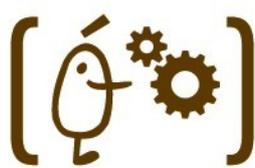


VNIVERSITAT  
E VALÈNCIA [  ]

GRADO EN INGENIERÍA  
MULTIMEDIA



VNIVERSITAT  
E VALÈNCIA

TRABAJO FIN DE GRADO

---

APLICACIÓN WEB PARA LA AYUDA Y  
CONCIENCIACIÓN DE LA IGUALDAD DE GÉNERO  
Y REDUCCIÓN DE DESIGUALDADES

---

**AUTOR: PEDRO ANTONIO RAMOS GUARDIOLA**

**TUTORA: INMACULADA COMA TATAY**

**SEPTIEMBRE 2022**





VNIVERSITAT  
DE VALÈNCIA



Escola Tècnica Superior  
d'Enginyeria **ETSE-UV**

## TRABAJO FIN DE GRADO

---

---

# APLICACIÓN WEB PARA LA AYUDA Y CONCIENCIACIÓN DE LA IGUALDAD DE GÉNERO Y REDUCCIÓN DE DESIGUALDADES

---

---

**AUTOR: PEDRO ANTONIO RAMOS GUARDIOLA**

**TUTORA: INMACULADA COMA TATAY**

---

### TRIBUNAL

PRESIDENTE/A:

VOCAL 1:

VOCAL 2:

FECHA DE DEFENSA:

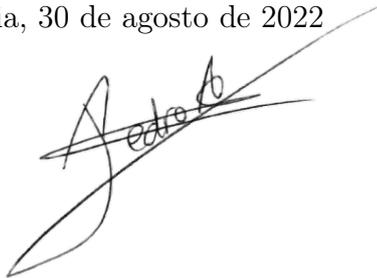
CALIFICACIÓN:



### **Declaración de autoría:**

Yo, Pedro Antonio Ramos Guardiola, declaro la autoría del Trabajo Fin de Grado titulado “Aplicación web para la ayuda y concienciación de la igualdad de género y reducción de desigualdades” y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual. El material no original que figura en este trabajo ha sido atribuido a sus legítimos autores.

Valencia, 30 de agosto de 2022

A handwritten signature in black ink, appearing to read 'Pedro A. Ramos Guardiola', written over a large, sweeping, stylized flourish that extends from the left side of the page.

Fdo: Pedro Antonio Ramos Guardiola



---

**Resumen:**

Este trabajo de fin de grado trata del desarrollo e implementación de una aplicación web y un videojuego para la ayuda y concienciación de la igualdad de género y reducción de desigualdades, que son dos de los objetivos de los Objetivos de Desarrollo Sostenible de la Agenda de 2030 propuesta por las Naciones Unidas.

El objetivo principal de este proyecto es recoger y brindar información sobre recursos disponibles dentro de la ciudad de Valencia, como al mismo tiempo concienciar a los más pequeños con un videojuego que se desarrolla dentro de la ciudad de Valencia.

La elaboración de este proyecto a mostrado la falta de acceso a información de recursos para estos temas y por lo tanto se creado la necesidad de acercar estos al público.

**Palabras clave:** ODS, aplicación web, videojuego, igualdad de género, reducción de desigualdades, valencia, recursos, gymkana

---



---

**Abstract:**

This final degree work deals with the development and implementation of a web application and a video game for the support and awareness of gender equality and reduction of inequalities, which are two of the objectives of the Sustainable Development Goals of the 2030 Agenda proposed by the United Nations.

The main objective of this project is to collect and provide information on resources available within the city of Valencia, and at the same time raise awareness among children with a video game that takes place within the city of Valencia.

The development of this project has shown the lack of access to information on resources for these issues and therefore created the need to bring them to the public.

**Keywords:** ODS, web application, videogame, gender equality, reduction of inequalities, valencia, resources, gymkana

---



---

**Resum:**

Aquest treball de fi de grau tracta del desenvolupament i implementació d'una aplicació web i un videojoc per a l'ajuda i conscienciació de la igualtat de gènere i reducció de desigualtats, que són dues dels objectius dels Objectius de Desenvolupament Sostenible de l'Agenda de 2030 proposta per les Nacions Unides.

L'objectiu principal d'aquest projecte és recollir i brindar informació sobre recursos disponibles dins de la ciutat de València, com al mateix temps conscienciar als més xicotets amb un videojoc que es desenvolupa dins de la ciutat de València.

L'elaboració d'aquest projecte ha mostrat la falta d'accés a informació de recursos per a aquests temes i per tant es crea la necessitat d'acostar aquests al públic.

**Paraules clau:** ODS, aplicació web, videojoc, igualtat de gènere, reducció de desigualtats, valència, recursos, gimcana

---



---

**Agradecimientos:**

En primer lugar quiero agradecer a mi tutora, Inmaculada Coma, por haberme ayudado en la realización de este proyecto y la redacción de la memoria.

En segundo lugar me gustaría agradecer a los profesores que me han impartido clase, no solo durante los años de la carrera, sino a todos los que me han motivado a seguir estudiando y por enseñarme todos los conocimientos que he necesitado para realizar este proyecto.

Y en tercer lugar, quiero agradecer a mi familia por haber dedicado todo su tiempo y esfuerzo en brindarme una buena educación, y a mis amigos por ayudarme y apoyarme durante todos estos años.

---



# Índice general

<b>1. Introducción</b>	<b>25</b>
1.1. Introducción . . . . .	25
1.2. Motivación . . . . .	25
1.3. Objetivos . . . . .	25
1.4. Organización de la memoria . . . . .	26
<b>2. Estado del arte</b>	<b>29</b>
2.1. Antecedentes . . . . .	29
2.1.1. Objetivos de Desarrollo Sostenible . . . . .	29
2.1.2. Recursos existentes . . . . .	32
2.2. Tecnologías . . . . .	36
2.2.1. HTML . . . . .	36
2.2.2. CSS . . . . .	37
2.2.3. JavaScript . . . . .	37
2.2.4. Unity . . . . .	38
2.2.5. GeoJSON . . . . .	42
2.3. Elección de tecnologías . . . . .	43
<b>3. Requisitos, especificaciones, coste, riesgos, viabilidad</b>	<b>45</b>
3.1. Requisitos . . . . .	45
3.1.1. Requisitos funcionales . . . . .	45
3.1.2. Requisitos no funcionales . . . . .	46
3.2. Especificaciones . . . . .	46
3.3. Planificación y estimación de costes . . . . .	47
3.3.1. Definición de actividades . . . . .	47
3.3.2. Estimación temporal . . . . .	48
3.3.3. Análisis de costes . . . . .	50
3.4. Análisis de viabilidad y riesgos . . . . .	52
3.4.1. Viabilidad económica . . . . .	53

3.4.2.	Viabilidad legal . . . . .	53
3.4.3.	Análisis de riesgos . . . . .	53
3.4.4.	Análisis de riesgos . . . . .	53
<b>4.</b>	<b>Análisis del sistema</b>	<b>55</b>
4.1.	Casos de uso . . . . .	55
4.2.	Diagramas de actividad . . . . .	58
<b>5.</b>	<b>Diseño del sistema</b>	<b>61</b>
5.1.	Diagrama de clases . . . . .	61
5.1.1.	Identificación de atributos . . . . .	62
5.2.	Interfaces de usuario . . . . .	63
5.3.	Estilo de interfaz . . . . .	68
<b>6.</b>	<b>Implementación del sistema</b>	<b>71</b>
6.1.	Herramientas de desarrollo . . . . .	71
6.2.	Texturas . . . . .	72
6.3.	Implementación del videojuego . . . . .	74
6.3.1.	Estructura del videojuego . . . . .	75
6.3.2.	Personalización de personaje . . . . .	76
6.3.3.	Movimiento del personaje . . . . .	77
6.3.4.	Sistema de dialogo . . . . .	79
6.3.5.	Sistema de puntuación de minijuegos . . . . .	81
6.4.	Implementación de ficheros con geolocalización . . . . .	84
6.5.	Implementación de la aplicación web . . . . .	86
6.5.1.	Estructura de la aplicación web . . . . .	86
6.5.2.	Interfaz del recurso . . . . .	87
6.5.3.	Inserción dinámica de los recursos . . . . .	88
6.5.4.	Mapa interactivo . . . . .	91
6.5.5.	Sistema de búsqueda . . . . .	93
<b>7.</b>	<b>Pruebas y resultados</b>	<b>95</b>
7.1.	Pruebas funcionales . . . . .	95
7.1.1.	Pruebas funcionales de la aplicación web . . . . .	95
7.1.2.	Pruebas funcionales del videojuego . . . . .	99
7.2.	Pruebas de usabilidad . . . . .	102
7.3.	Pruebas de velocidad . . . . .	103
7.4.	Resultados . . . . .	104

<b>8. Conclusiones</b>	<b>105</b>
8.1. Revisión de costes . . . . .	105
8.2. Conclusiones . . . . .	106
8.3. Trabajo futuro . . . . .	107
<b>A. Apéndice</b>	<b>109</b>
A.1. Diagrama de Gantt . . . . .	110
A.2. Diagrama de clases . . . . .	111
<b>Bibliografía</b>	<b>111</b>



# Índice de figuras

2.1. Logo de Objetivos de Desarrollo Sostenible [1] . . . . .	29
2.2. Captura 1 Sitio Web Ayuntamiento de Valencia [2] . . . . .	32
2.3. Captura 2 Sitio Web Ayuntamiento de Valencia [2] . . . . .	33
2.4. Captura 1 Sitio Web Ayuntamiento de Madrid [3] . . . . .	33
2.5. Captura 2 Sitio Web Ayuntamiento de Madrid [3] . . . . .	34
2.6. Captura 3 Sitio Web Ayuntamiento de Madrid [3] . . . . .	34
2.7. Captura 1 Sitio Web Ayuntamiento de Barcelona [4] . . . . .	35
2.8. Captura 2 Sitio Web Ayuntamiento de Barcelona [4] . . . . .	35
2.9. Editor de Sprites en Unity [5] . . . . .	38
2.10. Declaración de componente Text en Unity [5] . . . . .	40
2.11. Declaración de componente Image en Unity [5] . . . . .	40
2.12. Componentes interactivos en Unity [6] . . . . .	41
2.13. Animator del Animation Controller en Unity [7] . . . . .	42
2.14. Logo Bootstrap [8] . . . . .	43
2.15. Logo Leaflet [9] . . . . .	43
3.1. Diagrama de Gantt . . . . .	50
4.1. Diagrama de Casos de Uso . . . . .	55
4.2. Diagrama de Actividad de Página de Recursos . . . . .	59
4.3. Diagrama de Actividad de Información de Recurso . . . . .	59
4.4. Diagrama de Actividad del Videojuego . . . . .	60
5.1. Diagrama de clases . . . . .	62
5.2. Boceto interfaz de barra de navegación . . . . .	63
5.3. Boceto interfaz de página de inicio . . . . .	64
5.4. Boceto interfaz de página sobre mi . . . . .	65
5.5. Boceto interfaz de página juego . . . . .	65
5.6. Boceto interfaz de página de recursos . . . . .	66
5.7. Boceto interfaz de inicio videojuego . . . . .	67
5.8. Boceto interfaz de personalización de personaje videojuego . . . . .	67

5.9.	Boceto interfaz de minijuego general videojuego . . . . .	68
5.10.	Paleta de colores . . . . .	69
5.11.	Fuente 'PT Sans' . . . . .	69
5.12.	Fuente 'Minecraftia' . . . . .	69
6.1.	Texturas de Ayuntamiento, Palacio de las Artes y Parque Gulliver pixelados	72
6.2.	Sprites de Piel del Personaje . . . . .	73
6.3.	Sprites de Camiseta del Personaje . . . . .	73
6.4.	Sprites de Peinados del Personaje . . . . .	73
6.5.	Sprites para minijuego de memoria de cartas . . . . .	74
6.6.	Estructura del videojuego . . . . .	75
6.7.	Estructura de la aplicación web . . . . .	86
7.1.	Prueba funcional de barra de navegación . . . . .	96
7.2.	Prueba funcional de la página de inicio . . . . .	96
7.3.	Prueba funcional de la página de juego . . . . .	97
7.4.	Prueba funcional de la página sobre el autor . . . . .	97
7.5.	Prueba funcional de las páginas de recursos . . . . .	98
7.6.	Prueba funcional de las páginas de recursos al localizar un recurso en el mapa . . . . .	98
7.7.	Prueba funcional de las páginas de recursos al utilizar buscador por c.p. . . . .	99
7.8.	Prueba funcional de las pantallas de inicio y personalización del personaje	100
7.9.	Prueba funcional de la pantalla de información del mapa . . . . .	100
7.10.	Prueba funcional de las pantallas de mapa y minijuego del nivel 1 . . . . .	100
7.11.	Prueba funcional de las pantallas de mapa y minijuego del nivel 2 . . . . .	100
7.12.	Prueba funcional de las pantallas de mapa y minijuego del nivel 3 . . . . .	101
7.13.	Prueba funcional del sistema de dialogo . . . . .	101
7.14.	Prueba funcional de la pantalla de puntuación de minijuego . . . . .	101
7.15.	Prueba funcional de la pantalla final del videojuego . . . . .	101
7.16.	Prueba de velocidad de la página del videojuego . . . . .	103
7.17.	Prueba de velocidad de las páginas de recursos . . . . .	104

# Índice de tablas

3.1. Funcionalidades de la aplicación web . . . . .	46
3.2. Funcionalidades del videojuego . . . . .	47
3.3. Estimación temporal por días del Estado del Arte . . . . .	48
3.4. Estimación temporal por días del análisis de especificación, planificación, estimaciones y viabilidad . . . . .	48
3.5. Estimación temporal por días del análisis del proyecto . . . . .	49
3.6. Estimación temporal por días del diseño del proyecto . . . . .	49
3.7. Estimación temporal por días de la implementación del proyecto . . . . .	49
3.8. Estimación temporal general del proyecto . . . . .	49
3.9. Costes directos e indirectos anuales . . . . .	51
3.10. Costes personales . . . . .	51
3.11. Costes de software . . . . .	52
4.1. Especificación Caso de Uso “Ver Información Recurso” . . . . .	56
4.2. Especificación Caso de Uso “Localizar Recurso” . . . . .	56
4.3. Especificación Caso de Uso “Filtrar Recursos por C.P.” . . . . .	57
4.4. Especificación Caso de Uso “Videojuego” . . . . .	58
5.1. Identificación de atributos . . . . .	63
6.1. Ficheros del videojuego . . . . .	75
6.2. Recursos para la aplicación web . . . . .	85
6.3. Ficheros de la aplicación web . . . . .	86
8.1. Tiempo empleado del proyecto . . . . .	105
8.2. Costes finales de software . . . . .	106



# Índice de códigos

2.1. Código de ejemplo charset . . . . .	36
2.2. Código de ejemplo enlace HTML . . . . .	37
2.3. Código de ejemplo CSS . . . . .	37
2.4. Código de ejemplo JavaScript . . . . .	38
2.5. Ejemplo GeoJSON para un punto . . . . .	42
2.6. Ejemplo GeoJSON para un polígono . . . . .	42
6.1. Código para la personalización del personaje . . . . .	76
6.2. Código del movimiento del personaje . . . . .	77
6.3. Código de la animación de sprites del personaje . . . . .	78
6.4. Código del disparador del dialogo . . . . .	79
6.5. Código del sistema de dialogo . . . . .	80
6.6. Código para puntuación de los minijuegos 1 y 3 . . . . .	82
6.7. Código para puntuación del minijuego 2 . . . . .	82
6.8. Código de cronometro acabado . . . . .	83
6.9. Código de Sistema de puntuación . . . . .	83
6.10. Plantilla para ficheros GeoJSON . . . . .	84
6.11. Código de interfaz de recurso . . . . .	87
6.12. Código de funcionalidad del recurso . . . . .	87
6.13. Código para cargar los ficheros en aplicación web . . . . .	88
6.14. Código para mostrar los recursos . . . . .	88
6.15. Código para inicializar mapa interactivo . . . . .	91
6.16. Código para añadir marcador . . . . .	91
6.17. Código para mostrar marcador individual . . . . .	92
6.18. Código del sistema de búsqueda . . . . .	93
6.19. Código del apoyo al sistema de búsqueda . . . . .	93
6.20. Código de la interfaz del sistema de búsqueda . . . . .	93



# Capítulo 1

## Introducción

### 1.1. Introducción

Este trabajo de fin de grado trata del desarrollo e implementación de una aplicación web y un videojuego para la ayuda y concienciación de la igualdad de género y reducción de desigualdades, que son dos de los objetivos de los Objetivos de Desarrollo Sostenible de la Agenda de 2030 propuesta por las Naciones Unidas.

El desarrollo de este proyecto se basará en una aplicación web que recoja, muestre y localice aquellos recursos dentro de la ciudad de Valencia que estén relacionados con ayudas para la mujer, para personas con discapacidad, sin techo, minorías étnicas o migrantes y personas del colectivo LGBTI. Además, se complementará con el desarrollo de un videojuego en dos dimensiones que ayudará a la concienciación de estos temas.

### 1.2. Motivación

Se pretende crear una aplicación web que muestre información sobre los recursos disponibles en la ciudad de Valencia relativos a temas de igualdad de género y reducción de desigualdades, y que incluya tanto información de organismos oficiales como otro tipo de asociaciones u entidades. Al mismo tiempo, se pretende concienciar a los niños de dichos temas a través de un juego indexado en la página web, para ayudarles a entender mejor estos temas.

El principal motivo para la creación de esta aplicación web, es la carencia de aplicaciones similares, ya que no existen muchas páginas con un acceso fácil a la información y recursos disponible de la igualdad de género y reducción de desigualdades.

Por otro lado, no existen juegos online para reforzar la igualdad de género y la reducción de desigualdades a los más pequeños, por lo que se ha pensado en la utilidad de realizar un juego que sensibilice sobre estos temas.

### 1.3. Objetivos

El objetivo principal que se busca es crear una aplicación web, que a la vez de informar y recolectar recursos, también conciencie sobre la igualdad de género y la reducción de

desigualdades.

Para cumplir el objetivo principal tendremos que cumplir los siguientes objetivos específicos:

1. Realizar un juego para la concienciación de la igualdad de género y reducción de desigualdad.
2. Dicho juego constará de una gymkana virtual dentro de lugares emblemáticos de la ciudad de Valencia, dirigida a niños entre 6 y 11 años, donde realizaran diferentes minijuegos ambientados a la igualdad de género y reducción de desigualdades.
3. Crear ficheros de datos georreferenciados para almacenar los recursos disponibles para la ayuda a la mujer y a los grupos vulnerables, como son las minorías étnicas, personas migrantes, las personas con discapacidad, personas sin techo y personas que pertenecen al colectivo LGBTI que proporciona el ayuntamiento de Valencia y la consejería de Valencia y/o son encontradas en la ciudad de Valencia.
4. Indexar los recursos disponibles en la ciudad de Valencia dividiéndose entre las diferentes categorías recursos para la mujer, minorías étnicas y migrantes, personas con discapacidad, personas sin techo y personas del colectivo LGBTI. Asimismo mostrando toda la información, tanto de contacto y de localización de cada uno.
5. La aplicación web estará diseñada e implementada para diferentes plataformas, como PC, smartphones y tabletas.

## 1.4. Organización de la memoria

Este TFG se divide de ocho capítulos, donde el primero es una introducción a la temática de este trabajo, de también se expondrá la motivación y los objetivos de este.

En el segundo capítulo está el estado del arte, donde se hará un repaso a los antecedentes teóricos y sociales del proyecto, se comprará los recursos o proyectos existentes de índole similar, y se hará un repaso de las tecnologías utilizadas en el desarrollo de este proyecto.

En el tercer capítulo se realiza un análisis previo al desarrollo del proyecto, donde se definirán unos requisitos, se especificarán unas funcionalidades, se hará una estimación tanto temporal como de costes y se analizará la viabilidad, tanto económica, legal y de riesgos del proyecto.

En el cuarto capítulo se encuentra el análisis del sistema, que desempeña la primera fase del desarrollo del proyecto, en el cual se analizarán los casos de uso y las actividades que desempeñará el proyecto.

En el quinto capítulo se pasará al diseño del sistema, que es la segunda fase del desarrollo del proyecto, donde se expondrán las clases que utilizará el proyecto, como se hará un estudio del diseño de la aplicación web y del videojuego por medio de bocetos de las interfaces de usuario y el estilo que tomarán estas.

En el sexto capítulo se encuentra la última fase del desarrollo del proyecto, donde se hará un repaso de las herramientas que se utilizarán en el proyecto, como la implementación del videojuego y la aplicación web.

En el séptimo capítulo se mostrarán las diferentes pruebas realizadas en el desarrollo del proyecto y los resultados finales.

Finalmente, en el octavo capítulo se finalizará con las conclusiones sacadas a lo largo de la confección del proyecto y el posible trabajo futuro y mejoras que se podrán realizar en el futuro.



# Capítulo 2

## Estado del arte

### 2.1. Antecedentes

#### 2.1.1. Objetivos de Desarrollo Sostenible

Los Estados Miembros de las Naciones Unidas propusieron y aprobaron en 2015 una agenda para 2030, que se compondría de objetivos globales para el desarrollo sostenible para acabar con la pobreza, proteger el planeta y bienestar de todos, que terminarían llamándose los Objetivos de Desarrollo Sostenible (ODS).



Figura 2.1: Logo de Objetivos de Desarrollo Sostenible [1]

Son 17 dichos objetivos:

1. Fin de la pobreza
2. Hambre cero
3. Salud y bienestar
4. Educación de calidad
5. Igualdad de género
6. Agua limpia y saneamiento
7. Energía asequible y no contaminante
8. Trabajo decente y crecimiento económico
9. Industria, innovación e infraestructura
10. Reducción de desigualdades
11. Ciudades y comunidades sostenibles

12. Producción y consumo responsables
13. Acción por el clima
14. Vida submarina
15. Vida de ecosistemas terrestres
16. Paz, justicia e instituciones sólidas
17. Alianzas para lograr los objetivos

[1]

#### **2.1.1.1. Igualdad de género**

La igualdad de género no es solo un derecho humano básico, sino también un importante soporte para convivir en un mundo pacífico, próspero y sostenible.

En las últimas décadas han sido numerosos avances que se han logrado dentro de la igualdad de género, por ejemplo, más niñas escolarizadas, creación y modificación de leyes que fomentan la igualdad de género y más mujeres en altos cargos, pero aún queda un largo camino que recorrer, ya que aún existen grandes dificultades y desventajas para las mujeres en la actualidad. [10]

##### **2.1.1.1.1. Igualdad de género en España**

Desde 2005 el Instituto Europeo para la Igualdad de Género (EIGE) publica el Índice Europeo de Igualdad de Género, que muestra una visión general sobre la igualdad en Europa, comparando a países miembros y el estudio de su evolución, que permite exponer las áreas que necesitan mejoras para el cumplimiento de la igualdad de género.

Actualmente, España se sitúa en el puesto nº6 en el Índice Europeo de Igualdad de Género, que fue publicado en 2021, con una puntuación de 73.7 sobre 100. [11]

##### **2.1.1.1.1.1. Instituto de las Mujeres**

El Instituto de las Mujeres (IMs) es una organización autonómica, vinculada al Ministerio de Igualdad, que promueve el mismo trato y oportunidades para hombres y mujeres, formulando informes de utilización de las Directivas de la Unión Europea junto con la cooperación de otros Departamentos.

Su objetivo principal es promover y fortalecer los requisitos para asegurar la libertad, una igualdad real entre mujeres y hombres, como la participación en vida política, civil, laboral y económica de las mujeres, y eliminar y prevenir todo tipo de discriminación por razón de género.

El IMs se creó a partir del Instituto de la Mujer y la Dirección General para la Igualdad de Oportunidades con el propósito de racionalizar la organización de la gestión, y evitar duplicidad ante organismos administrativos a través de la Ley 15/2014, de 16 de septiembre (BOE, 17 de septiembre de 2014), se acuerda la incorporación de las competencias de la Dirección General para la Igualdad de Oportunidades dentro de los cometidos

y estructura del Instituto de la Mujer que paso a denominarse ‘Instituto de la Mujer y para la Igualdad de Oportunidades’. Finalmente, en la ley de Presupuestos Generales del Estado para 2021, la organización pasaría a denominarse Instituto de las Mujeres. [12]

#### **2.1.1.1.1.2. Comisión Jurídica y Social de la Mujer**

La Comisión de la Condición Jurídica y Social de la Mujer se reunió por primera vez en febrero de 1947, en Lake Success, Nueva York, poco después de la creación de las Naciones Unidas. En ese momento, los quince representantes gubernamentales que integraban el comité eran todas mujeres. Desde sus inicios la Comisión ha contado con el apoyo de una unidad de las Naciones Unidas que luego se convirtió en la División para el Adelanto de la Mujer, dependiente de la Secretaría de las Naciones Unidas.

Además, contribuyo al trabajo de las oficinas de las Naciones Unidas, como con el convenio relativo a la igualdad de remuneración entre la mano de obra masculina y femenina por un trabajo de igual valor de la Organización Internacional del Trabajo, que consagro el principio de igual salario por trabajo igual. En 1963, los esfuerzos consolidados de los derechos de las mujeres llevaron a la Asamblea General de las Naciones Unidas a pedir a la Comisión que de una declaración sobre la eliminación de la discriminación de las mujeres, que fue aprobada por la Asamblea en última instancia en 1967.

Esta declaración fue seguida en 1979 por la convención sobre la eliminación de todas las formas de discriminación contra la mujer, un instrumento jurídicamente vinculante, también redactado por la Comisión. En 1987, en el marco del seguimiento de la Tercera Conferencia Mundial sobre la Mujer, la Comisión asumió el liderazgo de las labores de coordinación y promoción del trabajo del sistema de las Naciones Unidas en los asuntos económicos y sociales para el empoderamiento de la mujer. Durante el mismo periodo, la Comisión ayudo a colocar el tema de la violencia contra la mujer en primer plano del debate internacional por la primera vez. Estos esfuerzos cristalizaron en la Declaración sobre la eliminación de la violencia contra la mujer, aprobada por la Asamblea General el 20 de diciembre de 1993.

