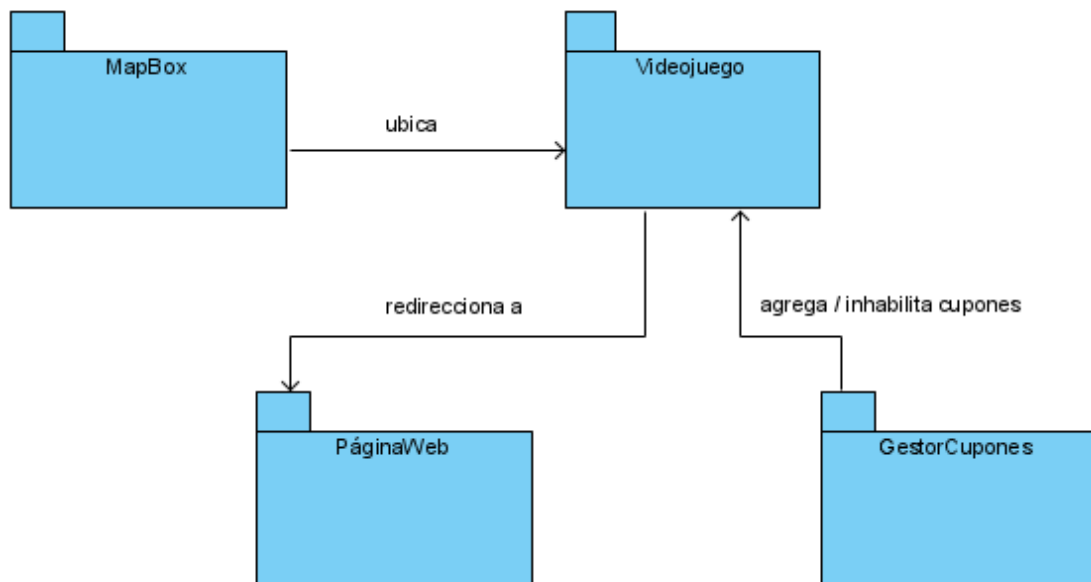


Ilustración 3: Diagrama de relación de sistemas

Para el alcance de nuestro proyecto, esta memoria se centrará en la parte correspondiente al videojuego, pilar fundamental dentro de este conjunto de sistemas.

El videojuego hará uso del sistema de MapBox [\[4\]](#), herramienta de mapas que se utilizará para situar al usuario y los puzles en la pantalla.

Además, el videojuego se encargará de redireccionar al jugador a la página web de la empresa, donde encontrará información relevante sobre el monumento o sitio turístico en el que se basa cada puzle una vez dada una respuesta correcta.

Por último, también se dará interacción entre este videojuego y otra aplicación externa desarrollada por la misma empresa, que ayudará a las empresas suscritas a la aplicación a gestionar sus cupones de descuento y publicidad.

Capítulo 2: Características del videojuego

2.1 Título del videojuego

El título de este videojuego (también título de lo que sería la empresa) es **CoolToTrip**.

Su traducción al español es “**Genial para viajar**”. Además, es un juego de palabras homófonas que vienen de **cultu - trip -> culture trip**, que en español significa “**viaje cultural**”.

2.2 Género del videojuego

El videojuego a desarrollar es un “**serious game**”, es decir, un videojuego cuyo propósito final no es el mero entretenimiento del usuario.

En este caso, este serious game será del género puzle. Estará basado en la localización del usuario y enfocado al fomento de lugares culturales de una región, siendo su propósito principal el aprendizaje por parte del usuario de parte de la historia de ese territorio a través de puzles que tendrá que ir resolviendo mientras conoce estos sitios.

Además de esto, se ha incluido un enfoque de “**in-game advertising**”. Para ello, el videojuego cuenta con un sistema de cupones que dan **publicidad** a aquellos locales y sitios turísticos de pago que se suscriban a la aplicación.

2.3 Clasificación PEGI

“Las clasificaciones por edades son sistemas que se utilizan para garantizar que el contenido de entretenimiento, como los juegos, pero también las películas, los programas de televisión o las aplicaciones móviles, esté claramente etiquetado con una recomendación de edad mínima basada en el contenido que tienen. **Estas clasificaciones por edad ofrecen orientación a los consumidores, en particular a los padres, para ayudarlos a decidir si comprar o no un determinado producto a un niño.**” [\[5\]](#)

Basándonos en las descripciones de las etiquetas PEGI disponibles en la web de *pegi.info* [6], podemos clasificar nuestro videojuego con las siguientes etiquetas:



Ilustración 4: Clasificación de edad PEGI [6]



Ilustración 5: Aviso PEGI "Incluye Compras" [6]

2.4 Influencia de otros videojuegos

Para una mejor comprensión del estilo de este videojuego se mencionarán otros videojuegos famosos que se asemejan a lo que se quiere llegar a crear con este proyecto:

- **Pokémon GO** [7]

Este videojuego para móviles basado en localización de la famosa franquicia de Pokémon Company y desarrollado por Niantic salió el 6 de julio de 2016 en las principales tiendas de aplicaciones móviles [Ilustración 6].



Ilustración 6: Imágenes de Pokémon GO [8]

En este videojuego el usuario tiene como objetivo la captura de estos Pokémon mientras se desplaza en el mundo real recogiendo a su vez objetos en ciertos puntos de interés de la ciudad.

- **Harry Potter: Wizards Unite** [9]

Este videojuego [Ilustración 7] salió al mercado el 25 de abril de 2018 de la mano de Niantic y WB Games San Francisco bajo la licencia de Portkey Games.



Ilustración 7: Imágenes de Harry Potter Wizards Unite [10]

Es un juego basado en localización donde se consiguen ciertos objetos en puntos concretos de la ciudad mientras el usuario debe de realizar ciertas acciones como lanzar hechizos.

- **Profesor Layton** [11]

Esta saga de videojuegos del género puzzle empezó su trayectoria con el lanzamiento de su primer videojuego Profesor Layton y La Villa Misteriosa para la consola Nintendo DS. Salió a la venta en Japón 15 de febrero de 2007 desarrollado y publicado por Level-5.



Ilustración 8: Pueblo de Profesor Layton [12]



Ilustración 9: Puzle de Profesor Layton [12]

En esta serie de videojuegos se cuenta una historia principal mientras que el jugador inspecciona los diferentes escenarios en 2D en busca de monedas y puzles que resolver [Ilustración 8][Ilustración 9].

2.5 La cámara

Para este proyecto, se usará una cámara en perspectiva **Top-Down**, también llamada vista de pájaro. Esta cámara irá **asociada a la posición del usuario** en la escena de forma que esta siga al jugador. De esta forma se verá mejor la localización del usuario en la pantalla principal donde se muestra el mapa.

Esta perspectiva es la usada comúnmente en los videojuegos basados en localización, como se puede ver, por ejemplo, en Pokémon GO [7].

A la hora de resolver puzles, esta cambiará mostrándose claramente la interfaz y el puzle en la misma pantalla.

2.6 Gráficos

Para los gráficos de este proyecto, se usarán un estilo de **gráficos 2.5D** para el mapa principal donde se mostrará el indicador de la posición del jugador, la guía y las señales de puzle. También se incluirá un **estilo 2D o 2.5D para la resolución de los puzles** según convenga.

En cuanto a la estética del videojuego, se busca un estilo sin necesidad de tanto realismo. En este caso se busca un **estilo cartoon** que haga más atractivo el videojuego.

2.7 Interfaz

Para la interfaz se ha pensado en un diseño lo más **simple** posible, intentando que no haya muchas opciones a la vez en pantalla.

Esta interfaz se irá mostrando a lo largo del proceso de desarrollo del proyecto en la fase de diseño de las diferentes funcionalidades que se irán añadiendo.

A continuación, se verán las principales ventanas de la interfaz del videojuego hechas a mano en las primeras fases del proyecto:

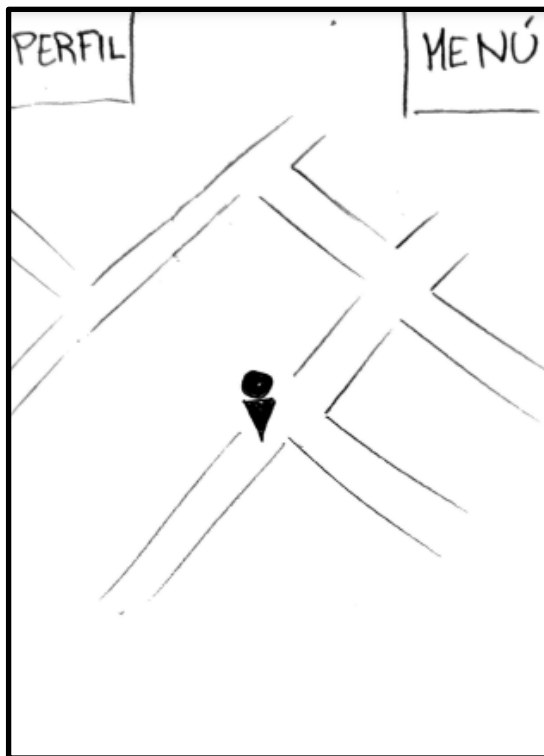


Ilustración 10: Prototipo de interfaz de mapa



Ilustración 11: Prototipo de interfaz de perfil

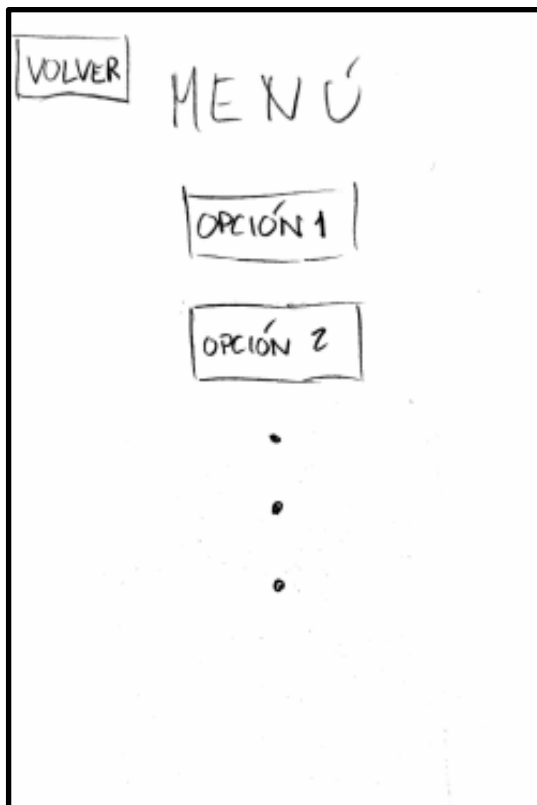


Ilustración 12: Prototipo de interfaz de menú



Ilustración 13: Prototipo de interfaz de puzzle

2.8 Modos de juego

A continuación, se analizarán los modos de juego disponibles y las posibles acciones dentro de estos:

- **Mapa:** Modo principal de la aplicación donde se muestra la posición del jugador en un mapa.

Acciones disponibles:

- Ir a Perfil o Menú
 - Ir a resolución de puzle encontrado
- **Resolver Puzle:** Escena de resolución de puzles.

Acciones disponibles:

- Volver al mapa
 - Dar una respuesta
 - Confirmar la respuesta
- **Perfil:** Se muestra información del usuario.

Acciones disponibles:

- Volver al mapa
- **Menú:** Se muestran varias opciones.

Acciones disponibles:

- Volver al mapa
 - Elegir opción
- **Selección de región:** Se muestran las regiones disponibles de la aplicación.

Acciones disponibles:

- Volver al menú
 - Elegir territorio

- **Selección de ruta:** Se muestran las rutas disponibles.

Acciones disponibles:

- Volver a selección de región
 - Elegir ruta
- **Ranking:** Se muestra el puesto del usuario según su puntuación y los 10 primeros puestos de un nivel territorial seleccionado.

Acciones disponibles:

- Volver al menú
 - Cambiar nivel territorial
- **Tienda de cupones:** Muestra el catálogo de cupones disponibles.

Acciones disponibles:

- Volver al menú
 - Seleccionar un cupón
- **Compra de cupón:** Muestra información de compra del cupón seleccionado.

Acciones disponibles:

- Volver a la tienda de cupones
 - Comprar cupón
- **Mis Cupones:** Se muestran los cupones conseguidos por el jugador.

Acciones disponibles:

- Volver al menú
 - Seleccionar un cupón
- **Ver info cupón:** Se muestra la información del cupón seleccionado.

Acciones disponibles:

- Volver a Mis Cupones
 - Mostrar el código QR del cupón

- **Mostrar QR:** Se muestra el código QR del cupón seleccionado.

Acciones disponibles:

- Volver a Mis cupones

A continuación, se muestra el diagrama de flujo donde se puede observar los cambios de estos modos de juego:

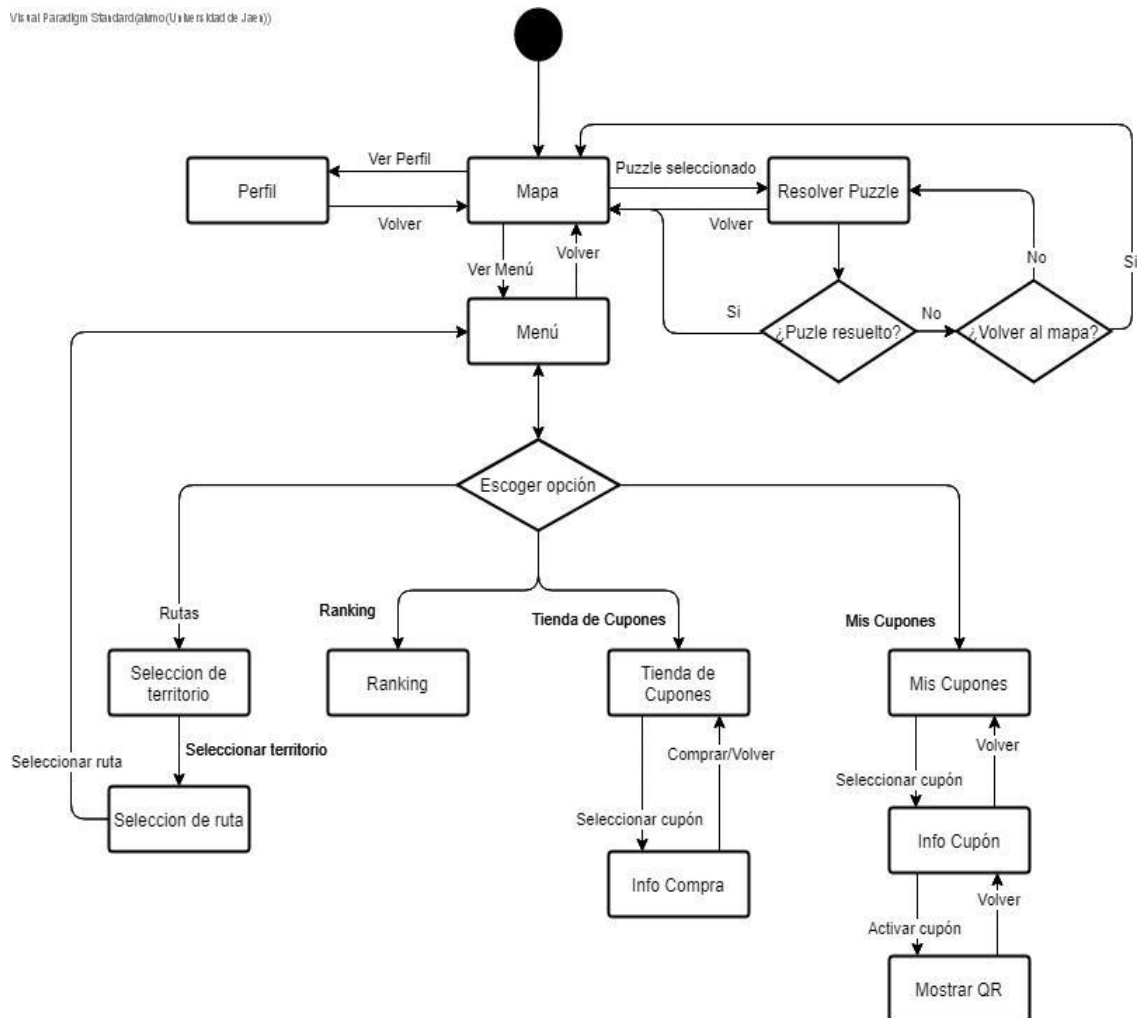


Ilustración 14: Flowchart

Enlace a la imagen: <Imágenes\Flowchart.jpg>

2.9 Mecánicas principales

- Si el jugador recorre una distancia determinada, conseguirá una moneda que puede canjear por una pista.
- Si el jugador elige una ruta sin empezar, se mostrará una notificación de “Nueva Ruta” mostrando el número de puzzles por resolver de dicha ruta. Se cambiará la ruta activa a la ruta seleccionada.
- Si el jugador elige una ruta ya empezada se le mostrará una notificación de “Continuar Ruta” mostrando el número de puzzles que le quedan por resolver de dicha ruta. Se cambiará la ruta activa a la ruta seleccionada.
- Si el jugador elige una ruta ya acabada, se le mostrará una notificación de “Ruta ya acabada” y no se cambiará la ruta activa.
- Al escoger una ruta, la flecha que deberá guiar al jugador apuntará en la dirección del siguiente puzzle a resolver.
- Al estar dentro de un radio determinado, el puzzle aparecerá en el mapa, pudiendo ser seleccionado para la resolución del mismo.
- Si el jugador tiene al menos una moneda, podrá desbloquear la siguiente pista de un puzzle.
- Si el jugador falla el puzzle, la cantidad de puntos máximos del puzzle se irá decrementando hasta que el número de intentos sea igual o menor que el máximo de intentos según la fórmula:
$$Puntuacion = maxPuntuacion / (numIntentos + 1)$$
- Si el jugador falla, podrá volver al mapa o intentarlo de nuevo.
- Si el jugador acierta, se le sumarán los puntos conseguidos a su puntuación máxima, que determina su puesto en el ranking, y a sus puntos canjeables.
- Si el jugador acierta, podrá consultar más información acerca del lugar donde se encuentra el puzzle.
- Si el puzzle resuelto es el último de la ruta, se le sumará una puntuación extra por completar la ruta. Esta puntuación será igual al número de puzzles de la ruta multiplicado por 100.

- El jugador podrá comprar un cupón en la tienda de cupones si la cantidad de puntos canjeables es mayor o igual que los puntos necesarios para la compra del cupón.
- Si el jugador compra un cupón, se le restarán los puntos necesarios para la compra de cupón a sus puntos canjeables.
- Si el jugador muestra el código QR de un cupón y la persona encargada lo activa, dicho cupón desaparecerá del apartado Mis Cupones.
- Si el jugador tiene una puntuación que se encuentra entre las diez puntuaciones más altas de su territorio, aparecerá en su pantalla de ranking y en la de los demás jugadores de ese territorio mostrando su puesto y puntuación.

2.10 Objetivos

El objetivo del jugador es completar los diversos puzles que conforman la ruta elegida.

Además, con la resolución de puzles, se ganarán puntos con los que se calculará en qué puesto se encuentra dentro del ranking del territorio, dando así cierta competitividad al juego.

Al completar cada puzle, se le dará información extra sobre el lugar donde se encuentra.

Al completar una ruta, se le darán puntos extra.

2.11 Economía

La economía de este videojuego recae directamente en los **puntos conseguidos** con cada puzle, así como en las monedas canjeables.

El jugador deberá tener cuidado a la hora de resolver los puzles, ya que **fallar en un puzle supone el decremento de los puntos** que ese puzle ofrece al ser resuelto.

Los puntos conseguidos, además de para el cálculo del puesto del jugador, servirán como moneda de cambio para la **compra de cupones** de descuento en los que el jugador esté interesado.

Por otra parte, el jugador tendrá **monedas que podrá canjear a cambio de pistas** para poder resolver un puzle. Estas monedas las conseguirá andando un número de kilómetros determinado. Cada puzle tiene 3 pistas, por lo que el jugador tendrá que tener esto en cuenta a la hora de gastar o no una moneda.

2.12 Criterios de éxito

A nivel de puzle, dependiendo del puzle en cuestión, el jugador tendrá que elegir entre varias opciones la opción correcta o conseguir el código o palabra clave que tendrá que escribir para resolver el puzle.

Estos criterios de éxito en el puzle se irán haciendo más variados a medida que incrementemos el número de tipos de puzle en la aplicación.

Se tiene también como criterio de éxito el completar todos los puzles de una ruta.

2.13 Criterios de fracaso

A nivel de puzle, el jugador habrá fracasado si la respuesta dada no es la correcta para el puzle en cuestión.

Capítulo 3: Tecnología y herramientas

3.1 Motor de Videojuegos

Para el desarrollo de este prototipo se ha optado por usar el motor de videojuegos Unity [\[13\]](#).

Unity es un motor de videojuegos muy usado actualmente por los desarrolladores de videojuegos debido a la gran cantidad de ventajas que tiene con respecto a otros motores gráficos. A continuación, se hace una enumeración de varias ventajas y desventajas de este motor de videojuegos.

Ventajas:

- Este motor está basado en componentes, lo que ayuda en la reutilización del código de scripts en diferentes entidades.
- Existe gran cantidad de vídeos explicativos y documentación, lo que ayuda considerablemente a la hora de desarrollar cualquier proyecto con este motor.
- Tiene una tienda de assets, donde se pueden encontrar multitud de soluciones ya creadas y muchas de ellas se pueden utilizar de forma gratuita.
- Permite el desarrollo en diferentes plataformas móviles de forma gratuita como Android e IOS.

Desventajas:

- En cuanto al rendimiento, Unity resulta ser algo exigente, manteniendo varias tareas activas en el equipo y haciendo que este pueda llegar a calentarse bastante.
- El espacio ocupado por el proyecto puede ser muy alto, sobre todo si se usan soluciones externas como los assets de la tienda.

Además de Unity, se han valorado otros motores de videojuegos como Unreal Engine [\[14\]](#), Source Engine 2 [\[15\]](#) o CryEngine [\[16\]](#).

Unreal Engine [\[14\]](#) era otra opción muy atractiva para el desarrollo del presente proyecto (gran potencia, uso gratuito...), pero dado que el estudio de la herramienta de mapas acaparará gran parte del tiempo, se ha estimado oportuno usar un motor de videojuegos ya conocido como Unity, motor de videojuegos que se usó en la asignatura de “Desarrollo de videojuegos” de este grado.

3.2 Plataforma de uso

Para este proyecto se ha optado usar **Android** con una **versión 8.0 o superior**. Siendo Android el sistema operativo móvil más utilizado, se ha optado por este para conseguir la mayor difusión y uso de esta aplicación.

También se ha barajado en un primer momento la opción de usar IOS, pero se ha preferido el desarrollo en Android por lo que se ha mencionado anteriormente.

3.3 Herramienta de mapas: MapBox

Para este proyecto es necesario utilizar alguna herramienta que nos permita visualizar mapas, así como movernos por ellos.

Tras investigar sobre este tipo de herramientas, las dos opciones que podrían ser de más utilidad al proyecto son Google Maps [\[17\]](#) o MapBox [\[4\]](#).

Google Maps [\[17\]](#) presenta una solución para videojuegos que cuesta **10\$ al mes por cada 1000 usuarios**. Aunque Google nos da 200\$ todos los meses para comenzar en el desarrollo, hace falta ingresar una tarjeta de crédito para poder hacer cargos en caso de que se pase dicho límite.

En el caso de MapBox [\[4\]](#), la herramienta da acceso gratuito siempre que no se sobrepase el **límite de 25.000 usuarios y 50.000 consultas**. Además, esta herramienta hará más fácil el trabajo en Unity dado que ya trae consigo algunas soluciones y prefabs para los distintos usos de la herramienta, siendo uno de ellos un prefab para un “Videojuego basado en localización”, que es precisamente el tipo de videojuego que se va a desarrollar.

MapBox es una herramienta que, si bien tiene una curva de aprendizaje algo elevada, llega a tener un paquete de recursos muy completo y variado.

Debido a lo anterior, además de buenas referencias por parte de otros compañeros informáticos, se ha decidido hacer uso de **MapBox**.

3.4 Otros programas usados

Además de Unity, para este proyecto se utilizarán otros programas que ayudarán al desarrollo del mismo:

- **GIMP v2.10** [\[18\]](#): Herramienta de edición de imágenes que se usará para el diseño del logo principal de la aplicación.
- **Inkscape** [\[19\]](#): Herramienta de creación de imágenes vectoriales que se usará para la creación de algunas imágenes de los puzles.
- **XMind 2020** [\[20\]](#): Herramienta para la creación de mapas mentales y diagramas que se usará para la presentación de requisitos y tareas del proyecto.
- **Visual Paradigm v16.2** [\[21\]](#): Herramienta de edición de diagramas que se usará para la creación de los diagramas de clases de diseño y secuencia del proyecto.
- **Blender v2.92** [\[22\]](#): Herramienta de modelado 3D que se usará para crear modelos básicos para indicadores como posición del jugador o posición del puzle.
- **JustInMind** [\[23\]](#): Herramienta de elaboración de mockups y prototipos que se usará para el diseño del prototipo inicial del proyecto.

Capítulo 4: Proceso de desarrollo de la aplicación

4.1 Metodología: Proceso Unificado

Para este proyecto se ha decidido usar la metodología de desarrollo del **Proceso Unificado** [24], aconsejada por el tutor, la cual facilitará el proceso de desarrollo dividiendo este en diferentes iteraciones.

El Proceso Unificado o RUP es una metodología de desarrollo la cual se centra en los casos de uso extraídos de los requerimientos. Estos casos de uso son los que guiarán el flujo de trabajo. En cada iteración se completan unas tareas relacionadas con varios casos de uso para la inclusión de nueva funcionalidad en el prototipo.

En este proyecto cada iteración tendrá una **duración de 2 semanas**. Además, el tutor también podrá comprobar el trabajo realizado a partir de las tareas que se han asignado para las diferentes iteraciones.

El proyecto, siguiendo esta metodología, consta de cuatro fases:

- **Inicio:**

Fase que da comienzo al desarrollo del proyecto definiéndose el alcance y los requisitos que servirán de base para la funcionalidad del mismo.

- **Elaboración:**

En esta fase se llega a una visión más completa del proyecto a realizar, la implementación iterativa del núcleo de la aplicación, nuevos requisitos y se ajustan las estimaciones.

- **Construcción:**

Esta abarca la evolución hasta convertirse en producto listo incluyendo requisitos mínimos.

- **Transición:**

En esta fase final, el programa debe estar listo para ser probado, instalado y utilizado por el cliente sin ningún problema. Una vez finalizada esta fase, se debe comenzar a pensar en futuras novedades para la misma.

4.2 Análisis del sistema

En este apartado, analizaremos el sistema a desarrollar en el proyecto definiendo su alcance y requisitos, y estimando su duración y costes.

4.2.1 Definición del proyecto

El videojuego a desarrollar será un serious game que tendrá la finalidad de dar a conocer lugares de interés cultural de un determinado territorio.

El juego consistirá en la **búsqueda y resolución de puzles** que tendrán relación con estos lugares. Para ello, el jugador deberá ingresar en la aplicación y elegir una determinada ruta disponible en el territorio deseado.

Una vez elegida la ruta, se le mostrará información acerca de la ruta elegida, como el número de puzles que la componen.

El jugador deberá guiarse a través del mapa de la ciudad mostrado en la pantalla con la ayuda de una “guía”. Esta guía, **indicará la dirección en la que se encuentra el siguiente puzle** a resolver. Una vez que el usuario esté cerca de dicho puzle, este aparecerá en la pantalla y podrá ser seleccionado, pasando entonces a la resolución del mismo.

Los puzles a resolver serán muy variados. Por ejemplo, habrá puzles en los que el usuario deberá elegir entre varias opciones basándose en las imágenes que aparecen o en la explicación que da el enunciado. También puede haber puzles donde la respuesta será un código o palabra que el jugador debe conseguir.

Para ayudar al jugador con la resolución de los puzles, dichos puzles también **tendrán disponibles unas pistas** que el jugador podrá desbloquear gastando monedas. Estas monedas podrá conseguirlas recorriendo cierta distancia.

Como recompensa, tras resolver un puzle, se le mostrará al jugador más información acerca del lugar donde se ubica dicho puzle. También conseguirá puntos al completar un puzle o ruta, los cuales determinarán su posición dentro del ranking de jugadores del territorio del que proviene.

Además de todo esto, también se estudia la posibilidad de añadir a la aplicación un **sistema de cupones**, dando a estos puntos una función extra.