La Comisión actuó como órgano preparatorio de la Cuarta Conferencia Mundial sobre la Mujer de 1995, en la que se aprobó la Declaración y Plataforma de Acción de Beijing. Después de la conferencia, la Asamblea General ordeno a la Comisión que desempeñará un papel central en la supervisión de la implementación de la Declaración y Plataforma de Acción de Beijing y de asesorar al Consejo Económico y Social en consecuencia. [13]

#### **2.1.1.2. Reducción de desigualdades**

1.1.2 Reducción de desigualdades La reducción de la desigualdad y es una parte integral dentro del seguimiento de los ODS. Dentro de los países y entre ellos persiste un problema de desigualdad, aunque existen algunos aspectos con signos positivos en la reducción de desigualdad, como es la reducción de diferencia de ingresos entre algunos países o las mejoras en la situación comercial en países de bajos ingresos, aún sigue existiendo una desigualdad abismal.

En el momento en el que dentro de un país la desigualdad de prominente, su inestabilidad social y posibilidad de conflicto también lo son, ya que en gran parte la población queda excluida del crecimiento económico del país, por lo que si persiste durante mucho tiempo se estrecharan las políticas redistributivas por la negación de trasvase de recursos

por las elites económicas. Todo puede llevar a la acumulación de problemas sociales en salud, vivienda o educación, y la ralentización del crecimiento económico de los países. [14]

### 2.1.1.2.1. Reducción de desigualdades en España

El desbalance de ingresos supone uno de los problemas socioeconómicos más importantes de España, que sería muy superior a la mayoría de los países europeos, y que persiste en el tiempo y hace a España vulnerable ante los posibles choques económicos. Una de las principales razones de este hecho es la menor capacidad redistributivas de los sistemas tributario y de seguridad social. La evolución de la desigualdad en España depende del ciclo económico ya que crece rápidamente en la recesión, pero disminuye levemente en el crecimiento, causando un impacto negativo, mientras crece el desempleo y baja el crecimiento económico, en los ingresos de hogares de bajos ingresos.

Una vez iniciada la crisis en 2008 se incrementó de manera trágica una desigualdad en los hogares debido a la duración e intensidad de esta, que llevo a una disminución en la proporción de la población con ingresos medios. Desde 2014 se observa una tendencia negativa en el crecimiento económico, que si no se soluciona se podría cronificar. [15]

### 2.1.2. Recursos existentes

Al buscar aplicaciones web con similitudes solo se han podido encontrar en las páginas web de los ayuntamientos. El primero de los ejemplos es el sitio web del ayuntamiento de Valencia en la sección de Infocidad, donde se ofrece, teniendo que navegar por diferentes secciones, diferentes categorías, siendo si los recursos se ofrecen en la ciudad, o si se ofrecen de forma municipal, dentro de estos se encuentran diferentes secciones como son ‘Mujeres’, ‘Sin Techo’ o ‘Minorías étnicas’. Dentro de esta aplicación web solo se puede acceder a la información de un recurso cada vez.

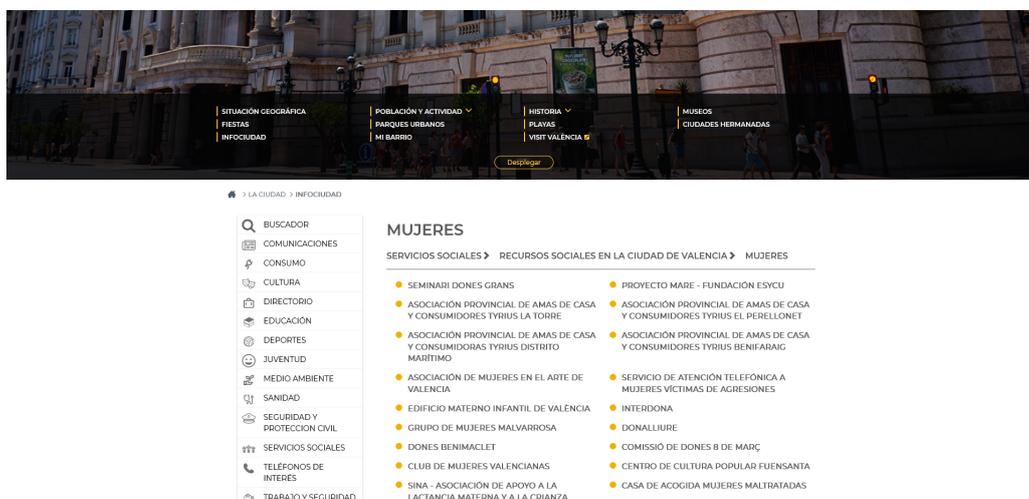


Figura 2.2: Captura 1 Sitio Web Ayuntamiento de Valencia [2]

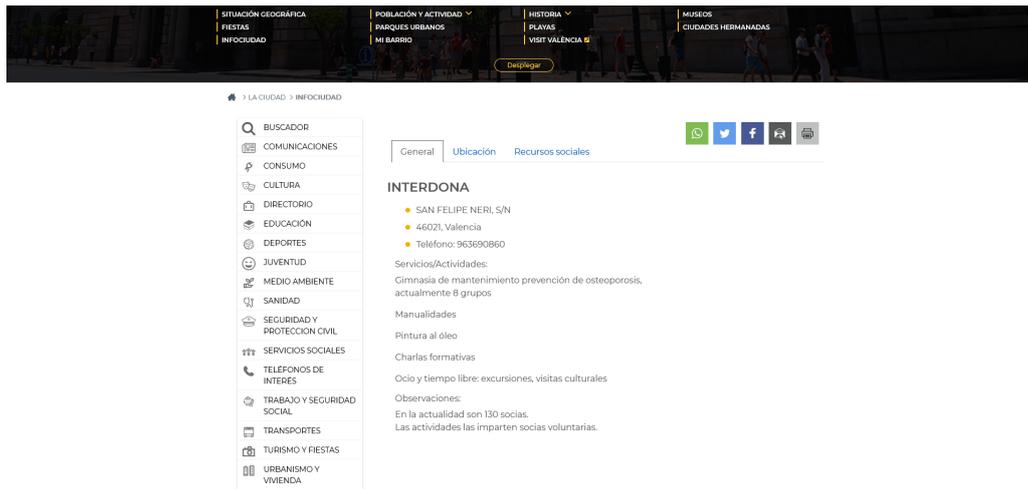


Figura 2.3: Captura 2 Sitio Web Ayuntamiento de Valencia [2]

El segundo ejemplo se trata del sitio web del ayuntamiento de Madrid, donde se muestra un listado dividido en páginas, con información básica de cada recurso como es la dirección y el número de teléfono, donde se puede acceder a la información completa si accedemos a un enlace, dejando solo ver un recurso completo cada vez. Esta página también ofrece un mapa con los recursos y sus localizaciones.

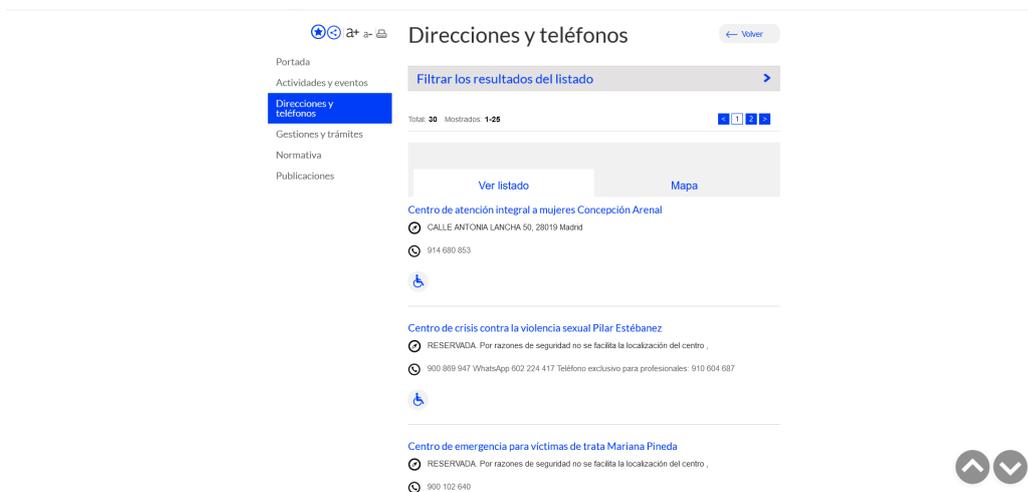


Figura 2.4: Captura 1 Sitio Web Ayuntamiento de Madrid [3]

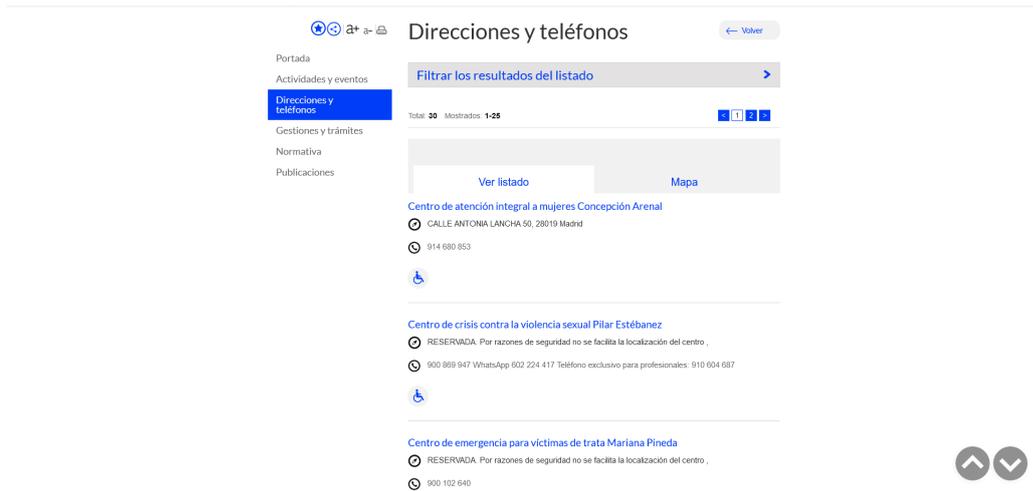


Figura 2.5: Captura 2 Sitio Web Ayuntamiento de Madrid [3]

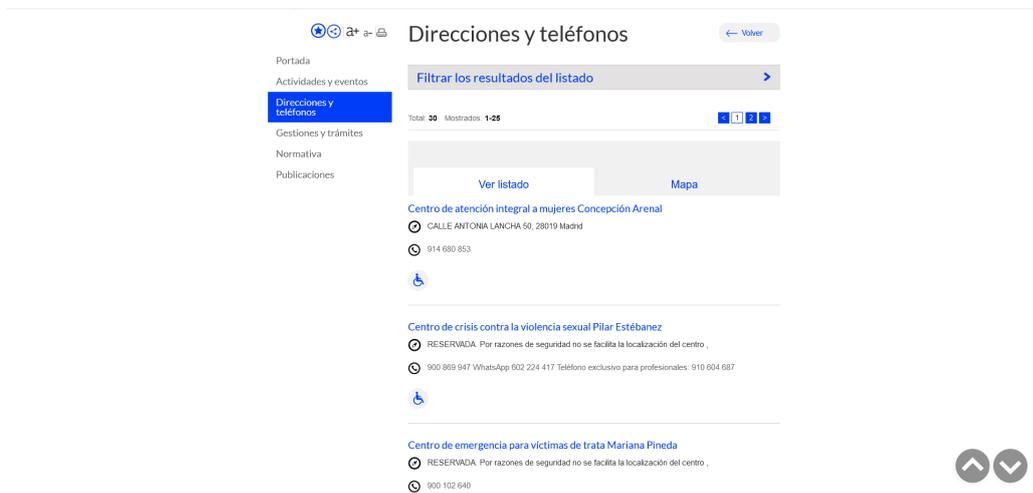


Figura 2.6: Captura 3 Sitio Web Ayuntamiento de Madrid [3]

Por último está el ejemplo del sitio web del ayuntamiento de Barcelona, específicamente la sección de Equidad de Género, donde se ofrece un mapa con las localizaciones de los recursos y un listado de ellos donde se muestra la dirección, contacto y enlace a su página web. También ofrece un filtrador por distrito, ámbito y/o tipo de acción.

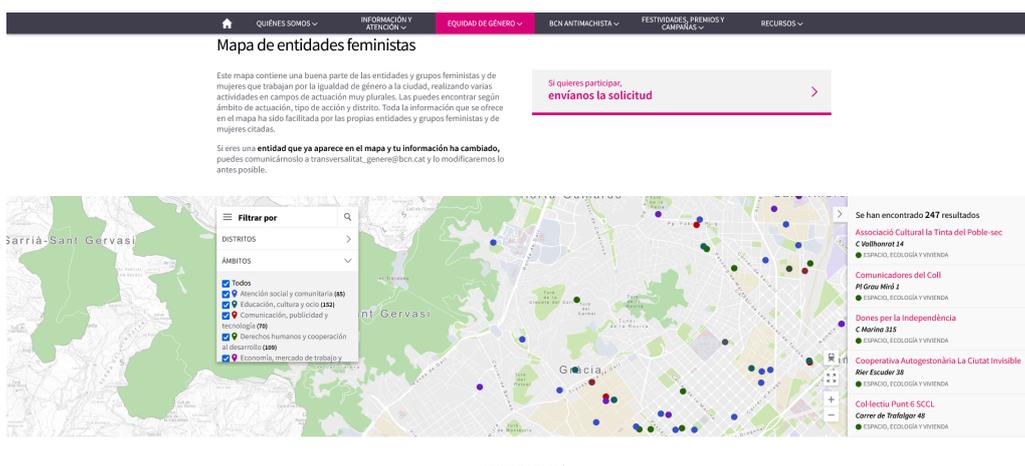


Figura 2.7: Captura 1 Sitio Web Ayuntamiento de Barcelona [4]

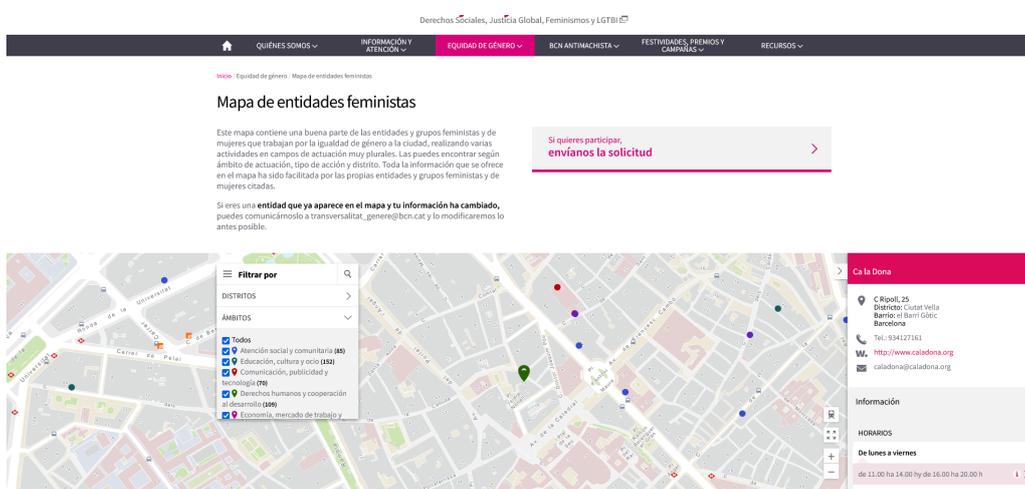


Figura 2.8: Captura 2 Sitio Web Ayuntamiento de Barcelona [4]

Con respecto al juego, no se ha encontrado videojuegos similares que cumplan con las características de Gymkana completamente virtual.

### 2.1.2.1. Aportaciones

Por un lado, para crear la aplicación web se ha buscado mezclar las características positivas de los recursos ya existentes y añadir nuevas características para poder componer una aplicación web que resulte más cómoda, intuitiva y con acceso a la información más rápido, por lo tanto, se cuenta con las siguientes características:

- Distribución de la aplicación web en diferentes categorías, de fácil acceso, diferenciando por sector al que van dirigidos los recursos.
- Un mapa al comienzo de cada sección donde se mostrarán las localizaciones de los recursos mostrados.
- Un filtrador por código postal, así facilitando la visualización de los recursos más relevantes.

- Todos los recursos son integrados en la misma página, así como toda su información.
- Los recursos son lanzados a partir de ficheros de datos externos al código de la aplicación web, por lo que una posible modificación de los datos almacenados o la inserción de nuevos recursos o datos se podría realizar con facilidad.

Y por otro lado, como se ha podido observar en el apartado anterior, no se ha encontrado videojuegos similares, por lo que se ha buscado crear un juego con las siguientes cualidades:

- Gymkana completamente virtual en la ciudad de Valencia.
- Personaje personalizable.
- Minijuegos educativos.

## 2.2. Tecnologías

### 2.2.1. HTML

Lenguaje de Marcas de Hipertexto (HyperText Markup Language, HTML) es el elemento más básico de la web, con él se define tanto el significado como la estructura del contenido de la aplicación web. El concepto de ‘hipertexto’ referenciaría a aquellos enlaces que entrelazan aplicaciones web, siendo estos uno de los valores fundamentales de la web. [16]

#### 2.2.1.1. HTML5

HTML5 es la última versión disponible de HTML la cual define reglas para el manejo de errores que conlleva a que la sintaxis sea compatible con implementaciones comunes. XHTML5 es otro tipo de sintaxis que no difiere de la sintaxis compatible con XML, y por ejemplo se puede utilizar para especificar el charset de un documento a partir de la etiqueta [17]:

```
1 <meta charset="utf-8" />
```

Código 2.1: Código de ejemplo charset

#### 2.2.1.2. DOCTYPE

El nuevo HTML5 requiere que se declare un elemento DOCTYPE en la parte superior de la página, de esta manera nos aseguramos de que el navegador muestre la página en modo básico. De lo contrario, para la versión XHTML5, este elemento es opcional por XML funciona de manera diferente en nuestro navegador [17].

### 2.2.1.3. Elementos y atributos

Un elemento se puede reconocer ya que se encuentra encerrado entre diples ‘<>’, el cual se puede definir con mayúsculas o minúsculas ya que no distingue mayúsculas. Un atributo por el otro lado son los valores específicos para un elemento que se sitúan detrás del título del elemento pero antes del cierre de diplo [16]. Un ejemplo puede ser la creación de un enlace, que se denomina con el elemento ‘a’ y que tiene un atributo ‘href’ que asigna la dirección del enlace:

```
1 <a href="https://www.example.com"> Enlace </a>
```

Código 2.2: Código de ejemplo enlace HTML

### 2.2.2. CSS

Hojas de Estilo en Cascada (Cascading Style Sheets, CSS), es un lenguaje que es utilizado para definir el estilo o visualización de una web, es decir, como se muestran los elementos organizados tanto en pantalla, forma impresa, voz u otros medios.

La especificación de CSS esta estandarizada por W3C (World Wide Web Consortium), desde la salida de su última versión, CSS3, las especificaciones han aumentado y han creado procesos más eficientes en el desarrollo y publicación de propuestas independientes [18, 19].

Un ejemplo de estructura y definición básico podría ser el siguiente, donde ‘.class’ hace referencia a la clase de un elemento, y se cambia el tamaño de la letra a 15px con ‘font-size’ y un margen de 10px con ‘margin’:

```
1 .class{
2     color: 15px;
3     margin: 10px;
4 }
```

Código 2.3: Código de ejemplo CSS

### 2.2.3. JavaScript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Aunque se conoce como un lenguaje de scripting para los sitios web y se utiliza en muchos entornos externos al navegador, como Node.js, Apache CouchDB y Adobe Acrobat. Java Script es un lenguaje de programación basado en prototipos, un multi-modelo de un solo hilo, dinámico, con soporte para la programación orientada a objetos, imperativa y declarativa.

Dentro de la sintaxis de JavaScript no se tienen en cuenta los espacios y saltos de línea, por lo que es necesario utilizar punto y coma ‘;’ para separar diferentes líneas de código, y no es preciso definir variables precisas, ya que cualquier variable puede utilizar cualquier tipo de atributo [20, 21]. Con el siguiente ejemplo, definimos una función ‘example’ donde definimos una variable de cadena de caracteres y devuelve dicha cadena anidada a otra, y donde también se definen diferentes comentarios, tantos de una sola línea como un bloque

de comentarios:

```
1 function example() {
2     //Comentario de una linea
3     var cadena = "Hello";
4
5     /*
6     * Bloque de comentario
7     */
8     return cadena + "world";
9 }
```

Código 2.4: Código de ejemplo JavaScript

## 2.2.4. Unity

Unity es un motor de desarrollo o motor de juegos, es decir, un software que contiene automatizaciones de programación para poder diseñar, crear y administrar entornos interactivos. Unity maneja variadas funcionalidades para ayudar al usuario a crear sus proyectos como un motor gráfico para renderizar gráficos 2D y 3D, un motor físico para simular acciones físicas, sonidos, animaciones y un largo etc.

En definitiva Unity es una herramienta que permite crear videojuegos para distintas plataformas, como son PC, teléfonos móviles, aplicaciones web, o videoconsolas, gracias a editores visuales y programación basada en scripts. [22]

### 2.2.4.1. Unity2D

Unity2D es el motor de Unity para proyectos bidimensionales, el cual se basa en Sprites, que son objetos gráficos en 2D o simples texturas estándar que se pueden combinar y procesar para mejorar el rendimiento y comodidad del desarrollo de un proyecto 2D.

Unity 2D precisa de un editor de Sprites incorporado con el que se pueden extraer secuencias de Sprites a partir de imágenes grandes, por lo que se puede editar múltiples Sprites a partir de una sola textura en un editor de imágenes. Estos Sprites se renderizan usando el componente Sprite Renderer en lugar de un Mesh Renderer, que se utilizaría para objetos tridimensionales.

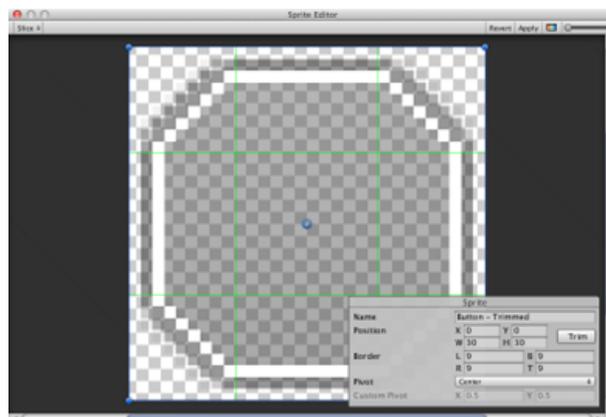


Figura 2.9: Editor de Sprites en Unity [5]

Asimismo, Unity2D tiene su propio motor de física independiente de otros componentes, para así aprovechar optimizaciones que solo son disponibles en 2D. Todos los componentes del motor físico son iguales a aquellos que se encuentran en la versión 3D, solo que estos son versiones ‘aplanadas’ y modificadas para objetos bidimensionales y añadiendo al nombre del componente un ‘2D’ al final, por lo que los Sprites se podrían equipar con los componentes Rigidbody 2D, Box Collider 2D y Hinge Joint 2D. [23]

#### **2.2.4.2. Unity WebGL**

Unity maneja diferentes compiladores para que los proyectos creados en Unity se puedan ejecutar en diferentes plataformas, uno de ellos es WebGL, el cual permite convertir el proyecto para poder ser ejecutado en aplicaciones web, donde encripta el proyecto, normalmente escrito en C, C++ Y C#), en ficheros JavaScript utilizando HTML5 y APU.

Es necesario que todo código este escrito en lenguaje JavaScript para que pueda funcionar con WebGL. El código es encriptado en sm.js, que es un subconjunto optimizado de JavaScript que permite que se ejecute el proyecto con facilidad. [24]

#### **2.2.4.3. Scripting**

Unity usa la implementación estándar del tiempo de ejecución de Mono para secuencias de comandos, pero tiene sus propias prácticas y técnicas para acceder al motor desde las secuencias de comandos. [25]

##### **2.2.4.3.1. EventSystem**

EventSystem es una forma de enviar eventos a objetos en la aplicación según el teclado, el mouse, el toque, la entrada del usuario y otros eventos. EventSystem consta de varios componentes que trabajan juntos para enviar eventos. [26]

#### **2.2.4.4. UI (Interfaz de Usuario)**

El sistema de UI permite crear interfaces de usuario rápidas. [27]

##### **2.2.4.4.1. Canvas**

Canvas es un componente o GameObject que debe estar presente para introducir todos los elementos UI, ya que todos los demás elementos UI deben ser secundarios a este. Canvas es representado como un área rectangular del lienzo, que facilita la colocación de elementos UI, los cuales se dibujan según el orden en el que aparecen en la jerarquía, es decir, el primer hijo se dibuja primero, y después el segundo hijo, etc. Pero en el caso de que dos elementos se superpongan el ultimo que aparece en la jerarquía se dibujará encima del anterior. [28]

##### **2.2.4.4.2. Componentes visuales**

Con la introducción del sistema UI se han agregado nuevos componentes para ayudar a crear funciones de interfaz gráfica de usuario específicas, como son el texto, imágenes y máscaras.

El componente de texto (Text) o Label tiene un área de texto para ingresar el texto a mostrar, al cual se le puede modificar fuentes, estilos, tamaño y formato de texto.

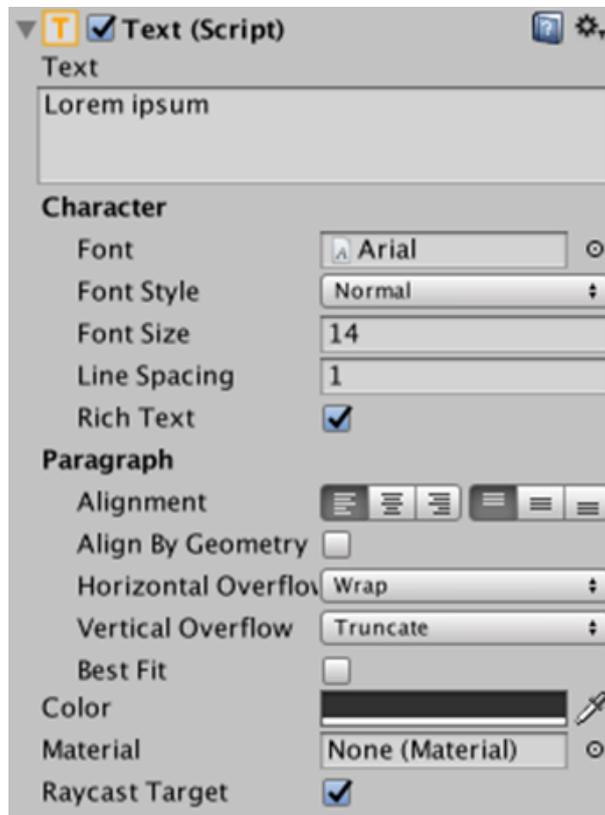


Figura 2.10: Declaración de componente Text en Unity [5]

Por otro lado están los componentes de imagen, en primero es Image, que se renderiza a partir de Sprites, que se pueden aplicar bajo el campo de TargetGraphic, el cual se le puede modificar el color y el material en el área de color y en los componentes de imagen respectivamente y el segundo es Raw Image, donde la diferencia es que este utiliza imágenes sin procesar en vez de Sprites.

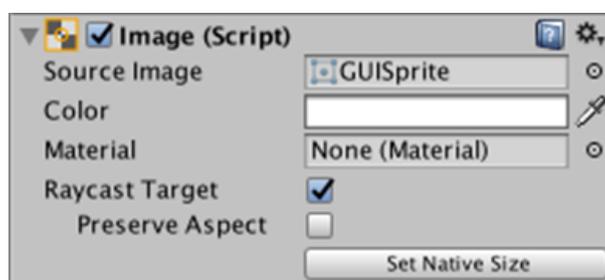


Figura 2.11: Declaración de componente Image en Unity [5]

Y por último está las máscaras (Mask), que aunque técnicamente no sea un control visible de UI, este modifica la apariencia de los elementos hijos de su jerarquía, ya que enmascara o restringe a dicho elemento hijo para que por más grande que sea que la máscara padre, este solo mostrara la sección que es visible dentro de la máscara padre. [5]

### 2.2.4.4.3. Componentes interactivos

Los componentes interactivos son aquellos componentes del sistema UI que maneja las interacciones, tanto de ratón-táctiles o de teclado.

Entre los componentes interactivos se encuentran botones (Button), que utiliza el evento `OnClick` para manejar la interacción, `Toggle` y `Toggle Group`, o casillas de verificación, que utiliza el evento `OnValueChanged`, también están el deslizador (Slider) y la barra de desplazamiento (Scrollbar), que utiliza también el evento `OnValueChanged`, y maneja los valores entre un máximo y un mínimo. El desplegable o `Dropdown`, utiliza el evento `OnValueChanged` y selecciona entre diferentes opciones. Igualmente nos encontramos el campo de `Input` (Input Field) que utiliza el elemento de texto editable y utiliza el evento cuando su campo ha sido modificado. Y finalmente se con el `Scroll Rect`, que suele ser la mezcla entre una máscara y un `Scrollbar` que muestra contenido desplazable. [6]



Figura 2.12: Componentes interactivos en Unity [6]

### 2.2.4.5. Animación

Unity utiliza un sistema de animación llamado Mecanim, donde a partir de clips de animación y controladores se puede animar el movimiento de los diferentes `GameObject` de la escena. Los clips de animación o `Animation Clips` son la línea de tiempo de animaciones o secuencias específicas, las cuales en 2D se crearían median diferentes `Sprites`, como la secuencia de diferentes `Sprites` simulando un movimiento, o el traslado dentro de la escena de un `GameObject`. Estos clips son la base para poder crear el controlador o `Animation Controller` que maneja al `GameObject` para que dependiendo de la situación o del input este ejecute un clip u otro. [29, 30, 31]

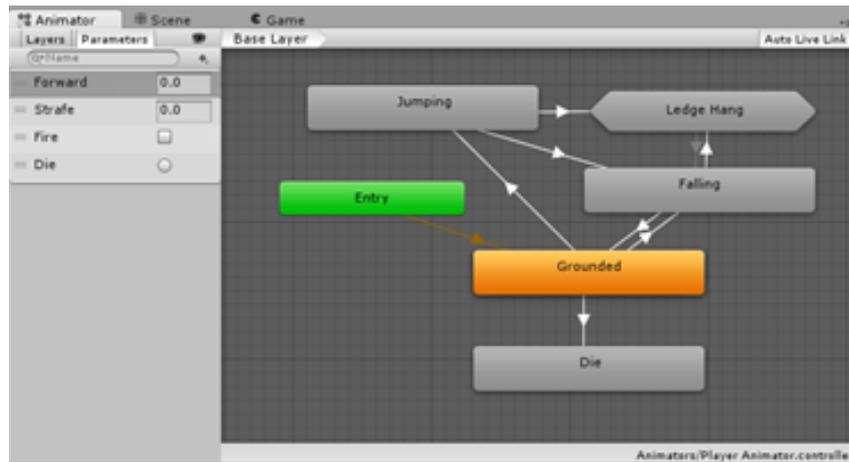


Figura 2.13: Animator del Animation Controller en Unity [7]

### 2.2.5. GeoJSON

JSON, o JavaScript Object Notation, es un formato de fichero de texto que se utiliza para almacenaje de objetos de JavaScript, aunque gracias a su extensa utilización en otros lenguajes, se considera como un lenguaje aparte de JavaScript.

GeoJSON es un formato específico de los ficheros JSON, que se utiliza para objetos con geolocalización, por ello es utilizado en aplicaciones web de cartografía por ser un estándar rápido, ligero y sencillo.

Con este formato se puede describir siete tipos de geometrías definidas en OpenGIS, que son puntos, líneas, polígonos, puntos múltiples, líneas múltiples, polígonos múltiples o colecciones [32]. Algunos ejemplos son:

```

1  {
2    "type": "Point",
3    "coordinates": [100.0, 0.0]
4  }
```

Código 2.5: Ejemplo GeoJSON para un punto

```

1  {
2    "type": "Polygon",
3    "coordinates": [
4      [
5        [100.0, 0.0],
6        [101.0, 0.0],
7        [101.0, 1.0],
8        [100.0, 1.0],
9        [100.0, 0.0]
10     ]
11   ]
12 }
```

Código 2.6: Ejemplo GeoJSON para un polígono

## 2.3. Elección de tecnologías

En el apartado anterior se ha explicado brevemente algunas las tecnologías que se han utilizado en el proyecto, por lo que se ha utilizado los lenguajes HTML, CSS y JavaScript para la aplicación web, mientras que para el desarrollo del videojuego se ha utilizado Unity.

Para la elaboración de la aplicación web aparte de utilizar HTML, CSS y JavaScript básico, también se han utilizado unas tecnologías más específicas que se utilizan en estos lenguajes. Entre estas tecnologías se encuentra Bootstrap, que es un toolkit que ayuda a la elaboración del desarrollo front-end, en especial a la arquitectura de la aplicación web, donde importando la librería CSS, por medio de clases, este podrá redistribuir la página para que sea más adaptable a los diferentes dispositivos. Asimismo también dispone de librerías de iconos de dominio público que se pueden añadir en la aplicación web fácilmente.



Figura 2.14: Logo Bootstrap [8]

Otra herramienta elegida ha sido Leaflet, una librería de JavaScript para crear mapas que también se he pueden utilizar desde dispositivos móviles. Con esta herramienta se puede cargar un mapa, como añadirle marcadores como texto, áreas o puntos. Gracias a esta herramienta se facilita la localización de lugares, ya que también se puede navegar por él, pudiendo hacer zoom y moverse por el mapa.



Figura 2.15: Logo Leaflet [9]

Para la realización del videojuego se ha utilizado cuatro herramientas dentro de Unity, que son Unity2D, WebGL, UI y Animator, que como se mencionó en el apartado anterior Unity2D es el motor para proyectos de 2 dimensiones de Unity, WebGL es compilador de Unity para aplicaciones web, que traduce los ficheros de funcionalidad e interfaz a lenguaje JavaScript para que funcione en aplicaciones web, UI es el creador de interfaz de usuario para los proyectos en Unity y Animator es el sistema de animación de Unity.

Y por último se encuentra la tecnología que se ha utilizado para los ficheros de recursos, en este caso se ha utilizado ficheros JSON con formato GeoJSON, ya que es un formato específico para almacenar puntos georreferenciados.



# Capítulo 3

## Requisitos, especificaciones, coste, riesgos, viabilidad

### 3.1. Requisitos

#### 3.1.1. Requisitos funcionales

- F1. La aplicación web será responsive, es decir, se adaptará al dispositivo desde donde se visualice.
- F2. La aplicación web mostrará los recursos que están almacenados en los ficheros georeferenciados.
- F3. Los ficheros de los recursos almacenaran datos como nombre, titularidad, correo electrónico, número de teléfono y fax, página web y, si procede, dirección, código postal y coordenadas.
- F4. Los recursos mostrados se dividirán en diferentes secciones, que formarán cada apartado del menú principal.
- F5. El usuario podrá acceder a las localizaciones de los recursos, en caso de que sea aplicable, a partir de un mapa indexado en la aplicación web.
- F6. El usuario podrá buscar recursos por Código Postal a partir de un buscador dentro de cada apartado de recursos de la aplicación web.
- F7. El videojuego será indexado dentro de la aplicación web por medio de WebGL de Unity.
- F8. El usuario podrá personalizar la apariencia del personaje del videojuego.
- F9. El usuario podrá mover el personaje en el videojuego por medio de las flechas del teclado del ordenador.
- F10. El videojuego dispondrá de un cuadro de dialogo que aparecerá al interactuar con los organizadores (o metas) en cada nivel.
- F11. El videojuego tendrá tres escenarios y tres minijuegos.

- F12. Los juegos recogerán una puntuación, y será necesario llegar a un mínimo de puntos para pasar de nivel.
- F13. El usuario podrá jugar a los minijuegos con el ratón del ordenador.
- F14. El usuario podrá reintentar cada minijuego, y en el caso de ganar dicho minijuego también podrá continuar al siguiente nivel.

### 3.1.2. Requisitos no funcionales

- NF1. La aplicación web tendrá una interfaz intuitiva que facilitará su uso.
- NF2. Los recursos serán guardados ficheros de formato JSON georreferenciado, es decir, en GeoJSON.
- NF3. Los recursos serán añadidos de forma dinámica a la página.
- NF4. Los recursos se encontrarán dentro de la ciudad de Valencia.
- NF5. La aplicación web tendrá secciones, una para el videojuego y una para cada categoría (o grupo vulnerable).
- NF6. El videojuego consistirá en una Gymkana Virtual dentro de la ciudad de Valencia.
- NF7. El videojuego estará enfocado a niños de entre 6 y 11 años.
- NF8. Entre los minijuegos se encontrará un juego de cartas de memoria, un quiz y un juego de selección.

## 3.2. Especificaciones

A partir de los requisitos especificados en el apartado anterior, se han obtenido las siguientes funcionalidades para la aplicación web, mostradas en las siguientes tablas (Tablas 3.1 y 3.2), donde la categoría identifica si son visibles para el usuario (Evidente) o son simplemente funcionales o no visibles para el usuario (Oculta):

Funcionalidad	Categoría
Insertar los recursos de una categoría en una misma página	Evidente
Insertar los recursos a partir de los diferentes archivos JSON	Oculta
Estilo y disposición responsive	Evidente
Insertar marcadores de posición de los recursos en el mapa	Evidente
Insertar buscador de recursos por Código Postal	Evidente
Insertar videojuego por HTML en la aplicación	Evidente
Mostrar la ubicación en el mapa desde la información de cada recurso	Evidente

Tabla 3.1: Funcionalidades de la aplicación web

Las funcionalidades del videojuego son:

Funcionalidad	Categoría
Movimiento del personaje a partir de las flechas del teclado	Evidente
Insertar manager para pasar de nivel a nivel	Oculto
Insertar manager de cada videojuego para ejecutar cada minijuego	Evidente
Insertar cámara que siga al personaje cuando se mueve	Evidente

Tabla 3.2: Funcionalidades del videojuego

### 3.3. Planificación y estimación de costes

#### 3.3.1. Definición de actividades

Para poder ayudar a la realización de un proyecto es necesario tener una planificación previa al comienzo de la elaboración de este, para ello, se ha dividido el proyecto en las siguientes actividades:

1. Estado del Arte
  - a) Recopilar información sobre aplicaciones similares
  - b) Estudiar las herramientas de software para el desarrollo del proyecto
2. Especificación, planificación, estimaciones y viabilidad.
  - a) Definir requisitos funcionales y no funcionales
  - b) Especificar funcionalidades del proyecto
  - c) Planificar etapas del proyecto
  - d) Estimar y analizar costes
  - e) Analizar viabilidad y riesgos
3. Desarrollo del proyecto
  - a) Análisis del sistema
    - Determinar casos de uso del proyecto
    - Crear diagramas de casos de uso, actividad y estados
    - Recopilar datos de recursos
  - b) Diseño del sistema
    - Crear diagramas de clases y de secuencia.
    - Diseñar interfaces de usuario
  - c) Implementación del sistema
    - Crear texturas para el videojuego
    - Implementar interfaz del videojuego
    - Implementar funcionalidad del videojuego
    - Crear y rellenar ficheros geolocalizados de recursos
    - Implementar interfaz de la aplicación web
    - Implementar funcionalidad de la aplicación web
    - Insertar videojuego en aplicación web
4. Pruebas y resultados

### 3.3.2. Estimación temporal

Una vez definidas las actividades que se van a realizar en el proyecto, es necesario hacer una estimación del tiempo que tardaremos en realizar las actividades identificadas, para ello existen diferentes técnicas como son:

- Asesoramiento de expertos: Se trata de consultar a profesionales con experiencia y conocimiento en ejecución de actividades similares, que proporcionaran una estimación más fiable.
- Estimación por analogía: Estimar el tiempo de las actividades a partir del tiempo utilizado en proyectos con actividades similares que ya estén finalizados.
- Estimación por tres valores: En esta técnica se busca el tiempo estimado de cada actividad a partir de tres variables o estimaciones temporales, que son el tiempo más optimista, el tiempo más probable y el tiempo más pesimista, utilizando la siguiente formula:

$$t_e = \frac{(t_o + 4*t_m + t_p)}{6}$$

Donde  $t_e$  es el tiempo estimado,  $t_o$  es el tiempo más óptimo,  $t_m$  es el tiempo más probable y  $t_p$  es el tiempo más pesimista.

Para este proyecto se va a utilizar la técnica de estimación por tres valores calculado en días de trabajo, la estimación ha sido calculada por el autor únicamente sin utilizar ningún grupo de expertos. Las estimaciones se separan en la tabla 3.3 para el Estado del Arte, la tabla 3.4 para especificación, planificación, estimaciones y viabilidad, y las tablas 3.5, 3.6, 3.7, correspondientes a las diferentes fases del desarrollo del proyecto, es decir, análisis, diseño e implementación del proyecto, y finalmente la tabla 3.8 con el resumen de todas las tablas y pruebas y resultados.

Actividad	$t_o$	$t_m$	$t_p$	$t_e$
Recopilar información sobre aplicaciones similares	1	1.5	3	1.66
Estudiar las herramientas de software para el desarrollo del proyecto	2	3	5	3.16
<b>Total de tiempo estimado en días</b>	<b>3</b>	<b>4.5</b>	<b>8</b>	<b>4.83</b>

Tabla 3.3: Estimación temporal por días del Estado del Arte

Actividad	$t_o$	$t_m$	$t_p$	$t_e$
Definir requisitos funcionales y no funcionales	1	1	3	1.33
Especificar funcionalidades del proyecto	1	1	3	1.33
Planificar de etapas del proyecto	1	1.5	3	1.66
Estimar y analizar costes	1	1.5	3	1.66
Analizar viabilidad y riesgos	1	2	4	2.16
<b>Total de tiempo estimado en días</b>	<b>5</b>	<b>7</b>	<b>16</b>	<b>8.16</b>

Tabla 3.4: Estimación temporal por días del análisis de especificación, planificación, estimaciones y viabilidad

Actividad	$t_o$	$t_m$	$t_p$	$t_e$
Determinar casos de uso del proyecto	0.5	1	2	1.08
Crear diagramas de casos de uso, actividad y estados	1.5	2	4	2.25
Recopilar datos de recursos	1	2	4	2.16
<b>Total de tiempo estimado en días</b>	<b>3</b>	<b>5</b>	<b>10</b>	<b>5.5</b>

Tabla 3.5: Estimación temporal por días del análisis del proyecto

Actividad	$t_o$	$t_m$	$t_p$	$t_e$
Crear diagramas de clases y de secuencia	2	3	5	3.16
Diseñar interfaces de usuario	4	6	10	6.33
<b>Total de tiempo estimado en días</b>	<b>6</b>	<b>9</b>	<b>15</b>	<b>9.5</b>

Tabla 3.6: Estimación temporal por días del diseño del proyecto

Actividad	$t_o$	$t_m$	$t_p$	$t_e$
Crear texturas para el videojuego	5	7	15	8
Implementar interfaz del videojuego	2	3	5	3.16
Implementar funcionalidad del videojuego	7	10	20	11.16
Crear y rellenar ficheros geolocalizados de recursos	2	3	5	3.16
Implementar interfaz de la aplicación web	5	10	15	10
Implementar funcionalidad de la aplicación web	5	7	15	8
Insertar videojuego en aplicación web	1	2	4	2.16
<b>Total de tiempo estimado en días</b>	<b>27</b>	<b>42</b>	<b>79</b>	<b>45.66</b>

Tabla 3.7: Estimación temporal por días de la implementación del proyecto

Actividad	$t_o$	$t_m$	$t_p$	$t_e$
Estado del Arte	3	4.5	8	4.83
Especificación, planificación, estimaciones y viabilidad	5	7	16	8.16
Análisis del proyecto	3	5	10	5.5
Diseño del proyecto	6	9	15	9.5
Implementación del proyecto	27	42	79	45.66
Pruebas y resultados	3	5	10	5.5
<b>Total de tiempo estimado en días</b>	<b>47</b>	<b>72.5</b>	<b>138</b>	<b>79.16</b>

Tabla 3.8: Estimación temporal general del proyecto

Por lo que una vez planificado el tiempo estimado de cada actividad, se obtiene el resultado de 79.16 días de trabajo, pero es necesario planificar la temporalización de las actividades mediante un diagrama de Gantt, como en este caso todos los roles y partes has sido realizadas por una sola persona, el diagrama representara la organización para una sola persona. Versión ampliada en apéndice [Apéndice A.1 ].

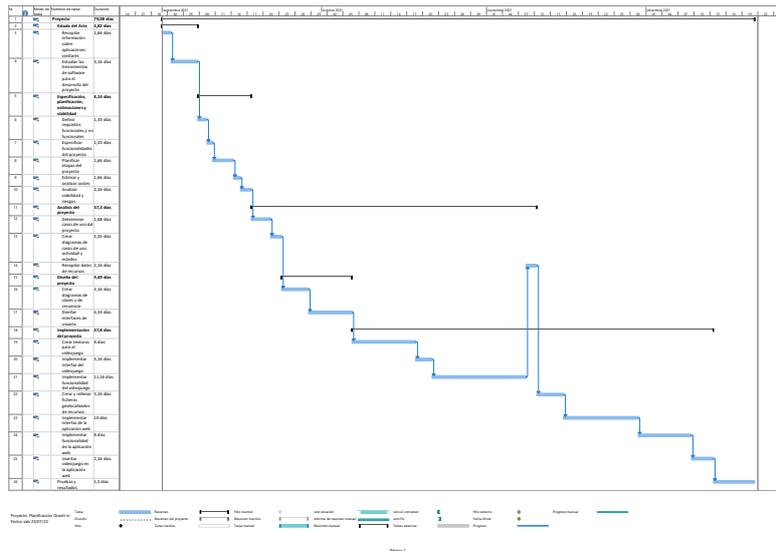


Figura 3.1: Diagrama de Gantt

### 3.3.3. Análisis de costes

Los costes se dividirán entre costes personales y costes de material o software. Dentro de los costes personales se encuentran los profesionales empleados en el proyecto, es decir, el gestor de proyecto, el analista, el artista, el desarrollador de videojuegos y el desarrollador web, donde se tendrá en cuenta tanto el sueldo que cobran los empleados como los costes indirectos de IRPF y Seguridad Social. Y dentro de los costes de material o software se sitúan los programas utilizados para la realización del proyecto.

#### 3.3.3.1. Costes personales

Comenzando por los costes personales, primero se ha de tener en cuenta el sueldo bruto, como su división en sueldo neto, IRPF y Seguridad Social. Primero.

Una vez planificadas que actividades se van a realizar en este proyecto, se ha dividido las actividades en diferentes profesiones, resultando en los siguientes profesionales:

1. Gestor de proyecto que se encargará de la especificación, planificación, estimaciones y viabilidad.
2. Analista programador que trabajará en el análisis del proyecto.
3. Diseñador gráfico que creará las texturas para el videojuego.
4. Desarrollador de videojuegos que se encargará de estudiar las herramientas de software que se utilizarán para el videojuego, diseñar e implementar las interfaces del videojuego e implementar la funcionalidad del videojuego.
5. Desarrollador web que recopilará información de aplicaciones similares, estudiará las herramientas de software que se utilizarán para la aplicación web, diseñará las interfaces de usuario, implementará la interfaz y funcionalidad de la aplicación web, como de insertar el videojuego en la aplicación web.

Para este caso se utilizara los porcentajes obtenidos de un empleado de 30 años, con 5 años de experiencia que reside en la Comunidad Valenciana, con la categoría profesional de Ingenieros y Licenciados, sin hijos ni personas a cargo, por lo que a partir de la calculadora de IRPF de El País [33] y los sueldos brutos se obtienen a partir de las medias encontradas sobre las profesiones en España:

Profesional	Sueldo Neto	IRPF	Seguridad Social	Sueldo Bruto
Gestor de proyectos	29596.10€	7863.90€	2540€	40000€
Analista programador	22904.20€	4816.20€	1879.60€	29600€
Diseñador gráfico	22304.90€	4572.60€	1822.50€	28700€
Desarrollador de videojuegos	24568.80€	5492.80€	2038.40€	32100€
Desarrollador Web	24235.90€	5357.50€	2006.60€	31600€

Tabla 3.9: Costes directos e indirectos anuales

Una vez aplicado el tiempo de empleo de cada profesional, el coste final de los costes personales contando con jornadas de 8 horas diarias son:

Profesional	Horas Totales	Sueldo Neto	IRPF	Seguridad Social	Sueldo Bruto
Gestor de proyectos	34.56	491.75€	130.66€	42.20€	664.62€
Analista programador	74.48	820.15€	172.46€	67.30€	1059.91€
Diseñador gráfico	64	686.31€	140.70€	56.08€	883.08€
Desarrollador de videojuegos	215.76	2548.54€	569.77€	211.44€	3329.76€
Desarrollador Web	311.49	3629.44€	802.31€	300.50€	4732.25€
				<b>Total</b>	<b>10669.62€</b>

Tabla 3.10: Costes personales

### 3.3.3.2. Costes de material o software

Y por último se encuentran los costes de material o software, que corresponderían a aquellos gastos en licencias de programas de software que se han utilizado para la realización del proyecto. Los programas utilizados para cada parte del proyecto con su coste mensual son:

- Visual Studio Code: Licencia Gratuita. Para las actividades:
  - Implementar interfaz del videojuego
  - Implementar funcionalidad del videojuego
  - Crear y rellenar ficheros geolocalizados de recursos
  - Implementar interfaz de la aplicación web

- Implementar funcionalidad de la aplicación web
- Insertar videojuego en aplicación web
- Visual Paradigm: Licencia Standard (19\$/mes o 18.68€/mes). Para las actividades:
  - Casos de uso del proyecto
  - Crear diagramas de casos de uso, actividad y estados
  - Crear diagramas de clases y de secuencia.
- Adobe XD: Plan Mensual (12.09€/mes). Para las actividades:
  - Diseñar interfaces de usuario
- Adobe Photoshop: Plan Mensual(24.19€/mes). Para actividades:
  - Crear texturas para el videojuego
- Procreate iPad: Pago único 9.99€. Para actividades:
  - Crear texturas para el videojuego
- Unity: Plan Profesional (1656€/año). Para actividades:
  - Implementar interfaz del videojuego
  - Implementar funcionalidad del videojuego

Una vez identificados los programas hay que calcular el precio, se cuenta con que los planes se pagaran completos, es decir, si un programa cuesta 10€ cada mes, significa que si sus actividades duran un total de 10 días, es decir, menos de un mes, el coste será de 10€, mientras que si se supera, por ejemplo, a un mes y medio de uso el coste de este pasara a dos meses, es decir, a 20€. Por lo que los gastos resultantes son:

Software	Dias de uso	Meses con suscripción	Precio
Visual Studio Code	37.64	∞	0€
Visual Paradigm	6.49	1	18.68€
Adobe XD	6.33	1	12.09€
Adobe Photoshop	8	1	24.19€
Procreate iPad	8	∞	9.99€
Unity	14.32	12	1656€
<b>Total</b>			<b>1720.95€</b>

Tabla 3.11: Costes de software

### 3.4. Análisis de viabilidad y riesgos

Una vez estimado tanto el tiempo como el coste del proyecto se debe verificar la viabilidad del proyecto, es decir, si es rentable realizar este proyecto tanto en aspectos económicos y legales, como analizar los posibles riesgos y la contención de estos.

### 3.4.1. Viabilidad económica

Primero hay que tener en cuenta el contexto del proyecto, al ser un proyecto social y de ayuda no se obtienen beneficios directos por el uso de este, por lo que se tiene entre manos un proyecto poco viable económicamente ya que los costes son claramente superiores a los beneficios.

Conociendo el contexto del proyecto una de las posibles soluciones a la poca viabilidad económica es pedir una subvención a la Administración pública, ya que toda la población puede disponer de la información de este proyecto

### 3.4.2. Viabilidad legal

Contemplando la Ley de Protección de Datos (RGPD) se puede determinar que el proyecto no incumple con ninguno de sus apartados, ya que no se recoge información del usuario, no es necesario el registro ni almacenaje de datos del usuario para el uso de la aplicación web y los datos de los recursos almacenados se pueden encontrar en sus sitios web públicos.