El jugador podría canjear sus puntos, conseguidos al resolver puzles y completar rutas, por cupones de descuento que ofrezcan otras empresas en la tienda de cupones de la aplicación. Esto permitirá que cualquier empresa que ofrezca un producto o servicio se pueda publicitar en la aplicación

4.2.2 Extracción de requisitos

Para definir el listado de requisitos de nuestro proyecto se clasificarán en Funcionales y No Funcionales.

Además, también se clasificarán según su prioridad mediante el método MoSCoW [\[25\]](#), con las siguientes etiquetas:

- Imprescindible (Must Have)
- Importantes (Should Have)
- Interesante (Could Have)
- Opcional (Won't Have and Would Have)

Número	Requisito	Prioridad	Tipo
1	El sistema deberá mostrar al jugador el mapa del territorio mientras lo recorre	Imprescindible	Funcional
2	El jugador podrá escoger una ruta dentro de un territorio concreto para su realización	Imprescindible	Funcional
3	El sistema mostrará la una notificación con información sobre la ruta escogida	Importante	Funcional
4	El sistema deberá mostrar una "guía" que ayude al usuario a encontrar los puzles	Imprescindible	Funcional
5	El sistema mostrará permitirá la selección de un puzle siempre que el jugador se encuentre dentro de una distancia determinada	Importante	Funcional
6	El jugador podrá canjear una moneda a cambio de una pista que le permita resolver el puzle en cuestión	Imprescindible	Funcional
7	El jugador podrá conseguir monedas andando una distancia determinada	Imprescindible	Funcional
8	El sistema deberá de sumar la puntuación asociada al puzle si el jugador resuelve este	Imprescindible	Funcional
9	El sistema deberá de decrementar la puntuación asociada al puzle hasta un mínimo si el jugador falla con sus respuestas	Interesante	Funcional
10	El jugador podrá consultar su información personal dentro de la aplicación	Imprescindible	Funcional

11	El jugador podrá ver en qué puesto se encuentra dentro del ranking del territorio al que pertenece	Imprescindible	Funcional
12	El jugador podrá canjear sus puntos a cambio de cupones de descuento que ofrezcan otras empresas	Interesante	Funcional
13	El tiempo de respuesta de la aplicación permitirá la interacción de forma fluida	Imprescindible	No funcional
14	El mapa se visualizará de forma clara para que el jugador pueda ubicarse fácilmente	Importante	No funcional
15	La aplicación deberá utilizar la herramienta MapBox para la creación de mapas	Imprescindible	No funcional
16	El dispositivo móvil debe de tener sistema Android 8.0 o superior	Imprescindible	No funcional

Tabla 1: Requisitos

Tras clasificar los diferentes requisitos, se construirá el Requirements Breakdown Structure (RBS) [Ilustración 15], que incluirá los anteriores requisitos y algunas funciones importantes para la consecución de los mismos.

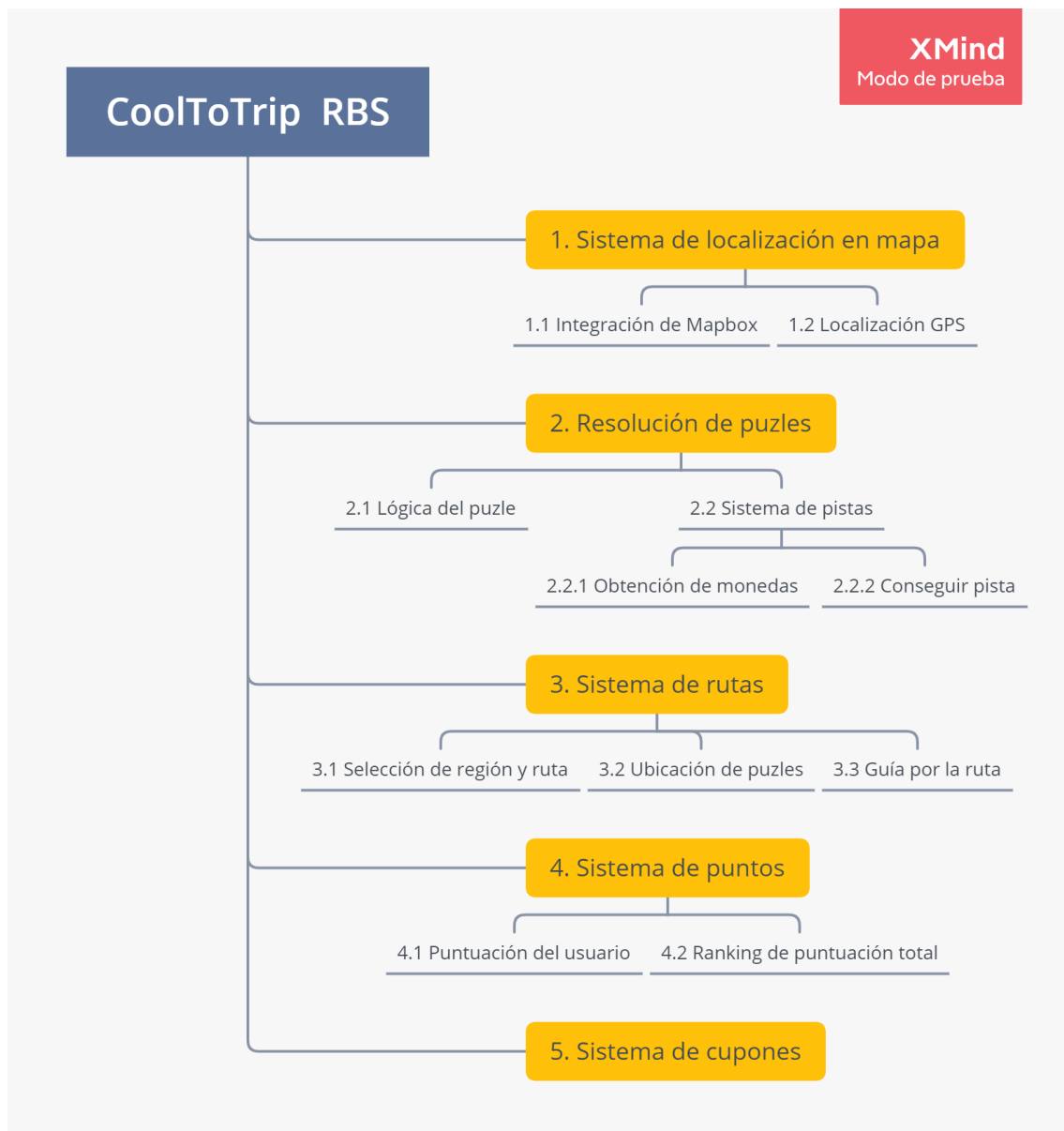


Ilustración 15: RBS

Enlace a la imagen: [Imágenes\RBS.png](#)

4.2.3 Declaración general del proyecto

En este documento se pretende conseguir la aprobación por parte del cliente y el equipo de desarrollo. Una vez aprobado se podrá pasar a la fase de planificación de tareas y costes.

DECLARACIÓN GENERAL DEL PROYECTO	Nombre de proyecto	Proyecto N°	Gestor de proyecto
	Cool To Trip		Álvaro Morales García
Problema / Oportunidad Se da una bajada muy importante en el sector del turismo dado el estado de pandemia. Se quiere dar solución a este problema fomentando el turismo a través de un videojuego.			
Meta Obtener el prototipo de un videojuego que fomente el turismo al final de la fase de desarrollo de este proyecto cumpliendo una serie de requisitos dados por el cliente.			
Objetivos Obtener un prototipo básico que contenga al menos lo siguiente: ● Integración de la herramienta MapBox ● Un sistema de localización del jugador ● Varios puzles a resolver ● Un sistema de pistas ● Un sistema de elección de rutas ● Un sistema de puntuación por región ● Un sistema de cupones			

Criterios de éxito • Que al menos un 75% de los usuarios que han descargado la aplicación la valoren positivamente. • Que al menos un 25% de las empresas del territorio, en este caso Jaén, se unan a la aplicación.			
Suposiciones, riesgos y obstáculos • Posibilidad de que el equipo informático usado para el proceso de desarrollo tenga algún fallo irreparable. • Disposición constante de conexión a internet. • Aprender en poco tiempo lo necesario de la herramienta MapBox para su integración en el proyecto.			
Hecho por	Fecha	Aprobado por	Fecha
Álvaro Morales García	25/02/2021	D. Juan Roberto Jiménez Pérez	26/02/2021

Tabla 2: Declaración general del proyecto

4.3 Planificación del proyecto

Para facilitar el proceso de desarrollo del proyecto, se dividirán los requisitos recogidos anteriormente en el RBS [Ilustración 15] y se dividirán en tareas que compondrán el Work Breakdown Structure (WBS) [Ilustración 16]. Con esto, se conseguirá una mayor precisión a la hora de estimar tiempos y costes del proyecto, así como una mejor organización a la hora de abordar las diferentes funcionalidades que debe de tener nuestra aplicación.

4.3.1 Definición de tareas

Tras esto, se construye el WBS [Ilustración 16], donde se recogen todas las tareas derivadas de los requisitos obtenidos.

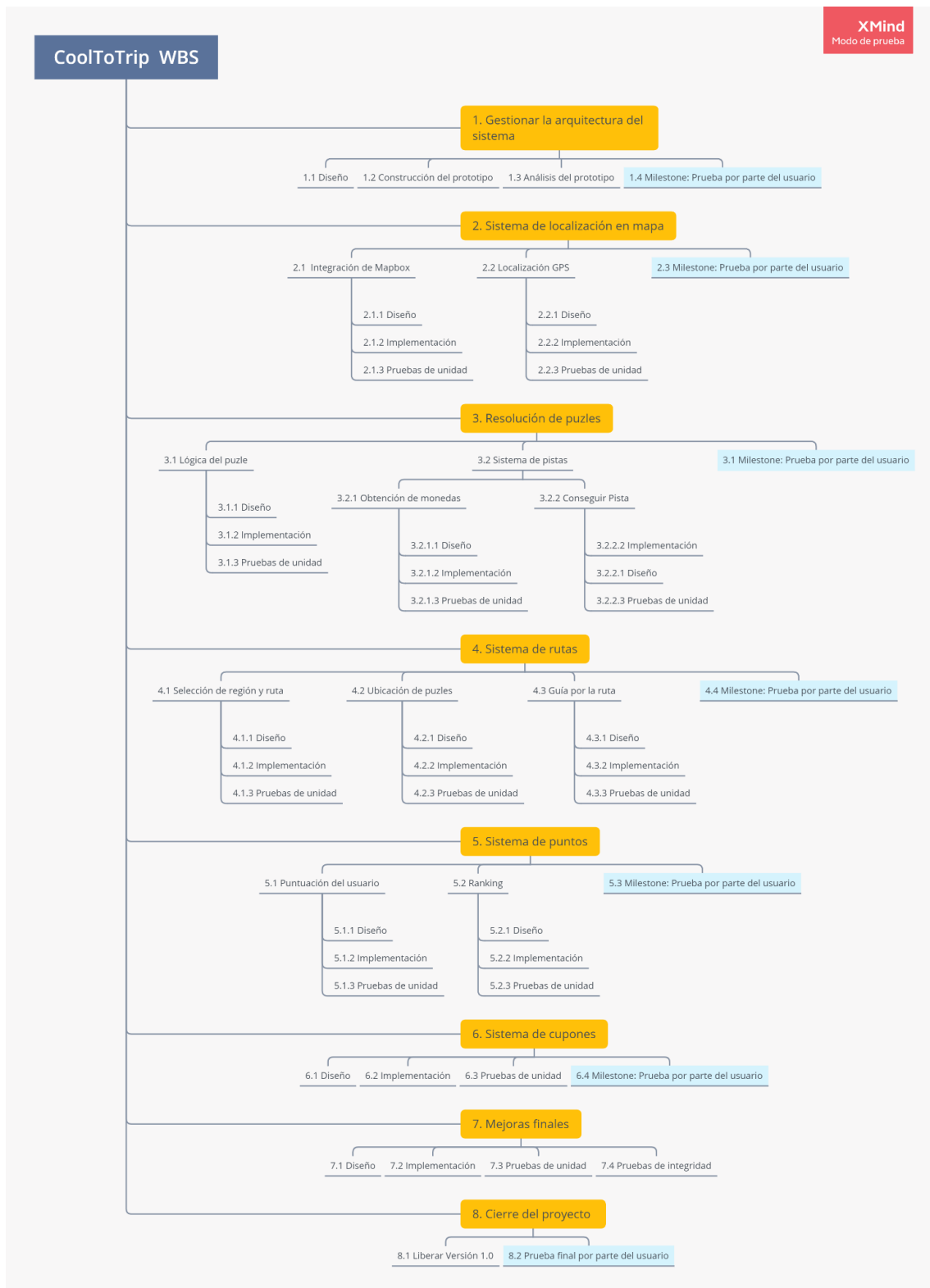


Ilustración 16: WBS

Enlace a la imagen: <Imágenes\WBS.png>

4.3.2 Estimación de tiempos

A continuación, se hará una estimación en horas del tiempo necesario aproximado que llevará cada tarea del WBS, obteniendo la cantidad de horas necesarias para la finalización de este proyecto (sin contar la creación de la memoria):

CoolToTrip WBS	
Tareas	Estimación de tiempo en horas
1. Gestionar la arquitectura del sistema	
1.1 Diseño	2
1.2 Construcción del prototipo	2
1.3 Análisis del prototipo	2
1.4 Milestone: Prueba por parte del usuario	1
2. Sistema de localización en mapa	
2.1 Estudio de la herramienta: Mapbox	20
2.2 Integración de Mapbox	
2.2.1 Diseño	10
2.2.2 Implementación	15
2.2.3 Pruebas de unidad	5
2.3 Localización GPS	
2.3.1 Diseño	4

2.3.2 Implementación	8
2.3.3 Pruebas de unidad	2
2.4 Pruebas de integridad	2
2.5 Milestone: Prueba por parte del usuario	3
3. Resolución de puzles	
3.1 Creación de los puzles	
3.1.1 Diseño	20
3.1.2 Implementación	15
3.1.3 Pruebas de unidad	10
3.2 Sistema de pistas	
3.2.1 Obtención de monedas	
3.2.1.1 Diseño	2
3.2.1.2 Implementación	5
3.2.1.3 Pruebas de unidad	2
3.2.2 Conseguir Pista	
3.2.2.1 Diseño	1
3.2.2.2 Implementación	2
3.2.2.3 Pruebas de unidad	2
3.3 Pruebas de integridad	3
3.4 Milestone: Prueba por parte del usuario	2

4. Selección de ruta	
4.1 Ubicación de puzles en la ruta	
4.1.1 Diseño	2
4.1.2 Implementación	4
4.1.3 Pruebas de unidad	4
4.2 Selección de región y ruta	
4.2.1 Diseño	3
4.2.2 Implementación	5
4.2.3 Pruebas de unidad	4
4.3 Pruebas de integridad	5
4.4 Milestone: Prueba por parte del usuario	2
5. Guía por la ruta	
5.1 Diseño	3
5.2 Implementación	5
5.3 Pruebas de unidad	2
5.4 Pruebas de integridad	3
5.4 Milestone: Prueba por parte del usuario	5
6. Sistema de puntos	
6.1 Puntuación del usuario	
6.1.1 Diseño	2

6.1.2 Implementación	3
6.1.3 Pruebas de unidad	2
6.2 Ranking	
6.2.1 Diseño	3
6.2.2 Implementación	4
6.2.3 Pruebas de unidad	3
6.3 Pruebas de integridad	5
6.4 Milestone: Prueba por parte del usuario	4
7. Sistema de cupones	
7.1 Diseño	4
7.2 Implementación	6
7.3 Pruebas de unidad	4
7.4 Pruebas de integridad	6
7.5 Milestone: Prueba por parte del usuario	3
8. Mejoras finales	
8.1 Diseño	3
8.2 Implementación	5
8.3 Pruebas de unidad	1
8.4 Pruebas de integridad	6
9. Cierre del proyecto	

9.1 Liberar Versión 1.0	1
9.2 Prueba final por parte del usuario	15

Tabla 3: Estimación de tiempo de las tareas

Tabla de iteraciones			
Iteraciones	Grupo de tareas	Horas	Etapas
Iteración 1	1 y 2.1	27	Inicio
Iteración 2	2.2 y 2.3	49	Elaboración
Iteración 3	3.1	45	Elaboración
Iteración 4	3.2, 3.3, 3.4 y 4	48	Construcción
Iteración 5	5 y 6	44	Construcción
Iteración 6	7, 8 y 9	54	Construcción y Transición
Horas totales estimadas		267	

Tabla 4: Estimación de tiempo de las iteraciones

Como podemos ver, el proyecto ocupará unas **267 horas**. El resto de horas hasta llegar a las 300 horas se dedicarán a la elaboración del contenido de esta memoria.

Habiendo dividido el proyecto en **6 iteraciones de 2 semanas** cada una, se obtiene que se debe trabajar en este unas **3,20 horas al día todos los días** para completarlo en dicho tiempo.

4.3.2 Estimación de recursos y costes

Para la creación de este proyecto es necesario estimar los costes que conllevarán el desarrollo del mismo. Para el cálculo de costes se harán los cálculos de **coste por hora** de los recursos **materiales y personal disponible**.

Se comenzará por estimar el coste de los **recursos digitales**. En este caso el coste se calculará **por mes**, ya que el precio será fijo independientemente de las horas de uso.

En el caso de **XMind** [\[20\]](#), no hay disponible una suscripción por meses, por lo que se calculará el coste con el plan para 6, siendo este el más corto. En este plan, el precio en euros sería de **32,80€ (40 dólares)**, que al mes sería de **5,46€**.

Con el programa **Visual Paradigm** [\[21\]](#) haremos los cálculos teniendo una suscripción mensual como “Modeler” siendo esta de **4,92€ (6 dólares) al mes**.

Dado que estos programas también pueden ser usados para otros proyectos, se dividirá este gasto entre el número de proyectos abiertos que actualmente tenga la empresa. Para hacer un cálculo estimado, supondremos que la empresa tiene **dos proyectos más** además de este (la página web y la aplicación de gestión de cupones) que usarán también algunos de estos programas (XMind y VisualParadigm), por lo que el coste por mes calculado anteriormente **se dividirá entre tres**.

Programa	Coste/mes	Coste total
GIMP	0€	0€
Inkscape	0€	0€
XMind	1.82€	5.46€
Visual Paradigm	1.64€	4.92€
JustInMind	0€	0€
Blender	0€	0€
Unity*	0€	0€
MapBox*	0€	0€
TOTAL:		10.38€

Tabla 5: Recursos digitales del prototipo

*Unity y MapBox son gratuitos hasta un límite de ingresos o usuarios. En el caso de Unity [\[13\]](#), será gratuito si se ha ingresado menos de 82.022,50€ (100.000\$) en los últimos 12 meses. MapBox SDK para Unity [\[4\]](#) también será gratuito siempre y cuando no se sobrepase de 25.000 usuarios y las 50.000 consultas de mapas.

Estos cálculos se han hecho teniendo en cuenta el desarrollo de este proyecto **sin licencias académicas**, ya que en ese caso el coste sería de 0€.

Para continuar con la estimación de costes, se debe de calcular los costes materiales de este proyecto. Para este apartado se tendrán en cuenta el coste del equipo de desarrollo, así como el material de oficina utilizado para ello.

El equipo de desarrollo se compone por **un ordenador de sobremesa, un portátil y una tableta gráfica**. Estos equipos se irán alternando ya que el espacio de trabajo no será el mismo todos los días. Se estima que la mitad del tiempo se utiliza el ordenador de sobremesa y la otra mitad el portátil.

El ordenador de sobremesa tiene un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. Dado que el proyecto son unas 300 horas, el tiempo empleado en el desarrollo con el

ordenador de sobremesa es de 150 horas. **El coste calculado del equipo por hora es de 0,021€**, por lo que el coste total para este proyecto es de **3,15€**.

El ordenador portátil tiene un coste de 800€ y una vida útil de 4 años. Haciendo el cálculo anterior, **el coste del equipo por hora es de 0,023€**, por lo que el coste total es de **3,42€**.

La tableta gráfica tiene un coste de 75€ con una vida útil de 4 años. Esta tableta se usará solo para la edición de algunas imágenes y su uso es como máximo de unas 3 horas. **El coste de la tableta por hora es de 0,002€** por lo que prácticamente el coste total **no llega al céntimo**.

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño de ventanas se aproxima a **4€**.

Para el cálculo del consumo eléctrico se ha hecho una estimación del gasto con la herramienta online OuterVision Power Supply Calculator [\[26\]](#).

Para el primer equipo se ha obtenido un gasto calculado por la herramienta de **435,98€ al año** si el equipo estuviera encendido siempre, quedando en **0,049€ la hora**. Multiplicando por la mitad de las horas totales del proyecto resulta en **6,644€ en total**

Para el segundo equipo se ha obtenido un gasto calculado por la herramienta de **222,10€ al año** si el equipo estuviera encendido siempre, quedando en **0,025€ la hora**. Multiplicando por la mitad de las horas totales del proyecto resulta en **3,385€ en total**

La suma total sería de **10,029€** para todo el prototipo.

Para el cálculo de gasto en fibra óptica, se ha considerado la contratación actual de Jazztel de fibra óptica 100mb por **28,95€ al mes** [\[27\]](#).

Dado que es un gasto mensual que no depende de su uso en horas, se multiplicará por los tres meses aproximados que se tardará en crear el prototipo, quedando en 86,76€. Al ser utilizado por los demás proyectos, se divide entre tres dando en total un gasto de **28,95€ en total** para este proyecto.

Concepto	Uso	Precio total
Ordenador sobremesa	Desarrollo del proyecto	3,15€
Ordenador portátil	Desarrollo del proyecto	3,42€
Tableta gráfica	Edición de imágenes	0€
Material de oficina	Creación de diagramas e interfaces	4€
Electricidad	Consumo del equipo de desarrollo	10,03€
Internet	Consultas, investigación y desarrollo	28,95€
Total:		49,55€

Tabla 6: Recursos materiales del prototipo

Por último, se debe calcular el coste del personal. Para este proyecto, los distintos roles serán asumidos por el alumno, aunque para el cálculo de costes se ha estimado oportuno hacer los cálculos con un salario medio en cada uno de los roles, sin contar con la experiencia del alumno en dichos roles.

Para calcular el gasto, se obtendrá el **salario por hora** de cada trabajador, teniendo en cuenta su sueldo por año. Para hacer los cálculos estimados, se dividirá este entre los **251 días laborables** (quitando sábados, domingos y festivos) que tiene este año 2021. Después se dividirá entre las **8 horas que tiene una jornada completa** normalmente.

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net* [28], *Indeed.com* [29] y *Jobted.es* [30].

En cuanto al rol de jefe de proyecto se cobra alrededor de unos **40.000€ al año**. El sueldo por hora es de unos **19,92€**.

Un programador de videojuegos cobra **al año unos 32.100 €**. Haciendo el cálculo, el coste por hora de un programador de videojuegos con este dato es de **15,99€/h**.

Para el diseño e ilustración de imágenes, modelos y la interfaz, el salario de un diseñador gráfico se aproxima a los **19.297€ al año**. Según esto, el coste por hora será de **9,61€/h**.

Rol	Horas	Precio total
Jefe de proyecto	90	1.792,80€
Programador de videojuegos	147	2.350,53€
Diseñador gráfico	30	288,30€
Total:		4.431,63€

Tabla 7: Personal del prototipo

Por último, debemos sumar todos los gastos calculados para obtener el coste total del proyecto:

Concepto	Coste
Recursos digitales	10,38€
Costes materiales	49,55€
Personal	4.431,63€
Total:	4.491,56€

Tabla 8: Estimación del coste total del prototipo

La suma total de todos los gastos calculados es de **4.491,56€**. Para tener en cuenta posibles cambios o gastos imprevistos, se sumará un 15% del total sirviendo como colchón para estas situaciones. Tras esto, **el gasto total estimado para el proyecto será de 5.165,30€**.

4.4 Ejecución del proyecto

En este apartado se describirá de forma detallada el proceso de desarrollo de la aplicación a través de cada una de sus iteraciones.

Para el diseño de las funcionalidades incluidas en la aplicación, en cada iteración se muestran diagramas simplificados de la parte de la aplicación que

se está desarrollando. Los diagramas completos resultantes de todo el proceso se encuentran en el [ANEXO: DIAGRAMAS DE DISEÑO CREADOS](#).

4.4.1 Iteración 1

Para la primera iteración, se incluyen las tareas del **grupo de tareas 1 y 2.1**.

Esta iteración se centrará en la **creación de un prototipo** muy simple de uno de los puzles para mostrar la esencia de este videojuego. Además, también se empezará a **estudiar la herramienta de mapas** a utilizar, en este caso MapBox.

GRUPO DE TAREAS 1: GESTIONAR LA ARQUITECTURA DEL SISTEMA

En el grupo de tareas 1 se construiría el prototipo del proyecto que queremos hacer. Como nuestro proyecto es ya un prototipo, lo que se va a construir será la forma más simple del proyecto y en lo que se basa, que en nuestro caso será un prototipo de uno de los puzles.

- **DISEÑO**

Para tener un punto de partida, se ha creado un mockup de lo se observaría en la ventana de resolución del puzle. A continuación, se muestran las ventanas creadas en el mockup:

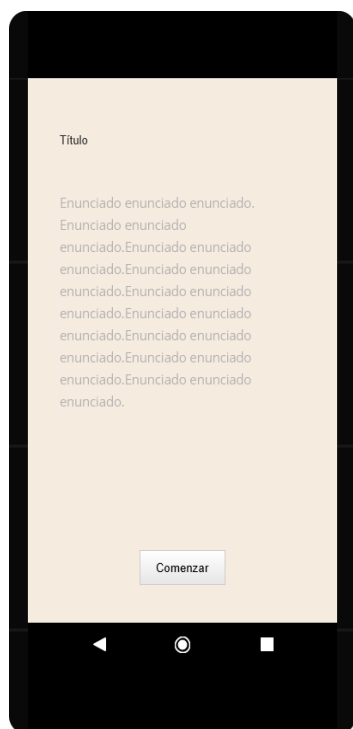


Ilustración 17: Imagen 1 de prototipo de puzle

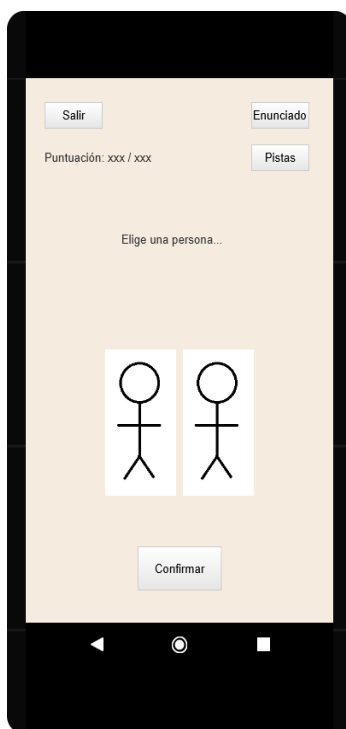


Ilustración 18: Imagen 2 de prototipo de puzle

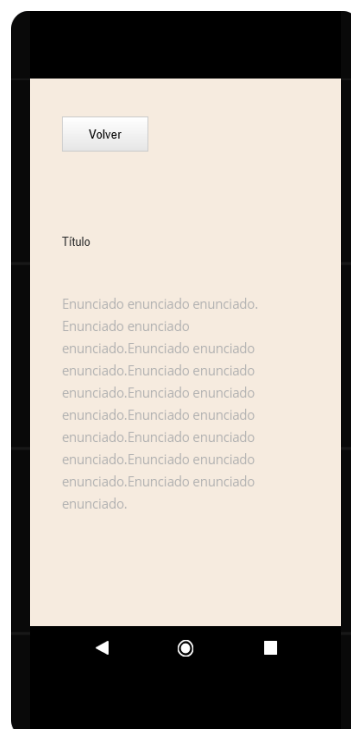


Ilustración 19: Imagen 3 de prototipo de puzle



Ilustración 20: Imagen 4 de prototipo de puzle

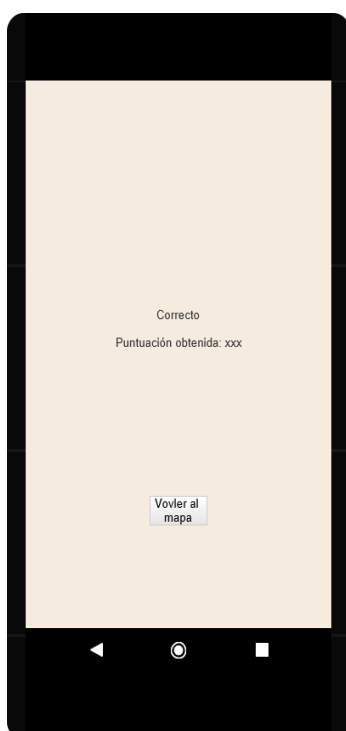


Ilustración 21: Imagen 5 de prototipo de puzle

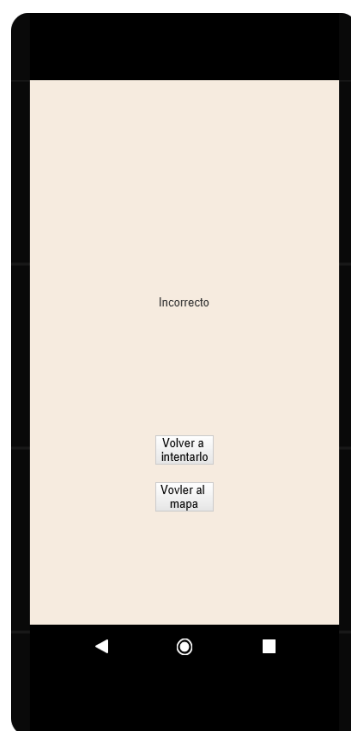


Ilustración 22: Imagen 6 de prototipo de puzle

- CONSTRUCCIÓN Y ANÁLISIS:

Inicialmente se ha creado una interfaz que será común a todos los puzles:

- ☐ Pantalla de inicio:

- ☐ Enunciado: Se muestra en texto el enunciado del puzle a resolver.
- ☐ Botón Iniciar: Se pasa a la pantalla de puzle

- ☐ Pantalla de puzle:

- ☐ Botón Salir: permite al usuario salir del puzle en cuestión para volver al mapa.
- ☐ Botón Enunciado: permite al usuario volver a visualizar el enunciado del puzle.
- ☐ Botón Pistas: permite al usuario ver las pistas ya obtenidas, así como canjear monedas para conseguir más pistas.
- ☐ Puntuación: refleja la puntuación que se va a obtener en caso de resolver el puzle y la puntuación máxima que tenía el puzle.