### 3.4.3. Análisis de riesgos

Para poder analizar los riesgos primero hay que identificar los activos del proyecto, entre ellos se encuentran los recursos almacenados, el sistema para mostrar los recursos, el sistema de búsqueda de recursos y el videojuego. Una vez determinados los activos hay que determinar unas posibles amenazas y vulnerabilidades:

1. Una de las posibles amenazas es en fallos o errores en los datos de los recursos almacenados, por lo que llevaría a desinformación sobre el recurso que tenga el error.
2. Otra sería fallos en el videojuego, como por ejemplo fallos en la disposición del videojuego, dejando así al videojuego sin disponibilidad por lo tanto no pudiéndose ejecutar cuando un usuario quiera ejecutarlo.
3. Una tercera amenaza se encontraría en el sistema para mostrar los recursos, como un fallo en el mapa que resultaría en incapacitar al usuario de poder localizar el recurso.
4. Y por último, otra posible amenaza estaría en el sistema de búsqueda, resultando en no poder filtrar los recursos por código postal.

### 3.4.4. Análisis de riesgos

Comenzando por el primer riesgo, este tiene una probabilidad de que ocurra bastante alta, ya sea porque una asociación ha actualizado algún dato, o esta ya no se encuentre disponible, por lo que para solucionarlo sería la notificación previa por parte del recurso de los cambios para poder actualizar los ficheros de datos.

Y finalizando con los tres últimos riesgos, la probabilidad de todos es baja, ya que una vez se ha incorporado al código y su funcionalidad esta activa es difícil que dejen

de funcionar, pero para solucionar estos riesgos es comprobar las actualizaciones de las diferentes herramientas que se utilizan para estos activos y cambiar los ficheros pertinentes para que estas actualizaciones no supongan un fallo en el sistema.

# Capítulo 4

## Análisis del sistema

Para analizar el sistema hay que contar con las siguientes especificaciones, como son que todos los usuarios son iguales y no necesitan registro, todas las páginas de recursos estar estructuradas de la misma forma y tienen la misma funcionalidad, por lo que cuando se refiera a ‘Recursos’ se identifican todas y cada una de las páginas/secciones de recursos.

### 4.1. Casos de uso

Primero comenzando con el diagrama de casos de uso solo es necesario especificar un actor que representa a cualquier usuario, ya que todos los usuarios son anónimos y pueden realizar las mismas funciones. El usuario puede acceder tanto a la página del videojuego con a las páginas de recursos, donde se pueden localizar los recursos y también ver la información relevante sobre ellos. Asimismo dentro de recursos se encuentra un sistema de búsqueda para filtrar los recursos por código postal. A continuación se muestra el diagrama y el desglose de las relaciones.

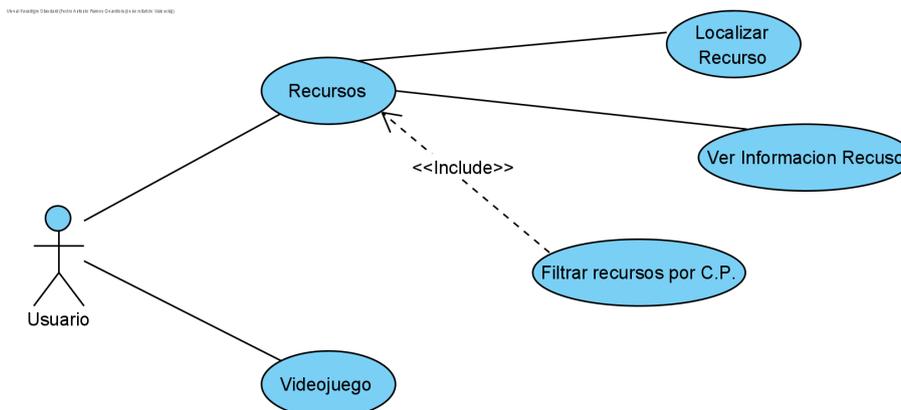


Figura 4.1: Diagrama de Casos de Uso

<b>Identificador</b>	CU-01
<b>Nombre</b>	Ver Información Recurso
<b>Actor</b>	Usuario
<b>Propósito</b>	Mostrar información sobre los recursos.
<b>Resumen</b>	Este caso de uso accede a la información de los recursos y los muestra en pantalla.
<b>Tipo</b>	Primario
<b>Referencias</b>	F1, F2, F3, F4, NF3 y NF5.
<b>Precondiciones</b>	Abrir página de recursos.
<b>Postcondiciones</b>	Se ha insertado los recursos a la página.
<b>Flujo de eventos principal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página de recursos.</li> <li>2. El sistema busca los recursos de la categoría abierta.</li> <li>3. El sistema inserta los recursos en la página de recursos.</li> </ol>
<b>Flujo de eventos alternativo</b>	<ol style="list-style-type: none"> <li>4. El usuario abre un recurso para mostrar la información del recurso.</li> </ol>

Tabla 4.1: Especificación Caso de Uso “Ver Información Recurso”

<b>Identificador</b>	CU-02
<b>Nombre</b>	Localizar Recurso
<b>Actor</b>	Usuario
<b>Propósito</b>	Localizar recursos en el mapa.
<b>Resumen</b>	Este caso de uso busca poder localizar los recursos en el mapa.
<b>Tipo</b>	Primario
<b>Referencias</b>	F5, NF2 y NF4.
<b>Precondiciones</b>	Abrir página de recurso.
<b>Postcondiciones</b>	Recurso localizado.
<b>Flujo de eventos principal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página de recursos.</li> <li>2. El sistema busca los recursos de la categoría abierta.</li> <li>3. El sistema inserta los recursos en el mapa dentro de la página de recursos.</li> </ol>
<b>Flujo de eventos alternativo</b>	<ol style="list-style-type: none"> <li>4. El usuario abre un recurso para mostrar la información del recurso.</li> <li>5. El usuario presiona sobre el botón para mostrar recurso específico en mapa.</li> <li>6. El sistema muestra el mapa el recurso seleccionado y su información.</li> </ol>

Tabla 4.2: Especificación Caso de Uso “Localizar Recurso”

<b>Identificador</b>	CU-03
<b>Nombre</b>	Filtrar Recursos por C.P.
<b>Actor</b>	Usuario
<b>Propósito</b>	Mostrar recursos con un código postal específico.
<b>Resumen</b>	Este caso de uso filtrará todos los recursos de la categoría en base si coinciden con el código postal buscado.
<b>Tipo</b>	Primario
<b>Referencias</b>	F5, F6, NF3 y NF4.
<b>Precondiciones</b>	Iniciar búsqueda por C.P.
<b>Postcondiciones</b>	Recursos insertados que coinciden con el C.P. buscado.
<b>Flujo de eventos principal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página de recursos.</li> <li>2. El usuario inserta en el input de búsqueda el código postal deseado.</li> <li>3. El sistema muestra la información de los recursos con el código postal deseado.</li> <li>4. El sistema añade al mapa los recursos con el código postal deseado.</li> </ol>
<b>Flujo de eventos alternativo</b>	

Tabla 4.3: Especificación Caso de Uso “Filtrar Recursos por C.P.”

<b>Identificador</b>	CU-04
<b>Nombre</b>	Videojuego
<b>Actor</b>	Usuario
<b>Propósito</b>	Jugar al videojuego.
<b>Resumen</b>	Este caso de uso busca que el usuario pueda iniciar y jugar al videojuego.
<b>Tipo</b>	Primario
<b>Referencias</b>	F7, F8, F9, F10, F11, F12, F13, F14, NF6, NF7 y NF8.
<b>Precondiciones</b>	Abrir página de juego
<b>Postcondiciones</b>	
<b>Flujo de eventos principal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la página de juego.</li> <li>2. El usuario inicia el videojuego.</li> <li>3. El sistema carga la pantalla inicial y de personalización.</li> <li>4. El usuario personaliza el personaje.</li> <li>5. El usuario comienza el videojuego.</li> <li>6. El sistema carga el mapa del primer nivel.</li> <li>7. El usuario encuentra al primer organizador.</li> <li>8. El sistema carga el primer minijuego.</li> <li>9. Mientras que el usuario no obtenga la victoria en el minijuego <ol style="list-style-type: none"> <li>9.1. El usuario realizara el minijuego</li> <li>9.2. El sistema obtendrá puntuación del juego</li> <li>9.3. El sistema mostrara resultado obtenido</li> </ol> </li> <li>10. El sistema carga el mapa del segundo nivel.</li> <li>11. El usuario encuentra al segundo organizador.</li> <li>12. El sistema carga el segundo minijuego.</li> </ol>

	<ul style="list-style-type: none"> <li>13. Mientras que el usuario no obtenga la victoria en el minijuego <ul style="list-style-type: none"> <li>13.1. El usuario realizara el minijuego</li> <li>13.2. El sistema obtendrá puntuación del juego</li> <li>13.3. El sistema mostrara resultado obtenido</li> </ul> </li> <li>14. El sistema carga el mapa del tercer nivel.</li> <li>15. El usuario encuentra al tercer organizador.</li> <li>16. El sistema carga el tercer minijuego.</li> <li>17. Mientras que el usuario no obtenga la victoria en el minijuego <ul style="list-style-type: none"> <li>17.1. El usuario realizara el minijuego</li> <li>17.2. El sistema obtendrá puntuación del juego</li> <li>17.3. El sistema mostrara resultado obtenido</li> </ul> </li> <li>18. El sistema carga la pantalla final.</li> <li>19. El usuario finaliza el videojuego.</li> </ul>
<b>Flujo de eventos alternativo</b>	

Tabla 4.4: Especificación Caso de Uso “Videojuego”

## 4.2. Diagramas de actividad

Una vez determinados los casos de uso, es necesario especificar las actividades que se pueden realizar en la aplicación web. Comenzando con las páginas de recursos, que como ya se ha mencionado antes todas son iguales cambiando únicamente los recursos que se muestran, por lo que tanto el diagrama de actividad de los recursos como de información de recurso aplica a todas las páginas de recursos.

En este primer diagrama de actividad se encuentra la página de recursos donde siempre comienza cargando los recursos pasando a mostrar al mismo tiempo tanto la información de estos como su localización en el mapa. La acción de mostrar información de recursos a su vez se puede calificar como una actividad en sí, que será mostrada posteriormente. Una vez mostrados los recursos se puede filtrar al ingresar un código postal en el sistema de búsqueda, lo que volvería a empezar la carga de recursos pero solo mostrando aquellos que coincidan con el filtro.

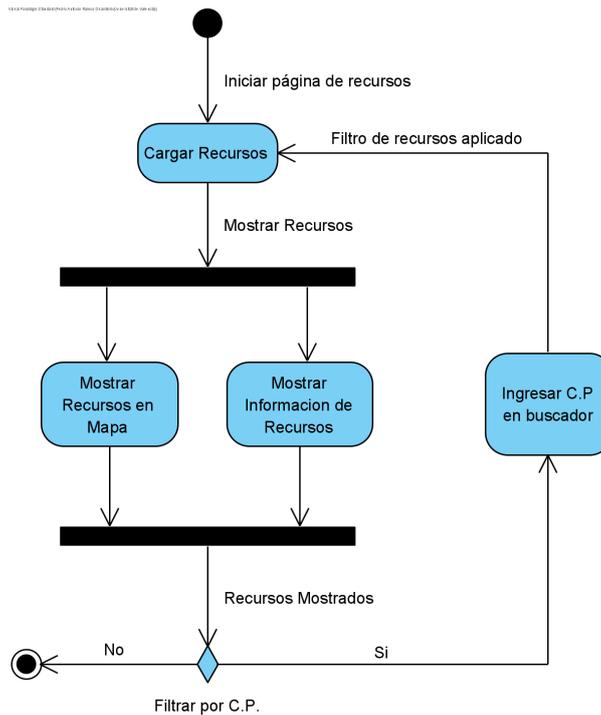


Figura 4.2: Diagrama de Actividad de Página de Recursos

Como mencionado antes, la acción de ‘mostrar información de recursos’ también se puede considerar como una actividad con diferentes acciones, que como se muestra en el siguiente diagrama consiste en cada vez que se abre el recurso este muestra la información relevante, además si este recurso tiene localización en el mapa, puedes localizar este por medio de un botón.

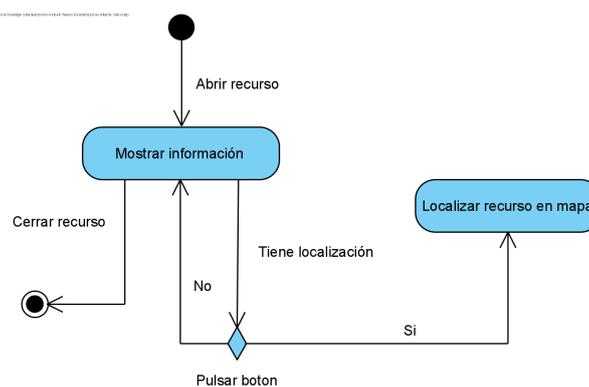


Figura 4.3: Diagrama de Actividad de Información de Recurso

Por último se encuentra las actividades que va a ejecutar nuestro videojuego. Cabe destacar como todos los niveles del videojuego mantienen la misma estructura, por lo que en el diagrama de actividad será una cadena de actividades idénticas, solo cambiando el número del nivel. El diagrama de actividad del videojuego, muestra como al iniciar el juego se abre la ventana de personalización de personaje, donde simultáneamente se puede personalizar tanto el color de piel, el estilo y color de pelo y el color de la camiseta del personaje. Una vez finalizada la personalización se comenzará la partida. Seguidamente se cargará el mapa del primer nivel, donde se podrá recorrer el mapa creado, una vez

encontrado al organizador o meta será necesario hablar o interactuar con él para así pasar al minijuego de este nivel, una vez finalizado el minijuego se verá una puntuación en el que si se ha superado, es decir, se ha obtenido una victoria, se puede continuar al siguiente nivel, en caso de no obtener la victoria se tendría que repetir el minijuego. Como mencionado antes todos los niveles tienen la misma estructura, por lo que la diferencia esta al superar el último nivel, donde al pasarlo se cargará la pantalla final, concluyendo así el videojuego.

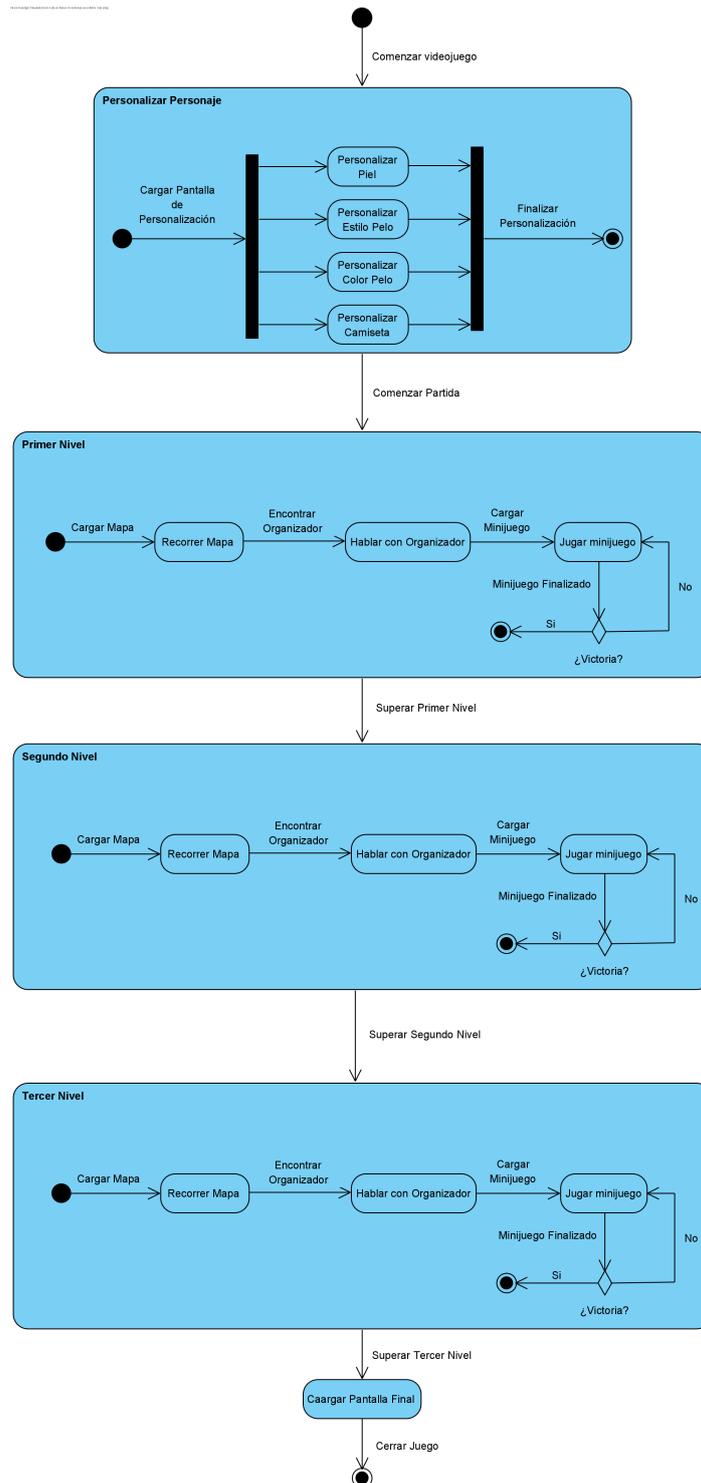


Figura 4.4: Diagrama de Actividad del Videojuego

# Capítulo 5

## Diseño del sistema

De acuerdo en el análisis del sistema realizado en el apartado anterior, se procederá a diseñar el sistema del proyecto donde se especificará las clases utilizadas en el proyecto, como los bocetos previos al diseño de la aplicación web y el videojuego como un estudio de fuentes, colores y estilos utilizados en el proyecto.

### 5.1. Diagrama de clases

Primero se definirán las clases del sistema, donde la central es la aplicación web, por un lado se encuentra los recursos, que dentro de la aplicación web pueden haber  $n$  páginas de recursos, que contara con un mapa y un conjunto de recursos, donde por cada grupo de recursos puede haber  $n$  Recurso, estos recursos tienen diferentes variables, como son nombre, titularidad, teléfono, email, web, web alternativa (alternativaWeb), código postal (cp), dirección y coordenadas, donde estas últimas se dividen en longitud y latitud dentro del mapa, y solo se puede tener un grupo de coordenadas por cada recurso.

Al otro lado, se hallan las clases relativas a la página de juego, la cual contiene un videojuego. Este videojuego contiene un personaje que tiene una piel, camiseta, color de pelo y estilo de pelo personalizables, el cual se puede desplazar por un mapa y puede hablar con el organizador de cada mapa. Dicho organizador contiene un dialogo que consiste en un sistema de texto que muestra diferentes secuencias de caracteres, y también puede cargar el minijuego del nivel. En el videojuego habrá  $n$  organizadores dependiendo de cuantos niveles existan.

Dentro del videojuego también están los niveles, mencionados en los organizadores, que cuentan con el mapa que tiene un organizador y unas texturas. Dentro de cada nivel se encuentra un minijuego, el cual se puede puntuar, repetir y continuar al siguiente nivel del videojuego. Versión ampliada en apéndice [Apéndice A.2 ].

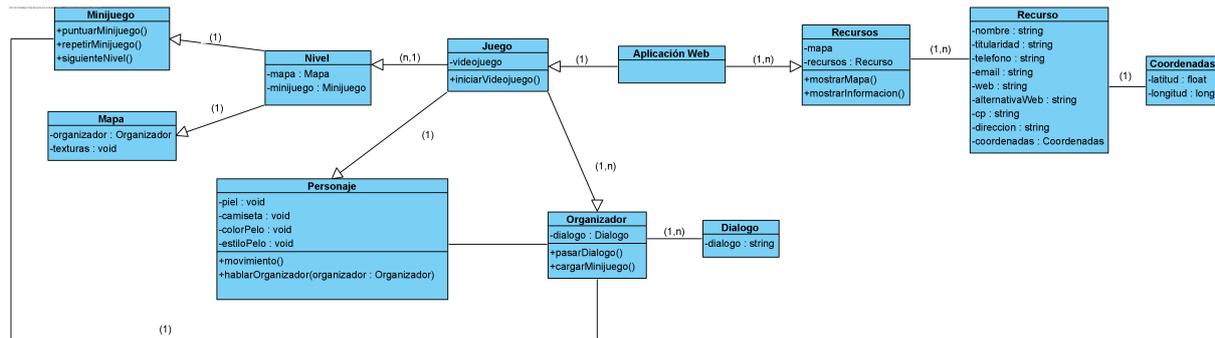


Figura 5.1: Diagrama de clases

### 5.1.1. Identificación de atributos

Clase	Atributo	Definición del atributo
<b>Recursos</b>	mapa	Mapa donde se insertarán los marcadores de los recursos.
	recursos	Listado de recursos.
<b>Recurso</b>	nombre	Nombre del recurso.
	titularidad	Titularidad del recurso.
	telefono	Teléfono/s del recurso.
	email	Correo/s electrónico/s del recurso.
	web	Enlace al sitio web del recurso.
	alternativaWeb	Enlace al sitio web alternativo del recurso.
	cp	Código Postal del recurso.
	direccion	Dirección (Calle, Numero, etc.) del recurso.
<b>Coordenadas</b>	coordenadas	Vector de coordenadas del recurso.
	latitud	Posición de la latitud de las coordenadas del recurso.
	longitud	Posición de la longitud de las coordenadas del recurso.
<b>Juego</b>	videojuego	Visualización del videojuego.
<b>Personaje</b>	piel	Identificador de la textura utilizada para la piel del personaje.
	camiseta	Identificador de la textura utilizada para la camiseta del personaje.
	colorPelo	Identificador de la textura utilizada para el color del pelo del personaje.
	estiloPelo	Identificador de la textura utilizada para el estilo del pelo del personaje.
<b>Organizador</b>	dialogo	Sistema de dialogo.
<b>Dialogo</b>	dialogo	Secuencia de cadenas de caracteres para el dialogo.
<b>Nivel</b>	mapa	Objetos y texturas del mapa del nivel.
	minijuego	Minijuego del nivel.

<b>Mapa</b>	organizador	Organizador con el dialogo del nivel que cargará el minijuego del nivel.
	texturas	Texturas utilizadas en los objetos del mapa.

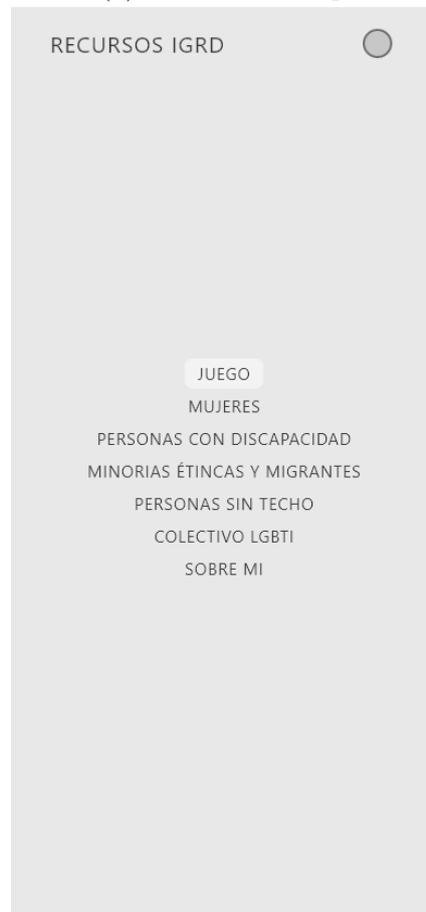
Tabla 5.1: Identificación de atributos

## 5.2. Interfaces de usuario

Una vez contempladas las clases del sistema, hay que estudiar un diseño de interfaz de usuario para tanto la aplicación web como el videojuego. Comenzando por la aplicación web, se ha buscado un diseño simple y sencillo, el cual facilite su utilización, al mismo tiempo siendo responsive, es decir, que se pueda adaptar al dispositivo desde donde se esté visualizando. Todas las páginas de la aplicación web contarán con la misma barra de navegación, que destacará la página actual, y el mismo footer con enlaces de interés. Esta barra de navegación sufrirá cambios dependiendo del dispositivo que este visualizando la aplicación.



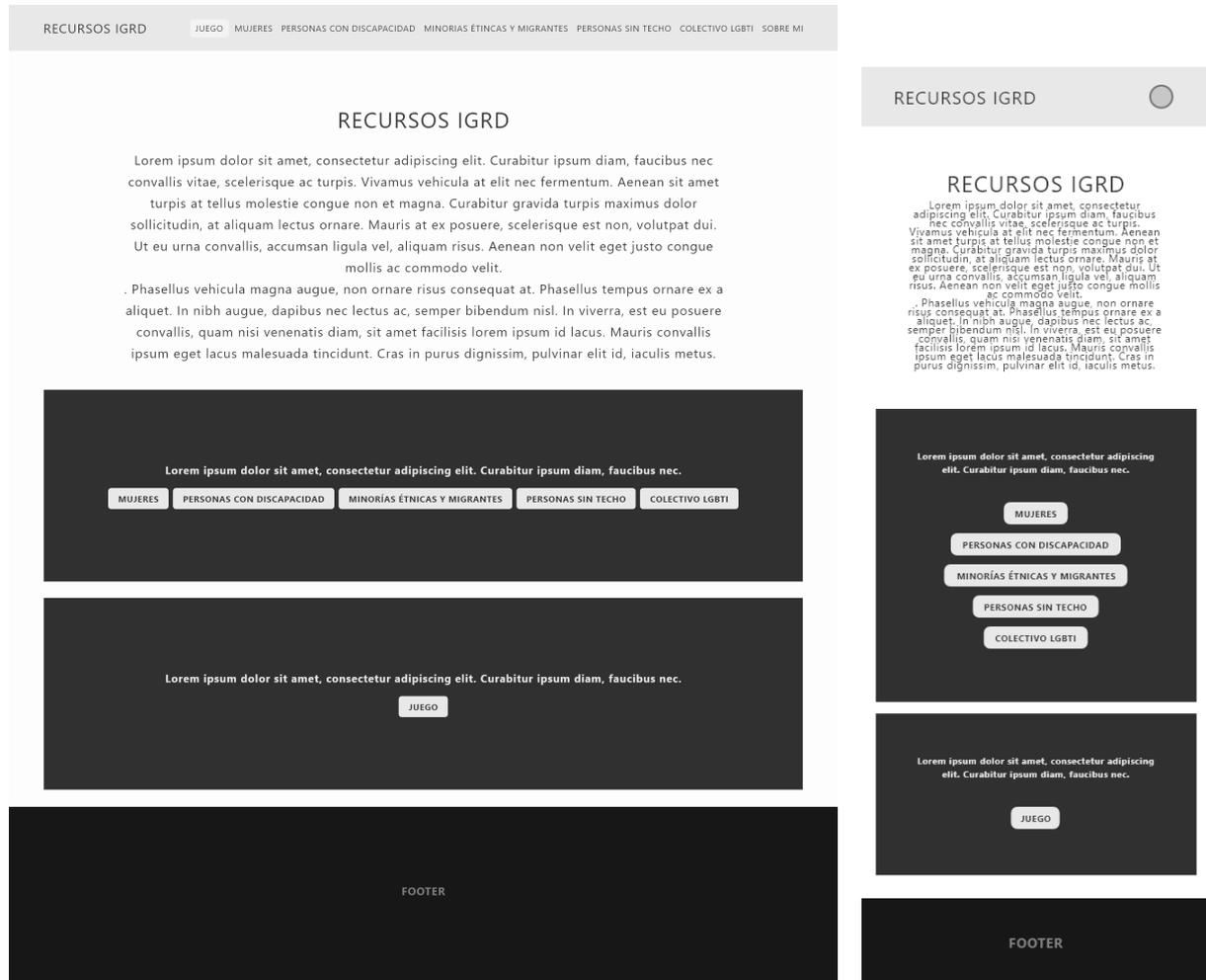
(a) Versión desktop



(b) Versión móvil

Figura 5.2: Boceto interfaz de barra de navegación

La aplicación web constará de una página de inicio, en la que se encontrará un texto describiendo la finalidad de la aplicación web al comienzo, y a continuación dos secciones, una para las páginas de recursos con botones enlace a estas y otra con otro botón enlace a la página del juego.

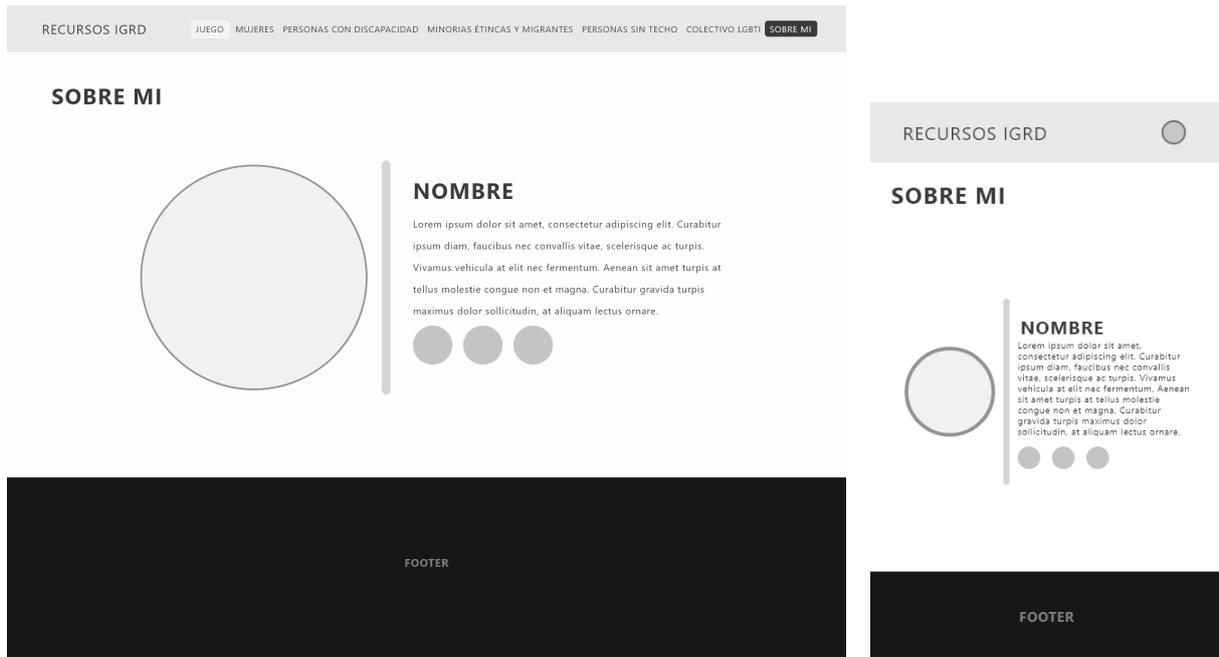


(a) Versión desktop

(b) Versión móvil

Figura 5.3: Boceto interfaz de página de inicio

También se dispondrá de una página acerca de la información del autor de la aplicación web, donde se mostrará un icono en forma de imagen, como un título con el nombre del autor, una descripción y una serie de iconos que hacen referencia a enlaces a formas de contacto o redes sociales.

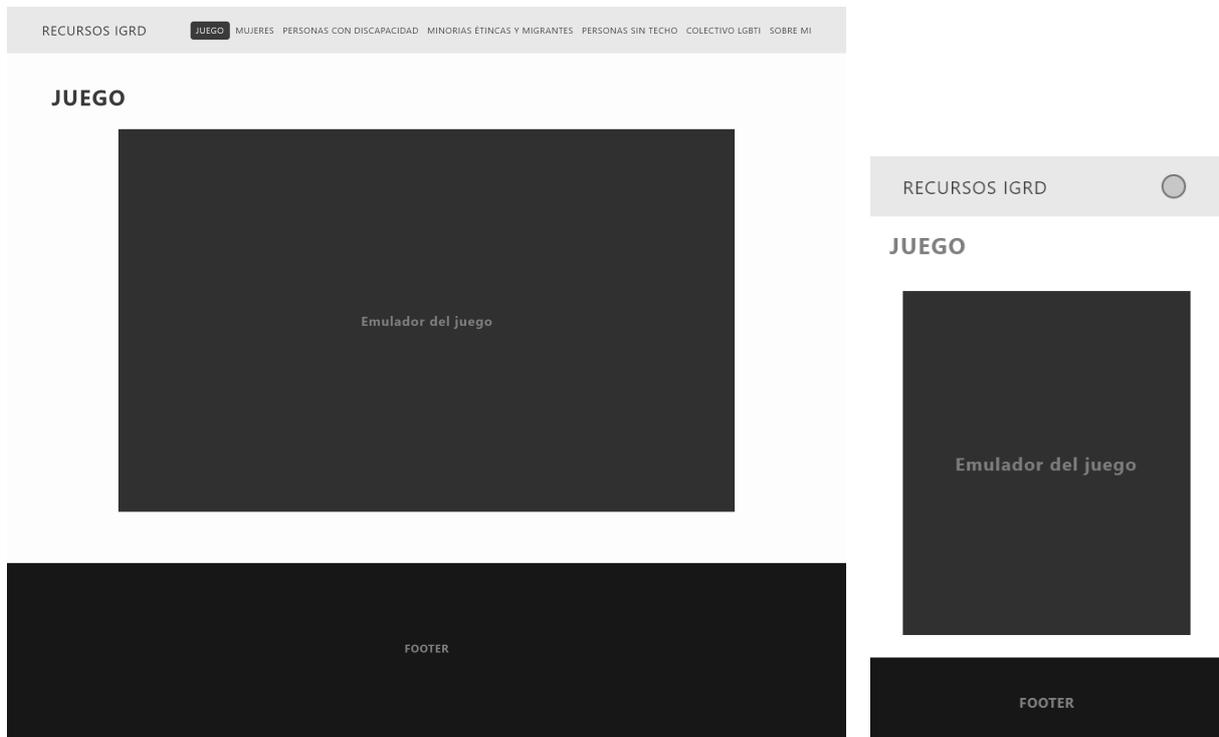


(a) Versión desktop

(b) Versión móvil

Figura 5.4: Boceto interfaz de página sobre mi

La página del juego dispone de un bloque central donde se insertará el videojuego creado, aunque este no estará disponible para en dispositivos móviles debido al compilador de este.

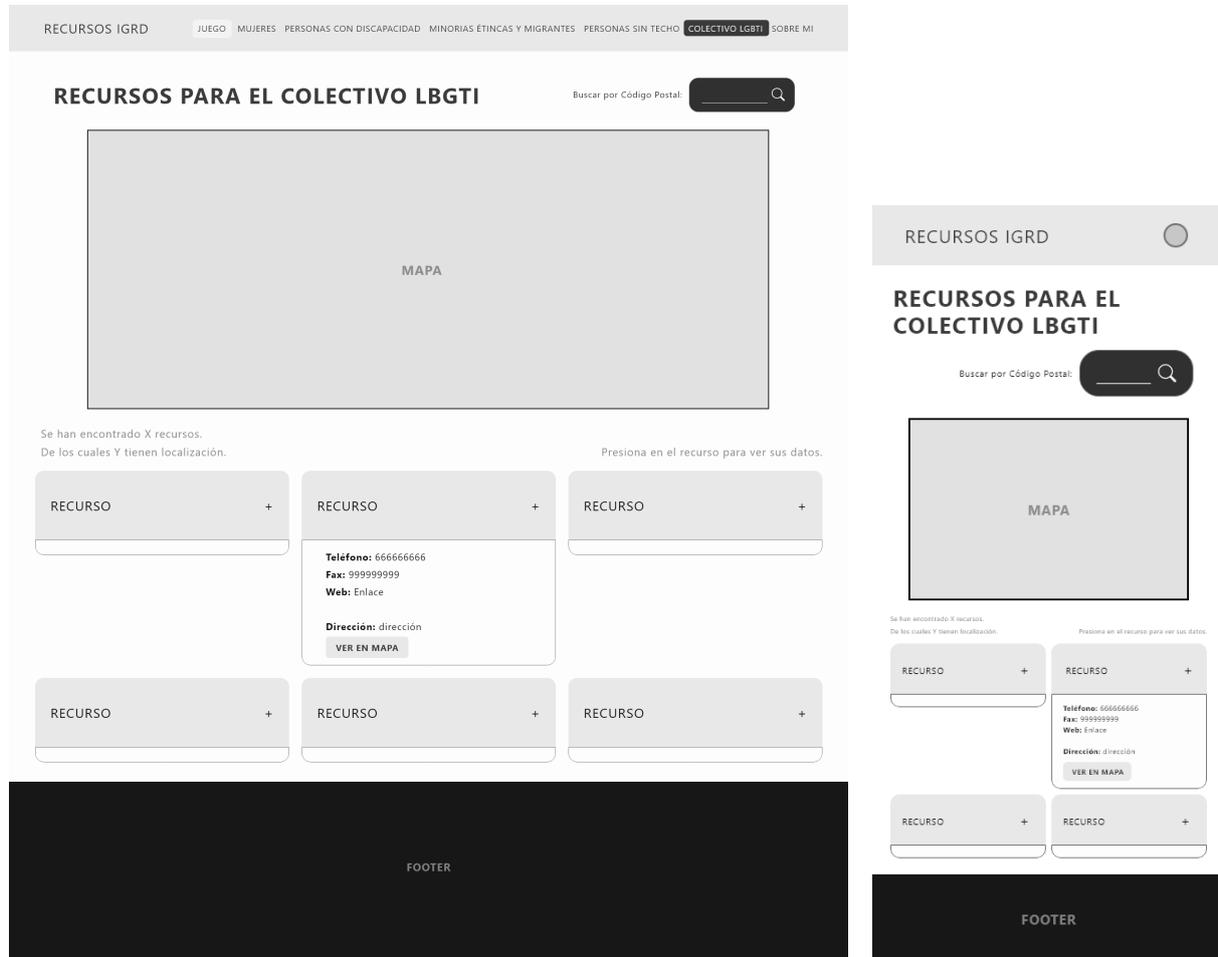


(a) Versión desktop

(b) Versión móvil

Figura 5.5: Boceto interfaz de página juego

Por último en la aplicación web se encuentran las páginas de recursos, que como se ha comentado antes, todas mantendrán la misma estructura. Dentro de estas se halla al comienzo un buscador por código postal, a continuación un mapa para localizar los recursos y seguidamente un listado de botones que representan los recursos, que al pulsarlos se extenderán mostrando toda la información relevante de los recursos.



(a) Versión desktop

(b) Versión móvil

Figura 5.6: Boceto interfaz de página de recursos

Finalmente, se ha diseñado las interfaces de usuario del videojuego, donde se han mostrado tanto la interfaz inicial, como el menú de personalización, como el tablero de los minijuegos.

La interfaz inicial consta de tres botones, uno para activar el panel de controles, otro para activar el panel de créditos y otro para comenzar la partida. También se compone de un logo y de un texto de bienvenida donde se explica cómo ganar el videojuego.

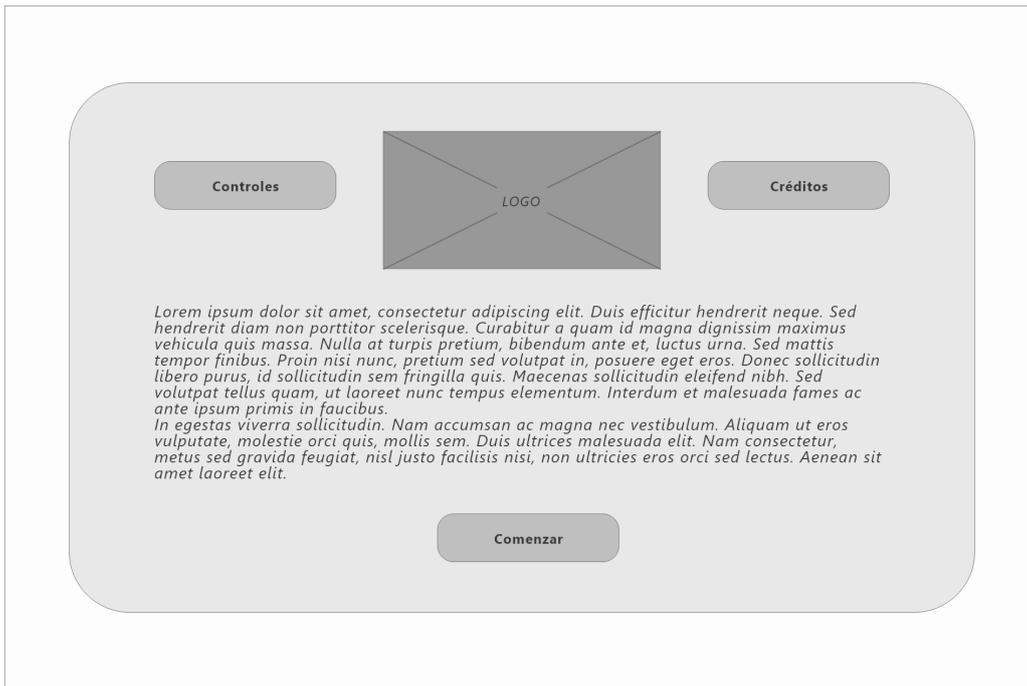


Figura 5.7: Boceto interfaz de inicio videojuego

La pantalla de personalización de personaje constará de cuatro secciones de botones, donde se puede mostrar el siguiente o el anterior elemento. Estas cuatro secciones personalizarán el color de piel, el estilo y color de pelo, y el color de la camiseta. Y por último un botón abajo que dará comienzo a la partida.

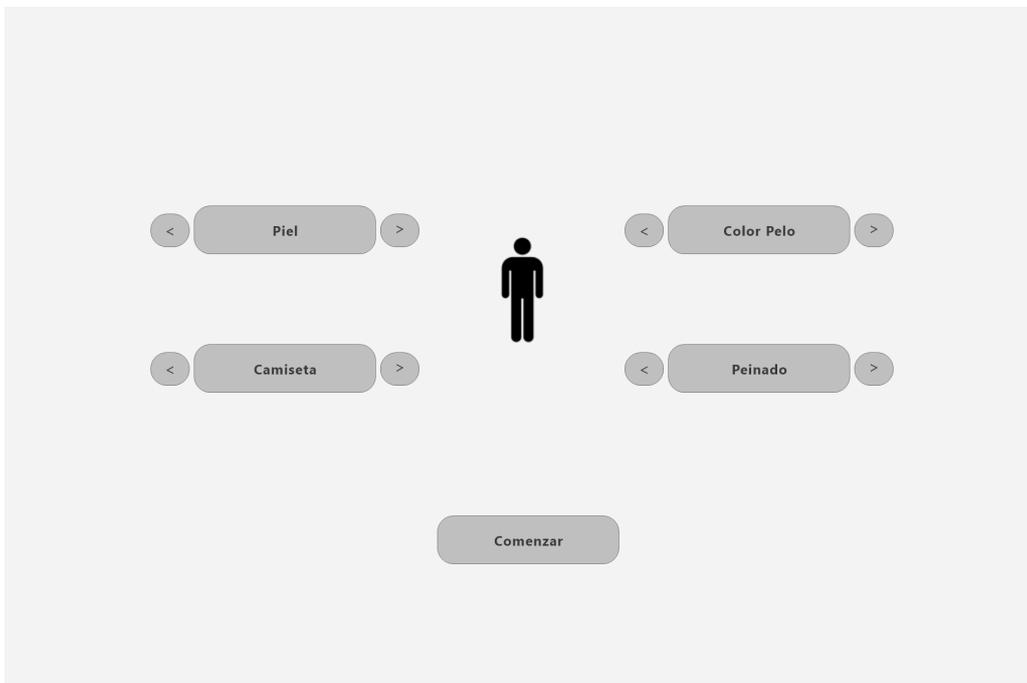


Figura 5.8: Boceto interfaz de personalización de personaje videojuego

Finalmente, la pantalla del tablero de minijuego, que consta de una sección con el tiempo restante, los intentos utilizados y el número de aciertos en el minijuego. Seguidamente un bloque donde se distribuirá el minijuego, este se modificara dependiendo del minijuego que se vaya a mostrar, en el primero que consta de un Quiz este mostrará un panel con la pregunta, y cuatro botones para las diferentes opciones, para el segundo minijuego que trata del juego de cartas de memoria, se distribuirá en un solo panel que cubre todo el bloque y dentro de este se distribuirán en tamaños iguales todas las cartas. Finalmente, para el tercer minijuego, es decir, el juego 'I am can' tratará de un panel con un texto y debajo tres botones que corresponderán a 'Niños', 'Niñas' y 'Ambos' .



Figura 5.9: Boceto interfaz de minijuego general videojuego

### 5.3. Estilo de interfaz

Cuando se habla de estilo se puede identificar la paleta de colores utilizada en el diseño, para este proyecto se ha querido representar una extensión del sitio web del ayuntamiento de Valencia, por lo que el color principal será el amarillo. Para hacer más accesible la aplicación web se ha seleccionado el negro y el blanco como colores secundarios, formando ellos el color de letra y el color de fondo respectivamente. Resultado así la paleta de colores utilizada en el proyecto.

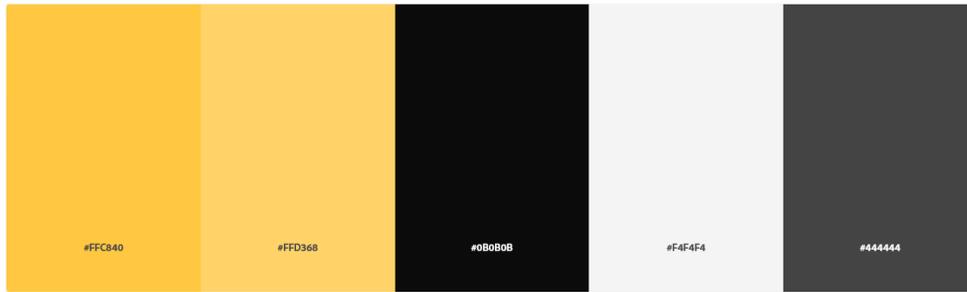


Figura 5.10: Paleta de colores

A parte de la paleta de colores, dentro del estilo de la interfaz también se encuentran las fuentes utilizadas. Para la aplicación web se utilizará la fuente ‘PT Sans’ de ParaType [34] ya que al ser una fuente sin serifa o gracia es más fácil leerla por lo tanto haciendo la lectura de los recursos accesible para todos los usuarios.

abcdefghijklmnopqrstuvwxyZ ABCDEFGHIJKLMNOPQRSTUVWXYZ  
 1234567890.,; ' " (!?) +-\*/=

Figura 5.11: Fuente ‘PT Sans’

Finalmente la fuente utilizada para el videojuego es ‘Minecraftia’ de Andrew Tyler [35] debido a que los gráficos del videojuego están inspirados en los videojuegos de pixeles así ayudando al ambiente general del videojuego.

ABCDEFGHIJKLMN0PQRSTUVWXYZ abcdefghijklmnopqrstuvwxyZ 0123456789  
 ABCDEFGHIJKLMN0PQRSTUVWXYZ abcdefghijklmnopqrstuvwxyZ 0123456789

Figura 5.12: Fuente ‘Minecraftia’



# Capítulo 6

## Implementación del sistema

Una vez se ha analizado y diseñado el sistema es momento de implementarlo, donde se hará un repaso de las tecnologías utilizadas, los lenguajes de programación y librerías utilizadas, revisando fragmentos del código importantes como la recapitulación de la resolución de aquellos problemas más complejos. Además se mostrará una estructura el proyecto, mostrando la división en ficheros.

### 6.1. Herramientas de desarrollo

La implementación de este proyecto se ha repartido entre videojuego y aplicación web. Para desarrollar la interfaz del videojuego se ha utilizado Unity, y para las texturas del videojuego se ha utilizado Procreate y Photoshop. Se ha utilizado el programa Visual Studio Code para desarrollar tanto interfaz como funcionalidad de la aplicación web y la funcionalidad del videojuego. Los lenguajes de programación empleados han sido el lenguaje orientado a objetos C# para la funcionalidad del videojuego, y HTML, CSS y JavaScript para la aplicación web.

Dentro de la implementación del desarrollo del videojuego se han utilizado las siguientes librerías:

- `UnityEngine.SceneManagement`: Utilizada para manejar el cambio de escenas o niveles a partir de scripts.
- `UnityEngine.UI`: Utilizada para modificar la interfaz de usuario (UI) a partir de scripts.
- `UnityEngine.UIElements`: Utilizada para transformar los elementos de la UI a partir de scripts.

Y las librerías utilizadas para implementar la interfaz y el funcionamiento de la aplicación web son:

- `bootstrap-icons.css`: Empleada para utilizar iconos en formato SVG dentro de la aplicación web.
- `bootstrap.min.css`: Se utiliza para ayudar la implementación de la arquitectura y el estilo de la aplicación web, con la que la aplicación web se adapte a cualquier dispositivo, independientemente de su resolución.

- `font-awesome.css`: Utilizada para añadir iconos en formato SVG dentro de la aplicación web.
- `leaflet.js`: Necesitada para la implementación de mapas interactivos en la aplicación web.
- `leaflet.css`: Empleada para dotar de estilo a los mapas interactivos de la aplicación web.
- `leaflet.markercluster`: Para añadir iconos a los marcadores del mapa interactivo de la aplicación web.

Cabe destacar que todas las librerías han sido añadidas de forma remota, por lo que los archivos de estas no han sido almacenados en el proyecto, si no que se han obtenido de sus diferentes repositorios remotos.

## 6.2. Texturas

Para las texturas de los mapas de los niveles se han utilizado el tileset ‘Pokemon Esmeralda de GameBoy Adanvanced’ recreado por Heartlessdragon [36], asimismo se han utilizado las texturas del paquete ‘Zelda-like tileset and sprites’ por ArMM1998 [37] como base para los Sprites del personaje.

Los mapas de las localizaciones disponían de edificios que no se podían reconstruir a partir de los tileset ya mencionados, por lo que se han creado a partir de Procreate y Photoshop los siguientes sprites para los edificios del Ayuntamiento de Valencia del primer nivel, el Palacio de las Artes del segundo nivel y el Parque Gulliver en el tercer nivel.

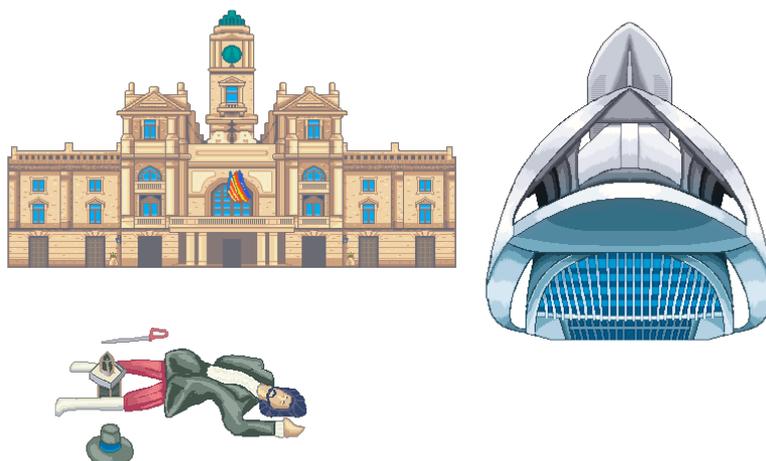


Figura 6.1: Texturas de Ayuntamiento, Palacio de las Artes y Parque Gulliver pixelados

Para la personalización del personaje, como se ha mencionado antes, se ha utilizado un Sprite de ‘Zelda-like tileset and sprites’ como base para el personaje, el cual se ha editado para poder diferentes tonos de piel, como se ha utilizado para crear ropa de diferente color y estilos de peinado de diferentes colores.