- ☐ Pantalla final:

- ☐ Botón Salir: Vuelve a la pantalla de mapa
- ☐ Botón Volver a intentar: Si el usuario ha fallado, puede volver a intentar el puzle.

A continuación, se han añadido los elementos concretos del puzle en cuestión:

- ☐ Imagen seleccionable 1
- ☐ Imagen seleccionable 2
- ☐ Botón Confirmar: Confirma la selección y da al usuario el resultado en la pantalla final.

Tras esto, se ha comprobado que todos los elementos tienen una correcta interacción y que estos podrían satisfacer requisitos obtenidos en el análisis como la cantidad de puntos obtenidos o tener un sistema de pistas.

- PRUEBAS INICIALES POR EL USUARIO

Para finalizar con este grupo de tareas, se ha procedido a hacer una prueba de usuario del prototipo. En general, para las pruebas de usuario se

dejará el prototipo en manos de otras personas, algún familiar o amigo, para que puedan comprobar que todo va bien y dar también algunas recomendaciones al respecto.

En este caso, el prototipo apenas tiene funcionalidad más allá de poder elegir una de las dos imágenes y comprobar si la imagen es la correcta o no.

Para las personas que lo han probado, le falta cierta información con respecto al puzle, pero la interfaz les agrada al ser simple e intuitiva, por lo que seguiremos con este estilo a lo largo del proyecto.

GRUPO DE TAREAS 2.1: ESTUDIO DE LA HERRAMIENTA DE MAPAS MAPBOX

Para concluir esta primera iteración, se ha procedido al estudio de la herramienta de mapas a utilizar MapBox.

Para empezar con Mapbox, se ha comenzado por la visualización de diversos tutoriales para el aprendizaje del uso de esta herramienta. Basándome en esto, se ha optado por usar la opción de “Videojuego basado en localización” siendo esta la base que se utilizará para la pantalla principal del videojuego.

4.4.2 Iteración 2

Para la segunda iteración, se incluyen las tareas del **grupo de tareas 2.2 y 2.3**.

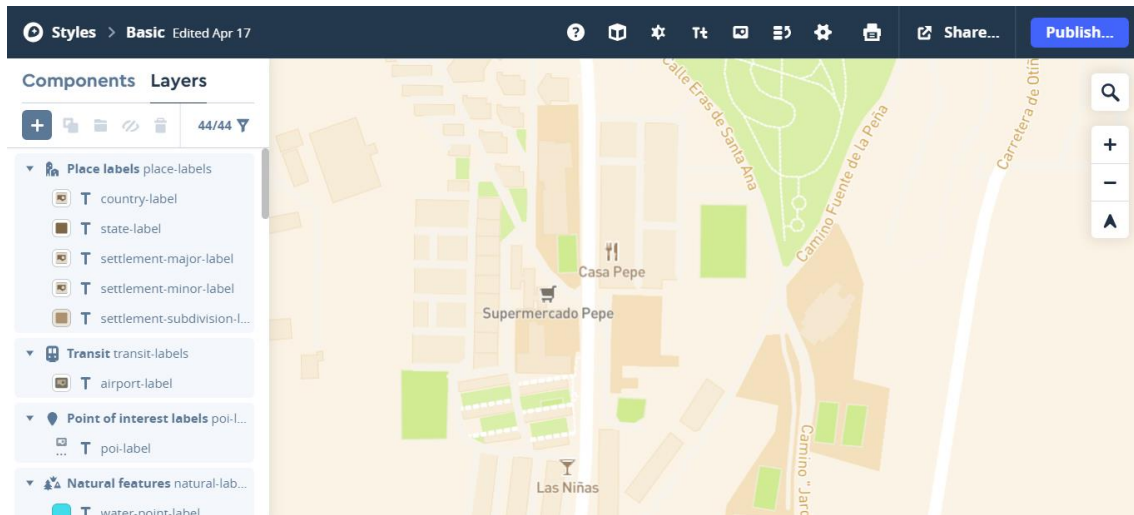
En esta iteración se trabajará en la **integración de MapBox** al proyecto y se empezará a desarrollar en Unity la parte **de localización del usuario**.

GRUPO DE TAREAS 2.2: INTEGRACIÓN DE MAPBOX

- **DISEÑO**

MapBox nos permite crear nuestro propio tipo de mapa que queremos utilizar, pudiendo añadir puntos de interés como localización de restaurantes, edificios públicos, etc.

Utilizando su herramienta de construcción de mapas, MapBox Studio [\[31\]](#), se ha construido un mapa simple, donde solamente se verá la información indispensable, el cual utilizaremos para nuestro videojuego [Ilustración 23].

Ilustración 23: MapBox Studio [\[31\]](#)

- IMPLEMENTACIÓN

Para comenzar a trabajar con MapBox se ha descargado de su página web e instalado los assets necesarios para su utilización en Unity.

Tras esto, como se ha comentado anteriormente, se ha utilizado la opción de “Videojuego basado en localización”. Esto colocará en nuestra escena un prefab que servirá de base a nuestro videojuego, incluyendo este la colocación del mapa en la escena y un indicador de dónde se encuentra el jugador.

Para una mejor visualización, se ha añadido una cámara al indicador del jugador con cierta inclinación, de tal forma que el jugador siempre se vea en el centro de la pantalla y el mapa se vea bien.

- PRUEBAS DE UNIDAD

Para las pruebas, se han introducido ubicaciones conocidas dentro de la herramienta de pruebas que tiene el propio MapBox. Se ha comprobado que, efectivamente, el indicador del jugador se posiciona en las ubicaciones dadas a la herramienta a través de un documento de texto.

GRUPO DE TAREAS 2.3: LOCALIZACIÓN GPS

- DISEÑO

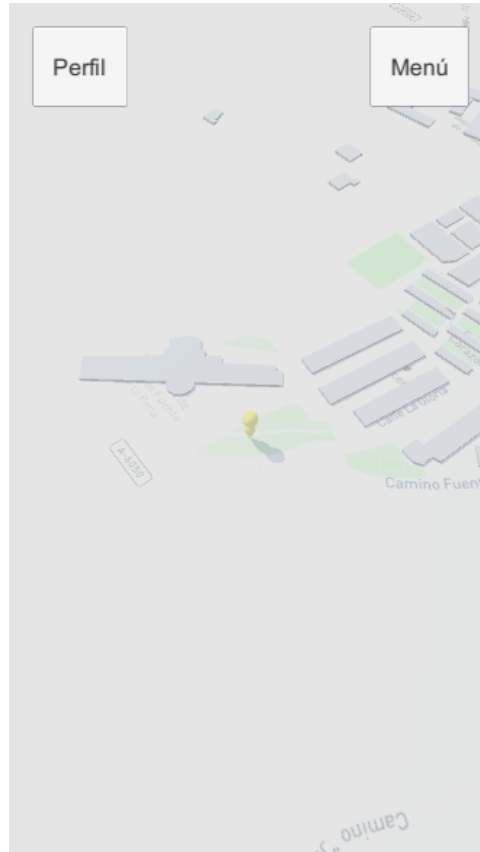


Ilustración 24: Localización GPS

MapBox nos da directamente la localización GPS y la muestra en la escena [Ilustración 24], pero para poder gestionar mejor esa localización, debemos hacer que MapBox pase la ubicación a un script que se llamará “Mapa” y que forma parte del sistema que se va a crear, separando así la herramienta del propio sistema del videojuego [Ilustración 25].

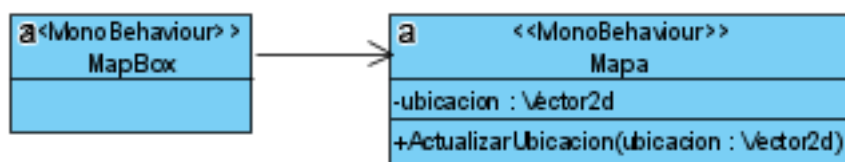


Ilustración 25: Diseño de Localización GPS

- IMPLEMENTACIÓN

Para gestionar de forma adecuada la localización GPS se ha creado el script mencionado “Mapa”, el cual será actualizado por MapBox. Para ello, se ha investigado sobre los scripts que usa MapBox y, una vez encontrado el script que detecta la ubicación, se ha modificado incluyendo una llamada al método “ActualizarUbicacion” del script Mapa de forma que actualice la información del script cada vez que MapBox detecte un cambio en la ubicación del jugador.

- PRUEBAS DE UNIDAD

Para probar esto, simplemente se ha utilizado el log de manera que muestra la actualización de la ubicación en el script creado “Mapa”. Una vez que el script muestra buenos resultados con la herramienta de pruebas, se procede a las pruebas de integridad.

- PRUEBAS DE INTEGRIDAD

Se ha creado el ejecutable para móvil para comprobar que todo funciona correctamente.

La ubicación se muestra de forma adecuada y el mapa se visualiza en pantalla de forma correcta. La ubicación se actualiza sin problema mostrando el movimiento en pantalla sin problema.

- PRUEBAS POR PARTE DEL USUARIO

El usuario tras haberlo usado unos minutos comprueba que la ubicación es correcta, pero la rotación que detecta la aplicación parece no ir bien.

El problema es resuelto tras cambiar una opción de la herramienta MapBox que permite establecer el giro usando el ángulo negativo, dando como resultado la rotación esperada.

4.4.3 Iteración 3

Para la tercera iteración, se incluyen las tareas del **grupo de tareas 3.1**. Esta iteración se centrará en la creación del pilar central de nuestro videojuego, los puzles.

GRUPO DE TAREAS 3.1: CREACIÓN DE LOS PUZZLES

- DISEÑO

Para el diseño de los puzzles, en cuanto a su interfaz, se ha seguido con el diseño que tenía el primer prototipo de puzzle, siguiendo la estética simple que el usuario externo probó en su momento. Para ello, se ha creado un prefab en Unity que se usará para instanciar cada uno de los puzzles, agregando en su caso los elementos particulares del puzzle en cuestión (dentro del gameobject resaltado en azul) [Ilustración 26][Ilustración 27].

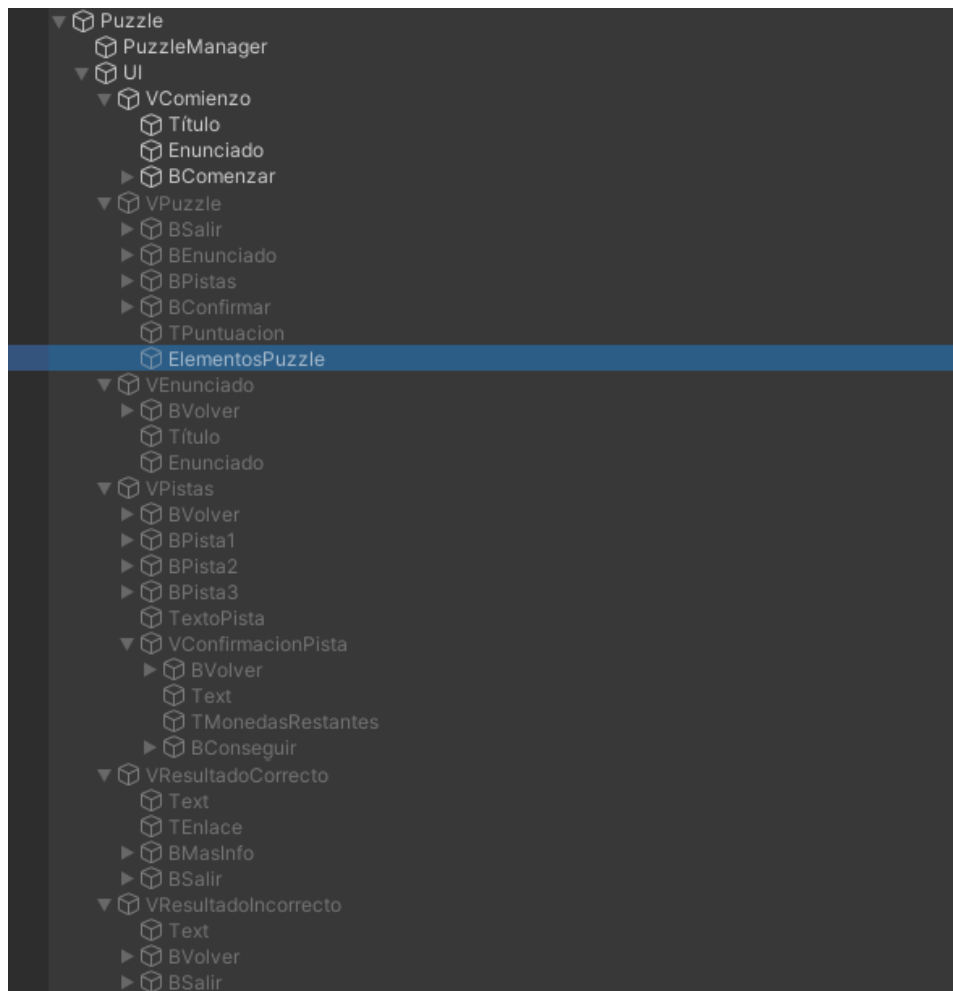


Ilustración 26: Jerarquía de objetos en Unity

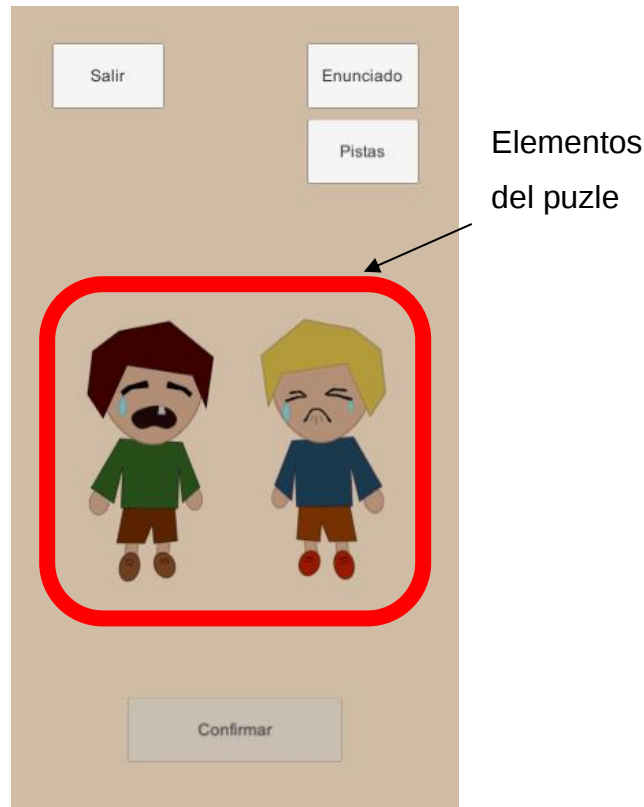


Ilustración 27: Elementos del puzle en la escena

Para un diseño correcto de los puzles del videojuego a nivel de clases, se debe tener en cuenta que el número de puzles que se pueden llegar a crear puede ser muy elevado, así como el número de tipos de puzle. Por esto, se ha dedicado una gran parte del tiempo a diseñar de forma correcta una generalización de la clase Puzzle y ControladorPuzzle [Ilustración 28], para que la inclusión de nuevos puzles sea lo más sencilla posible.

En este prototipo, se mostrarán dos tipos de puzle:

- Puzle de selección de opción
- Puzle de inserción de texto/código

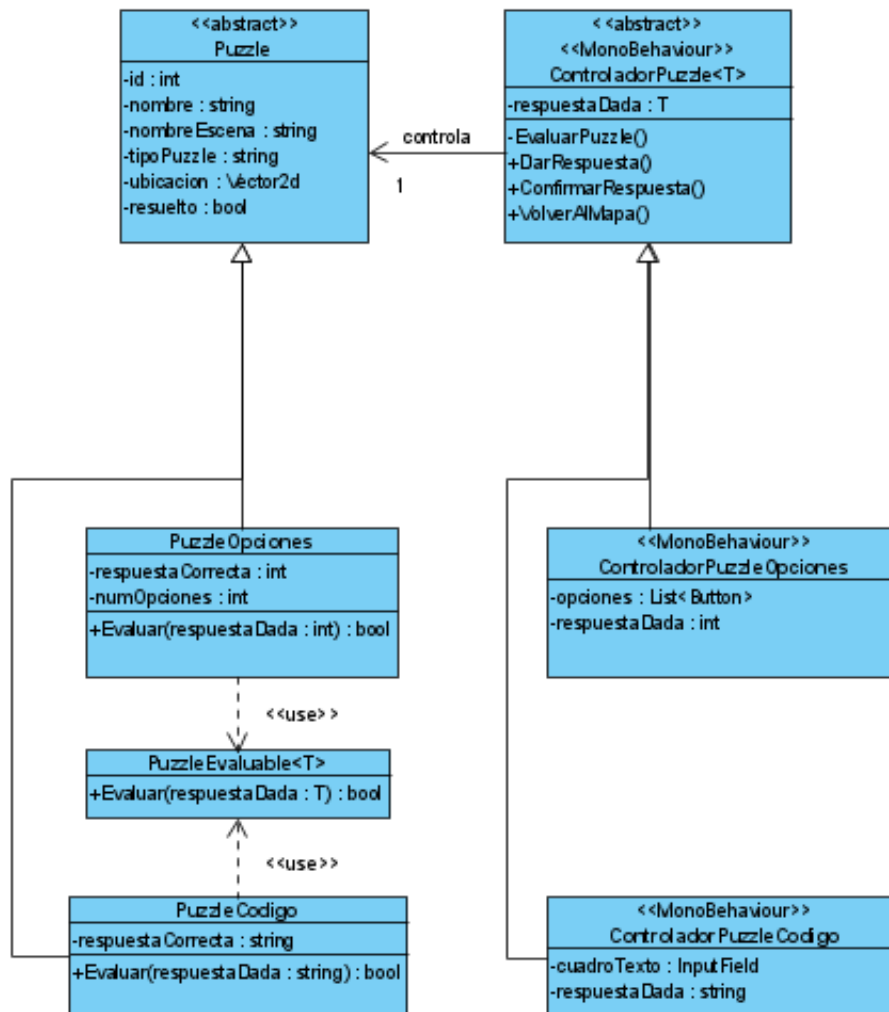


Ilustración 28: Diseño de los puzles y su controlador

Para el diseño de los puzles a nivel de contenido y temática, debemos tener en cuenta que este proyecto se basa en dar a conocer lugares de interés del territorio, teniendo que basar todos los puzles en resolver problemas que tengan relación con dichas ubicaciones. Por esto, se ha decidido centrarnos en una temática concreta dentro del gran abanico de lugares turísticos de la localidad de Jaén, en concreto, las leyendas urbanas.

En este proyecto se mostrarán los siguientes puzles:

- **Puzle “El niño y el espectro”:** Puzle de tipo selección de opción basado en la leyenda de El espectro de la Fuente de la Peña en el cual el jugador deberá guiarse por la información dada en la leyenda local para distinguir de forma visual cuál de los dos niños mostrados realmente es un niño y no la criatura transformada en infante.

- **Puzle “Una declaración de amor escalofriante”:** Puzle de tipo inserción de texto/código basado en la leyenda del Palacio de los Vélez en el que el jugador deberá analizar la nota anónima mostrada y averiguar el nombre de la persona “anónima”.
- **Puzle “Un lagarto explosivo”:** Puzle de tipo selección de opción basado en la leyenda de El Lagarto de la Magdalena en el cual el jugador deberá de escoger el detonador correcto para hacer explotar de nuevo a este fiero reptil.

La visualización y resolución de estos puzles se pueden consultar en el [ANEXO 2: RESOLUCIÓN DE LOS PUZZLES](#).

Dado que las imágenes mostradas deben contener elementos muy específicos para la resolución del puzle, se debería de contar con un artista gráfico que las dibuje. Al no ser así, se ha procedido a escanear dichas imágenes hechas a mano, pudiendo modificarse estas al final del desarrollo de este proyecto si el tiempo lo permite.

- IMPLEMENTACIÓN

Para la implementación de los puzles, se ha creado una clase abstracta llamada “Puzzle<T>” que tendrá todos los elementos comunes necesarios y un “ControladorPuzzle<T>”, también clase abstracta, que gestione los datos de dicho puzle según las acciones del jugador y el tipo de dato que maneja el puzle. Dado que se debe instanciar un tipo de puzle concreto, se ha creado la especificación para dichos tipos, “PuzzleOpciones” y “PuzzleCodigo”, los cuales deberán de implementar la interfaz “Evaluable<T>” para poder ser correctamente evaluados según el tipo de dato que reciban.

Estos puzles deben de ser gestionados por un controlador que maneje un tipo de dato concreto según el puzle, ya que la solución a este puede ser desde una opción o un texto, como el caso de este prototipo, hasta condiciones de victoria mucho más complejas, como una posición concreta de un modelo 3D dentro de la escena (posible trabajo futuro). Por esto, se ha creado también la especificación de cada tipo de controlador de puzle (“ControladorPuzzleOpciones” y “ControladorPuzzleCodigo”).

Para este prototipo, la carga de cada uno de los puzles se hará en escenas diferentes, de forma que el puzle ya está construido de antemano. El diseño creado permitiría, como trabajo futuro, la creación de la escena de un puzle concreto en tiempo real, sabiendo el tipo de puzle que se cargará y con la ayuda de un fichero de configuración para ajustar, por ejemplo, qué imágenes aparecen y la posición de las mismas.

- PRUEBAS DE UNIDAD

Para las pruebas de unidad, se ha procedido a comprobar si las respuestas dadas se evaluaban de forma correcta, mostrando la ventana “Incorrecto” en caso de fallar y mostrando la ventana de “Correcto” en caso de acertar con la respuesta. Tras comprobar que todo va como se esperaba, se procede a dar paso a la nueva iteración.

4.4.4 Iteración 4

Para la tercera iteración, se incluyen las tareas del **grupo de tareas 3.2.1, 3.2.2, 3.3, 3.4, 4.1 y 4.2.**

Para comenzar esta iteración, se va a incluir un **sistema de pistas** dentro de los puzles que ayudará al usuario a resolver el puzle, así como una nueva funcionalidad que permita la **elección de diferentes rutas.**

GRUPO DE TAREAS 3.2.1: OBTENCIÓN DE MONEDAS

- DISEÑO

Para la creación del sistema de pistas, antes será necesario obtener el canjeable que gastará el jugador a cambio de una pista dentro de los puzles. Estos canjeables serán **monedas**. El jugador podrá conseguir estas monedas andando un cierto número de kilómetros. Para ello se necesitará una clase Jugador que almacene dichas monedas [Ilustración 29].

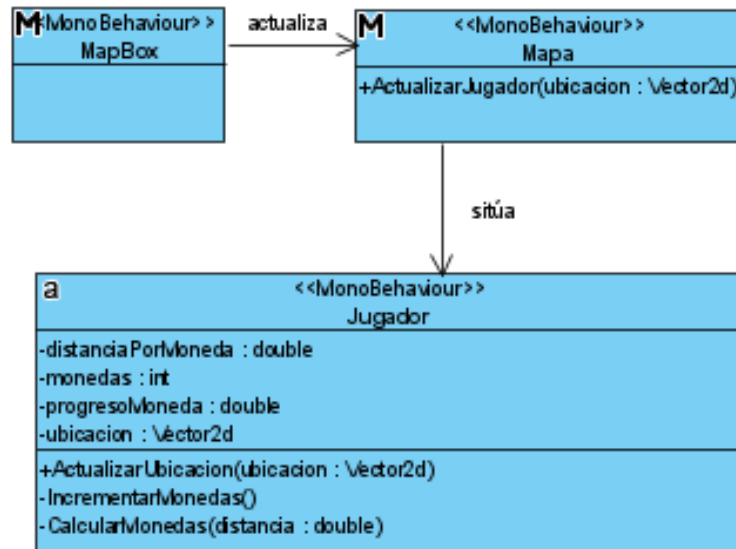


Ilustración 29: Diseño de obtención de monedas

● IMPLEMENTACIÓN

Para obtener dichas monedas, lo primero que tenemos que conseguir es calcular la distancia entre dos puntos.

Para calcular la distancia entre dos puntos dada su latitud y longitud, se ha optado por usar la fórmula de Haversine:

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

La fórmula Haversine nos da como resultado la distancia del círculo máximo entre dos puntos en una esfera dada la longitud y latitud de los dos puntos.

Esta función estará disponible en una clase “Auxiliar”.

Para implementar la obtención de monedas, ahora el mapa actualizará la información del script llamado “Jugador” llamando a su método “ActualizarUbicación” al que le pasará la nueva ubicación. Este método a su vez llamará a otro que sumará la distancia recorrida hasta la nueva ubicación y comprobará con un contador si se ha llegado a la meta para conseguir una moneda, sumando una moneda al jugador en tal caso.

- PRUEBAS DE UNIDAD

Para las pruebas de unidad, simplemente se han introducido varias ubicaciones en el fichero de prueba de Mapbox y se ha mostrado en el log el número de monedas en caso de que se haya llegado a tal meta. Tras ver que todo funciona correctamente, se procede a seguir con el sistema de pistas.

GRUPO DE TAREAS 3.2.2: CONSEGUIR PISTA

- DISEÑO

Una vez conseguida una moneda, el usuario tendrá acceso a una pista en la ventana de pistas [Ilustración 30].

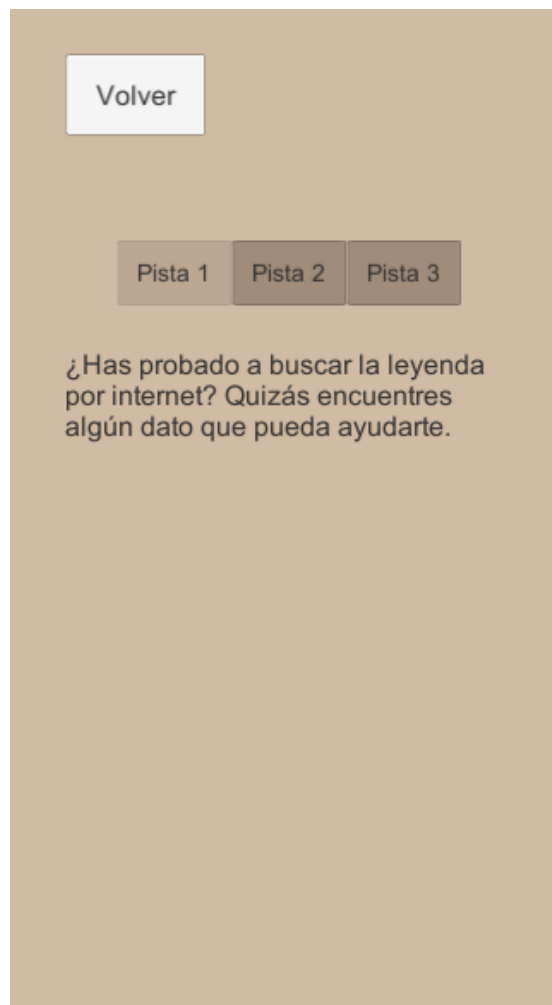


Ilustración 30: Ventana de pistas

Para ello se incluirá en el sistema una nueva clase “Pista” y un script “SistemaPistas” [Ilustración 31] que controlará las pistas del puzle presentado en pantalla.

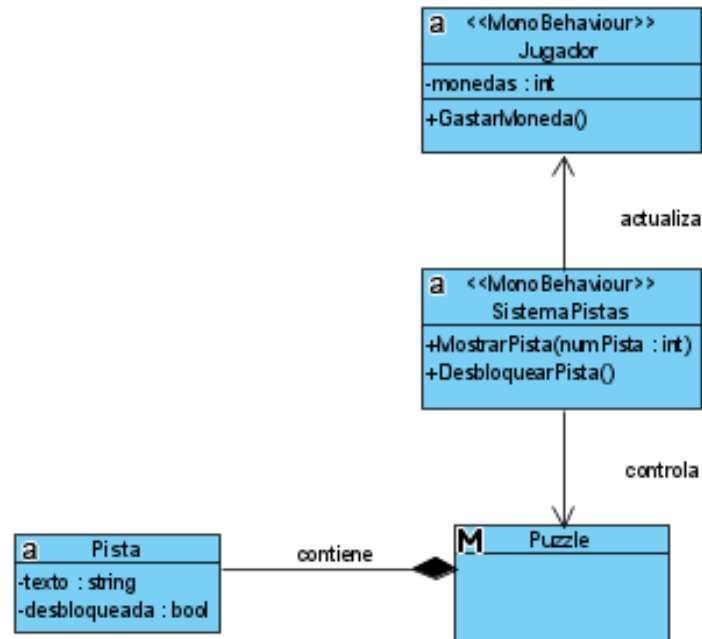


Ilustración 31: Diseño del sistema de pistas

Se ha decidido que el número de pistas por puzle siempre va a ser el mismo para que el jugador pueda reflexionar sobre si es conveniente gastar una moneda ahora, o más adelante. De esta forma se controlará mejor la economía de este canjeable. En este prototipo el número por defecto de pistas de cada puzle será de 3 pistas.

Antes de gastar una moneda, aparecerá una ventana de confirmación:

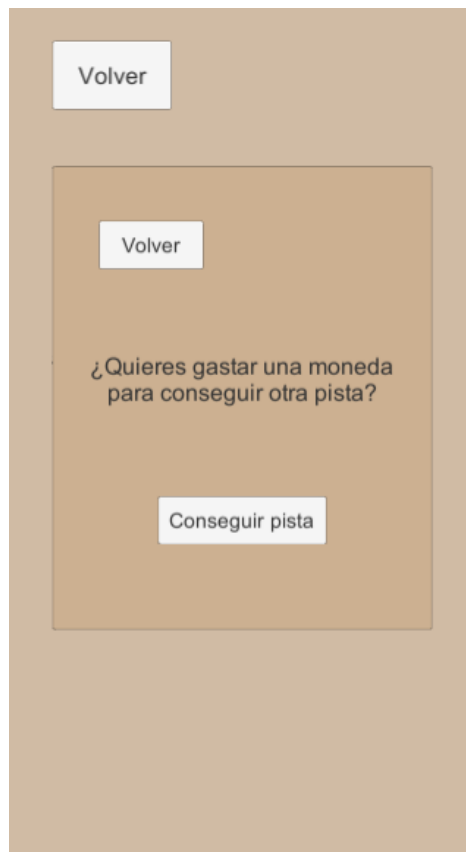


Ilustración 32: Ventana de confirmación "Conseguir pista"

- IMPLEMENTACIÓN

Para la implementación de este sistema de pistas, a la clase puzzle se le ha añadido un array con 3 pistas por defecto. Estas pistas por defecto estarán bloqueadas y el usuario podrá desbloquearlas canjeando una moneda. Para esto, el sistema de pistas habilitará el botón de "Conseguir pista" si el jugador tiene al menos una moneda. Al pulsarlo se decrementará una moneda al jugador llamando al método "GastarMoneda" de la clase "Jugador" y se mostrará la siguiente pista no desbloqueada hasta el momento.

- PRUEBAS DE UNIDAD

Se comprueba en las pruebas de unidad con ayuda del log que al pulsar el botón de confirmación la cantidad de monedas del jugador se decrementa y que la pista que se va a mostrar aparece como desbloqueada.

No se permite desbloquear pistas si no se tiene monedas.

- PRUEBAS DE INTEGRIDAD

Para las pruebas de integridad, se ha intentado conseguir monedas andando con el dispositivo. A la hora de resolver un puzle, el sistema de pistas muestra que hemos conseguido más monedas andando.

Se ha añadido al puzle “El niño y el espectro” la primera pista. Se ha probado a intentar conseguir una pista sin tener monedas, pero el botón no se puede pulsar al no tener monedas. Al conseguir una moneda se ha comprobado que el botón sí está activo y se muestra la primera pista.

También se ha intentado desbloquear una pista posterior sin tener la anterior desbloqueada. El resultado es positivo puesto que, aunque intentamos desbloquear en este caso la segunda pista sin tener la primera, el resultado del desbloqueo de pista nos da la primera de ellas, que es lo que se pretendía.

- PRUEBA POR PARTE DEL USUARIO

El usuario probará en esta ocasión todos los puzles, así como la obtención de monedas andando con el dispositivo y el sistema de pistas.

Según el usuario, el puzle “El niño y el espectro” le ha resultado especialmente difícil puesto que no conocía la leyenda. En este caso, el sistema de pistas que se construirá posteriormente dará al jugador la información necesaria para que el puzle sea más fácil. Una vez que se le ha dado la que sería la primera pista, el jugador ha podido resolverlo.

Para el puzle “Un lagarto explosivo”, el jugador ha podido resolverlo sin ningún problema.

Para el puzle “Una nota de amor escalofriante” el jugador ha detectado que el teclado no se activaba en el dispositivo, por lo que ha sido incapaz de resolverlo por problemas técnicos. Dicho error se ha subsanado llamando directamente a la apertura de teclado en el script “ControladorPuzzleCodigo”.

A la hora de probar el sistema de pistas, el usuario comprueba que puede gastar sus monedas sin ningún problema hasta acabarse. Aun así, dice sentirse incómodo al no mostrar en la escena del puzle el número de monedas que le quedan para gastar en pistas.

Para solucionar lo anterior, en la ventana de confirmación para desbloquear una pista, se muestra al usuario la cantidad de monedas actual que le quedan antes de gastar una.

GRUPO DE TAREAS 4.1: UBICACIÓN DE PUZZLES EN LA RUTA

- DISEÑO

Para continuar, los puzzles ahora deben tomar una ubicación concreta dentro del mapa. Al conjunto de esos puzzles ya ubicados lo llamaremos “Ruta”.

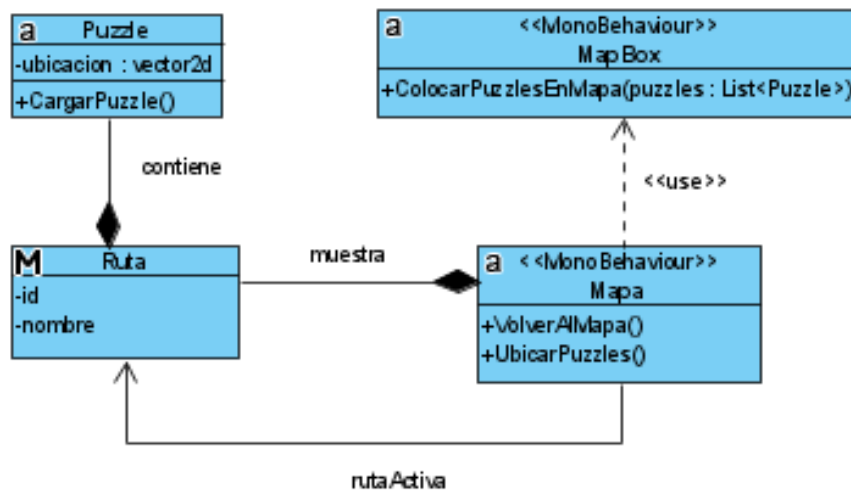


Ilustración 33: Diseño de ubicación de puzzles

El mapa contendrá todas estas rutas [Ilustración 33] , y deberá mostrar los puzzles de la misma en las ubicaciones correctas dentro del mapa. Para ello, debe enviar a MapBox la ubicación de todos los puzzles para que se muestre en pantalla un modelo simple creado en Blender que servirá de señalización para los puzzles [Ilustración 34], cargando el puzzle si se presiona el modelo.



Ilustración 34: Marca de puzle en el mapa

- IMPLEMENTACIÓN

Se ha creado la clase “Ruta” que contendrá todos los puzles que la componen. Además, para poder ubicarlos, se ha añadido a la clase “Puzzle” un atributo “ubicación” que tendrá las coordenadas de este.

Para ubicar los puzles de una sola ruta, ahora el script Mapa tiene una “rutaActiva” que indicará cuál de todas las rutas se está mostrando en el mapa.

Para ubicar los puzles, MapBox tiene una herramienta de pruebas que, dadas unas ubicaciones, muestra un determinado prefab en dichos lugares, sin embargo, esto no sirve a nivel de implementación.

Se ha investigado sobre las funciones internas de la herramienta, encontrando un método que crea una nueva capa de elementos. Se llama entonces desde el script Mapa a la herramienta MapBox con el método “UbicarPuzzles” para que cree una nueva capa que contendrá puntos de interés, en nuestro caso, los puzles.

Tras esto, se ha creado un script “poiPuzzle” en el que se incluye el método “OnClick” el cual hará que, una vez que el usuario pulse dicho prefab, se llame al método “CargarPuzzle” del puzle correspondiente.

Para terminar, debemos asegurarnos que tras resolver dicho puzle, el jugador puede volver al mapa. Para ello, el script “ControladorPuzzle” llamará al método de “Mapa” llamado “VolverAlMapa”.

- PRUEBAS DE UNIDAD

Se ha comprobado que, al inicio de la ejecución, el mapa muestra la ruta de prueba creada, colocando correctamente todos los puzles en sus respectivas ubicaciones.

También se ha comprobado que al pulsar el prefab del puzle se carga de forma correcta la escena del puzle correspondiente.

Se ha detectado un error crítico a la hora de volver a la escena del mapa al intentar iniciar de nuevo la herramienta MapBox. Para solucionarlo, se ha creado un script llamado “MapboxMap” en el que se ha añadido la función DontDestroyOnLoad. Esto hará que el servicio de localización no se pare en ningún momento, incluso cuando se cambie de escena.

Tras esto se ha comprobado que todo funciona correctamente.

GRUPO DE TAREAS 4.2: SELECCIÓN DE REGIÓN Y RUTA

- DISEÑO

Anteriormente se ha comprobado que los puzles se ubican según la ruta de prueba creada, pero en la aplicación que se desea crear, el número de rutas será mayor. Por esto, el usuario deberá poder elegir entre varias rutas dentro de diversos territorios.

Para que el jugador pueda elegir la ruta que desea hacer, se creará un menú en la pantalla principal [Ilustración 35], el cual tendrá una opción para elegir la ruta activa. Una vez que ha seleccionado dicha opción, deberá seleccionar el territorio concreto donde quiere hacer la ruta [Ilustración 36][Ilustración 37] y una vez seleccionado, podrá elegir la ruta que se desea [Ilustración 38].



Ilustración 35: Ventana 1 de selección de ruta



Ilustración 36: Ventana 2 de selección de ruta

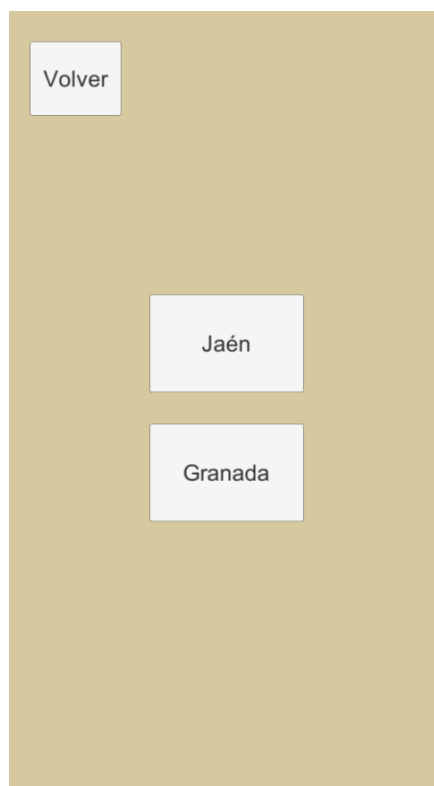


Ilustración 37 : Ventana 3 de selección de ruta

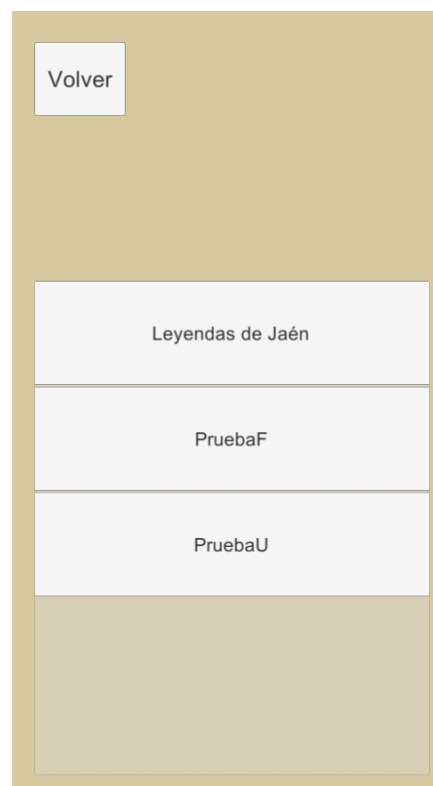


Ilustración 38: Ventana 4 de selección de ruta

En este caso, dado que es un prototipo a nivel local, no se ha incluido el listado entero de ciudades y regiones. Para simularlo, se han incluido botones de otros territorios sin ninguna funcionalidad, solo estando disponible la localidad de Jaén.

El diseño de esta parte se ha cambiado de manera que sea un DAO el que contenga las rutas y un controlador para la gestión de estas [Ilustración 39].

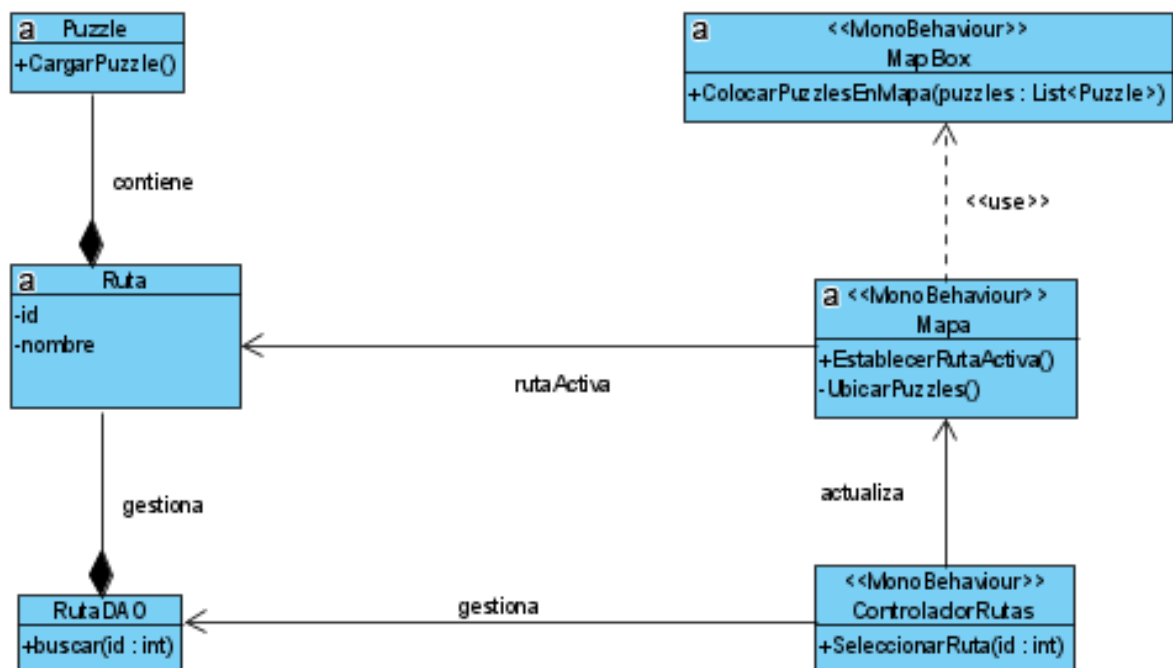


Ilustración 39: Diseño de selección de ruta

De esta forma se separarán los scripts del videojuego de la gestión de entidades que se debería hacer con una futura base de datos.

● IMPLEMENTACIÓN

Se han creado en la escena las ventanas correspondientes para la interfaz necesaria de selección de la ruta.

Para manejar de forma correcta la interacción con el usuario, se ha creado un script llamado "ControladorRutas" que mostrará el menú de selección de ruta. Una vez seleccionada la región, el controlador se encargará de mostrar en pantalla con un scroll las diferentes rutas disponibles.

Al seleccionar una ruta, el controlador llamará al método “EstablecerRutaActiva” del script Mapa pasándole como parámetro la ruta elegida por el jugador y mostrándose en el mapa la ruta seleccionada.

- PRUEBAS DE UNIDAD

Para las pruebas, se han creado unas rutas de prueba cerca de la ubicación del desarrollador y se ha comprobado si la elección de la ruta hace que cambie la ruta activa del mapa cambie a la ruta elegida. Tras comprobar que la elección de la ruta funciona correctamente se pasa a las pruebas de integridad.

- PRUEBAS DE INTEGRIDAD

En las pruebas de integridad se ha comprobado que el cambio de ruta se hace de forma correcta, pero se han detectado errores a la hora de la resolución de los puzles.

Al resolverse los puzles, estos no se ocultan o eliminan, por lo que el usuario podría resolverlos una y otra vez. Para resolverlo, se ha decidido cambiar la forma en que se muestran los puzles en las rutas.

Ahora las rutas tendrán un “puzzleActivo” que será el índice del siguiente puzle a resolver. Como consecuencia de este cambio, en el mapa solo se mostrará un puzle cada vez y, una vez que se resuelva dicho puzle, se llamará al nuevo método “SiguientePuzzle” que incrementará el índice de Activo y se eliminará el anterior de la capa por MapBox mostrando el siguiente a resolver.

En caso de haber acabado la ruta, el atributo puzzleActivo será mayor al número de puzles que componen la ruta. De ser así, el método “SiguientePuzzle” devolverá un null indicando que la ruta activa ya se ha terminado.

- PRUEBA POR PARTE DEL USUARIO

Se ha creado el ejecutable correspondiente y se ha dejado al usuario el dispositivo para probar la nueva selección de ruta.

El usuario ha podido resolver los puzles sin problema hasta la terminación de la ruta elegida sin detectarse fallos.

4.4.5 Iteración 5

Para la quinta iteración, se incluyen las tareas del **grupo de tareas 5, 6.1 y 6.2**.

En esta iteración se trabajará en una **guía que indique al usuario la situación de los puzles**, así como en un **sistema de puntuaciones** y un **ranking** que darán a este videojuego cierta competitividad incentivando al usuario a realizar más rutas.

GRUPO DE TAREAS 5: GUÍA POR LA RUTA

- DISEÑO

En pantalla se mostrará una guía que indicará al usuario la dirección del puzle que se encuentra actualmente activo. Se ha creado con este propósito un modelo en Blender de una flecha que hará de guía para el jugador [Ilustración 40].



Ilustración 40: Guía en el mapa

Para ofrecer un mayor reto al jugador, se ha decidido ocultar el en cuestión hasta que el jugador esté a menos de una distancia dada [Ilustración 41].

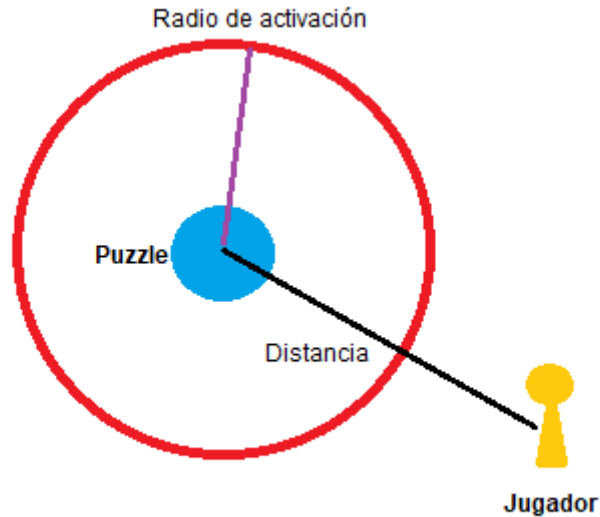


Ilustración 41: Radio de activación del puzle

Con esto, el jugador deberá guiarse exclusivamente por la indicación mostrada en el mapa hasta llegar a una distancia que permita al jugador activar el puzle. Esto fuerza al jugador a acercarse al lugar de interés turístico que se quiere destacar con el puzle en cuestión.

El diseño de clases se ha modificado para esta funcionalidad (se incluyen los cambios realizados para corregir los errores de la guía en las pruebas de usuario) [Ilustración 42].

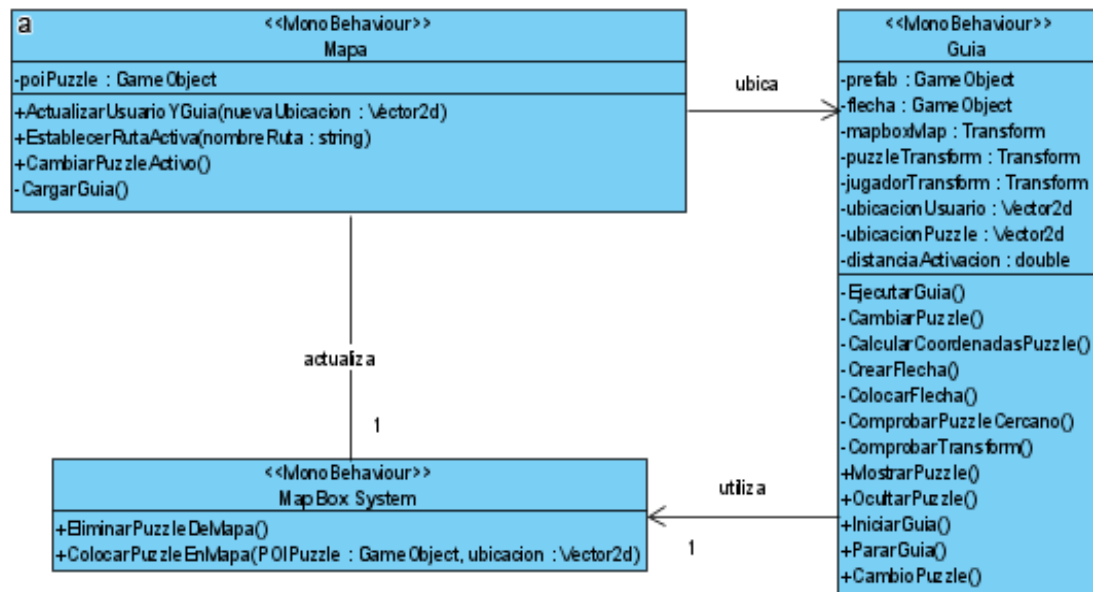


Ilustración 42: Diseño de la guía

● IMPLEMENTACIÓN

Se ha creado un nuevo script llamado “Guía” que controlará la flecha que indica la dirección al y la visibilidad del puzzle.

El script Mapa iniciará la guía una vez se haya elegido la ruta con el método “CargarGuia”. Este método hará una búsqueda dentro de la escena para encontrar el prefab del modelo “poiPuzzle” y el prefab del jugador.

Tras encontrar los prefabs, se iniciará una subrutina dentro del script Guía que colocará la flecha apuntando a la ubicación del puzzle activo. Se hará después un cálculo de la distancia y, si la distancia es menor o igual al radio de activación, la flecha se ocultará y el puzzle se hará visible permitiendo que el jugador pueda seleccionarlo para resolverlo.

Una vez que el puzzle activo es resuelto, el script Mapa tiene que llamar al método “CambioPuzzle” del script Guía para hacer una nueva búsqueda y encontrar el nuevo puzzle activo.

● PRUEBAS DE UNIDAD

Para las pruebas de unidad, se ha comprobado que la guía apunta en todo momento al puzzle activo. También se ha comprobado que encuentra el

puzle activo en la escena mediante el log y, una vez encontrado, se puede observar que el puzle activo aparece y desaparece según la distancia al usuario.

- PRUEBAS DE INTEGRIDAD

En las pruebas de integridad se observan fallos a la hora de volver a la escena del mapa tras completar una ruta, ya que en ese caso no se llama a “CambiarDePuzzle”. Para corregir esto, se ha creado un nuevo método en el script Guia llamado “PararGuia” que parará la subrutina. Ocultará también el último puzle para que no pueda resolverse de nuevo y destruirá la flecha.

Se volverá a iniciar la guía en caso de que se escoja otra ruta.

- PRUEBA POR PARTE DEL USUARIO

Para las pruebas de usuario se han cambiado las ubicaciones de una de las rutas para que ahora tenga que andar un poco más y así probar las indicaciones de la guía.

Se ha detectado un problema a la hora de encontrar el puzle cuando la ubicación del mismo se aleja mucho de la ubicación del usuario. Esto ocurre porque la herramienta MapBox no crea el mapa completo, sino que crea un mapa con un rango limitado que se va actualizando según se mueve el jugador. Al ser así, cuando se crea una nueva capa, Mapbox coloca el prefab que representa al puzle en el mapa solo si la ubicación del mismo se encuentra dentro de esos límites. De no ser así, espera a que dicha ubicación entre en los límites del mapa para crearlo en la escena.

Para corregir esto, la guía debe actualizarse con la ubicación del usuario y el puzle activo dadas las longitudes y latitudes de los mismos.

Ahora el script Mapa también deberá actualizar la guía cada vez que el jugador cambie de posición ya que la guía necesita saber en todo momento dónde se encuentra el jugador para ocultar o no el puzle activo. Para ello, el mapa llamará en su método modificado “ActualizarUsuarioYGuia” al método “ActualizarGuia” del script Guía pasando la nueva ubicación del jugador como parámetro.

Utilizando de nuevo la función auxiliar “CalcularDistanciaKM” obtenemos la distancia que hay entre el puzle y el jugador y, si esta es menor o igual que el radio de activación dado, la flecha se ocultará y el puzle se hará visible permitiendo que el jugador pueda seleccionarlo para resolverlo. Si el jugador vuelve a alejarse, el puzle volverá a no ser visible y la flecha aparecerá de nuevo.

Una vez que el puzle activo es resuelto, el script Mapa tiene que actualizar la guía con la ubicación del nuevo puzle activo. Para ello, el script Mapa llama al método “CambioPuzzle” del script Guía pasando la nueva ubicación del puzle como parámetro. De esta forma, la flecha cambiará de dirección apuntando al nuevo puzle activo.

GRUPO DE TAREAS 6.1: PUNTUACIÓN DEL USUARIO

- DISEÑO

Para hacer el sistema de puntuaciones, cada puzle tendrá una puntuación máxima asociada.

Esta puntuación del puzle se irá decrementando si el jugador falla al dar sus respuestas. Para calcular la puntuación cuando falla, se ha seguido la siguiente fórmula:

$$Puntuacion = maxPuntuacion / (numIntentos + 1)$$

Para ello, se ha incluido una nueva relación entre el controlador del puzle y el jugador [Ilustración 43].