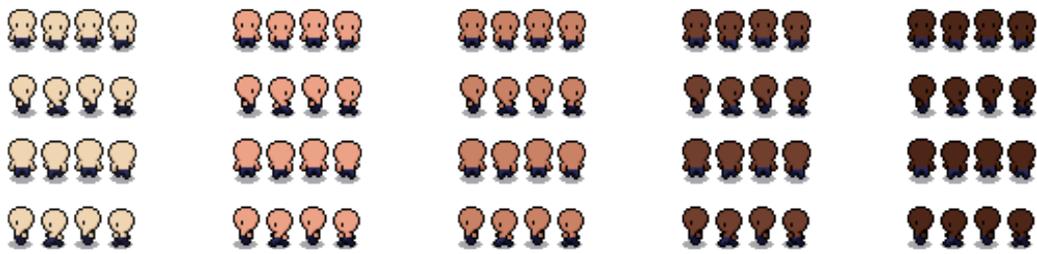


Figura 6.2: Sprites de Piel del Personaje



Figura 6.3: Sprites de Camiseta del Personaje



Figura 6.4: Sprites de Peinados del Personaje

Finalmente para el minijuego de memoria, se ha buscado crear un juego de cartas de buscar parejas, donde se buscan la combinación de profesiones, donde hay que buscar a la mujer y al hombre que ejercen la misma profesión. Para ello se han creado diferentes sprites para tanto la parte frontal como la trasera de la carta. Se han utilizado profesiones como las de profesor, ingeniero, cocinero, empresario, etc. Dando como resultado las siguientes cartas:

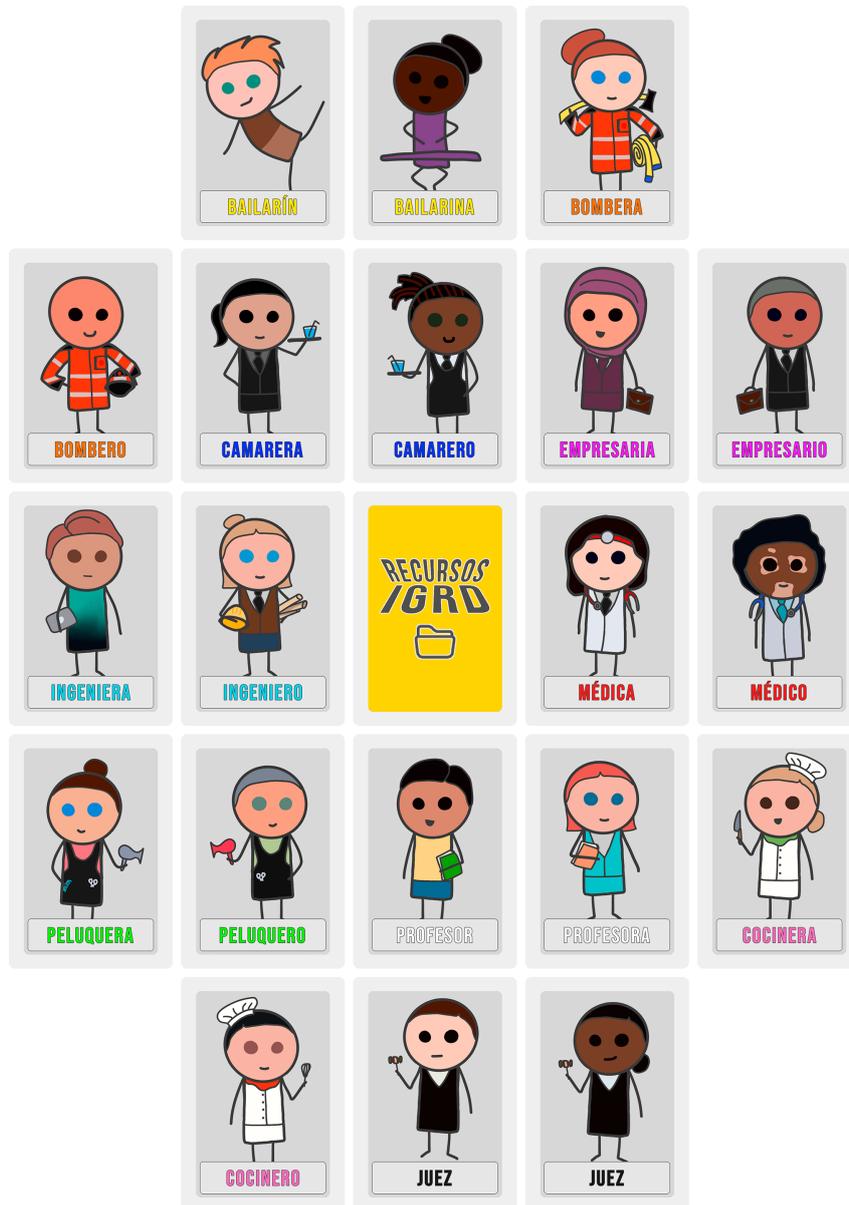


Figura 6.5: Sprites para minijuego de memoria de cartas

### 6.3. Implementación del videojuego

A la hora de implementar el videojuego han surgido diferentes inconvenientes como ha sido la personalización del personaje, el movimiento del personaje dentro de los mapas, el sistema de dialogo, y el sistema de puntuación de los minijuegos.

### 6.3.1. Estructura del videojuego

Los ficheros empleados en la implementación del juego se estructura tal que, la carpeta ‘Animations’ que guarda todas las secuencias de animación del personaje, del organizador y del dialogo, como los controladores de estos, la carpeta ‘Cards’ almacena los sprites de las cartas para el segundo minijuego, la carpeta ‘Fonts’ almacena las fuentes utilizadas, ‘gtx’ guarda los sprites, imágenes y texturas de los mapas de los niveles, la carpeta ‘Prefabs’ recoge los objetos que se repiten en diferentes escenas, la carpeta ‘Scenes’ guarda las escenas utilizadas, la carpeta ‘Scripts’ almacena todos los archivos C# que dan funcionalidad al videojuego, la carpeta ‘Sprites’ almacena los sprites utilizados para el personaje, y la carpeta ‘Tiles’ almacena los bloques que componen los mapas de los niveles, que son generados automáticamente al componer el mapa.

Tipo de fichero	Número de ficheros
Secuencias de animación (.anim)	227
Controladores (.controller)	31
Imágenes y Sprites (.png .jpg .svg)	88
Fuentes (.ttf)	1
Materiales (.mat)	2
Prefabs (.prefab)	9
Escenas (.unity)	8
Scripts (.cs)	23
Assets ó Tiles (.asset)	9334

Tabla 6.1: Ficheros del videojuego

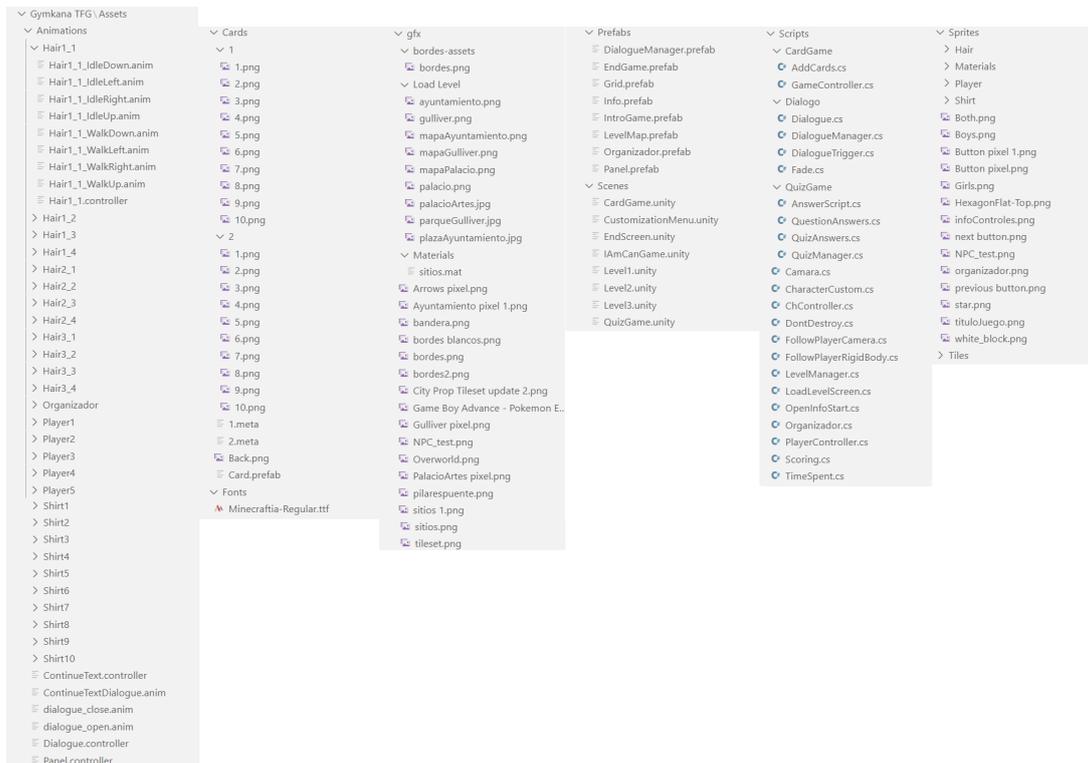


Figura 6.6: Estructura del videojuego

### 6.3.2. Personalización de personaje

Comenzando con la personalización, se ha utilizado un sistema donde se almacenan todos los sprites de personalización en objetos diferentes, es decir, cada objeto tiene un sprite con su animación de movimiento correspondiente, todos estos objetos son añadidos a la jerarquía del personaje por medio de tres carpetas, una para piel, otra para camiseta y otra para peinado. Con esto se ha creado un sistema donde a partir de las carpetas contenedoras de los objetos se ha añadido o quitado la visibilidad del objeto en función de si es la opción seleccionada.

Estas opciones se seleccionan por medio de dos botones para seleccionar el siguiente objeto o el objeto previo para cada uno de los tipos de objetos de personalización. Todos los objetos se distinguirán por un índice, de tal modo que la primera piel correspondería a 'Player1' y la piel número 3 correspondería a "Player3". Los peinados contienen dos índices uno para el estilo y otro para el peinado, por lo que si queremos el peinado con el segundo estilo pero con el cuarto color el objeto correspondiente será "Hair2\_4".

Con todo esto se nos quedaría el siguiente código, el cual solo muestra la selección de camiseta, pero todos los objetos se seleccionan de la misma forma:

```
1  void Update()
2  {
3      //Shirt Modify
4      shirtFile = "Shirt" + (shirtIndex + 1);
5      for (int i = 0; i < shirt.Length; i++)
6      {
7          if (shirt[i].name == shirtFile)
8          {
9
10             shirt[i].gameObject.SetActive(true);
11             shirt[i].GetComponent<Animator>().enabled = true;
12         }
13         else
14         {
15             shirt[i].GetComponent<Animator>().enabled = false;
16             shirt[i].gameObject.SetActive(false);
17         }
18     }
19 }
20
21 public void NextShirt()
22 {
23     //GetComponent<PlayerController>.GetComponent<Animator>().SetFloat("moveX",
24     -1);
25     if (shirtIndex < shirt.Length - 1)
26     {
27         shirtIndex++;
28     }
29     else
30     {
31         shirtIndex = 0;
32     }
33 }
34 public void PreviousShirt()
35 {
36     if (shirtIndex > 0)
```

```
36     {
37         shirtIndex--;
38     }
39     else
40     {
41         shirtIndex = shirt.Length - 1;
42     }
43 }
44
45 //Del fichero CharacterCustom.cs
```

Código 6.1: Código para la personalización del personaje

### 6.3.3. Movimiento del personaje

Siguiendo con el segundo de los inconvenientes, el movimiento del personaje tiene que al mismo tiempo que mover el objeto principal del personaje, es decir, aquel que almacena todos los sprites, también tiene que animar todos los sprites (piel, camiseta y peinado) dependiendo de la tecla de dirección que pulsemos.

Por lo que se han creado dos funciones similares, una para el movimiento del personaje y otra para las animaciones de los sprites. Primero si vemos la función para el movimiento se obtiene el valor del input al pulsar las teclas desplazamiento, primero se bloqueará el movimiento diagonal, y en el caso de que el input no sea 0, es decir, se presione una tecla se almacena la posición a la que se va a dirigir el personaje, si se puede desplazar a esa dirección entonces moverá el personaje a esa dirección.

La diferencia entre las funciones de movimiento del personaje y de animación de los sprites está dentro de que hace con los valores del input, ya que en movimiento se emplea para desplazarse y en animación se emplean como variable para el animator del sistema de animación el cual ejecuta la animación del paso a X dirección dependiendo de la tecla pulsada. Teniendo en cuenta todo lo anterior, se obtiene el siguiente código:

```
1 //Movimiento del personaje
2 void Update()
3 {
4     if (!isMoving)
5     {
6         input.x = Input.GetAxisRaw("Horizontal");
7         input.y = Input.GetAxisRaw("Vertical");
8
9         //elimina movimiento diagonal
10        if (input.x != 0)
11            input.y = 0;
12
13        if (input.y != 0)
14            input.x = 0;
15
16        if (input != Vector2.zero)
17        {
18            var targetPos = transform.position;
19            targetPos.x += input.x;
20            targetPos.y += input.y;
21
22            if (IsWalkable(targetPos))
```

```

23         StartCoroutine(Move(targetPos));
24     }
25 }
26 }
27 IEnumerator Move(Vector3 targetPos)
28 {
29     isMoving = true;
30     while ((targetPos - transform.position).sqrMagnitude > Mathf.Epsilon)
31     {
32         transform.position = Vector3.MoveTowards(transform.position, targetPos,
33             moveSpeed * Time.deltaTime);
34         yield return null;
35         isMoving = false;
36     }
37     transform.position = targetPos;
38     isMoving = false;
39 }
40 private bool IsWalkable(Vector3 targetPos)
41 {
42     if (Physics2D.OverlapCircle(targetPos, 0.1f, solidObjectsLayer | interactLayer)
43         != null)
44     {
45         return false;
46     }
47     return true;
48 }
49
50 //Del fichero ChController.cs

```

Código 6.2: Código del movimiento del personaje

```

1 //Animacion de los Sprites
2 void Update()
3 {
4     if (!isMoving)
5     {
6         input.x = Input.GetAxisRaw("Horizontal");
7         input.y = Input.GetAxisRaw("Vertical");
8
9         //elimina movimiento diagonal
10        if (input.x != 0)
11            input.y = 0;
12
13        if (input.y != 0)
14            input.x = 0;
15
16        if (input != Vector2.zero)
17        {
18            var targetPos = transform.position;
19            targetPos.x += input.x;
20            targetPos.y += input.y;
21
22            GetComponent<Animator>().SetFloat("moveX", input.x);
23            GetComponent<Animator>().SetFloat("moveY", input.y);
24
25            StartCoroutine(Move(targetPos));
26        }

```

```

27     }
28     GetComponent<Animator>().SetBool("isMoving", isMoving);
29 }
30
31 IEnumerator Move(Vector3 targetPos)
32 {
33     isMoving = true;
34     while ((targetPos - transform.position).sqrMagnitude > Mathf.Epsilon)
35     {
36         transform.position = Vector3.MoveTowards(transform.position, targetPos,
37             moveSpeed * Time.deltaTime);
38         yield return null;
39         isMoving = false;
40     }
41     transform.position = targetPos;
42     isMoving = false;
43 }
44 //Del fichero PlayerController.cs

```

Código 6.3: Código de la animación de sprites del personaje

### 6.3.4. Sistema de dialogo

El sistema de dialogo se basa en un área de texto que tiene que ser modificada al pulsar una tecla y así dando lugar a una secuencia de oraciones. Para que este sistema funcione se ha necesitado un trigger que ejecute el dialogo, el cual será los organizadores, por lo que el primer problema se encuentra en cómo se consigue que el dialogo se dispare cuando el personaje este al lado del organizador con dirección hacia este y además necesite ser presionada una tecla para ello.

Para solucionar este primer problema se han creado tanto el personaje como los organizadores como objetos que pueden colisionar entre sí, esto quiere decir, que el momento en el que ambos choquen se podrá ejecutar el sistema de dialogo, para comprobar que el personaje está en dirección hacia el organizador se obtienen los parámetros de ‘moveX’ y ‘moveY’ que corresponden con los valores de la animación del personaje, por lo que si la posición del personaje actual sumada a la dirección da lugar a la posición del organizador, y se ha presionado, en este caso, la tecla ‘Z’ entonces se ejecutara el sistema de dialogo.

```

1  void Update()
2  {
3      if (Input.GetKeyDown(KeyCode.Z) && !talking)
4      {
5          Interact();
6      }
7  }
8
9  void Interact()
10 {
11     var faceDir = new Vector3(GetComponent<Animator>().GetFloat("moveX"),
12         GetComponent<Animator>().GetFloat("moveY"));
13     var interactPos = transform.position + faceDir;
14
15     var collider = Physics2D.OverlapCircle(interactPos, 0.3f, interactLayer);
16     if(collider != null)

```

```

16     {
17         Debug.Log("ChController Interact --> " + collider.name);
18         collider.GetComponent<DialogueTrigger>().TriggerDialogue();
19     }
20 }
21
22 //Del fichero ChController.cs

```

Código 6.4: Código del disparador del dialogo

Una vez ya se ha implementado el disparador es necesario implementar el sistema de dialogo, por lo que se ha implementado un DialogueManager que inicia el dialogo, puede mostrar la siguiente sentencia u oración y finaliza el dialogo. Para iniciar el dialogo primero da paso a la animación de entrada del dialogo, después borra cualquier instancia restante de cualquier otro dialogo previo y finalmente recoge todas las sentencias que se han añadido por medio del organizador y muestra la primera sentencia.

Para mostrar la siguiente sentencia del dialogo se utiliza la tecla 'Z', entonces una vez presionada la tecla la función saca la siguiente oración de la cola de sentencias, borrándola también de esta cola, dando paso a una función que da el efecto de máquina de escribir, la cual muestra carácter por carácter la sentencia completa. Una vez finalizada la sentencia se puede volver a presiona la tecla 'Z' para pasar a la siguiente sentencia, donde en el caso de que ya no queden sentencias en la cola, pasaría a la función de cierre de dialogo.

Dentro de la función de cierre se llama al animador del dialogo para que ejecute la animación de cierre, por lo que una vez terminado el dialogo se pasará a la siguiente escena, que coincide con el minijuego del nivel.

```

1  void Start()
2  {
3      sentences = new Queue<string>();
4  }
5
6  void Update()
7  {
8      //Pasa a la siguiente frase al presionar Z
9      if (Input.GetKeyDown(KeyCode.Z) && !isTalking)
10     {
11         DisplayNextSentence();
12     }
13 }
14
15 public IEnumerator StartDialogue(Dialogue dialogue, bool org)
16 {
17     yield return new WaitForEndOfFrame();
18
19     organizador = org;
20     animator.SetBool("open", true);
21     player.GetComponent<ChController>().talking = true;
22
23     sentences.Clear();
24     foreach (string sentence in dialogue.sentences)
25     {
26         sentences.Enqueue(sentence);
27     }
28
29     DisplayNextSentence();

```

```
30 }
31
32 public void DisplayNextSentence()
33 {
34     if (sentences.Count == 0)
35     {
36         StartCoroutine(EndDialogue());
37         return;
38     }
39
40     string sentence = sentences.Dequeue();
41     StopAllCoroutines();
42     StartCoroutine(TypeSentence(sentence));
43     Debug.Log(sentence);
44 }
45
46 //Efecto de maquina de escribir (una letra cada vez)
47 IEnumerator TypeSentence(string sentence)
48 {
49     isTalking = true;
50     dialogueText.text = "";
51     foreach (char letter in sentence.ToCharArray())
52     {
53         dialogueText.text += letter;
54         yield return null;
55     }
56     isTalking = false;
57 }
58
59 IEnumerator EndDialogue()
60 {
61     animator.SetBool("open", false);
62
63     yield return new WaitForEndOfFrame();
64     player.GetComponent<ChController>().talking = false;
65
66     if (organizador)
67     {
68         levelManager.NextScene();
69         organizador = false;
70     }
71 }
72
73 //Del fichero DialogueManager.cs
```

Código 6.5: Código del sistema de dialogo

### 6.3.5. Sistema de puntuación de minijuegos

El sistema de puntuación para los minijuegos se basa en puntuación de tres estrellas, por lo que para que el videojuego pueda continuar hay que obtener tres estrellas en cada minijuego. En el plano donde se muestra la puntuación al finalizar el minijuego también se encuentra un texto que en el caso de no haber llegado a las tres estrellas te indica como llegar. Para obtener la puntuación en el manager de cada minijuego se ha necesitado almacenar una variable con la puntuación obtenida. Para el primer y último minijuego tiene una división diferente que para el segundo minijuego.

Empezando por cómo se almacena la puntuación en el primer y último minijuego, es decir, un quiz y el juego ‘I am can’, donde hay que seleccionar una opción o respuesta entre varias por medio de botones. En el manager de estos juegos la puntuación deriva dependiendo de cuantos aciertos tengas en comparación de cuantas preguntas haya.

```

1  if (preguntasRespondidas == totalPreguntas)
2  {
3      EnabledButtons();
4      timer.stopTime();
5      endGame.GetComponent<Scoring>().anotation.text = timer.getTimeSpent();
6      endGame.GetComponent<Scoring>().ScorePoints(aciertos, totalPreguntas);
7      endGame.SetActive(true);
8  }
9
10 //Del fichero QuizManager.cs

```

Código 6.6: Código para puntuación de los minijuegos 1 y 3

Por otro lado se encuentra el almacenaje de puntuación del segundo minijuego, es decir, el minijuego de cartas de memoria. Como en este minijuego trata de buscar parejas se tendrá en cuenta a parte de si se ha encontrado las parejas también el número de intentos que se han utilizado. Como hay un total de 10 parejas de cartas se ha determinado que el número máximo de intentos para conseguir tres estrellas sean 24, para dos estrellas 30 intentos y para una estrella 36.

```

1  void CheckGameFinished()
2  {
3      correctGuesses++;
4      matchesTest.text = correctGuesses + "/" + gameGuesses;
5      float sentGuesses = countGuesses;
6
7      if (correctGuesses == gameGuesses)
8      {
9          timer.stopTime();
10         Debug.Log("Finished!" + " It took " + countGuesses + " guesses");
11
12         if(countGuesses <= 24)
13         {
14             sentGuesses = correctGuesses;
15             endGame.GetComponent<Scoring>().anotation.text = "Enhorabuena ganaste!"
16             ;
17         }
18         else if(countGuesses > 24 && countGuesses < 30)
19         {
20             sentGuesses = correctGuesses * 0.66f;
21             endGame.GetComponent<Scoring>().anotation.text = "Para ganar tienes que
22             usar 24 intentos o menos";
23         }
24         else if(countGuesses > 30 && countGuesses < 36)
25         {
26             sentGuesses = correctGuesses * 0.33f;
27             endGame.GetComponent<Scoring>().anotation.text = "Para ganar tienes que
28             usar 24 intentos o menos";
29         }
30     }
31     else
32     {
33         sentGuesses = correctGuesses * 0.33f;
34         endGame.GetComponent<Scoring>().anotation.text = "Para ganar tienes que
35         usar 24 intentos o menos";
36     }
37 }

```

```

29         sentGuesses = 0;
30         endGame.GetComponent<Scoring>().anotation.text = "Para ganar tienes que
           usar 24 intentos o menos";
31     }
32
33     if (timer.secondLeft <= 0)
34     {
35         sentGuesses = 0;
36         endGame.GetComponent<Scoring>().anotation.text = "Te quedaste sin
           tiempo!";
37     }
38
39     endGame.GetComponent<Scoring>().ScorePoints(sentGuesses, gameGuesses);
40
41     endGame.SetActive(true);
42 }
43 }
44
45 //Del fichero GameController.cs

```

Código 6.7: Código para puntuación del minijuego 2

También se cuenta con la opción de que el jugador no consiga terminar el minijuego a tiempo, ya que se dispone de un cronometro donde si llegas a cero se terminará la partida del minijuego automáticamente, dando como resultado cero estrellas.

```

1 private void NoTimeLeft()
2 {
3     endGame.transform.GetComponent<Scoring>().anotation.text = "Te quedaste sin
           tiempo!";
4     endGame.transform.GetComponent<Scoring>().ScorePoints(0, gameGuesses);
5     endGame.SetActive(true);
6 }
7
8 //De los ficheros QuizManager.cs y GameManager.cs

```

Código 6.8: Código de cronometro acabado

Por último, hay que observar cómo se maneja la puntuación, una vez se envía la información a la función de puntuación se distribuye en porcentajes, donde la proporción entre los puntos conseguidos y los puntos totales superen el 30% se conseguirá una estrella, cuando se supere el 60% serán dos estrellas, y finalmente cuando los puntos conseguidos sean iguales a los puntos totales se conseguirá la tercera estrella, activando así el botón para continuar al siguiente nivel.

```

1 public void ScorePoints(float achievedPoints, float totalPoints)
2 {
3     float percentage = achievedPoints * 100 / totalPoints;
4     retryButton.SetActive(true);
5
6     //Supera el 33% de la puntuacion maxima gana una estrella
7     if (percentage > 30)
8     {
9         star1.GetComponent<Image>().color = new Color(0,225,248);
10        Debug.Log("Estrella1");
11    }

```

```
12
13 //Supera el 66% de la puntuacion maxima gana dos estrellas
14 if (percentage > 60)
15 {
16     star2.GetComponent<Image>().color = new Color(0, 225, 248);
17     Debug.Log("Estrella2");
18 }
19
20 //Supera el 100% de la puntuacion maxima gana tres estrellas y pasa al
    siguiente nivel
21 if (percentage >= 100)
22 {
23     star3.GetComponent<Image>().color = new Color(0, 225, 248);
24     Debug.Log("Estrella3");
25     continueButton.SetActive(true);
26 }
27 }
28 }
29
30 //Del fichero Scoring.cs
```

Código 6.9: Código de Sistema de puntuación

## 6.4. Implementación de ficheros con geolocalización

Para poder insertar los recursos a la aplicación web ha sido necesario el uso de ficheros JSON, como estos ficheros necesita tener unas coordenadas se ha utilizado el formato GeoJSON, con el que se puede añadir a parte de información relevante como el nombre, teléfono, etc. También se puede añadir un apartado que geolocaliza un punto por medio de coordenadas, para así posteriormente poder añadirlos al mapa. Toda la información de los recursos ha sido obtenida por los sitios web oficiales de cada recurso como desde el sitio web de Infociedad del Ayuntamiento de Valencia.