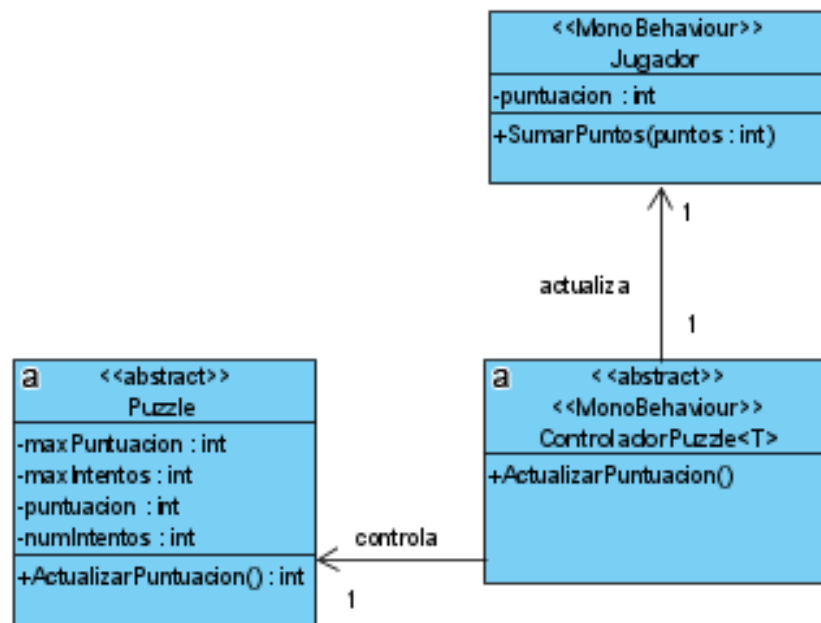


Ilustración 43: Diseño del control de puntuación del jugador

- IMPLEMENTACIÓN

Para implementar esto, el puzle ahora tendrá dos atributos nuevos: “puntuación” y “maxPuntuacion”. El atributo “puntuación” se sumará al jugador una vez que se resuelva con el método “SumarPuntos” del script Jugador y tendrá como valor inicial el de “maxPuntuacion”. Esta puntuación se irá decrementando según la fórmula anterior, pero hasta un mínimo de veces.

Para que no se reduzca demasiado la puntuación, siguiendo la misma lógica, el puzle tendrá también otros dos atributos: “intentos” y “maxIntentos”. Los intentos se irán incrementando según el usuario falle, decrementando la puntuación sólo si el número de intentos no ha llegado al máximo de intentos [Ilustración 44].

```

public void ActualizarPuntuacion()
{
    if (numIntentos <= maxIntentos)
    {
        puntuacion = maxPuntuacion / (numIntentos + 1);
        numIntentos++;
    }
}

```

Ilustración 44: Cálculo del decremento de puntuación

- PRUEBAS DE UNIDAD

Para las pruebas, se ha dado a los puzzles una puntuación inicial y un máximo de intentos. Tras resolver los puzzles, se puede observar que la puntuación varía si se ha fallado. Se comprueba que la puntuación se suma correctamente al jugador con ayuda del log.

GRUPO DE TAREAS 6.2: RANKING

- DISEÑO

Para el diseño del ranking, se ha incluido una nueva opción en la ventana del menú y se ha creado una ventana para el ranking.

Esta ventana [Ilustración 45] mostrará los diez primeros puestos del tipo de ranking elegido (local, regional, nacional o mundial) junto con la posición del jugador dentro de este.

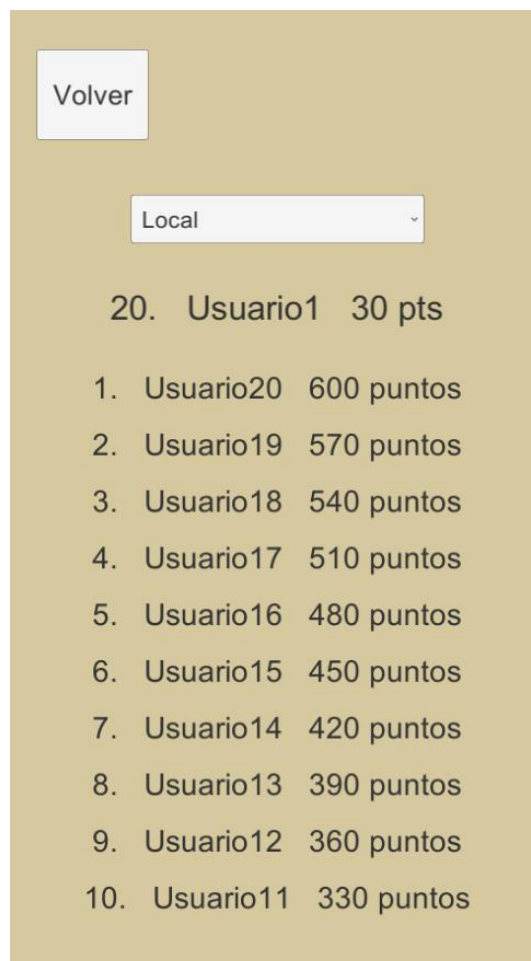


Ilustración 45: Ventana de ranking

A nivel de clases, se ha creado una nueva clase llamada “Usuario” [Ilustración 46]. Esta clase pasará a tener ciertos atributos que tenía antes el jugador, como los datos personales y la puntuación. También se ha creado un “ControladorUsuarios” y un “DAOUsuario” que servirán como base para un trabajo futuro de gestión de usuarios de la aplicación.

Ahora el script Jugador tendrá la información de ubicación del jugador y tendrá función de controlador que gestionará a un “usuarioConectado”.

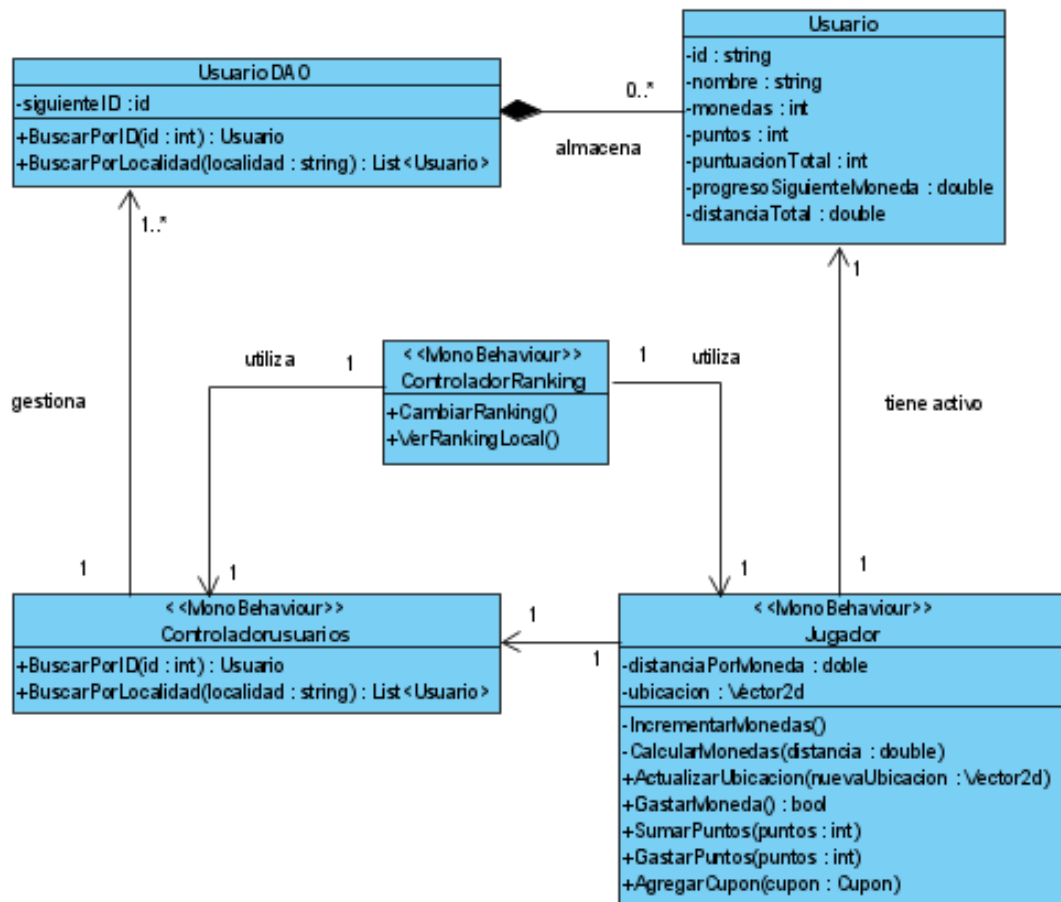


Ilustración 46: Diseño del ranking

El nuevo “ControladorRanking” será un script que maneje la información del usuario conectado y del resto de usuarios para ordenarlos por puntuación y mostrarlos por pantalla.

● IMPLEMENTACIÓN

Para la implementación de esta funcionalidad solo se ha tenido en cuenta el ranking a nivel local, creando el código suficiente para extender esto a

territorios de mayor área cuando se integre en la aplicación una base de datos con tablas de países, regiones y localidades.

Al mostrarse el ranking, se llamará al método “CambiarRanking” [Ilustración 47] que tomará el valor del scroll para mostrar un tipo u otro de ranking.

```
public void CambiarRanking()
{
    // Comprobamos la selección y mostramos el ranking según el
    territorio
    // Actualmente solo se ve el ranking local. Queda listo para
    ampliación.

    string seleccion = opciones.options[opciones.value].text;

    switch (seleccion)
    {
        case "Local":
            VerRankingLocal();
            break;

        case "Regional":
            break;

        case "Nacional":
            break;

        case "Mundial":
            break;
    }
}
```

Ilustración 47: Implementación del cambio de territorio en el ranking

A la hora de ver el ranking local, el método “VerRankingLocal” tomará la localidad del jugador y buscará a todos los usuarios con esa misma localidad. Tras esto, los ordenará según la puntuación obtenida de forma descendente, mostrando los 10 primeros y al propio jugador.

- PRUEBAS DE UNIDAD

Para las pruebas de unidad, se han creado usuarios adicionales con puntuaciones diversas y se ha mostrado el ranking local. El usuario conectado se muestra en el último puesto al no tener puntuación, por lo que no está entre los diez primeros.

Después de cambiarle manualmente la puntuación al usuario, se puede ver que su puesto es diferente y que se muestra entre los diez primeros.

- PRUEBAS DE INTEGRIDAD

Para estas pruebas, se ha dejado al jugador con una puntuación de 0, solo pudiendo aumentar si resuelve los puzles. Tras resolver los puzles se observa que la puntuación del usuario va variando y, con ella, su puesto en el ranking.

- PRUEBA POR PARTE DEL USUARIO

Para que el usuario pueda comprobar la nueva funcionalidad, se ha creado un nuevo ejecutable y se le ha instalado en el dispositivo.

En este caso el usuario observa cómo varía las puntuaciones y los puestos del ranking sin errores.

4.4.6 Iteración 6

Para la sexta y última iteración, se incluyen las tareas del **grupo de tareas 7, 8 y 9**.

En esta última iteración se creará un **sistema de cupones** donde el usuario encontrará descuentos para las empresas que quieran publicitarse en la aplicación y, por último, se harán **mejoras en la interfaz principal** para la etapa final del desarrollo de este proyecto.

GRUPO DE TAREAS 7: SISTEMA DE CUPONES

- DISEÑO

Para la creación del sistema de cupones, se han creado dos nuevas opciones en la ventana del menú que nos permitirán acceder a la tienda de cupones y a una lista de los cupones ya conseguidos.

Para comprar cupones en la tienda, el usuario gastará los mismos puntos que ha conseguido en la resolución de los puzles. Para que al gastar puntos no baje su puesto en el ranking, el jugador tendrá una puntuación máxima, usada para calcular el ranking, y unos puntos canjeables, para la compra de cupones.

En la tienda de cupones, encontraremos los cupones que ofertan las empresas en la aplicación [Ilustración 48].

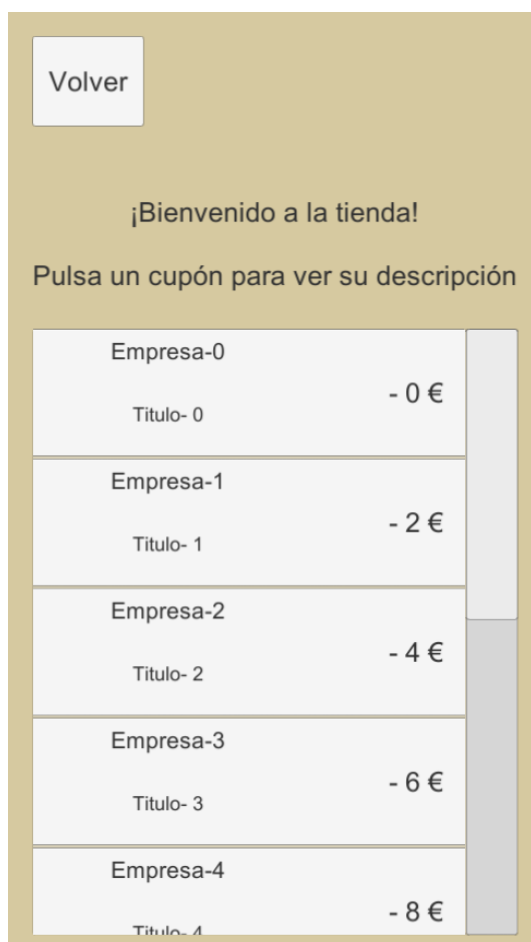


Ilustración 48: Ventana Tienda de cupones

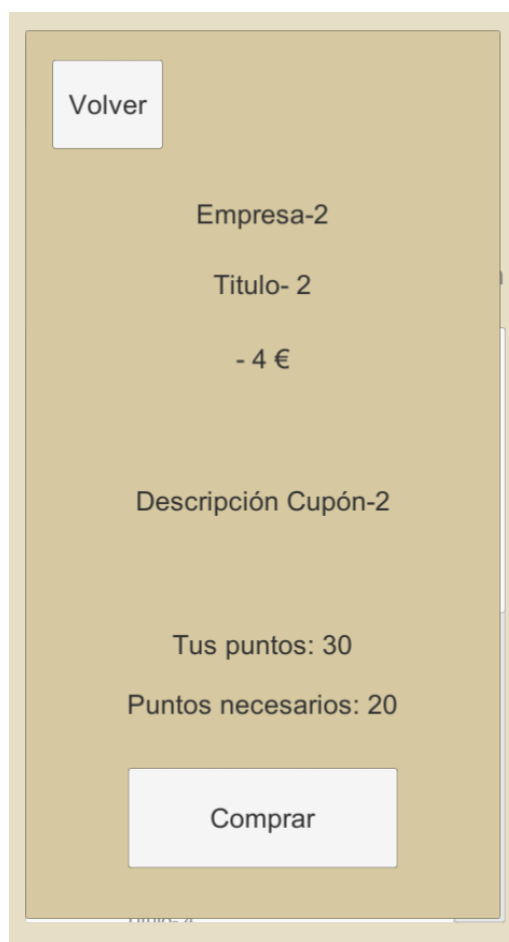


Ilustración 49: Ventana Información de compra

Una vez en la tienda, el usuario seleccionará un cupón y se abrirá una nueva ventana con una descripción más detallada del cupón elegido [Ilustración 49]. Para que el usuario pueda administrarse mejor, también se mostrarán los puntos que se necesitan para comprar dicho cupón, así como los puntos que tiene el usuario en el momento de la compra.

Una vez que el usuario compra un cupón, dicho cupón se almacena en ese usuario y se podrá ver en la ventana de “Mis Cupones” [Ilustración 50].

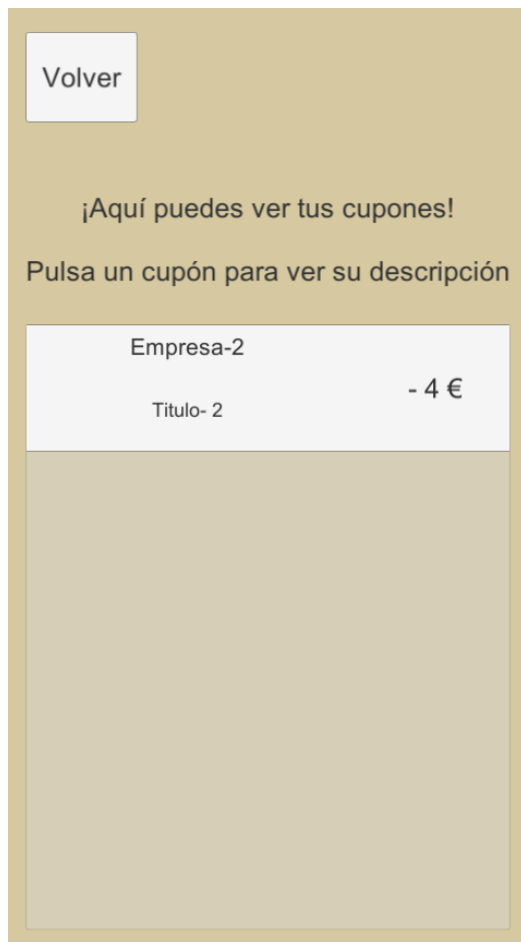


Ilustración 50: Ventana Mis cupones

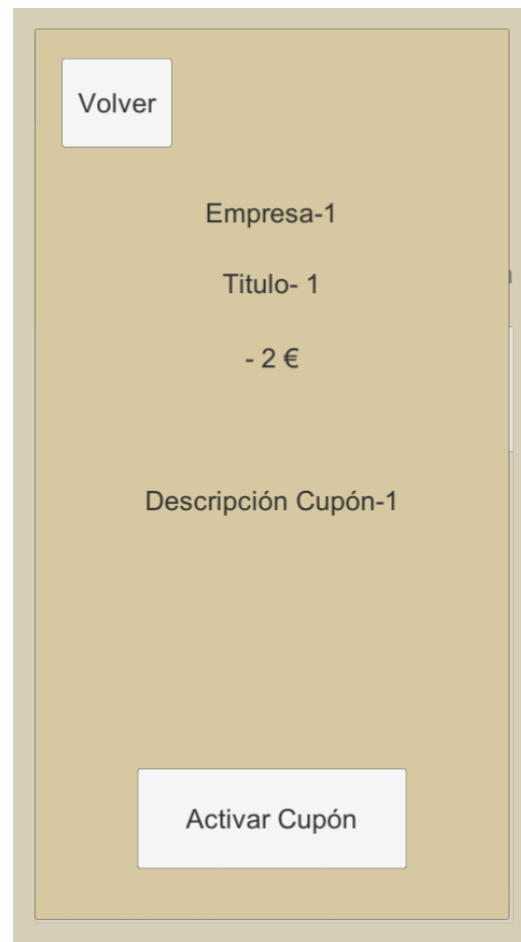


Ilustración 51: Ventana Información de cupón

Tras seleccionar un cupón dentro de la ventana “Mis Cupones”, el usuario podrá ver de nuevo la información detallada de dicho cupón [Ilustración 51] y podrá activarlo para conseguir un descuento. Al pulsar “Activar Cupón” aparecerá la pantalla de activación con código QR [Ilustración 52].



Ilustración 52: Ventana Código QR de cupón

En la pantalla aparecerá un código QR [Ilustración 52] que el usuario debe mostrar a la persona encargada para que esta, con una aplicación secundaria que se haría como futuro trabajo, pueda activar dicho cupón para que se aplique tal descuento.

Un diseño simplificado a nivel de clases de esta última parte a nivel de clases sería el siguiente:

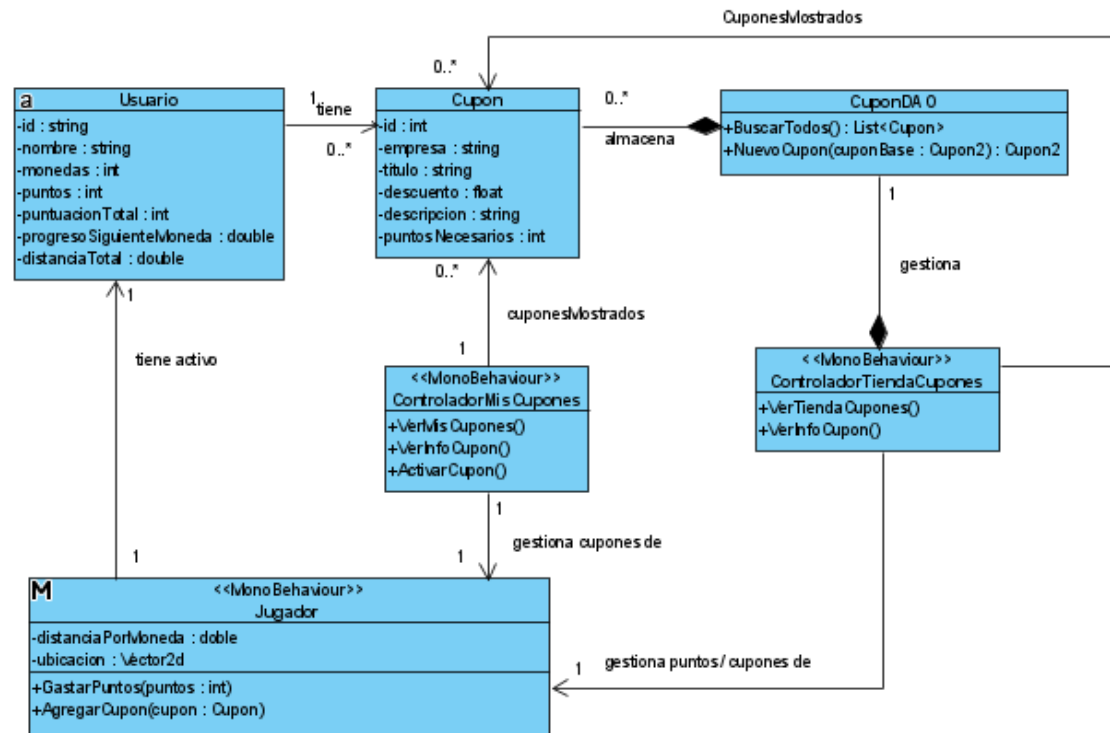


Ilustración 53: Diseño del sistema de cupones

● IMPLEMENTACIÓN

Para la implementación de la tienda de cupones, se ha creado un controlador que mostrará por pantalla los cupones base que están almacenados en el DAO.

Estos cupones base no son los que se utilizarán directamente, sino que, como su nombre indica, servirán de base para crear un nuevo cupón con un constructor copia que posteriormente se asignará al usuario. De esta forma, evitamos que desaparezcan cupones de la tienda porque se usan directamente.

Los cupones requieren un número mínimo de puntos para su compra, por lo que se deshabilitará el botón de “Comprar Cupón” en la ventana de “Tienda de Cupones” en caso de que el número de puntos canjeables del usuario sea menor que el requerido.

Una vez comprado, se añadirá a la lista de cupones que puede manejar el usuario.

Para canjear los cupones obtenidos, el usuario mostrará la ventana del código QR [Ilustración 52] y, una vez que la persona encargada active dicho

cupón con la aplicación secundaria mencionada, se eliminará de la lista de cupones del jugador.

Dado que la aplicación para canjear los cupones no se creará en este proyecto, se simulará la activación de los cupones activando una cuenta atrás de unos segundos que harán que los cupones se eliminen y desaparezcan de la ventana de “Mis cupones”.

- PRUEBAS DE UNIDAD

Para las pruebas de unidad de esta funcionalidad, se ha comprobado a través del log que el número de puntos canjeables que tiene el usuario se decrementa con la compra del cupón y que el botón de compra no se puede pulsar en caso de que el número de puntos canjeables sea inferior a los puntos requeridos para la compra.

En el caso de la activación del cupón, se ha comprobado que este desaparece tras unos segundos al activarlo.

- PRUEBAS DE INTEGRIDAD

En las pruebas de integridad comprobamos que la puntuación del usuario varía correctamente con la compra de usuarios y que esto no afecta al ranking. Al resolver s, la puntuación también varía y deja de nuevo al usuario comprar cupones al superar de nuevo el mínimo requerido.

- PRUEBA POR PARTE DEL USUARIO

Se ha creado un ejecutable con esta funcionalidad y se ha puesto a prueba por parte del usuario.

Tras usarlo unos minutos, el usuario no ha encontrado ningún error en la aplicación. El usuario comprende perfectamente el funcionamiento de los cupones con la interfaz que se ha creado, por lo que este apartado no necesitará cambios.

GRUPO DE TAREAS 8: MEJORAS FINALES

En este grupo de tareas, se crearán **nuevas ventanas** y se harán **pequeñas modificaciones** que mejorarán la interfaz haciendo que sea algo más intuitiva para el usuario.

- DISEÑO

Para empezar, se ha diseñado una nueva ventana accesible desde la ventana del mapa con el botón “Perfil”. Esta ventana mostrará información personal del usuario, así como la cantidad de monedas que tiene actualmente, su puntuación total y sus puntos canjeables [Ilustración 54].



Ilustración 54: Ventana de perfil

Además de esta ventana, se han creado varios tipos de notificaciones que aparecerán cuando el usuario comience una nueva ruta [Ilustración 55], continúe con una ya empezada [Ilustración 56] o complete una [Ilustración 57].

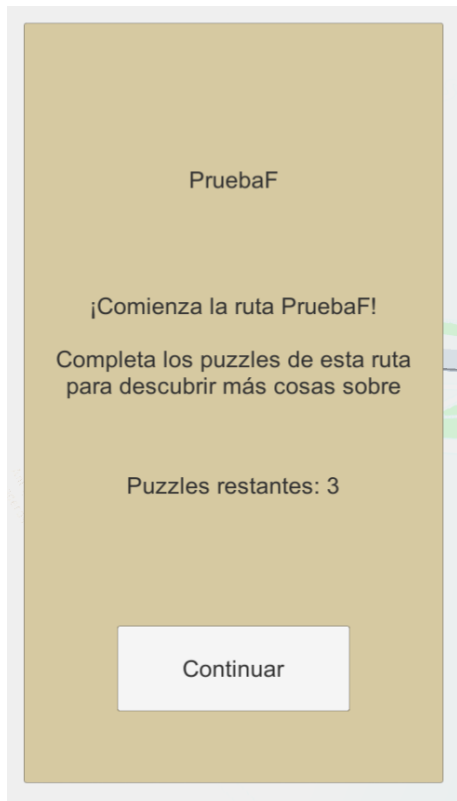


Ilustración 55: Notificación de comienzo de ruta

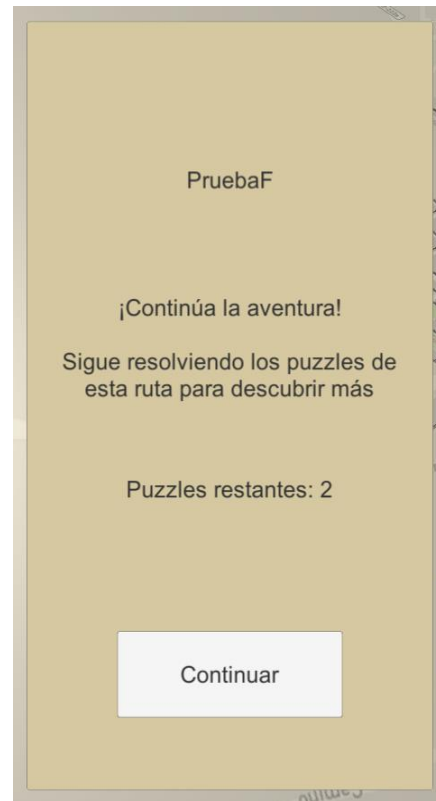


Ilustración 56: Notificación de continuación de ruta



Ilustración 57: Ventana de finalización de ruta

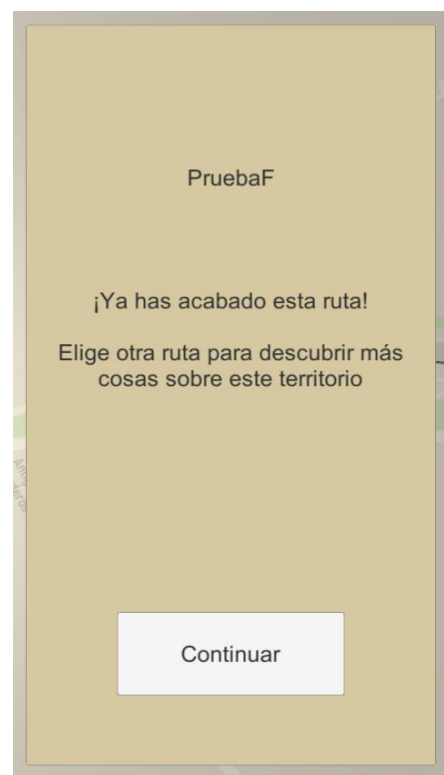


Ilustración 58: Notificación de selección de ruta ya acabada

También se informará al usuario del número restante de puzles que le quedan por resolver de la ruta elegida [Ilustración 55][Ilustración 56]. Además, al completar una ruta se sumarán puntos extra al jugador [Ilustración 57]. Si el jugador escoge una ruta que ya había sido completada también se le mostrará una notificación [Ilustración 58].

Además de estas mejoras, se han sustituido las antiguas imágenes en los puzles utilizando el programa de creación de imágenes vectoriales InkScape.

Los diagramas creados para el diseño de la aplicación no se han modificado más. Estos se encuentran en el [ANEXO 1: DIAGRAMAS DE DISEÑO FINALES](#).

- IMPLEMENTACIÓN

Para implementar la nueva ventana del perfil, se ha creado un nuevo controlador, “ControladorPerfil”, que mostrará la información del usuario por pantalla. En un futuro, este controlador también contendrá la funcionalidad para cambiar información personal y la imagen de usuario.

Para implementar las nuevas notificaciones, la clase Ruta contendrá dos nuevos atributos booleanos que indicarán si ha sido empezada y si ha sido finalizada. En caso de que la ruta no se haya empezado nunca, se mostrará la notificación de nueva ruta y se pondrá el atributo “empezada” a false. Si la ruta ha sido empezada pero no finalizada, se mostrará la notificación de continuar ruta al ser elegida.

Tras la resolución del último puzle de la ruta, el atributo “finalizada” pasará a tener el valor true y aparecerá una notificación que avisará al usuario de que la ruta ha acabado y se le mostrará la cantidad de puntos obtenidos por completar dicha ruta.

Si el jugador escoge una ruta que ya ha sido completada, se le mostrará una notificación avisando de que la ruta ya se ha terminado con anterioridad y la ruta activa no cambiará en tal caso.

- PRUEBAS DE UNIDAD

Se ha comprobado que todas las ventanas funcionan correctamente y que la información mostrada tanto en el perfil como en las notificaciones sea correcta.

- PRUEBAS DE INTEGRIDAD

No se observan fallos al ir resolviendo puzles y seleccionando otras rutas. La información mostrada también cambia de forma correcta.

GRUPO DE TAREAS 9: CIERRE DEL PROYECTO

Por último, se ha creado un logo simple para la aplicación con GIMP, se han comprobado todos los parámetros del ejecutable para que pueda ser ejecutado correctamente en cualquier dispositivo con Android 8 o superior y se ha creado el ejecutable final.

En este se ha incluido una ruta “Prueba Puzles” para que se puedan probar los puzles del prototipo las veces que se desee sin necesidad de andar.

- PRUEBA FINAL POR PARTE DEL USUARIO

El usuario ha probado la aplicación con toda la funcionalidad disponible y no ha encontrado problema alguno a excepción de pequeños fallos en la resolución de la aplicación al haber cambiado de dispositivo.

Para solucionar esto se han retocado todas las ventanas haciendo que se adapten de forma dinámica al dispositivo.

Tras comprobar que todo funciona correctamente se ha procedido al cierre del proyecto.

Capítulo 5: Proyección empresarial y comercialización

En este apartado se analizarán algunos aspectos a tener en cuenta para la comercialización del videojuego, así como un pequeño estudio de este proyecto dentro de la empresa.

5.1 Investigación de mercado

Este proyecto va más allá de ser un simple videojuego, ofreciendo un servicio publicitario muy potente para locales comerciales, bares y restaurantes, hoteles, etc.

La aplicación actualmente se puede definir como un videojuego basado en la localización del usuario que sirve como guía turística en un determinado territorio y que además integra publicidad y descuentos de empresas del territorio en cuestión.

Se ha buscado posible competencia, siendo esta quizás videojuegos del mismo estilo, como *Pokémon GO* [7] y algunas páginas web de descuentos como *Groupon.es* [32], pero no se han encontrado resultados de ninguna aplicación parecida a la que se desarrolla en este trabajo de fin de grado. Esto es indicador de que la idea es innovadora y puede llegar a convertirse en un proyecto empresarial de gran calibre.

Nuestro público objetivo serán todas aquellas personas apasionadas por conocer la parte turística de su ciudad y otras ciudades, así como locales comerciales, bares y otras empresas que deseen publicitarse en la aplicación.

5.2 Segmentos de mercado

Se estudiará el perfil de los clientes basados en los siguientes criterios:

- **Geográfico:** El proyecto desarrollado actualmente solo contiene rutas y puzzles para la provincia de Jaén, pero se ha mostrado que fácilmente se puede expandir a cualquier nivel territorial, por lo que cualquiera con un

móvil Android (se incluirán más plataformas en un futuro) podrá ser parte de la experiencia CoolToTrip.

- **Demográfico:** La edad objetivo de este proyecto se comprende entre un mínimo de 7 años, definido por la clasificación PEGI [\[5\]](#), y una edad de unos 75 años, dado que este videojuego depende mucho del nivel de movilidad de una persona. Esto lo podemos observar en el gráfico que muestra la capacidad de movilidad física según la edad de la persona [\[Ilustración 59\]](#).

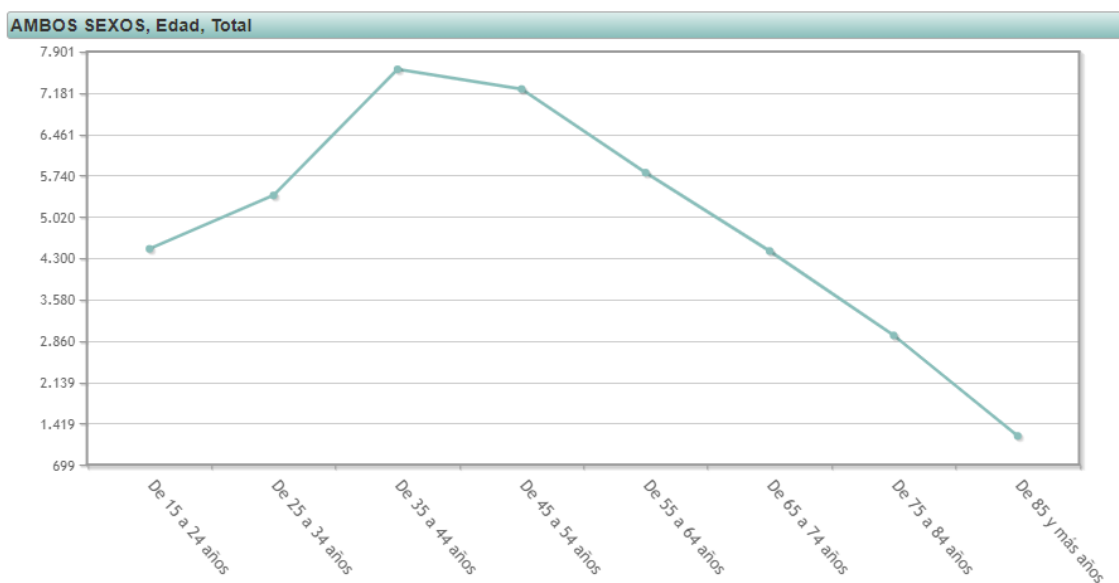


Ilustración 59: Grafica de la capacidad de movilidad según la edad de la persona [\[33\]](#)

5.3 Estrategia de promoción y marketing

Para la promoción de la empresa se pondrán en práctica las siguientes medidas:

- **Página Web:** la propia página web creada es de vital importancia para la publicidad del videojuego, además de ser la principal vía para captar empresas que se quieran unir a la plataforma.
- **Posicionamiento web SEO:** SEO o Search Engine Optimization consiste en una serie de técnicas y estrategias para mejorar el posicionamiento de la página web en los buscadores. SEO es una

estrategia que no depende de pagos para conseguir un buen posicionamiento [\[34\]](#).

Para conseguir esto tendremos en cuenta las siguientes medidas:

- **Definir y usar pocas palabras clave** evitando que Google detecte estas como spam, lo que afectaría de forma negativa.
- **Referenciar otros sitios** que pudieran complementar nuestro servicio.
- **Presentar contenido de la forma más limpia y sencilla posible**, intentando conquistar la posición cero de Google, los Featured Snippets. Estos son cuadros que aparecerán encima de todos los resultados de una consulta por parte del usuario dando una respuesta muy concreta a este. Además, es de vital importancia dado que el resultado de una búsqueda por voz normalmente es un Featured Snippet.
- **Redes sociales:** sin duda alguna, la mayor publicidad a día de hoy se encuentra en redes sociales como Twitter, Instagram o Facebook. Estar en redes sociales permitirá la publicitación de la plataforma llegando a un número muy alto de lo que podrían ser potenciales usuarios de la aplicación.
- **Plataformas multimedia:** La presencia de cortos publicitarios y vídeos explicativos de la plataforma en YouTube o Twitch hará que la empresa llegue a aún más personas.

5.4 Fuentes de ingresos

La aplicación desarrollada tendrá en cuenta las siguientes fuentes de ingresos:

- **Usuarios:** Si bien la aplicación es gratuita, se incluirán en un futuro micropagos para la obtención de nuevos puzzles y rutas con los que el usuario podrá continuar explorando el territorio deseado

- **Empresas:** Uno de los principales ingresos se hará por parte de todas aquellas empresas interesadas en publicitarse en la aplicación, pudiendo gestionar sus cupones de descuento dentro de esta y apareciendo en el mapa que se le muestra al usuario mientras recorre las calles de la ciudad.
- **Instituciones:** Esta aplicación podrá ser subvencionada por parte de los ayuntamientos e instituciones para la expansión de esta a aquellos territorios interesados en el proyecto.

5.5 Expansión del proyecto

En este proyecto se ha tenido muy presente que para que funcione en el mercado debe ser fácilmente escalable, pero hay algunos aspectos que es conveniente explicar para una mayor facilidad a la hora de añadir novedades al proyecto.

5.5.1 Ampliación de rutas y puzles

Explicaremos a continuación de forma simplificada lo que se debe de hacer a la hora de crear un nuevo puzle o una nueva ruta.

Para la creación de puzles tenemos dos opciones, escoger un tipo de puzle ya creado o crear un nuevo tipo:

- **Tipo de puzle ya creado:** Para crear un nuevo puzle a partir de los ya disponibles, simplemente debemos indicar el id del puzle, el número de intentos máximos, la puntuación máxima y la ubicación del mismo, así como sus pistas (en el prototipo se generan automáticamente).
- **Tipo de puzle aún no creado:** Para un puzle de un nuevo tipo, el proceso es algo más complicado. Se debe de crear una nueva especificación de la clase `Puzzle<T>` que implemente la interfaz “`Evaluable<T>`” indicando el tipo de dato que manejará el puzle. También se debe crear una especificación de la clase `ControladorPuzzle<T>` de forma que solo controle puzles del tipo de la nueva especificación de `Puzzle<T>`. En este controlador se debe de codificar su comportamiento con la escena. Después, simplemente debemos hacer los mismos pasos que en el puzle ya creado.

Para finalizar, en el prefab Puzzle, debemos agregar los elementos necesarios en el prefab “ElementosPuzzle” [Ilustración 60] de forma que se muestren en la ventana de puzzle.

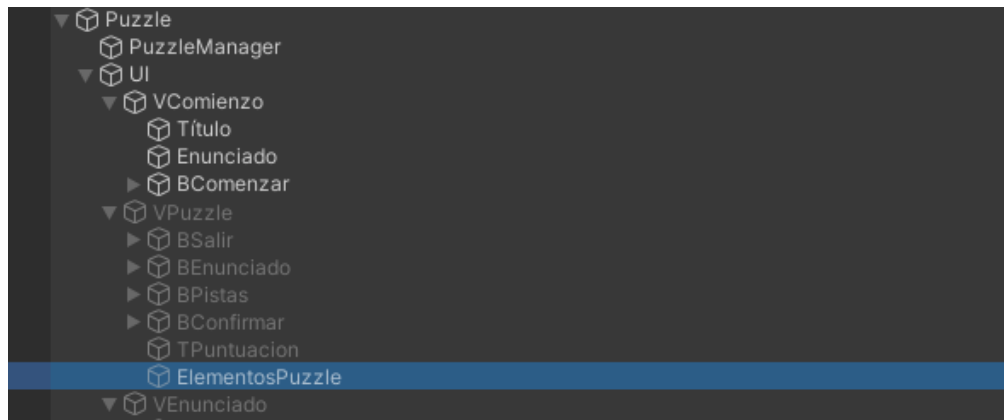


Ilustración 60: Situación de los elementos del puzzle en la jerarquía de objetos de Unity

Para la inclusión de nuevas rutas simplemente debemos indicar el id de la ruta, su nombre, la localidad donde se encuentra y la lista de puzzles asociados a esta.

5.5.2 Factores a tener en cuenta

Se expondrá a continuación algunos factores a tener en cuenta para la inclusión de nuevos puzzles y rutas.

Para empezar, se debería de disponer de un grupo amplio de personas que se encarguen de hacer un listado de todos aquellos sitios con interés cultural que sirvan de base para los puzzles. Esto, llevado a una expansión mayor como puede ser el mundo entero, necesitaría de varios equipos que hagan este trabajo en diferentes territorios de todos los países del mundo donde se quiera expandir el mercado.

Sería necesario también contar con un equipo de traductores e intérpretes que ayuden con la tarea de llevar a cabo la internacionalización de la aplicación para su uso en otros países.

Por último, un factor muy importante es la cultura dentro de estos países, teniendo que adecuar los puzzles a sus costumbres y tradiciones de forma que no resulten impactantes para el usuario de ese país. Esto puede tenerse en

cuenta a nivel de interfaz con el tipo de imágenes expuestas, la forma de transmitir el enunciado del puzzle o, incluso, los colores utilizados.

5.6 Estimación de costes del proyecto completo

En este apartado se hará una estimación de costes, igual que se ha hecho para este trabajo de fin de grado, pero orientado al proyecto en su totalidad, teniendo en cuenta **la terminación de este videojuego, la creación de la página web y la creación de la aplicación de gestión y lectura de cupones, así como su mantenimiento.**

En esta estimación se harán los cálculos sin tener en cuenta un trabajo de internacionalización de la empresa, siendo su expansión solo a nivel de España.

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net*, *Indeed.com* y *Jobted.es*.

5.6.1 Completitud del videojuego

Para completar el videojuego, haría falta hacer gran parte de los puntos mencionados en el apartado de [TRABAJO FUTURO](#).

Teniendo en cuenta esto, se estima que la cantidad de horas que faltan hasta su completitud es de otras **1200 horas**, entrando en estos nuevos roles:

- Un compositor/productor para la música.
- Un experto en seguridad informática.

Todos los gastos calculados a continuación se estiman para las 1200 horas extra, por lo que se sumarán al final a los gastos estimados del prototipo.

Empezaremos estimando los gastos de recursos digitales teniendo en cuenta que **1200 horas son aproximadamente 5 meses**, trabajando ahora a jornada completa, es decir, **8 horas al día**.

En el caso de **XMind** [\[20\]](#), se calculará el coste con el plan para 6 meses cuyo precio en euros sería de **32,80€ (40 dólares)**, que al mes serían **5,46€**.

Con el programa **Visual Paradigm** [21] haremos los cálculos teniendo una suscripción mensual como “Modeler” siendo esta de **4,92€ (6 dólares) al mes**.

El coste de XMind y Visual Paradigm **se dividirán entre tres**, número de proyectos que se estarían desarrollando en paralelo, y se multiplicará por los 5 meses que duraría este proyecto.

También hay que añadir a la lista de programas utilizados el programa Avid Pro Tools First [35] que utilizará el compositor/productor de sonido.

Programa	Coste/mes	Coste total
GIMP	0€	0€
Inkscape	0€	0€
Avid Pro Tools First	0€	0€
XMind	1,82€	9,10€
Visual Paradigm	1,64€	8,20€
JustInMind	0€	0€
Blender	0€	0€
Unity*	0€	0€
MapBox*	0€	0€
TOTAL:		17,30€

Tabla 9: Estimación de costes de recursos digitales del videojuego

*Para el coste de **Unity** [13] y **MapBox** [4] se seguirán usando los planes **gratuitos** ya que no se habría lanzado el proyecto aún y no se estarían superando los límites de estos planes. Esto sí cambiará cuando se calcule el mantenimiento del proyecto.

A continuación, se estimarán los costes asociados a los recursos materiales.

Se tendrá en cuenta que se necesitarán dos ordenadores de sobremesa iguales que el usado para la creación del prototipo.

El ordenador de sobremesa tiene un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. **El coste calculado de un ordenador por hora es de 0,021€.** Dado que se dispondrá de 2 ordenadores, el número de horas a tener en cuenta será de **600 por ordenador.**

Seguimos teniendo un **coste por electricidad de 435,98€ al año** si el equipo estuviera encendido siempre, quedando en **0,049€ la hora por equipo.**

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño de ventanas se aproximará a otros **7€.**

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes** [27]. Hay que multiplicarla por 5 meses de desarrollo y dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **48,25€.**

Concepto	Uso	Precio total
Ordenador 1	Desarrollo del proyecto	12,06€
Ordenador 2	Desarrollo del proyecto	12,06€
Tableta gráfica	Edición de imágenes	0€
Material de oficina	Creación de diagramas e interfaces	7€
Electricidad	Consumo del equipo de desarrollo	58,80€
Internet	Consultas, investigación y desarrollo	48,25€
Total:		138,17€

Tabla 10: Estimación de costes de recursos materiales del videojuego

Por último, se estimará el coste por personal según los diferentes roles. El cálculo se hará de la misma forma que la estimación de costes por personal del prototipo teniendo en cuenta el salario promedio al año junto con los 251 días laborables de 2021 y las 8 horas de una jornada completa.

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net* [28], *Indeed.com* [29] y *Jobted.es* [30].

Como se ha calculado anteriormente, un jefe de proyecto cobra alrededor de unos 40.000€ al año. Su **sueldo por hora** es de unos **19,92€**.

Un programador de videojuegos cobra al año unos 32.100 €. Haciendo el cálculo, el coste por hora de un programador de videojuegos con este dato es de **15,99€/h**. Se tendrá en cuenta que se contratará a un segundo programador por lo que el trabajo de programación ahora se dividirá aproximadamente entre dos.

Para el diseño e ilustración de imágenes, modelos y la interfaz, el salario de un diseñador gráfico es de unos 19.297€ al año. Según esto, el coste por hora será de **9,61€/h**.

Un experto en ciberseguridad cobra de media al año unos 45.000€, por lo que su salario medio sería de **22,41€ por hora**.

Por último, un productor musical cobra entre 14.000€ y 22.000€ al año según la página de *campustraining.es* [36]. Por lo tanto, el salario medio de productor musical es de unos 18.000€ al año, lo que sería unos **8,96€ la hora**.

A continuación, se muestra una tabla con el número de horas estimadas que trabajará cada rol junto con su coste dentro del proyecto.

Rol	Horas	Precio total
Jefe de proyecto	250	4.980€
Programador de videojuegos	600	9.594€
Diseñador gráfico	90	864,90€
Experto en ciberseguridad	190	4.257,90€
Compositor/Productor musical	70	627,20€
Total:		20.324€

Tabla 11: Estimación de costes de personal implicado en el videojuego

Una vez calculados, se hará la suma total de todos los gastos de recursos digitales, recursos materiales y personal implicado:

Concepto	Coste
Recursos digitales	17,30€
Costes materiales	138,17€
Personal	20.324€
Total:	20.479,47€

Tabla 12: Estimación total de costes del videojuego

La suma total de los gastos sería de unos **20.597.46€** a lo que le sumaremos un **15% del total para gastos imprevistos** en el proyecto, quedando un presupuesto de **23.551,39€**.

Por último, para el videojuego debemos sumar también el coste asociado al prototipo desarrollado por el alumno quedando el presupuesto total en **28.716,69€**.

5.6.2 Página web

Esta página web contendrá en principio la información acerca de todos los lugares turísticos que aparecerán en el videojuego e información sobre las rutas disponibles y eventos especiales de los territorios donde ha llegado la aplicación.

Esta página se estima que estará acabada en unas **240 horas**, es decir, **1 mes** trabajando 8 horas al día.

Para esto, además de algunos de los programas ya usados anteriormente, se debe incluir un **servicio de hosting** para nuestra página web. Para ello se contratará la tarifa Standard de Hostalia.[\[37\]](#)*

Programa	Coste/mes	Coste total
XMind	1,82€	1,82€
Visual Paradigm	1,64€	1,64€
JustInMind	0€	0€
Hosting Web de Hostalia*	0€	0€
TOTAL:		3,46€

Tabla 13: Estimación de costes de recursos digitales de la página web

*Hostalia ofrece 3 meses gratis. Tras esto, serán 3,99€ al mes, que se reflejarán en el mantenimiento de la página.

A continuación, se estiman los costes de los recursos materiales utilizados.

Seguiremos utilizando un ordenador de sobremesa que tiene un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. **El coste calculado de un ordenador por hora es de 0,021€.**

El coste por electricidad es de 435,98€ al año si el equipo estuviera encendido siempre, quedando en 0,049€ la hora por equipo.

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño de la página se aproximará a **3€**.

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes** [27]. Hay que dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **9,65€**.

Concepto	Uso	Precio total
Ordenador	Desarrollo de la aplicación	5,04€
Material de oficina	Creación de diagramas e interfaces	3€
Electricidad	Consumo del equipo de desarrollo	11,76€
Internet	Consultas, investigación y desarrollo	9,65€
Total:		29,45€

Tabla 14: Estimación de costes de recursos materiales de la página web

Por último, se estimarán los costes asociados al personal.

Para la creación de la página web, se necesitará una persona encargada como **jefe de proyecto y un programador web**.

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net* [28], *Indeed.com* [29] y *Jobted.es* [30].

Como se ha calculado anteriormente, un jefe de proyecto cobra alrededor de unos 40.000€ al año. Su **sueldo por hora** sería de unos **19,92€**.

Un programador web con experiencia media cobra al año unos 24.678€. Haciendo el cálculo, el coste por hora de un programador web es de **12,28€/h**.

Un experto en ciberseguridad cobra de media al año unos 45.000€, por lo que su salario medio sería de **22,41€ por hora**.

Rol	Horas	Precio total
Jefe de proyecto	60	1.195,20€
Programador web	140	1.719,20€
Experto en ciberseguridad	40	896,40€
Total:		3.810,80€

Tabla 15: Estimación de costes de personal implicado en la página web

A continuación, se muestra en la tabla la suma de todos los costes dando el coste total estimado del proyecto:

Concepto	Coste
Recursos digitales	3,46€
Costes materiales	29,45€
Personal	3.810,80€
Total:	3.843,71€

Tabla 16: Estimación total de costes de la página web

La suma total de los gastos sería de unos **3.843,71€** a lo que le sumaremos un **15% del total para gastos imprevistos** en el proyecto, quedando un presupuesto de **4.420,27€**.

5.6.3 Creación del gestor de cupones

Esta aplicación será descargada por las empresas que se suscriban para publicitarse dentro del videojuego. Con esta aplicación la empresa podrá añadir, borrar, ver y editar los cupones de descuento que publiquen, así como los datos de la empresa. Con esta misma aplicación, la persona encargada podrá también leer el código QR de un cupón, reflejando el descuento en la factura e inhabilitando dicho cupón.

Esta aplicación será muy simple, teniendo una interfaz básica para las funciones de la misma.

Para esta aplicación se estima que la duración del desarrollo de la misma será de unas **240 horas (1 mes a jornada completa)**.

Programa	Coste/mes	Coste total
XMind	1,82€	1,82€
Visual Paradigm	1,64€	1,64€
JustInMind	0€	0€

TOTAL:	3,46€
---------------	--------------

Tabla 17: Estimación de costes de recursos digitales del gestor de cupones

A continuación, se estimarán los costes asociados a los recursos materiales.

El ordenador de sobremesa tiene un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. **El coste calculado de un ordenador por hora es de 0,021€.**

Seguimos teniendo un **coste por electricidad de 435,98€ al año** si el equipo estuviera encendido siempre, quedando en **0,049€ la hora por equipo.**

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño de ventanas se aproximará a **1,5€.**

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes** [27]. Hay que dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **9,65€.**

Concepto	Uso	Precio total
Ordenador	Desarrollo de la aplicación	5,04€
Material de oficina	Creación de diagramas e interfaces	3€
Electricidad	Consumo del equipo de desarrollo	11,76€
Internet	Consultas, investigación y desarrollo	9,65€
Total:		29,45€

Tabla 18: Estimación de costes de recursos materiales del gestor de cupones

Para esta aplicación, se necesitará una persona encargada como **jefe de proyecto y un programador**, así como de un nuevo **experto en ciberseguridad.**

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net* [28], *Indeed.com* [29] y *Jobted.es* [30].

Como se ha calculado anteriormente, un jefe de proyecto cobra alrededor de unos 40.000€ al año. Su **sueldo por hora** sería de unos **19,92€**.

Un programador con experiencia media cobra al año unos 29.800€. Haciendo el cálculo, **el coste por hora de un programador es de 14,84€/h**.

Un experto en ciberseguridad cobra de media al año unos 45.000€, por lo que su salario medio sería de **22,41€ por hora**.

Rol	Horas	Precio total
Jefe de proyecto	60	1.195,20€
Programador	125	1.855€
Experto en ciberseguridad	55	1.232,55€
Total:		4.282,75€

Tabla 19: Estimación de costes de personal implicado en el gestor de cupones

A continuación, se muestra en la tabla la suma de todos los costes dando el coste total estimado del proyecto:

Concepto	Coste
Recursos digitales	3,46€
Costes materiales	29,45€
Personal	4.282,75€
Total:	4.315,66€

Tabla 20: Estimación total de costes del gestor de cupones

La suma total de los gastos sería de unos **4.315,66€** a lo que le sumaremos un **15% del total para gastos imprevistos** en el proyecto, quedando un presupuesto de **4.963€**.

5.7 Estimación de costes de mantenimiento

A continuación, se hará una estimación sobre los costes de mantenimiento de los tres sistemas mencionados anteriormente (el videojuego, la página web y el gestor de cupones).

Para el mantenimiento, se estimarán los gastos mensuales que están asociados a los tres proyectos (y no por un número de horas concretas como se ha hecho anteriormente). Gastos como los programas utilizados e internet se dividirán entre estos tres.

Además, para el mantenimiento de los mismos, estos gastos mensuales se harán teniendo en cuenta que se contratan a las personas necesarias por rol, ocupando ahora el alumno solo el rol de uno de los programadores del videojuego.

Se tendrá en cuenta el siguiente personal:

- Personal para el videojuego:
 - Dos programadores de videojuegos
 - Un diseñador gráfico
- Personal para la página web:
 - Un programador web
- Personal para el gestor de cupones:
 - Un programador de software
- Personal común:
 - Experto en seguridad
 - Jefe del servicio al cliente
 - Dos operarios de atención al cliente

El coste asociado al personal común como sus sueldos y sus gastos materiales se calcularán y se sumarán al final de este apartado.

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net*, *Indeed.com* y *Jobted.es*.

5.7.1 Videojuego

Para el videojuego, el mantenimiento del mismo se basará en la corrección de fallos e incidencias que pueda haber, así como la adición de algunas nuevas funcionalidades que vayan mejorando y actualizando la aplicación.

De nuevo, se necesitarán algunos de los programas que hemos estado utilizando durante el desarrollo del mismo y que ayudan en la organización de tareas y diseño como XMind, Visual Paradigm y JustInMind.

Además de esto, es necesario contar para el mantenimiento del mismo con la tarifa anual de Unity Plus [\[38\]](#), teniendo en cuenta que al usar la aplicación más usuarios y ganar más, pasaremos del límite de Unity gratuito. Esto costaría **al mes 66,5€** por los dos programadores.

También hay que hacer el cálculo para la aplicación de mapas MapBox [\[4\]](#) al pasarnos del límite de usuarios y consultas. Para ello estimaremos que tenemos unos 200.000 usuarios en la aplicación, teniendo que pagar **640€ al mes**.

Programa	Coste/mes
XMind	1,82€
Visual Paradigm	1,64€
JustInMind	0€
Unity	66,5€
MapBox	640€
TOTAL:	709,96€

Tabla 21: Estimación de costes de recursos digitales del mantenimiento del videojuego

Para el mantenimiento se calcularán también los costes materiales al mes que se tendrán por el equipo informático, internet, etc.

Los ordenadores de sobremesa tienen un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. El coste

calculado de un ordenador será de 15,33€ al mes. Teniendo en cuenta que contaremos con un, el gasto es de **45,99€ por los tres ordenadores al mes.**

El coste por electricidad sería de **142,39€ al año por equipo** si este estuviera encendido las **8 horas** de la jornada completa. Calculando el coste total de los tres equipos serían unos **35,60€ al mes.**

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño se aproximará a **4€ al mes.**

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes.** Hay que dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **9,65€.**

Concepto	Uso	Coste/Mes
Ordenadores	Mantenimiento de la aplicación	45,99€
Material de oficina	Creación de diagramas e interfaces	4€
Electricidad	Consumo del equipo de desarrollo	35,60€
Internet	Consultas, investigación y desarrollo	9,65€
Total:		95,24€

Tabla 22: Estimación de costes de recursos materiales del mantenimiento del videojuego

Para un correcto mantenimiento, se seguirá contando con el trabajo de los programadores de videojuegos que cobran 32.100€ al año, que entre 14 pagas son **2292,86€ al mes.**

También se tendrá contratado un diseñador gráfico que cobra al año unos 19.297€ que serán unos **1378,36€ al mes.**

Los sueldos medios por rol se han consultado en las páginas de [InfoJobs.net](#) [28], [Indeed.com](#) [29] y [Jobted.es](#) [30].

Rol	Sueldo/mes
Programador de videojuegos	2.292,86€
Diseñador gráfico	1.378,36€
TOTAL:	3.671,22€

Tabla 23: Estimación de costes del personal implicado en el mantenimiento del videojuego

A continuación, se muestra en la tabla la suma de todos los costes dando el coste total estimado del mantenimiento por mes:

Concepto	Coste
Recursos digitales	709,96€
Costes materiales	95,24€
Personal	3.671,22€
Total:	4.476,42€

Tabla 24: Estimación total de costes del mantenimiento del videojuego

Esto se sumará al final a los costes por el personal común para obtener el coste del mantenimiento total.

5.7.1 Página Web

El mantenimiento de la página web consistirá en la corrección de errores por parte del programador web e inclusión de nueva información si fuera necesario.

Para esta aplicación, además de algunos de los programas ya usados anteriormente, se debe incluir un **servicio de hosting** para nuestra página web. Para ello se contratará la tarifa Standard de Hostalia [\[37\]](#), que serían **3,99€ al mes**.

Programa	Coste/mes	Coste total
XMind	1,82€	1,82€
Visual Paradigm	1,64€	1,64€
JustInMind	0€	0€
Hosting Web de Hostalia*	0€	3,99€
TOTAL:		7,45€

Tabla 25: Estimación de costes de recursos digitales del mantenimiento de la página web

* Hostalia ofrece los 3 primeros meses gratuitos, pero para nuestros cálculos, omitiremos este dato para calcular el mantenimiento en un año normal de la página web.

Para el mantenimiento se calcularán también los costes materiales al mes que se tendrán por el equipo informático, internet, etc.

Un ordenador de sobremesa tiene un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. El coste calculado de un ordenador será de **15,33€ al mes**.

El coste por electricidad sería de **142.39€ al año por equipo** si este estuviera encendido las **8 horas** de la jornada completa, es decir **11,87€ al mes**.

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño se aproximará a **1,5€**.

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes** [27]. Hay que dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **9,65€**.

Concepto	Uso	Coste/Mes
Ordenadores	Mantenimiento de la aplicación	15,33€
Material de oficina	Creación de diagramas e interfaces	1,5€
Electricidad	Consumo del equipo de desarrollo	11,87€
Internet	Consultas, investigación y desarrollo	9,65€
Total:		38,35€

Tabla 26: Estimación de costes de recursos materiales del mantenimiento de la página web

Para el mantenimiento de la página web habrá un programador web encargado. Un programador web cobra 24.678€ al año, que entre 14 pagas son **1.762,71€ al mes.**

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net* [28], *Indeed.com* [29] y *Jobted.es* [30].

Rol	Sueldo/mes
Programador web	1.762,71€
TOTAL:	1.762,71€

Tabla 27: Estimación de costes del personal implicado en el mantenimiento de la página web

A continuación, se muestra en la tabla la suma de todos los costes dando el coste total estimado del mantenimiento por mes:

Concepto	Coste
Recursos digitales	7,45€
Costes materiales	38,35€
Personal	1.762,71€
Total:	1.808,51€

Tabla 28: Estimación total de costes del mantenimiento de la página web

Esto se sumará al final a los costes por el personal común para obtener el coste del mantenimiento total.

5.7.3 Gestor de cupones

Para la aplicación de gestión de cupones que tendrán las empresas asociadas, el mantenimiento se basará en corrección de los errores de la misma y adición de alguna funcionalidad nueva si se diera el caso.

De nuevo, se necesitarán algunos de los programas que hemos estado utilizando durante el desarrollo del mismo y que ayudan en la organización de tareas y diseño como XMind, Visual Paradigm y JustInMind.

Programa	Coste/mes
XMind	1,82€
Visual Paradigm	1,64€
JustInMind	0€
TOTAL:	3,46€

Tabla 29: Estimación de costes de recursos digitales del mantenimiento del gestor de cupones

Se calcularán también los costes materiales al mes que se tendrán por el equipo informático, internet, etc.

Se contará con un ordenador de sobremesa igual que los utilizados para los demás proyectos que tiene un coste de 920€ y cuya vida útil se estima que

es de unos 5 años sin cambiar ningún componente. El coste calculado r será de **15,33€ al mes.**

El coste por electricidad sería de **142.39€ al año por equipo** si este estuviera encendido las **8 horas** de la jornada completa, es decir **11,87€ al mes.**

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño se aproximará a **1,5€.**

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes.** Hay que dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **9,65€.**

Concepto	Uso	Coste/Mes
Ordenadores	Mantenimiento de la aplicación	30,16€
Material de oficina	Creación de diagramas e interfaces	1,5€
Electricidad	Consumo del equipo de desarrollo	23,73€
Internet	Consultas, investigación y desarrollo	9,65€
Total:		65,04€

Tabla 30: Estimación de costes de recursos materiales del mantenimiento del gestor de cupones

Para un correcto mantenimiento, se seguirá contando con el trabajo de un programador, Este programador de aplicaciones tiene de media un sueldo de unos 29.800€ al año, que dividido entre 14 pagas son unos **2.128,57€ al mes.**

Los sueldos medios por rol se han consultado en las páginas de [InfoJobs.net](#) [28], [Indeed.com](#) [29] y [Jobted.es](#) [30].

Rol	Sueldo/mes
Programador	2.128,57€
TOTAL:	2.128,57€

Tabla 31: Estimación de costes del personal implicado en el mantenimiento del gestor de cupones

A continuación, se muestra en la tabla la suma de todos los costes dando el coste total estimado del mantenimiento por mes:

Concepto	Coste
Recursos digitales	3,46€
Costes materiales	65,04€
Personal	2.128,57€
Total:	2.197,07€

Tabla 32: Estimación total de costes del mantenimiento del gestor de cupones

Esto se sumará al final a los costes por el personal común para obtener el coste del mantenimiento total.

5.7.4 Gasto total de mantenimiento

Para los gastos comunes de mantenimiento se calcularán estos tomando como personal común al experto en ciberseguridad y el equipo de servicio técnico, formado por el jefe de servicio al cliente y dos operarios.

Se contará con cuatro ordenadores de sobremesa igual que los utilizados para los demás proyectos que tienen un coste de 920€ y cuya vida útil se estima que es de unos 5 años sin cambiar ningún componente. El coste calculado será de **61,33€ al mes** por todos los ordenadores.

El coste por electricidad sería de **142,39€ al año por equipo** si este estuviera encendido las **8 horas** de la jornada completa, es decir 11,87€ al mes. Esto multiplicado por cuatro ordenadores daría un total de **47,48€ al mes**.

El material de oficina utilizado como bolígrafos, lápices y folios para la realización en sucio de diagramas y diseño se aproximará a **4€**.

Se seguirá teniendo el mismo contrato para la fibra óptica, siendo el gasto de **28,95€ al mes** [27]. Hay que dividirla teniendo en cuenta que también harán uso de esta otros dos proyectos, dando un total de **9,65€**.

Concepto	Uso	Coste/Mes
Ordenadores	Mantenimiento de la aplicación	61,33€
Material de oficina	Creación de diagramas e interfaces	4€
Electricidad	Consumo del equipo de desarrollo	47,48€
Internet	Consultas, investigación y desarrollo	9,65€
Total:		122,46€

Tabla 33: Estimación de costes de recursos materiales del personal común de mantenimiento

Se calculará ahora el coste total por el personal común al mes.

Los sueldos medios por rol se han consultado en las páginas de *InfoJobs.net* [28], *Indeed.com* [29] y *Jobted.es* [30].

Un experto en ciberseguridad cobra de media al año unos 45.000€, dividido entre 14 pagas da como resultado unos **3.214,29€ al mes**.

Un jefe de servicio al cliente gana aproximadamente unos 24.772€ al año, dividido entre 14 nos daría unos **1.769,43€ al mes**.

Un operario de servicio al cliente cobra de sueldo unos 16.682€ al año de media, que entre 14 pagas serían unos 1.191,57€ al mes. Multiplicando esto por dos personas serían **2.383,14€ al mes**.

Rol	Sueldo/mes
Experto en ciberseguridad	3.214,29€
Jefe de servicio al cliente	1.769,43€
Operarios de servicio al cliente	2.383,14€
TOTAL:	7.366,86€

Tabla 34: Estimación de costes de costes del personal común de mantenimiento

Sumando todo esto, los gastos comunes quedan en **7.489,32€ al mes.**

A continuación, se muestra en la tabla la suma de todos los costes dando el coste total estimado del mantenimiento de todo por mes:

Concepto	Coste
Videojuego	4.476,42€
Página web	1.808,51€
Gestor de cupones	2.128,57€
Gastos comunes	7.489,32€
Total:	15.902,82€

Tabla 35: Estimación total de costes de mantenimiento

El coste total del mantenimiento estimado por el conjunto del videojuego, la página web y la aplicación de gestión de cupones es de **15.902,82€ al mes.**

Capítulo 6: Conclusiones

6.1 Problemas encontrados

En este apartado se comentarán los problemas más importantes que se han dado en la etapa de desarrollo del proyecto.

Uno de los problemas a resolver ha sido la gestión de la localización del jugador.

En este caso, la curva de aprendizaje de la herramienta MapBox es complicada y, aunque se disponía de recursos como videotutoriales, al tener que gestionar de forma externa la localización, se han tenido que hacer ciertas modificaciones en la herramienta como se ha mencionado en el proceso de desarrollo. Si bien el modificarla no ha sido problema, el entendimiento de la herramienta de forma interna ha resultado algo más difícil, aunque finalmente se han encontrado los métodos encargados de comprobar la localización y modificarlos para lo que se deseaba.

Otro de los problemas que más tiempo ha llevado solucionar ha sido la **localización del puzle en la escena del mapa** para que la guía funcione correctamente.

Como en el caso anterior, la gestión interna que tiene MapBox de los puntos de interés es algo complicada. En cuanto estos puntos de interés, MapBox **solo los posiciona en la escena si su localización se encuentra dentro de los límites del mapa en la escena**, por lo que encontrar el puzle en la escena es imposible si la ubicación de este está fuera de dichos límites.

Para resolver esto, se ha debido de gestionar en el propio proyecto desarrollado **la ubicación exacta de los puzles y las coordenadas que tendría dicho puzle en la escena dada dicha ubicación**, para así hacer que la guía muestre de forma correcta la indicación de dirección o, en su caso, muestre el puzle si se está dentro del radio de activación.

Un problema actual sin resolver es el **guardado del progreso del usuario**, reiniciándose si la aplicación se cierra. Esto se debe a que el prototipo

guarda en memoria todo el progreso. Se solucionará en un futuro trabajo **creando una base de datos y la aplicación cliente servidor.**

Uno de los mayores obstáculos a la hora de realizar este proyecto ha sido el desarrollo del **apartado de [proyección empresarial](#)**, dado que el alumno **no ha cursado la asignatura de “Creación de empresas TIC”** en la cual se explican conceptos clave para un buen desarrollo empresarial.

6.2 Conclusiones

Este proyecto ha supuesto un gran esfuerzo y dedicación por parte del alumno, pero se ha desarrollado con gran ilusión.

Ha sido un proceso muy completo en el que el alumno ha podido poner en práctica todo lo aprendido en el grado debido a que los videojuegos requieren de muchas áreas de la informática, como la gestión de proyectos, ingeniería de requisitos y diseño software, entre muchas otras.

Además, si bien este trabajo de fin de grado ha sido tutorizado por D. Juan Roberto Jiménez Pérez, la libertad que ha tenido el alumno para realizarlo ha sido plena, teniendo que enfrentar a condiciones que no se han dado en el paso por este grado hasta ahora. Por ejemplo, la investigación desde cero de herramientas nuevas como MapBox o el simple hecho de no tener guiones como en las prácticas de las asignaturas que ayuden al alumno a desarrollar este proyecto.

El desarrollo de esta propuesta ha resultado en un gran enriquecimiento para el alumno poniendo a prueba, ya no solo el conocimiento adquirido durante estos años, sino también la capacidad de organización y autodisciplina que supone emprender un proyecto de mayores dimensiones.

Sin duda, lo que más se agradece de todo este proceso ha sido el haberlo disfrutado tanto, trabajando con una idea propia que se espera poder seguir desarrollando en un futuro para dar un servicio que sin duda puede llegar a ser muy necesario en estos momentos.

Se puede concluir con esto que no hacen falta recursos descomunales para hacer ese proyecto soñado por uno mismo y que, con dedicación y

esfuerzo, todo se puede llevar a buen puerto consiguiendo todo lo que uno se proponga.

6.3 Trabajo futuro

En este apartado final, se comentarán algunas de las cuestiones que podrían añadirse a este proyecto para completar dicho prototipo y llevarlo a un ámbito más comercial.

A continuación, se expone un listado de tareas y funcionalidades que se podrían realizar para la completitud del mismo:

- Creación de una base de datos para la gestión de los datos de la aplicación como usuarios, puzles, cupones, empresas, etc.
- Implantación de un sistema cliente-servidor.
- Desarrollo para otras plataformas móviles (IOS, Windows Phone, etc.)
- Mejora de los gráficos generales de la aplicación con la contratación de artistas gráficos.
- Composición e inclusión de música en el videojuego.
- Creación de un login a la aplicación.
- Inclusión de nueva funcionalidad para la gestión de usuarios (dar de alta usuarios, cambiar datos personales, etc.).
- Creación de una aplicación secundaria externa para la activación de cupones por parte del personal encargado de las empresas publicitadas en la aplicación.
- Inclusión de nuevos puzles y rutas para más localidades.
- Inclusión de rutas que abarcan más de una localidad, cambiando con ello el tipo de guía en este caso.
- Inclusión de filtros en la elección de ruta para lo comentado en el anterior punto.
- Parametrización de los elementos de la escena del puzle con un fichero de configuración
- Inclusión de puntos de interés que muestren en el mapa las empresas publicitadas si están cercanas y los cupones disponibles de dichas empresas.

- Creación de la página web de la empresa de este proyecto. El botón “Más Info” al resolver un puzzle, llevaría una página dentro de este dominio para comprobar toda la información sobre el lugar turístico en cuestión.
- Creación de un sistema de micropagos para la obtención de nuevas rutas o rutas “especiales”.
- Nueva opción en la ventana de menú que permita volver a resolver aquellos puzzles que ya han sido resueltos por el jugador. En este caso los puzzles no cambiarían la puntuación del usuario y todas las pistas estarían desbloqueadas

Estas son sólo algunas de las ideas que se pueden llevar a cabo para la expansión de esta aplicación, pero, sin duda, se podrían añadir muchas más.

Bibliografía

- [1] «Aportación del turismo a la economía española,» [En línea]. Available:
https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=estadistica_C&cid=1254736169169&menu=ultiDatos&idp=1254735576863. [Último acceso: 2021].

- [2] «La aportación del turismo al PIB español se desploma,» [En línea]. Available:
<https://www.europapress.es/turismo/nacional/noticia-aportacion-turismo-pib-espanol-desploma-124-55-2020-avanza-estudio-20210223144922.html>. [Último acceso: 2021].

- [3] «La industria del videojuego en España Anuario 2020,» [En línea]. Available: http://www.aevi.org.es/web/wp-content/uploads/2021/04/AEVI_Anuario_2020.pdf. [Último acceso: 2020].

- [4] «MapBox,» [En línea]. Available: <https://www.mapbox.com/>. [Último acceso: 2021].

- [5] «PEGI INFO,» [En línea]. Available: <https://pegi.info/es/node/19>. [Último acceso: 2021].

- [6] «PEGI ¿Qué significan las etiquetas?,» [En línea]. Available: <https://pegi.info/es/que-significan-las-etiquetas>. [Último acceso: 2021].

- [7] «PokémonGO,» [En línea]. Available: <https://pokemongolive.com/es/>. [Último acceso: 2021].

- [8] «Pokémon GO alcanza los 10 millones de descargas en Play Store,» [En línea]. Available: <https://portalhoy.com/pokemon-go->

alcanza-los-10-millones-de-descargas-en-play-store/. [Último acceso: 2021].

- [9] «Harry Potter Wizards Unite,» [En línea]. Available: <https://www.harrypotterwizardsunite.com/es/>. [Último acceso: 2021].
- [10] «Review in Progress: Harry Potter: Wizards Unite,» [En línea]. Available: <https://www.destructoid.com/review-in-progress-harry-potter-wizards-unite/>. [Último acceso: 2021].
- [11] «Portal para la serie de Layton,» [En línea]. Available: <https://www.nintendo.es/Juegos/Portal-de-Nintendo/Portal-para-la-serie-Layton/Portal-de-la-serie-Layton-627605.html>. [Último acceso: 2021].
- [12] «Imágenes de El Profesor Layton y la Villa Misteriosa,» [En línea]. Available: <https://www.nosolobits.com/es/juego/318/el-profesor-layton-y-la-villa-misteriosa/imagenes/ds>.
- [13] «Unity.com,» [En línea]. Available: <https://unity.com/es>. [Último acceso: 2021].
- [14] «UnrealEngine.com,» [En línea]. Available: <https://www.unrealengine.com/en-US/>. [Último acceso: 2021].
- [15] «SourceEngine2,» [En línea]. Available: https://developer.valvesoftware.com/wiki/Source_2. [Último acceso: 2021].
- [16] «CryEngine,» [En línea]. Available: <https://www.cryengine.com/>. [Último acceso: 2021].
- [17] «Google Maps Platform,» [En línea]. Available: <https://developers.google.com/maps?hl=es>. [Último acceso: 2021].
- [18] «gimp.org,» [En línea]. Available: <http://www.gimp.org.es/>. [Último acceso: 2021].

- [19] «inkscape.org,» [En línea]. Available: <https://inkscape.org/es/>. [Último acceso: 2021].
- [20] «xmind.net,» [En línea]. Available: <https://www.xmind.net/desktop/>. [Último acceso: 2021].
- [21] «VisualParadigm.com,» [En línea]. Available: <https://www.visual-paradigm.com/>. [Último acceso: 2021].
- [22] «blender.org,» [En línea]. Available: <https://www.blender.org/>. [Último acceso: 2021].
- [23] «JustinMind.com,» [En línea]. Available: https://www.justinmind.com/?k=justinmind&a=252192201828&adg=52001997837&cmp=1063145459&match=e&adposition=&utm_medium=cpc&utm_source=google&utm_campaign=1063145459&utm_term=justinmind_e&gclid=CjwKCAjw8uGBhBAEiwAayu_9RDK60z_VwyJxBdYPBcWXY7k4klejg_PCA8oG4. [Último acceso: 2021].
- [24] G. B. y. J. R. Ivar Jacobson, El proceso unificado de desarrollo de software, Madrid: Assison Wesley, 2000.
- [25] «Método MoSCoW,» [En línea]. Available: <https://www.ticportal.es/glosario-tic/metodo-moscow>. [Último acceso: 2021].
- [26] «OuterVision® Power Supply Calculator,» [En línea]. Available: <https://outervision.com/power-supply-calculator>. [Último acceso: 2021].
- [27] «Fibra Jazztel 100Mb,» [En línea]. Available: https://www.mijazztel.com/internet_fibra/fibra_50?ta=true&tsource=jazgm_exacta_005_fibra_n2_ex_PC_55084307221&distribuidor=google&gclid=CjwKCAjwwqaGBhBKEiwAMk-FtF2zYx2Gc2jcFGWdkxpnPTjfXnDXJMz0Uor7Y4Mv5xYcrsvnCtqn2BoCeCsQAvD_BwE. [Último acceso: 2021].

- [28] «InfoJobs.net,» [En línea]. Available: <https://www.infojobs.net/>. [Último acceso: 2021].
- [29] «Salarios Indeed.com,» [En línea]. Available: <https://es.indeed.com/career/salaries>. [Último acceso: 2021].
- [30] «Salarios Jobted.es,» [En línea]. Available: <https://www.jobted.es/salario>. [Último acceso: 2021].
- [31] «MapBox Studio,» [En línea]. Available: <https://studio.mapbox.com/styles/amg00156/ckkwkx23d0seb17mpzl2dks9e/edit/#15.8/37.758549/-3.792722>. [Último acceso: 2021].
- [32] «Groupon.es,» [En línea]. Available: <https://www.groupon.es/>. [Último acceso: 2021].
- [33] «Gráfica de dificultad en la movilidad según sexo y grupo de edad. Población de 15 y más años,» [En línea]. Available: <https://www.ine.es/jaxi/Datos.htm?path=/t15/p419/a2017/p01/I0/&file=09001.px#!tabs-grafico>. [Último acceso: 2021].
- [34] « SEO: la guía completa para que conquistes la cima de Google en el 2021,» [En línea]. Available: <https://rockcontent.com/es/blog/que-es-seo/>. [Último acceso: 2021].
- [35] «AvidProTools,» [En línea]. Available: <https://www.avid.com/es/pro-tools>. [Último acceso: 2021].
- [36] «Cuánto cobra un productor musical: sueldos de la industria de la música,» [En línea]. Available: <https://www.campustraining.es/noticias/cuanto-cobra-productor-musical-sueldos-industria-musica/>. [Último acceso: 2021].
- [37] «Hostalia.com,» [En línea]. Available: <https://www.hostalia.com/hosting/#planes>. [Último acceso: 2021].

- [38] «UnityStore,» [En línea]. Available:
https://store.unity.com/es/products/unity-plus?_gl=1*r37ios*_gcl_aw*R0NMLjE2MjQxMjc1NjAuQ2p3S0NBa ndxN2FHQmhBREVPd0E2dUdacDRncjBNUmZIME5BRktNd2h6Tk VPeEdJWklNaHotVHRRLXBZbEN5c1BUX3laY1hVRVVQakxSb0N HeU1RQXZEX0J3RQ..*_ga*MTA2MjY4ODY2MC4xNjE1MTk2OD Mz*_ga_1S78E. [Último acceso: 2021].

Anexos

ANEXO 1: DIAGRAMAS DE DISEÑO FINALES

En este anexo, se verán el **diagrama de casos de uso**, los **diagramas de secuencia** y el **diagrama final de diseño**. Dado que los diagramas no se ven lo suficientemente grandes como para distinguir ciertos elementos, se incluye un enlace a cada imagen.

DIAGRAMA DE CASOS DE USO

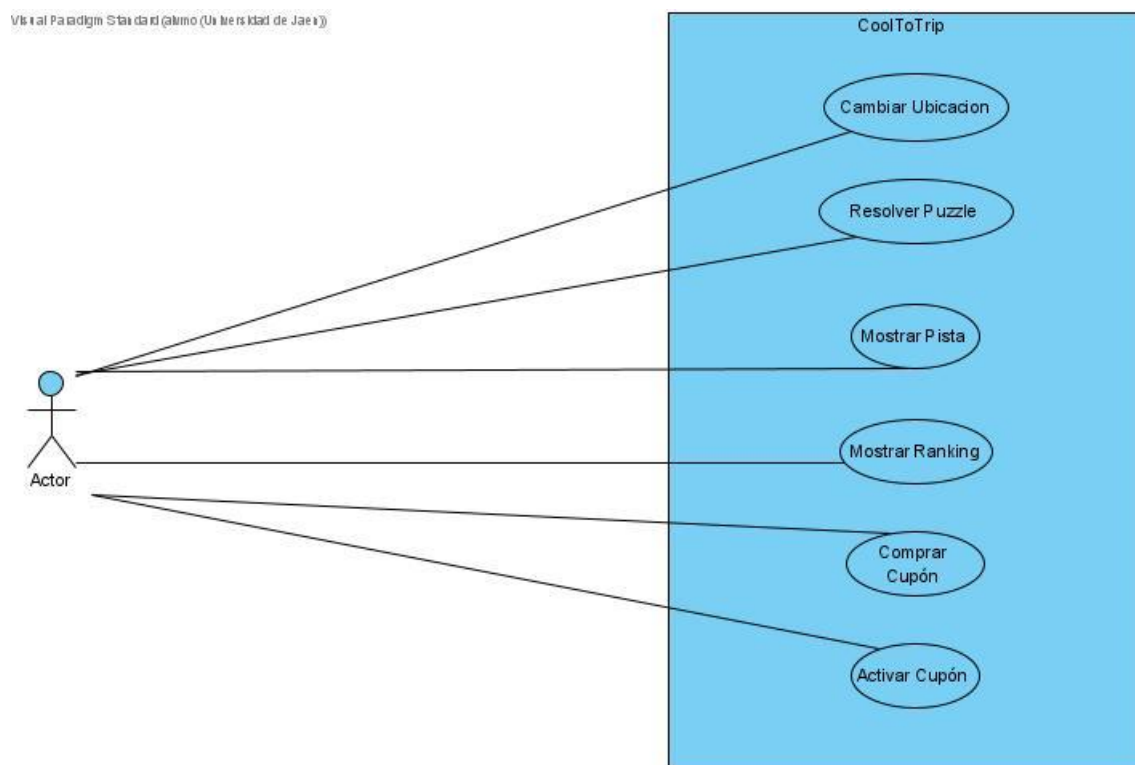


Ilustración 61: Diagrama de casos de uso

Enlace a la imagen: [Imágenes\Diagrama de casos de uso.jpg](#)

DIAGRAMAS DE SECUENCIA

CASO DE USO: Cambiar de ubicación

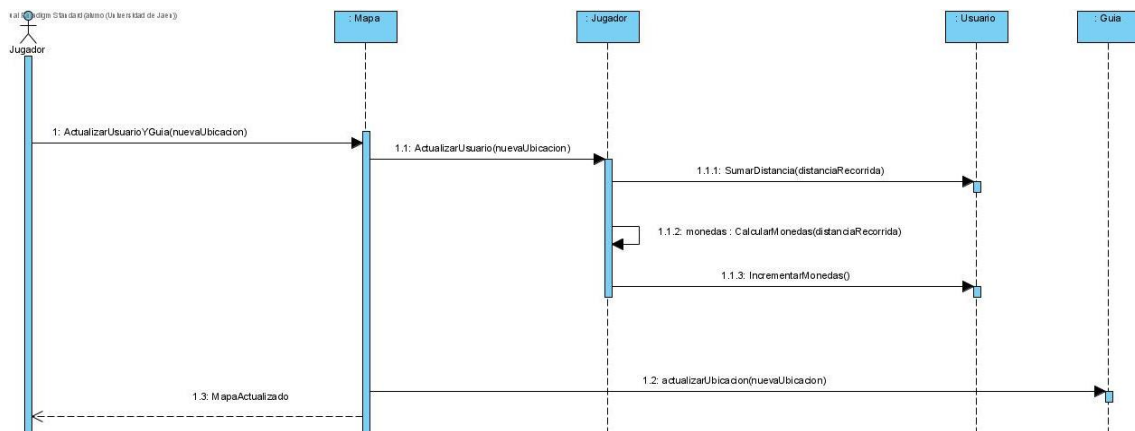


Ilustración 62: Diagrama de secuencia del caso de uso "Cambiar de ubicación"

Enlace a la imagen: Imágenes\Diagrama de secuencia_Cambiar de ubicacion.jpg

CASO DE USO: Resolver puzle

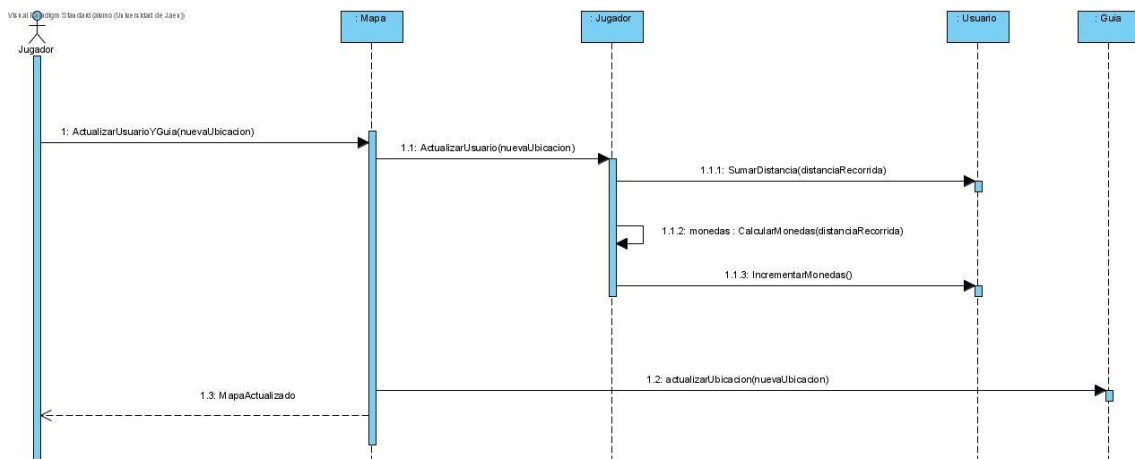


Ilustración 63: Diagrama de secuencia del caso de uso "Resolver puzle"

Enlace a la imagen: Imágenes\Diagrama de secuencia_Resolver puzle.jpg

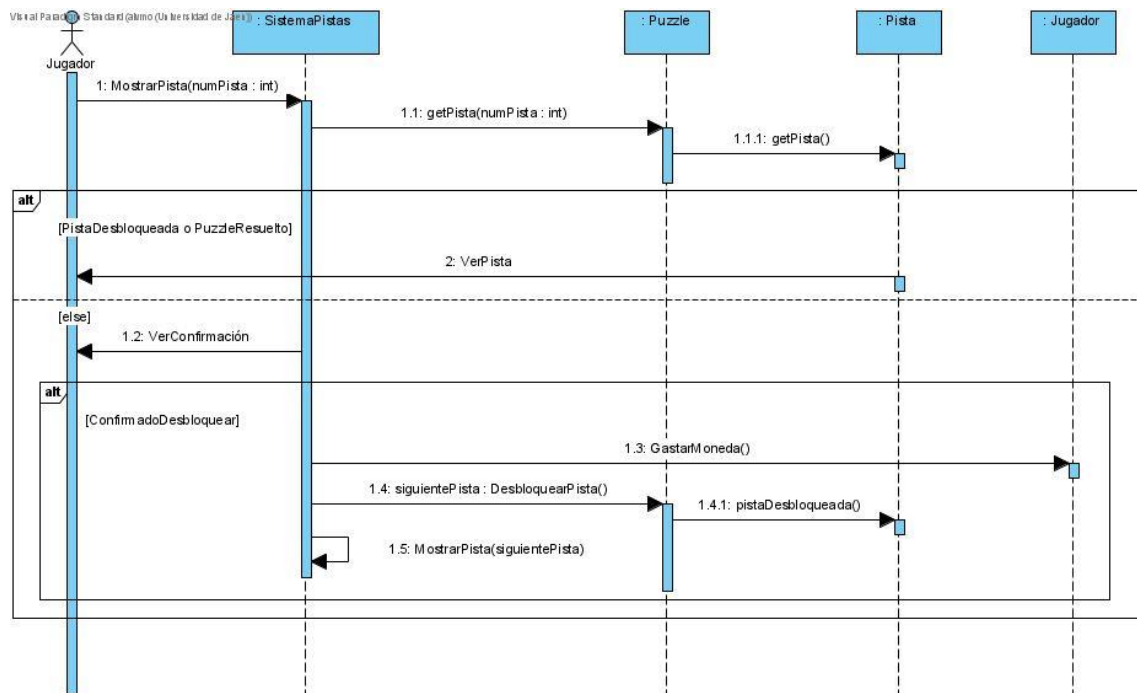
CASO DE USO: Mostrar pista

Ilustración 64: Diagrama de secuencia del caso de uso "Mostrar pista"

Enlace a la imagen: [Imágenes\Diagrama de secuencia_Mostrar pista.jpg](#)

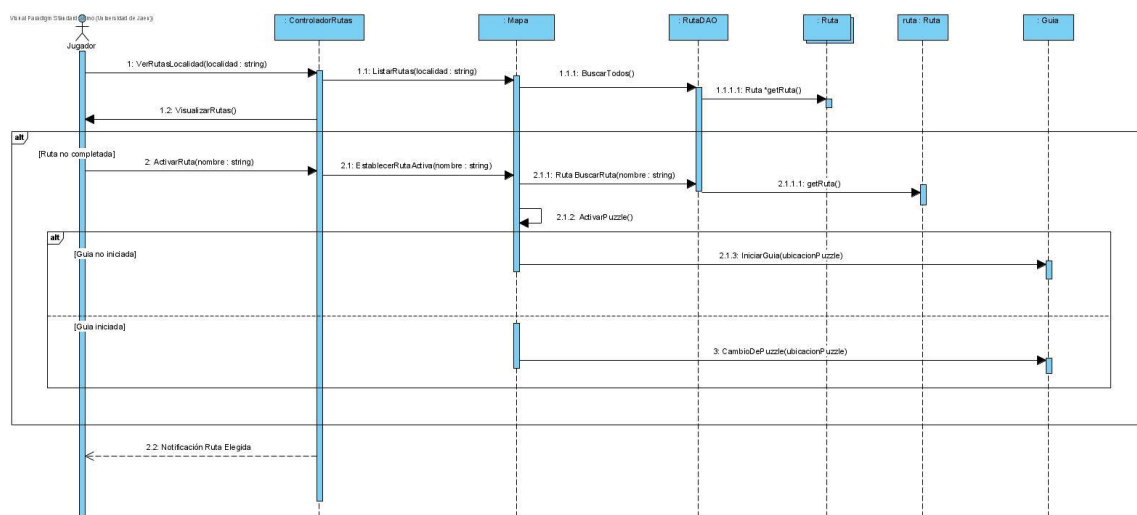
CASO DE USO: Seleccionar ruta

Ilustración 65: Diagrama de secuencia del caso de uso "Seleccionar ruta"

Enlace a la imagen: [Imágenes\Diagrama de secuencia_Seleccionar ruta.jpg](#)

CASO DE USO: Mostrar ranking

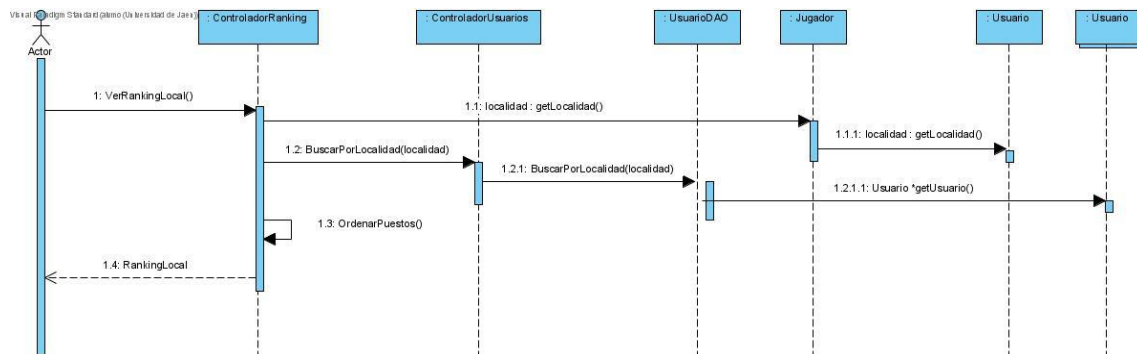


Ilustración 66: Diagrama de secuencia del caso de uso "Mostrar ranking"

Enlace a la imagen: [Imágenes\Diagrama de secuencia_Mostrar ranking.jpg](#)

CASO DE USO: Comprar cupón

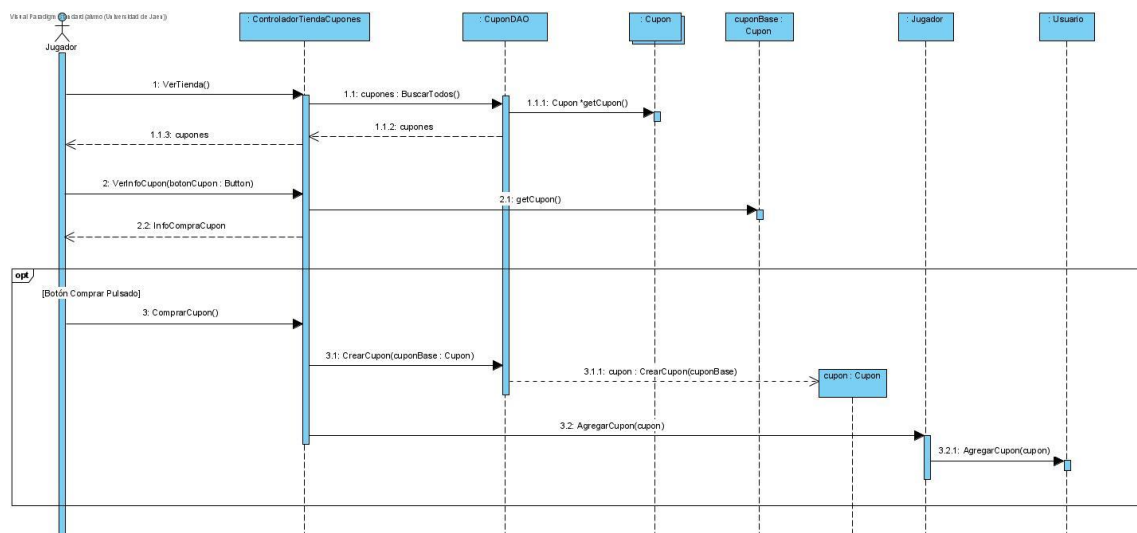


Ilustración 67: Diagrama de secuencia del caso de uso "Comprar cupón"

Enlace a la imagen: [Imágenes\Diagrama de secuencia_Comprar cupon.jpg](#)

CASO DE USO: Activar cupón

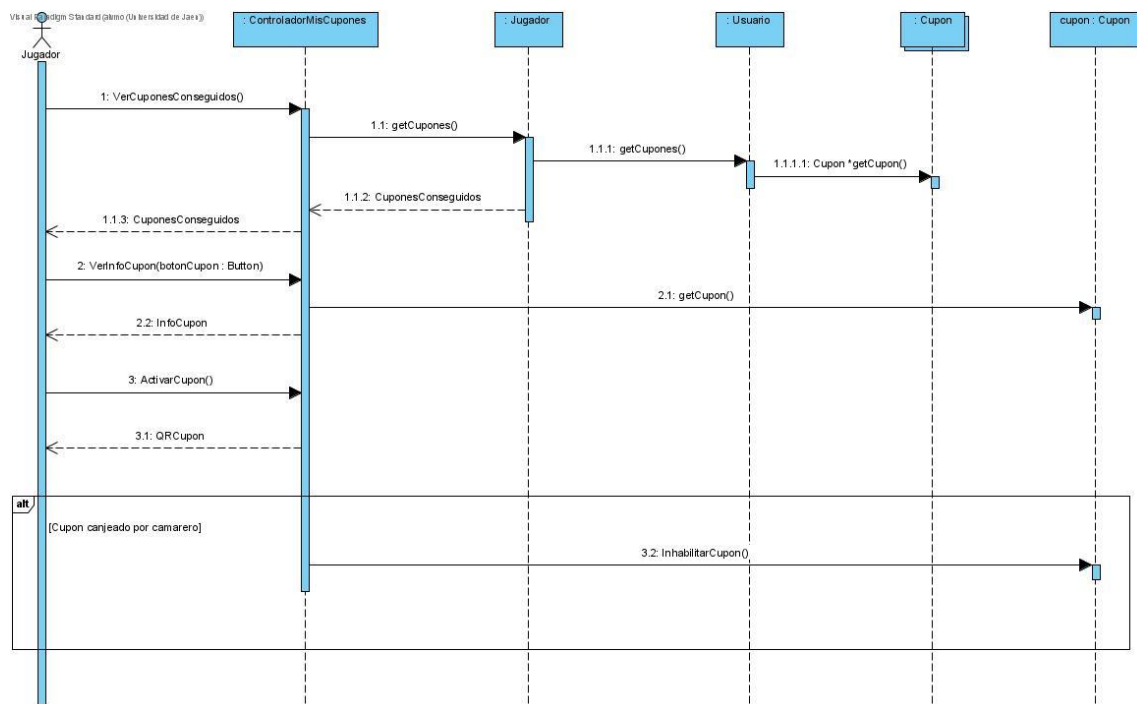


Ilustración 68: Diagrama de secuencia del caso de uso "Activar cupón"

Enlace a la imagen: Imágenes\Diagrama de secuencia_Activar cupon.jpg

Visual Paradigm Standard Edition (Universidad de Jaén)



138

ANEXO 2: RESOLUCIÓN DE LOS PUZZLES

En este anexo, se explicará la forma en que se han de resolver los puzzles incluidos en este prototipo.

EL CHICO Y EL ESPECTRO

Al empezar este puzzle se nos muestra lo siguiente:

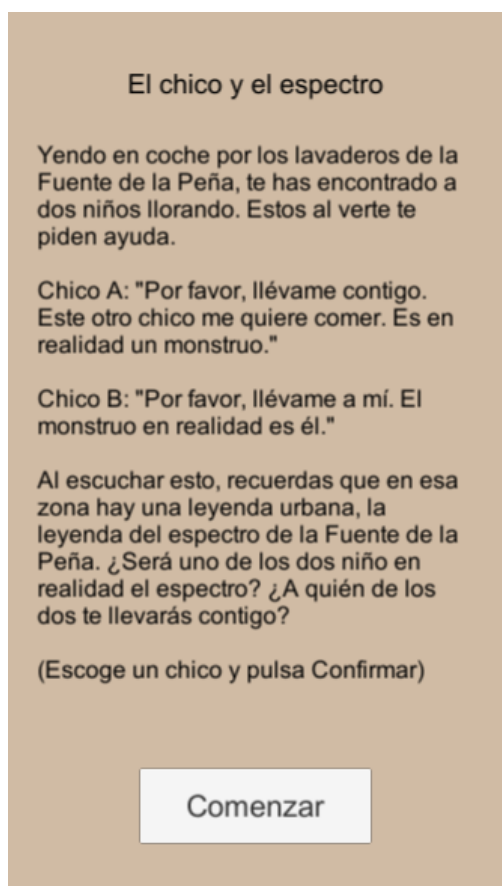


Ilustración 70: Enunciado del puzzle "El chico y el espectro"



Ilustración 71: Elementos del puzzle "El chico y el espectro"

La dificultad de este puzzle radica en que, para resolverlo, hace falta saber qué se dice en esta leyenda. Para ello, podemos buscar por internet la leyenda y leerla.

A continuación, se redactará un resumen de la misma:

Un arriero se dirigía en su carro hacia la capital desde Los Villares. Al pasar por los lavaderos de la Fuente de la Peña, escuchó los llantos de un chico.

El arriero pensó que podría ser el hijo perdido de alguna de las mujeres que lavaban allí sus ropas, así que lo tranquilizó y lo montó en su carro.

Entrando por el barrio de San Felipe, el arriero notó como sus mulas se cansaban cada vez más, como si cargaran con un peso enorme.

Al girar su cabeza hacia atrás, vio que en lugar del chico que había recogido en su carro había montado un monstruo gigantesco, con un terrible rostro y unos dientes enormes.

El monstruo con una diabólica sonrisa le preguntó:

- “¿Tienes dientes como yo?”

Tras esto, el arriero salió corriendo, dejando sus mulas y su carro atrás.

Conociendo ya la leyenda, debe de destacarse el dato crucial para distinguir al espectro del chico que no lo es, **sus enormes dientes**.

Sabiendo esto, podemos caer en la trampa de elegir al chico de la derecha creyendo que el espectro es el de la izquierda porque se le ve un diente. La respuesta correcta es justo lo contrario, el espectro tiene dientes **muy grandes**, los cuales está escondiendo el chico de la derecha, por lo que el chico que nos debemos llevar, y por lo tanto la respuesta correcta, es **el chico de la izquierda**.

UNA DECLARACIÓN DE AMOR ESCALOFRIANTE

Al empezar este puzle se nos muestra lo siguiente:

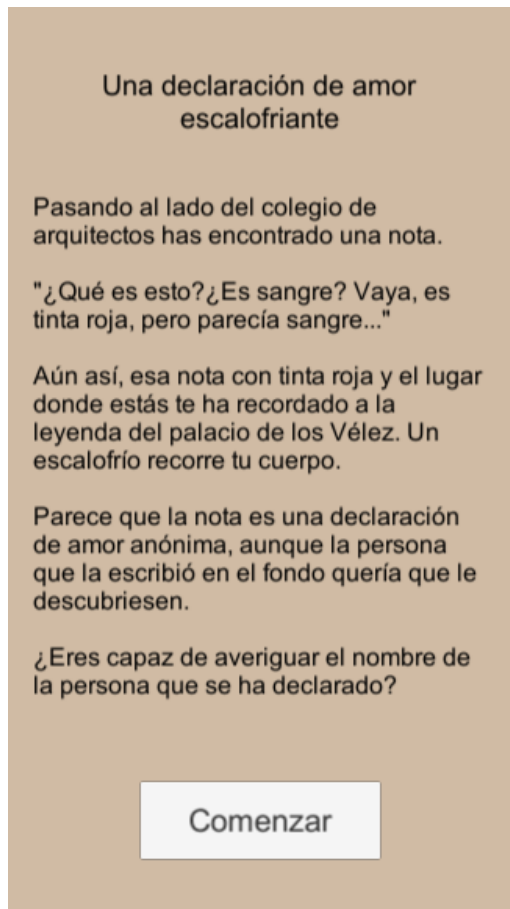


Ilustración 72: Enunciado del puzle "Una declaración de amor escalofriante"

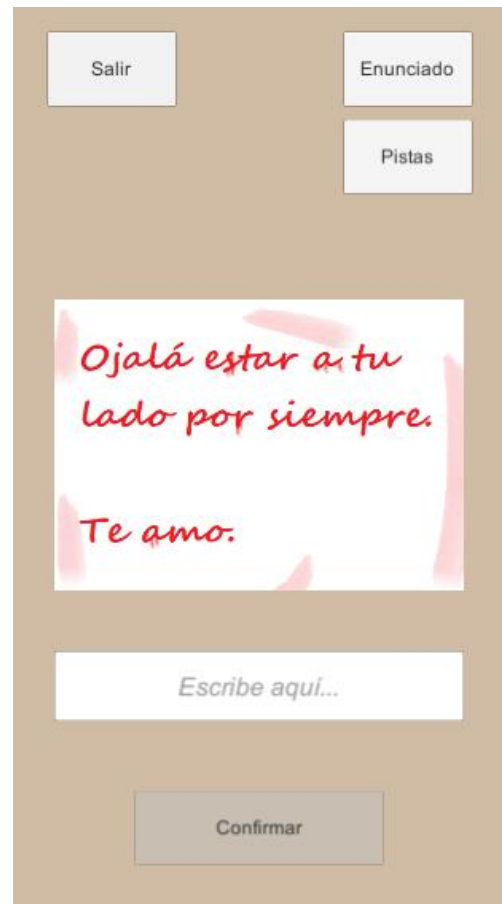


Ilustración 73: Elementos del puzle "Una declaración de amor escalofriante"

La respuesta a este puzle se encuentra en las letras que tienen un poco de tinta corrida [Ilustración 74] siendo estas las siguientes:

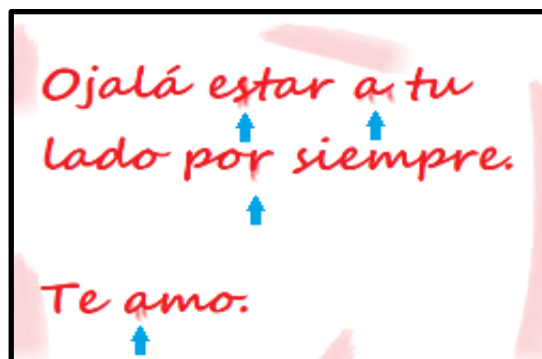


Ilustración 74: Solución del puzle "Una nota de amor escalofriante"

Por lo tanto, la respuesta a este puzle es: **SARA**

UN LAGARTO EXPLOSIVO

Al empezar este puzle se nos muestra lo siguiente:

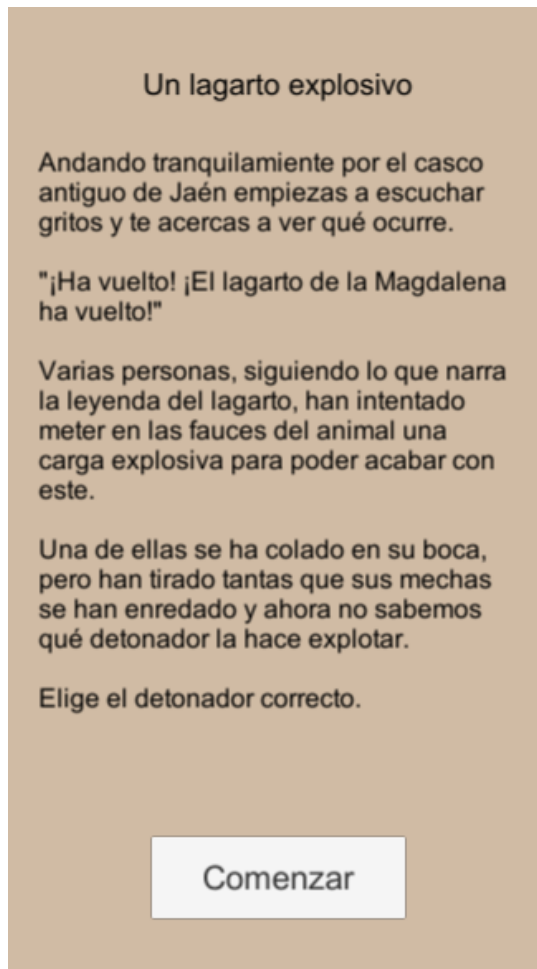


Ilustración 75: Enunciado del puzle "Un lagarto explosivo"

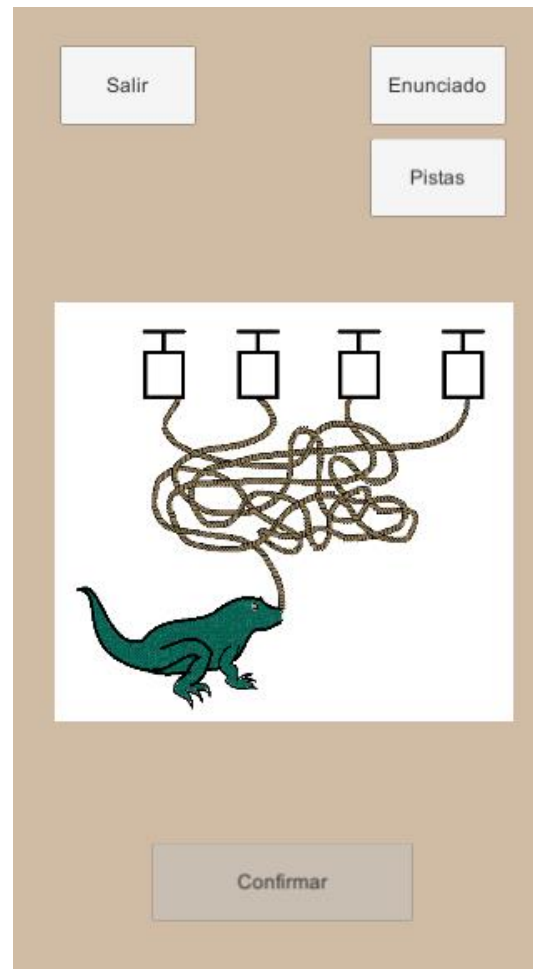


Ilustración 76: Elementos del puzle "Un lagarto explosivo"

Para resolverlo, solo tenemos que seguir la mecha que sale de la boca del lagarto hasta dar con el detonador correcto [Ilustración 77], siendo este el marcado a continuación:

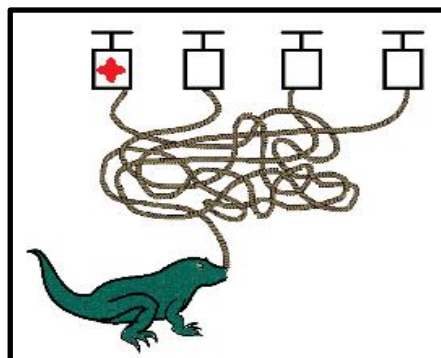


Ilustración 77: Resolución del puzle "Un lagarto explosivo"

ANEXO 3: MANUAL DE INSTALACIÓN

1. Conecte su dispositivo móvil al ordenador y meta el archivo CoolToTrip.apk en una carpeta de su dispositivo.
2. En su dispositivo, busque el archivo anterior con la aplicación de gestión de archivos del dispositivo [Ilustración 78] y presiónelo para instalarlo [Ilustración 79].

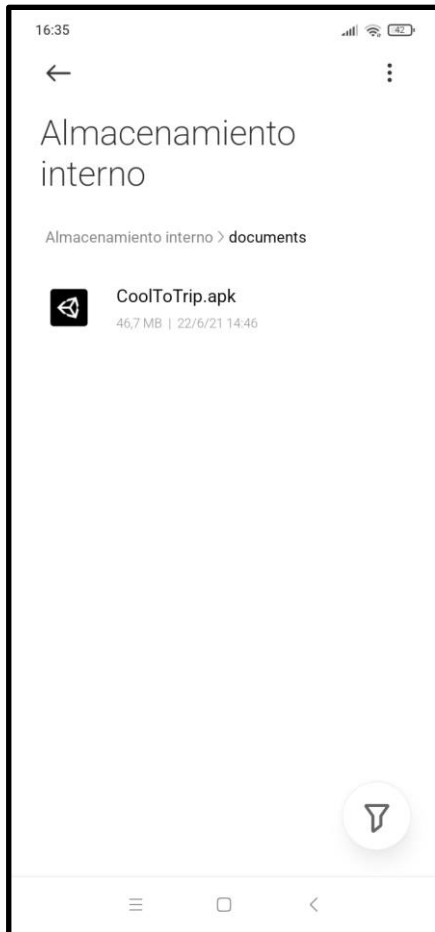


Ilustración 78: Gestor de archivos del dispositivo

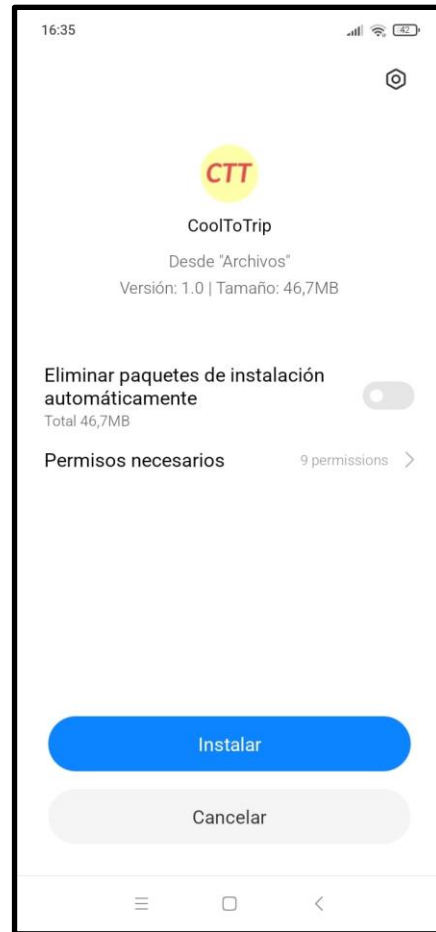


Ilustración 79: Instalación de la aplicación

3. Si no permite la instalación, deberá cambiar sus ajustes para poder instalar aplicaciones de terceros en su dispositivo.

4. Una vez instalado, dé los permisos necesarios a la aplicación para su localización en el mapa [Ilustración 80].

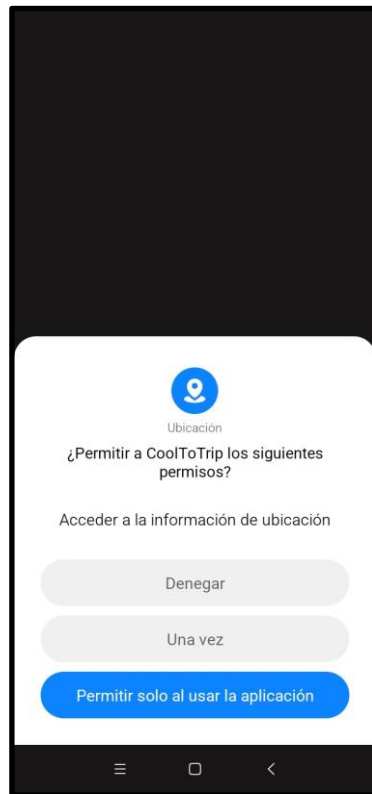


Ilustración 80: Permitir permisos

5. Es posible que tenga que reiniciar la aplicación tras dar los permisos de ubicación si no se muestra nada en pantalla.