A partir de esa información y al utilizar el formato GeoJSON se ha conseguido una plantilla de datos, donde se separa según el tipo, donde 'Feature' es un recurso que contiene una localización, mientras que 'Service' es aquel recurso que no tiene localización.

```
1 {
2     "type": "Feature",
3     "properties": {
4         "name": "name",
5         "ownership": "ownership",
6         "phone": [
7             "phone"
8         ],
9         "fax": [
10            "fax"
11        ],
12        "email": [
13            "email"
14        ],
15        "web": "url",
16        "valenciaEs": "url",
17        "cp": "cp",
```

```

18     "location": "location"
19   },
20   "geometry": {
21     "type": "Point",
22     "coordinates": [
23       0.0,
24       0.0
25     ]
26   }
27 },
28 {
29   "type": "Service",
30   "properties": {
31     "name": "name",
32     "ownership": "ownership",
33     "phone": [
34       "phone"
35     ],
36     "fax": [
37       "fax"
38     ],
39     "email": [
40       "email"
41     ],
42     "web": "url",
43     "valenciaEs": "url"
44   }
45 }

```

Código 6.10: Plantilla para ficheros GeoJSON

Una vez rellenos los ficheros de recursos georreferenciados, estos son todos los recursos que se van a mostrar en la aplicación web:

Temática de los recursos	Nº. de recursos con localización	Nº. de recursos sin localización	Nº. de recursos totales	Origen de los recursos
Mujeres	87	6	93	Sitio web de Infocidad del Ayuntamiento de Valencia.
Personas con discapacidad	131	1	132	Sitio web de Infocidad del Ayuntamiento de Valencia.
Minorías étnicas y migrantes	28	0	28	Sitio web de Infocidad y Repara del Ayuntamiento de Valencia.
Personas sin techo	12	0	12	Sitio web de Infocidad del Ayuntamiento de Valencia.
Colectivo LGBTI	3	3	6	Búsqueda propia en sitios web oficiales
<b>Total de recursos en la aplicación web</b>			<b>261</b>	

Tabla 6.2: Recursos para la aplicación web

## 6.5. Implementación de la aplicación web

Finalmente para la implementación de la aplicación web hay diferentes puntos importantes como es la interfaz que se va a implementar para cada uno de los recursos, como insertar dinámicamente los recursos a la página de recursos, como crear el mapa interactivo y añadir las localizaciones de los recursos en él, y el sistema de búsqueda o filtrado por código postal.

### 6.5.1. Estructura de la aplicación web

Para la aplicación web se ha empleado la siguiente estructura de ficheros, donde la carpeta ‘JS’ son los ficheros en formato JavaScript para la funcionalidad general de la aplicación web, la carpeta ‘Build’ y ‘TemplateData’ que determinan los archivos generados automáticamente por el compilador de Unity WebGL para poder insertar el juego en la aplicación web, la carpeta ‘JSON’ que incluye todos los ficheros que almacenan la información georreferenciada de los recursos, la carpeta ‘media’ que contiene las imágenes utilizadas, y finalmente los archivos HTML que contienen la arquitectura de todas las páginas de la aplicación web, y la hoja de estilo ‘style.css’ que contiene el estilo general de la aplicación web.

Tipo de fichero	Número de ficheros
HTML (.html)	8
Hojas de estilo (.css)	2
JavaScript (.js)	6
Imágenes (.png .jpg .svg .ico)	16
JSON (.json)	6
Data (.data)	1
WebAssembly (.wasm)	1

Tabla 6.3: Ficheros de la aplicación web

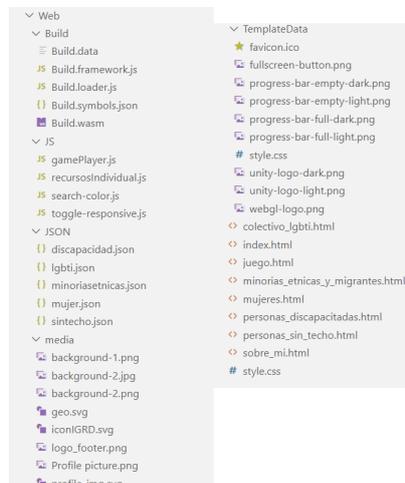


Figura 6.7: Estructura de la aplicación web

## 6.5.2. Interfaz del recurso

Para la interfaz del recurso se buscaba que al iniciar la página solo aparecerán botones con el nombre de los recursos, pero que al presionar sobre el botón desplegará, a modo de acordeón, toda la información del recurso. Para ello se ha creado una función donde recoge todos los botones que se han creado dinámicamente, que se profundizara sobre ello en el siguiente apartado, y a cada uno de ellos le ha dotado la funcionalidad de que al presionar sobre ellos modifica la altura del siguiente elemento, que resultaría ser un bloque con toda la información.

```

1 <div class="col-lg-4 col-md-6 col-recurso">
2   <button class="accordion">NOMBRE DEL RECURSO</button>
3
4   <div class="panel">
5     <p>
6       <b>Titularidad:</b> titularidad<br>
7       <b>Telefono:</b> numro teléfono<br>
8       <b>Fax:</b> numero fax<br>
9       <b>Web:</b> <a href="url" target="_blank"> <i class="bi-link-45deg"></i>
10         > Enlace <i class="bi-arrow-bar-right"></i> </a><br><br>
11       <a href="urlAlternativa" target="_blank"> <i class="bi-link-45deg"></i>
12         Mas Información <i class="bi-arrow-bar-right"></i></a><br><br>
13       <b>Dirección:</b>dirección<br>
14     </p>
15     <button class="recurso-button" data-cord-x="0.0" data-cord-y="10.0"
16       data-title="nombre recurso" data-link="url">Ver en mapa</button>
17     <p></p>
18   </div>
  </div>

```

Código 6.11: Código de interfaz de recurso

```

1 var acc = document.getElementsByClassName("accordion");
2 document.getElementById("totalRecursos").innerHTML = acc.length;
3 document.getElementById("recLocalizacion").innerHTML = numLocal;
4 var i;
5 for (i = 0; i < acc.length; i++) {
6   acc[i].addEventListener("click", function () {
7     this.classList.toggle("active");
8     var panel = this.nextElementSibling;
9     if (panel.style.maxHeight) {
10      panel.style.maxHeight = null;
11    } else {
12      panel.style.maxHeight = panel.scrollHeight + "px";
13    }
14  });
15 }
16
17 //Del fichero recursoIndividual.js

```

Código 6.12: Código de funcionalidad del recurso

### 6.5.3. Inserción dinámica de los recursos

Una vez que ya se ha diseñado como es la interfaz de cada uno de los recursos hay que añadir la lista completa a la aplicación web, pero hay que tener en cuenta que la disposición tiene que ser adaptativa, si se ha efectuado un filtrado de recursos y el tipo de recurso insertado. Para poder hacer más sencillo una disposición adaptativa se han utilizado las clases de la librería de Bootstrap.

Por lo tanto, para comenzar hay que recoger los recursos de los ficheros JSON, por lo que por medio de una petición Get del protocolo HTTP podemos retraer los recursos del fichero, y una vez se han cargado se envían a la función de mostrado.

```
1  function loadRecursos(input) {
2      const requestURL = document.querySelector('meta[name="recursos"]').content;
3      const request = new XMLHttpRequest();
4      request.open('GET', requestURL);
5      request.responseType = 'json';
6      request.send();
7      var features;
8      request.onload = function () {
9          showRecursos(recursosDiv, request.response, input);
10     }
11 }
12
13 //Del fichero recursoIndividual.js
```

Código 6.13: Código para cargar los ficheros en aplicación web

Para mostrar los recursos en la aplicación web se ha creado una función, a la que se le envía el id del bloque al que se van a cargar los recursos, un objeto con la información de los recursos y una cadena de caracteres que servirá para el sistema de búsqueda, que sería explicado posteriormente en otro apartado. Al comenzar se inicializaría el mapa interactivo, que será explicado en el siguiente apartado.

Para poder insertar dinámicamente los recursos primero hay que recorrer el objeto con la información de estos, y mientras se va recorriendo uno a uno, primero se comprueba si el recurso está vacío y así continuar con el siguiente. Acto seguido, se comprueba si el valor de búsqueda está vacío o si es un servicio el recurso que se va a mostrar, en el caso de que no sea servicio se añade el código postal del recurso a una lista que ayudara posteriormente en el filtrado de recursos. Para añadir la información a la interfaz comentada antes se utilizarán las funciones de `createElement()` y `appendChild()`, `createElement()` crea elementos en el documento HTML dentro de la funcionalidad en JavaScript, con este se irá creando los diferentes elementos dotándoles de las clases, estilos y contenido pertinentes, cabe destacar que para aquellos recursos que sean servicios se ha restringido buscar los valores de dirección y coordenadas, ya que estos no disponen de ellos. Una vez se han creado los elementos es necesario añadirlos unos dentro de otros para poder crear la estructura de la interfaz, que se realizara por medio de la función `appendChild()`.

Este código se repite posteriormente pero con el caso en que se haya añadido el filtrado de recursos, donde solo aparecerán los recursos cuyo código postal coincida con el introducido. Y por último todos los recursos se añaden al bloque de recursos, y se le añade la funcionalidad de acordeón ya explicada previamente.

```
1 function showRecursos(div, obj, search) {
2   typeof obj;
3   const data = obj['features'];
4
5   var numLocal = 0;
6
7   //Código de inicialización del mapa interactivo
8
9   const row = document.createElement("div");
10  row.className = "row px-2";
11
12  for (var i = 0; i < data.length; i++) {
13
14    if (typeof data[i] !== 'undefined') {
15
16      if (search == "" || data[i].type == "Service") {
17        if (data[i].type !== "Service") {
18          if (!pruebaCP.includes(data[i].properties.cp)) {
19            pruebaCP += data[i].properties.cp + " ";
20            CPs.innerHTML += "<option>" + data[i].properties.cp + "</option>";
21          }
22        }
23        const col = document.createElement("div");
24        col.className = "col-lg-4 col-md-6 col-recurso";
25
26        const button = document.createElement("button");
27        button.className = "accordion";
28        button.innerHTML = data[i].properties.name;
29
30        const panel = document.createElement("div");
31        panel.className = "panel";
32
33        var inner = "";
34        if (data[i].properties.ownership != "")
35          inner += "<b>Titularidad:</b> " + data[i].properties.ownership
36            + "<br>";
37
38        for (var t = 0; t < data[i].properties.phone.length; t++) {
39          inner += "<b>Telefono:</b> " + data[i].properties.phone[t] + "
40            <br>";
41        }
42
43        for (var f = 0; f < data[i].properties.fax.length; f++) {
44          inner += "<b>Fax:</b> " + data[i].properties.fax[f] + "<br>";
45        }
46
47        for (var e = 0; e < data[i].properties.email.length; e++) {
48          inner += "<b>Correo Electronico:</b> " + data[i].
49            properties.email[e] + "<br>";
50        }
51
52        if (data[i].properties.web != "")
53          inner += "<b>Web:</b> <a href='" + data[i].properties.web + "'
54            target='_blank' > <i class='bi-link-45deg'></i> Enlace <i
55            class='bi-arrow-bar-right' ></i> </a><br>";
56
57        if (data[i].properties.valenciaEs != "")
58          inner += "<br><a href='" + data[i].properties.valenciaEs + "'
```

```
target='_blank'> <i class='bi-link-45deg' ></i > Mas
Información <i class='bi-arrow-bar-right' ></i></a><br>";
54
55
56     if (data[i].type == "Feature") {
57
58         inner += "<br>";
59         inner += "<b>Dirección:</b> " + data[i].properties.location + "
        <br>";
60         addMarker(data[i]);
61         numLocal++;
62     }
63
64     const text = document.createElement("p");
65     text.innerHTML = inner;
66     panel.appendChild(text);
67
68     if (data[i].type == "Feature") {
69
70         var buttonMap = document.createElement('button');
71         buttonMap.className = "recurso-button";
72
73         buttonMap.dataset.cordX = data[i].geometry.coordinates[1];
74         buttonMap.dataset.cordY = data[i].geometry.coordinates[0];
75         buttonMap.dataset.title = data[i].properties.name;
76
77         if (data[i].properties.web != "") {
78             buttonMap.dataset.link = data[i].properties.web;
79         }
80         else if (data[i].properties.valenciaEs != "") {
81             buttonMap.dataset.link = data[i].properties.valenciaEs;
82         }
83         else {
84             buttonMap.dataset.link = "";
85         }
86
87         buttonMap.textContent = "Ver en mapa";
88         buttonMap.addEventListener('click', showOneCoordinate, false);
89
90
91         panel.appendChild(buttonMap);
92
93         var space = document.createElement('p');
94
95         panel.appendChild(space);
96     }
97
98     col.appendChild(button);
99     col.appendChild(panel);
100
101     row.appendChild(col);
102
103 } else {
104
105     if (data[i].properties.cp == search) {
106         //Mismo código de inserción de recurso pero si se ha filtrado
        por c.p.
107     }
108 }
```

```

109     }
110   }
111
112   const app = document.querySelector(div);
113   app.appendChild(row);
114
115
116   //Código de funcionalidad del acordeón
117 }
118
119 //Del fichero recursoIndividual.js

```

Código 6.14: Código para mostrar los recursos

### 6.5.4. Mapa interactivo

Como mencionado antes se ha creado un mapa interactivo donde se podrá localizar los recursos, este mapa ha sido creado con la librería de Leaflet, que ayuda a crear mapas interactivos. Para ello primero hay que inicializar el mapa, que es cargado en la función de mostrar recursos previamente explicada, donde primero se le añade el punto de vista principal y después se le añaden todos los atributos necesarios para iniciarlo. Y finalmente se crea un grupo de marcadores, donde se añadirán los marcadores que señalan a los recursos.

```

1  mymap = new L.map("mapid").setView([39.46989425180655, -0.3763457058638949], 12);
2  L.tileLayer('https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_
   token=pk.eyJ1IjoicGVkcm9yeG1vcyIsImEiOiJja204YmVjY2YwdHFOmBzOW91cW9pZnh5In0.
   Ggf0VEANAt_75Y6TkLKRvA', {
3    maxZoom: 20,
4    attribution: '<p style="font-size: 10px;margin-top:-4px;"> Map data &copy;<a
   href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
   contributors, ' +
5    'Imagery @<a href="https://www.mapbox.com/">Mapbox</a><p>',
6    id: 'pedrorxmos/ckm8cwm8mfey617o4uztrf0pt',
7    style: 'mapbox://styles/pedrorxmos/ckm8cwm8mfey617o4uztrf0pt',
8    tileSize: 512,
9    zoomOffset: -1
10  }).addTo(mymap);
11
12  markers = L.markerClusterGroup({ showCoverageOnHover: false });
13
14  //Del fichero recursoIndividual.js

```

Código 6.15: Código para inicializar mapa interactivo

Dentro del bucle que recorre todos los recursos se ha añadido una función que crea un marcador dentro del mapa interactivo con las coordenadas del recurso y le añade un título o nombre y un enlace dependiendo si tiene página web o página web alternativa.

```

1  function addMarker(data) {
2    var customIcon = new L.Icon({
3      iconUrl: 'media/geo.svg',
4      iconSize: [30, 30],
5      iconAnchor: [15, 30],
6      popupAnchor: [0, -30]

```

```

7     });
8
9     if (data.properties.web != "") {
10        markers.addLayer(new L.Marker(new L.LatLng(data.geometry.coordinates[1],
11           data.geometry.coordinates[0]), { icon: customIcon })
12           .bindPopup('<span style="text-transform: uppercase;">' +
13             data.properties.name + '</span> <br> <a href="' +
14             data.properties.web + '" style="text-transform: uppercase;" target=
15             "_blank"> <i class="bi-link-45deg" ></i> Mas Información <i class=
16             "bi-arrow-bar-right" ></i></a>'));
17    }
18    else if (data.properties.valenciaEs != "") {
19        markers.addLayer(new L.Marker(new L.LatLng(data.geometry.coordinates[1],
20           data.geometry.coordinates[0]), { icon: customIcon })
21           .bindPopup('<span style="text-transform: uppercase;">' +
22             data.properties.name + '</span> <br> <a href="' +
23             data.properties.valenciaEs + '" style="text-transform: uppercase;"
24             target="_blank"> <i class="bi-link-45deg" ></i> Mas Información <i
25             class="bi-arrow-bar-right" ></i></a>'));
26    }
27    else {
28        markers.addLayer(new L.Marker(new L.LatLng(data.geometry.coordinates[1],
29           data.geometry.coordinates[0]), { icon: customIcon })
30           .bindPopup('<span style="text-transform: uppercase;">' +
31             data.properties.name + '</span>'));
32    }
33 }
34
35 //Del fichero recursoIndividual.js

```

Código 6.16: Código para añadir marcador

Por último se ha añadido un botón ‘Ver en mapa’ en cada uno de los recursos, el cual por medio de diferentes datasets o grupo de datos acerca la vista del mapa interactivo al recurso seleccionado, los cuales se añaden por los atributos data-cord-x, data-cord-y, data-title y data-link, que corresponde a la coordenada X, coordenada Y, título o nombre de recurso y enlace respectivamente, y activa un popup con el nombre del recurso y el enlace a su página web.

```

1     function showOneCoordinate(event) {
2         var cX = event.currentTarget.dataset.cordX;
3         var cY = event.currentTarget.dataset.cordY;
4         var title = event.currentTarget.dataset.title;
5         var link = event.currentTarget.dataset.link;
6
7         var content = '<span style="text-transform: uppercase;">' + title + '</span>
8           <br> <a href="' + link + '" style="text-transform: uppercase;" target="
9           _blank"> <i class="bi-link-45deg" ></i> Mas Información <i class="
10          bi-arrow-bar-right" ></i></a>';
11
12        window.scrollTo(0, 0);
13
14        mymap.setView([cX, cY], 17);
15
16        var popup = L.popup({ offset: [0, -25] })
17          .setLatLng([cX, cY])
18          .setContent(content)
19          .openOn(mymap);

```

```

17 }
18
19 //Del fichero recursoIndividual.js

```

Código 6.17: Código para mostrar marcador individual

### 6.5.5. Sistema de búsqueda

Finalmente está el sistema de búsqueda o filtrado de recursos por código postal, el cual trata de un input de texto el cual mostrará los recursos que tenga un código postal que coincida con el introducido en el input de texto.

Para ello se ha creado una función que se ejecuta cada vez que se introduce algo en el input de texto, donde primero vacía el contenido del bloque que muestra los recursos y borra el mapa interactivo, donde volverá al final a llamar a la función de cargar recursos, pero esta vez con la variable de búsqueda no vacía, por lo que para transcurrir por los ficheros que tienen localización su código postal tiene que coincidir con el código postal buscado.

```

1  function searchCP(event) {
2      const app = document.querySelector(recursosDiv);
3      app.innerHTML = "";
4      mymap.remove();
5
6      loadRecursos(event.currentTarget.value);
7  }
8
9  //Del fichero recursoIndividual.js

```

Código 6.18: Código del sistema de búsqueda

Asimismo, si ha añadido un código que apoya al sistema de búsqueda, que se encuentra dentro del bucle que recorre los recursos, el cual crea una lista con todos los códigos postales únicos que se pueden encontrar en el listado de recursos, con lo que al empezar a buscar aparecerá una lista de códigos postales disponibles, aunque sigue con la posibilidad de que lo pueda escribir el usuario.

```

1  if (data[i].type !== "Service") {
2      if (!pruebaCP.includes(data[i].properties.cp)) {
3          pruebaCP += data[i].properties.cp + " ";
4          CPs.innerHTML += "<option>" + data[i].properties.cp + "</option>";
5      }
6  }
7
8  //Del fichero recursoIndividual.js

```

Código 6.19: Código del apoyo al sistema de búsqueda

```

1  <div class="growing-search">
2      <label class="search-label" id="search-label">Buscar por Código Postal: </
3      label>
4      <div class="input">
5          <!--Input de texto para la busqueda-->

```

```
5     <input type="text" name="search" id="search" list="CPs" autocomplete="off"
6         class="search" placeholder="Ej.: 46001" />
7     <!--Listado de apoyo a la búsqueda-->
8     <datalist id="CPs">
9     </datalist>
10    <div class="submit">
11        <span id="icon-search" class="fa fa-search icon-search"></span>
12    </div>
13 </div>
```

Código 6.20: Código de la interfaz del sistema de búsqueda

# Capítulo 7

## Pruebas y resultados

Para poder comprobar que tanto la aplicación web y el videojuego funcionan y se visualizan correctamente se ha necesitado realizar diferentes pruebas. Primero unas pruebas funcionales, que muestran la funcionalidad de la aplicación web y del videojuego, donde se comprueba el cumplimiento de los requisitos especificados en el capítulo tres. Y unas pruebas de usabilidad, donde se mostrarán las diferentes versiones por las que ha pasado el proyecto.

### 7.1. Pruebas funcionales

#### 7.1.1. Pruebas funcionales de la aplicación web

Primero, se comenzará con las pruebas funcionales de la aplicación web, para ello se va a ir enumerando los requisitos que se van comprobando en cada prueba. Para todas las pruebas de la aplicación web se han comprobado los requisitos F1 y NF1, que piden que la aplicación sea responsive e intuitiva, para ello, se muestra como ejemplo, la página de inicio (Figura 7.2).

Para probar el requisito F4 y NF5 que requieren que la aplicación web se divida en secciones se mostrará la barra de navegación de la aplicación web (Figura 7.1).

Para el requisito F7, que consta en indexar el videojuego en la aplicación web se muestra la página del juego (Figura 7.3).

Los requisitos F2, F3, NF2 y NF3 que corresponden con la inserción de los recursos en la aplicación web, como también la información que se muestra de estos. Para la prueba solo se mostrará la página de los recursos que caen en la categoría de ‘Personas Sin Techo’, ya que es una de las categorías con menos recursos, pero los suficientes para mostrar el funcionamiento correcto, y todas las categorías tienen el mismo funcionamiento y estructura (Figura 7.5).

Los requisitos F5 y NF4 piden poder localizar los recursos que tengan una dirección dentro del mapa interactivo (Figura 7.6).

Y finalmente el requisito F6, que trata del sistema de búsqueda por código postal, donde la prueba muestra la página con los recursos de la categoría de ‘Personas Sin Techo’ al aplicar el código postal ‘46003’ (Figura 7.7)

Figuras de las pruebas funcionales de la aplicación web:

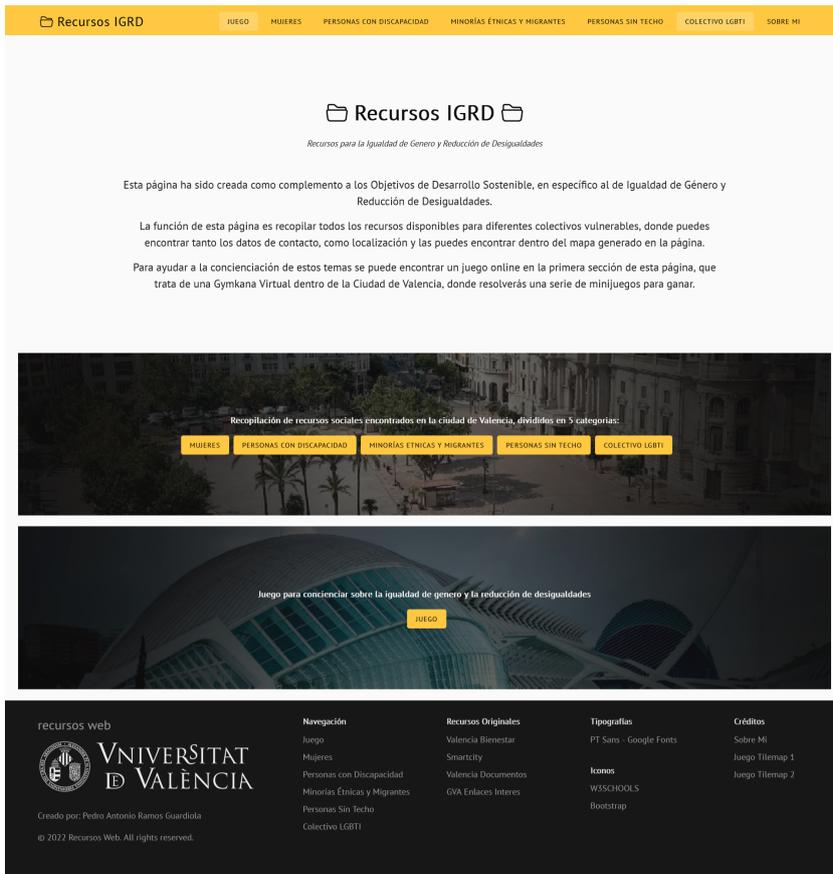


(a) Versión desktop



(b) Versión móvil

Figura 7.1: Prueba funcional de barra de navegación

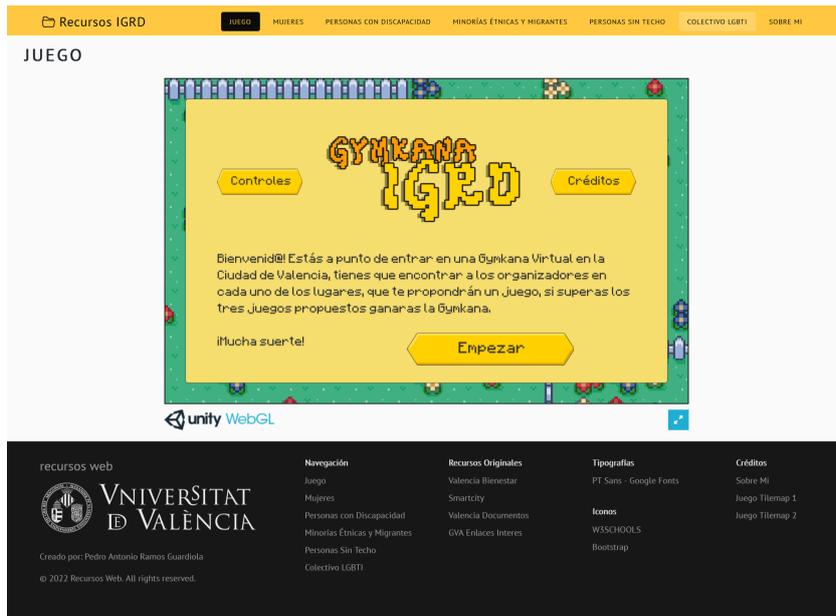


(a) Versión desktop

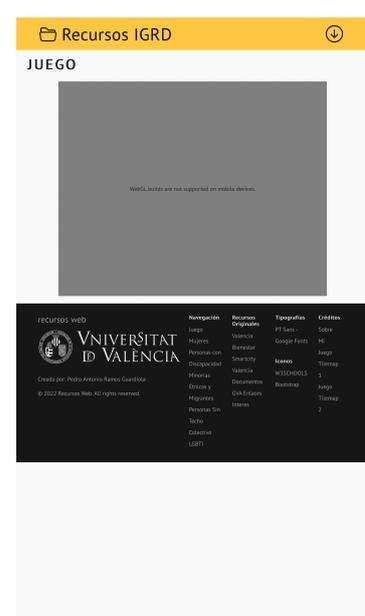


(b) Versión móvil

Figura 7.2: Prueba funcional de la página de inicio

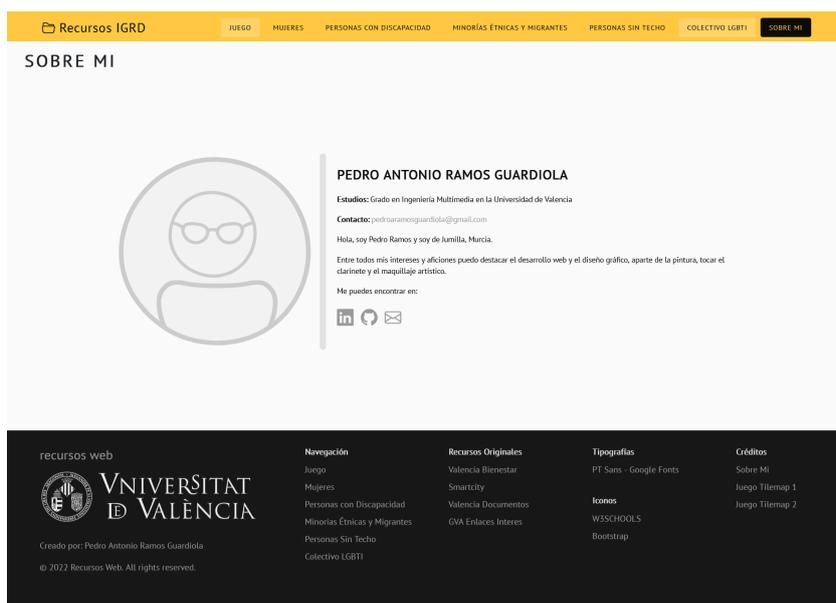


(a) Versión desktop

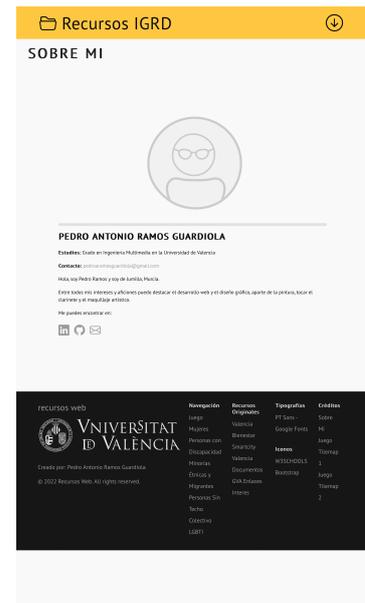


(b) Versión móvil

Figura 7.3: Prueba funcional de la página de juego

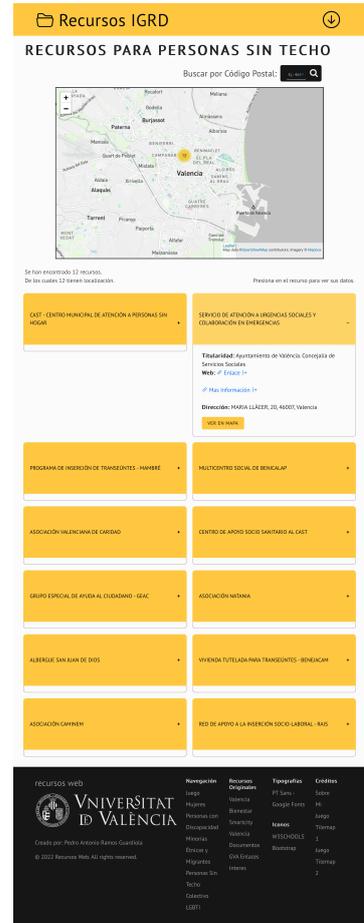
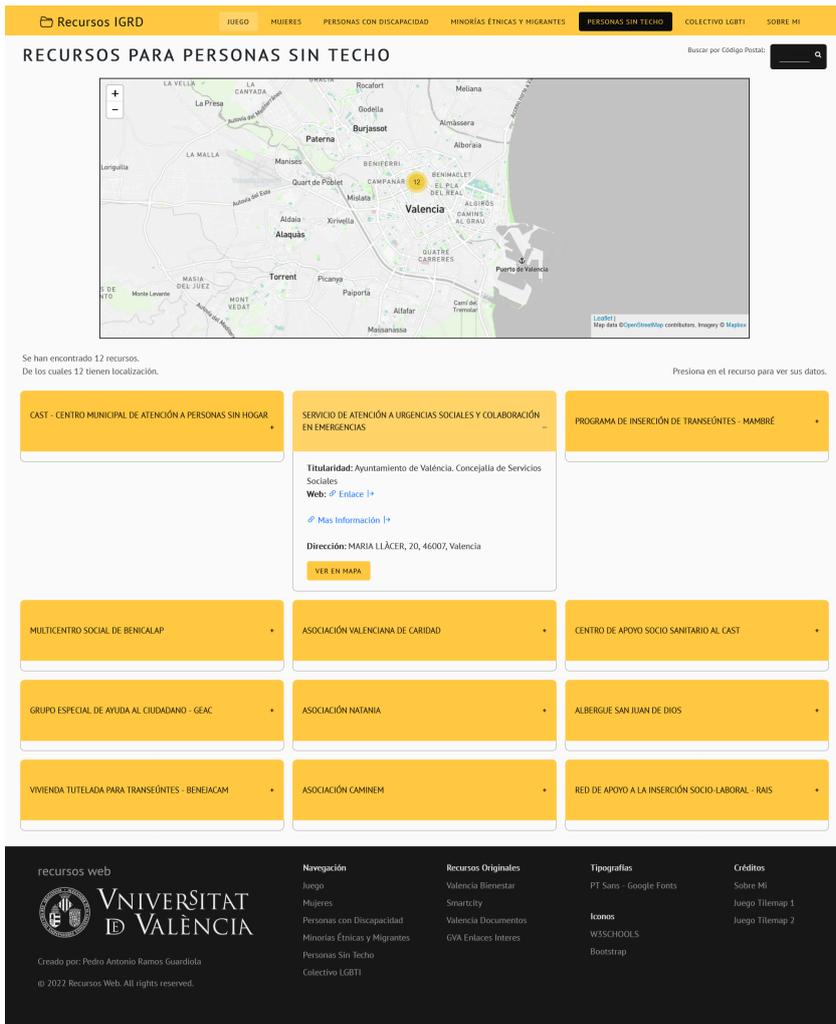


(a) Versión desktop



(b) Versión móvil

Figura 7.4: Prueba funcional de la página sobre el autor



(a) Versión desktop

(b) Versión móvil

Figura 7.5: Prueba funcional de las páginas de recursos

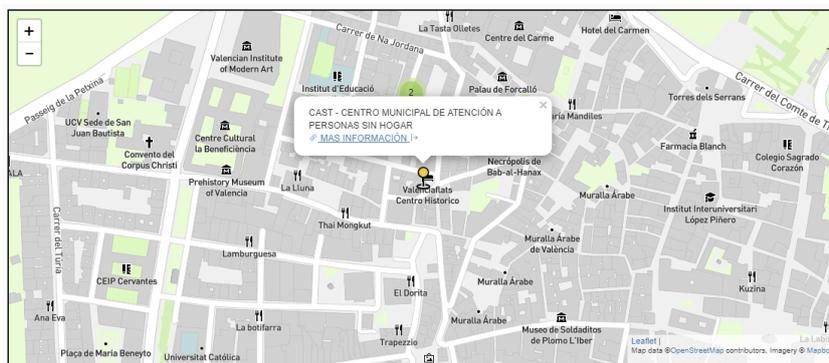


Figura 7.6: Prueba funcional de las páginas de recursos al localizar un recurso en el mapa

Recursos IGRD

JUEGO MUJERES PERSONAS CON DISCAPACIDAD MINORÍAS ÉTNICAS Y MIGRANTES PERSONAS SIN TECHO COLECTIVO LGBTI SOBRE MI

RECURSOS PARA PERSONAS SIN TECHO

Buscar por Código Postal: 46003

Se han encontrado 4 recursos. De los cuales 4 tienen localización.

Presiona en el recurso para ver sus datos.

CAST - CENTRO MUNICIPAL DE ATENCIÓN A PERSONAS SIN HOGAR

CENTRO DE APOYO SOCIO SANITARIO AL CAST

GRUPO ESPECIAL DE AYUDA AL CIUDADANO - GEAC

Titularidad: Ayuntamiento de Valencia, Concejalía de Policía  
 Teléfono: 962084757  
 Fax: 965520240  
 Correo Electrónico: cast@valencia.es  
 Web: [Enlace](#) <sup>+</sup>  
[Mas Información](#) <sup>+</sup>

Dirección: JARDINES, s/n (Esquina Soguers 20), 46003, Valencia

VER EN MAPA

RED DE APOYO A LA INSERCIÓN SOCIO-LABORAL - RAIS

recursos web

UNIVERSITAT DE VALÈNCIA

Creado por: Pedro Antonio Ramos Guardiola  
 © 2022 Recursos Web. All rights reserved.

Navegación  
 Juego  
 Mujeres  
 Personas con Discapacidad  
 Minorías Étnicas y Migrantes  
 Personas Sin Techo  
 Colectivo LGBTI

Recursos Originales  
 Valencia Bienestar  
 Smartcity  
 Valencia Documentos  
 GVA Enlaces Interes

Tipografías  
 P1 Sans - Google Fonts

Iconos  
 WISCHDOLS  
 Bootstrap

Créditos  
 Sobre Mi  
 Juego Tilemap 1  
 Juego Tilemap 2

Figura 7.7: Prueba funcional de las páginas de recursos al utilizar buscador por c.p.

## 7.1.2. Pruebas funcionales del videojuego

Por último, para las pruebas funcionales del videojuego se ha ejecutado el videojuego y las figuras que se muestran son como la sucesión de pantallas del videojuego, donde el requisito NF7 aplica a todo el videojuego, ya que solo explica a que público está este creado.

En la primera pantalla se encuentra la pantalla de inicio y personalización que cumple con el requisito F8 (Figura 7.8).

A continuación se encuentran los diferentes niveles del videojuego, donde primero se encuentra una pantalla que indica la localización del personaje, cumpliendo con el requisito NF6, que indica que el videojuego representará la ciudad de Valencia (Figura 7.9). Los niveles tienen un mapa que representa cada uno un lugar de la ciudad de Valencia, donde el personaje se podrá mover, que cumple con los requisitos F9, F11 y NF6, y un minijuego, que prueba los requisitos F11 y NF8 (Figuras 7.10, 7.11, 7.12).

Cana nivel tiene un organizador con dialogo, donde entraría la prueba del requisito F10, donde funciona con todos los organizadores, pero se mostrará solo el organizador del nivel 2 (Figura 7.13).

Y por último los requisitos F12, F13 y F14, que representan la puntuación de cada minijuego, para su prueba solo se muestra el minijuego del segundo nivel, aunque la puntuación de todos los niveles se muestra igual (Figura 7.14).

Las figuras de las pruebas del videojuego son:



Figura 7.8: Prueba funcional de las pantallas de inicio y personalización del personaje



Figura 7.9: Prueba funcional de la pantalla de información del mapa

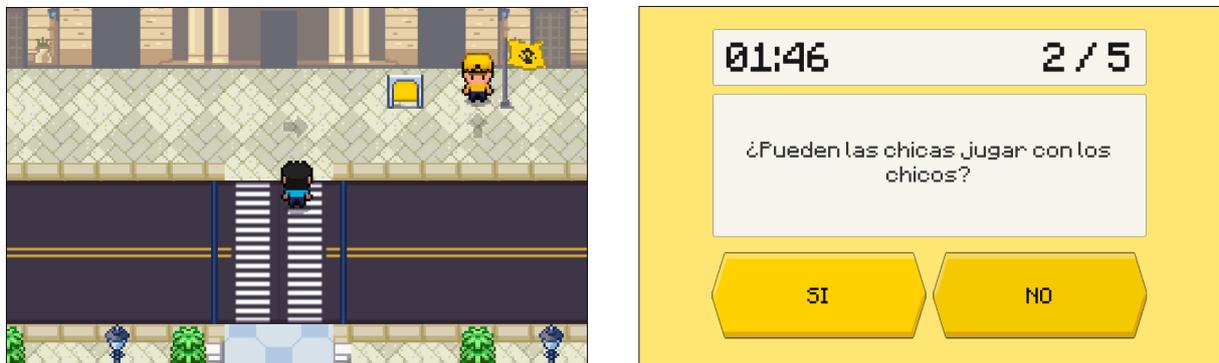


Figura 7.10: Prueba funcional de las pantallas de mapa y minijuego del nivel 1



Figura 7.11: Prueba funcional de las pantallas de mapa y minijuego del nivel 2



Figura 7.12: Prueba funcional de las pantallas de mapa y minijuego del nivel 3



Figura 7.13: Prueba funcional del sistema de dialogo

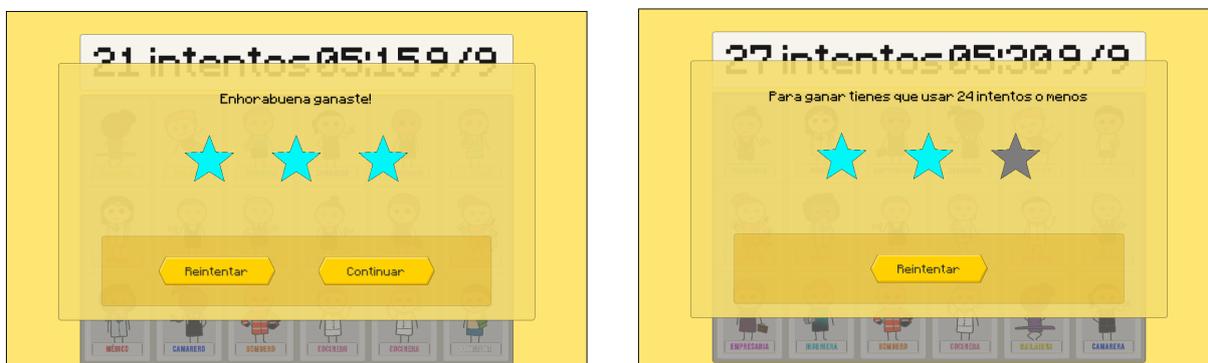


Figura 7.14: Prueba funcional de la pantalla de puntuación de minijuego



Figura 7.15: Prueba funcional de la pantalla final del videojuego

## 7.2. Pruebas de usabilidad

Para estas pruebas, a parte de revisar en propia persona las diferentes versiones, también se ha compartido con diferentes voluntarios de diferentes ámbitos y experiencias para que probasen las diferentes versiones y obtener un feedback para poder mejorar la aplicación web y el videojuego, estos voluntarios son diez personas que abarcan entre los 21 y 30 años de edad, donde cinco de estos tienen conocimientos informáticos básicos, uno de ellos tiene conocimientos más desarrollados en el ámbito web, otro tiene conocimientos avanzados de desarrollo de videojuegos y los tres son compañeros del Grado de Ingeniería Multimedia con los mismos conocimientos que el autor. Por ello tanto la aplicación web y el videojuego han pasado por las siguientes versiones, y han tenido los siguientes inconvenientes:

1. Primera versión del videojuego: En esta primera versión la aplicación web solo contaba con una página de juego, ya que ha sido lo primero en desarrollarse, donde al obtener el feedback de los voluntarios se ha concluido en los siguientes fallos:
  - Animación del personaje a destiempo, por lo que se modificaron las secuencias de animación para que coincidiese con el movimiento de este al presionar las teclas de movimiento.
  - Los intentos del segundo minijuego son escasos, ya que en un comienzo para poder conseguir las tres estrellas era necesario superarlo con menos o igual a 16 intentos, por lo que el número de intentos máximos para superar el minijuego paso a ser 24.
  - Falta de indicación en los mapas, por lo que se le añadió un panel inicial al cargar el mapa que explicaba en que localización de la ciudad de Valencia se encontraba el personaje y se añadió unas flechas en el mapa que indicaban la dirección para encontrar al organizador del minijuego.
  - Interferencias en el sistema de dialogo con la expansión de pantalla del compilador. Este problema se causaba ya que previamente la tecla de dialogo era 'Espacio', el cual tenía interferencias si habías expandido el videojuego a pantalla completa previamente, haciendo que se saliera de pantalla completa, por lo que se pasó a utilizar la tecla 'Z'.
  - Falta de sprites, ya que en esta versión no se contaba aún con todos los sprites que componían el videojuego, como eran los edificios o las cartas del segundo minijuego, que serían añadidos a la siguiente versión.
2. Última versión del videojuego: Esta segunda y ultima versión del videojuego ya se contaba con todos los fallos expuestos en la primera versión solucionados, donde al haber obtenido un nuevo feedback de los voluntarios, ya no se encontraban nuevos fallos, resultando así en una versión definitiva.
3. Primera versión aplicación web: Para la primera versión de la aplicación web donde ya se contaba con los recursos se encontraron los siguientes fallos:
  - Falta de adaptabilidad a diferentes resoluciones debido a que estas se ejecutaban por script no por hoja de estilo, por lo que el cambio entre una versión desktop y una versión móvil necesitaba que se refrescase la página para ver una diferencia y que se adaptase al nuevo ancho de la ventana, por lo que se

investigó mejor el uso de Bootstrap en HTML, resultado así una página completamente responsive o adaptable, con dos versiones diferentes de barra de navegación que se escondían o aparecían dependiendo del ancho de pantalla.

- Fallo en la funcionalidad del botón que redirige el mapa a la localización de un recurso específico, por lo que se acabo cambiando al actual proceso que envía datasets a la función de mostrar el marcador del recurso.
4. Segunda versión de la aplicación web: En esta segunda versión se solucionó el problema de la funcionalidad del botón de localización del recurso, y una vez recogido el feedback de esta nueva versión, se encontró un nuevo fallo:
    - Falta de sección sobre el autor de la página. Debido a que la aplicación web iba a ser subida al servidor de la catedrá de la Universidad de Valencia, por lo que se añadió una nueva sección en la barra de navegación titulada ‘Sobre Mi’ donde se muestra información relevante sobre el autor de la aplicación web.
  5. Ultima versión de la aplicación web: Una vez añadida la sección sobre el autor y modificada la aplicación web para mejorar la adaptabilidad se volvió a pedir un feedback de los voluntarios, resultando en un feedback completamente positivo, donde no se encontraron fallos, así dando lugar a la versión definitiva de la aplicación web.

### 7.3. Pruebas de velocidad

Una vez la aplicación web tiene una versión final se han realizado unas pruebas de velocidad a cada una de las secciones. Para ello se ha utilizado la herramienta PageSpeed Insights [38], que permite ver los tiempos de carga de cada uno de los archivos y también ofrece cambios para mejorar la velocidad de carga de la aplicación web.

Comenzando por la página del videojuego se puede observar la gran diferencia entre la versión móvil y la versión desktop, ya que el videojuego no se puede cargar en un dispositivo móvil, por lo que los tiempos de carga bajan drásticamente.

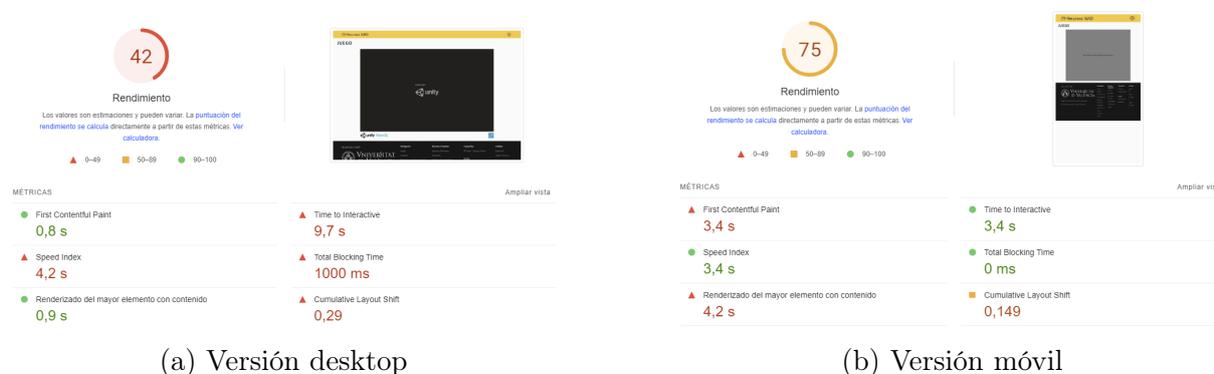


Figura 7.16: Prueba de velocidad de la página del videojuego

Como se puede observar en las capturas, el rendimiento de la versión desktop es mucho menor que la versión móvil y la mayoría errores encontrados tratan de los ficheros

JavaScript que se generaron con WebGL de Unity para poder compilar el videojuego en la aplicación web, por lo que no sería posible mejorar el rendimiento de esta página individual.

Por último se ha probado la velocidad de las páginas de recursos, en todas ellas se han encontrado los mismos errores solo cambiando el rendimiento general dependiendo de la página ya que unas muestran más recursos que las otras. Para mostrar un ejemplo se utilizará la página de los recursos para mujeres.

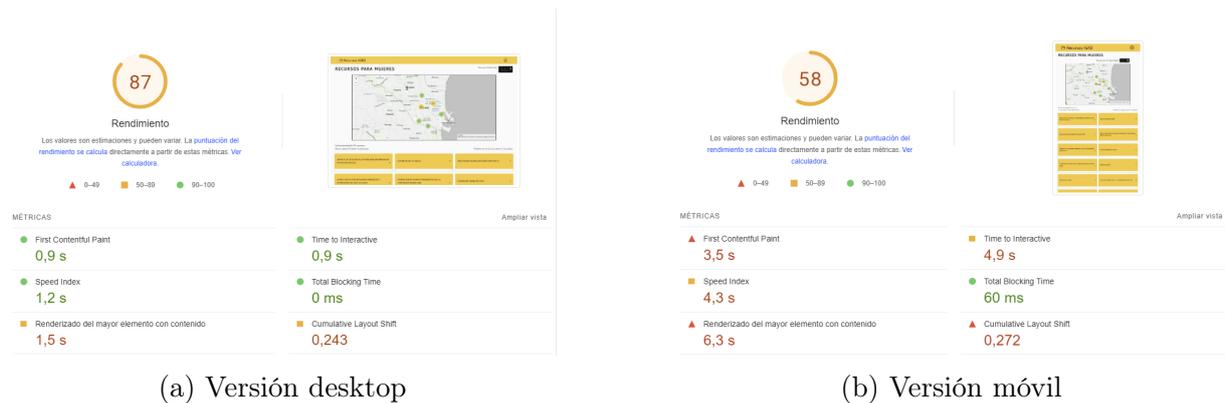


Figura 7.17: Prueba de velocidad de las páginas de recursos

Como se muestra en las capturas, esta vez la versión móvil tiene un rendimiento menor que la versión desktop, esto es debido a los ficheros que modifican la apariencia para la versión móvil, aunque la diferencia no es tan grande como en la página del videojuego. Entre los errores se encuentran las imágenes ocultas, es decir, los iconos de localización del mapa o los iconos de los enlaces de la información de cada recurso, y el tiempo de carga de los ficheros JSON, aunque no resultan ser un problema grave.

Al realizar las pruebas se han encontrado diferentes fallos generales, como son la falta de una etiqueta `< metaname = "viewport" >` que define el tamaño de ventana, como la ralentización por usar librerías online para Bootstrap y Leaflet, ya que en estos casos se cargan más ficheros de los necesarios, por lo que se recomienda tener las librerías de forma local, así solo utiliza aquellos ficheros que son necesarios.

A parte de errores, también se han encontrado auditorías positivas, como son los tiempos de ejecución de JavaScript, las hojas de estilo CSS escuetas, como el correcto tamaño y codificación de las imágenes utilizadas.

## 7.4. Resultados

Una vez ya finalizadas todas las pruebas y, por ende, tener las versiones definitivas tanto del videojuego como de la aplicación web, y que por último, se han subido al servidor.

El resultado completo de la aplicación web se puede comprobar en el siguiente enlace a la catedr a de la Universidad de Valencia: <https://ludificacion.uv.es/wp-content/uploads/recursosIGRD/> [39]

# Capítulo 8

## Conclusiones

### 8.1. Revisión de costes

En este apartado se comparará las estimaciones tanto temporal como de costes realizadas en el tercer capítulo con el tiempo y coste real, es decir, el tiempo y coste empleado al realizar el proyecto,

Para esta comparación hay que tener en cuenta estos factores:

1. Los costes personales no se tendrán en cuenta, ya que el proyecto solo ha sido realizado por una persona.
2. Las texturas que se han creado son menos de las estimadas, ya que se han utilizado texturas de terceros, por lo que solo se contará con el tiempo empleado para las texturas de las cartas, las texturas para el Ayuntamiento, el Palacio de las Artes y el parque Gulliver.
3. Aunque parte de del software esta proporcionado por la universidad, en esta revisión se mostrará como si no estuviera proporcionado por ella.

A la hora de realizar este proyecto, los tiempos utilizados en días para cada apartado han sido:

Actividad	Tiempo empleado
Estado del Arte	8
Especificación, planificación, estimaciones y viabilidad	12
Análisis del proyecto	5
Diseño del proyecto	8
Implementación del proyecto	62
Crear texturas para el videojuego	5
Implementación del videojuego	30
Implementación de los ficheros georreferenciados	7
Implementación de la aplicación web	25
Pruebas y resultados	5
<b>Total de tiempo empleado en días</b>	<b>105</b>

Tabla 8.1: Tiempo empleado del proyecto

Por lo que a partir de los tiempos empleados, y del software utilizado el cual se pagaría la suscripción completa especificada en el segundo capítulo, es decir, si un programa cuesta 10€ cada mes, significa que si sus actividades duran un total de 10 días, es decir, menos de un mes, el coste será de 10€, mientras que si se supera, por ejemplo, a un mes y medio de uso el coste de este pasara a dos meses, es decir, a 20€. Por lo que los gastos resultantes son:

Software	Dias de uso	Meses con suscripción	Precio
Visual Studio Code	42	$\infty$	0€
Visual Paradigm	5	1	18.68€
Adobe XD	8	1	12.09€
Adobe Photoshop	5	1	24.19€
Procreate iPad	5	$\infty$	9.99€
Unity	30	12	1656€
<b>Total</b>			<b>1720.95€</b>

Tabla 8.2: Costes finales de software

Por lo que el coste del software final del proyecto sería igual al estimado.

## 8.2. Conclusiones

Este trabajo de fin de grado ha consistido en la implementación de una aplicación web que disponga de recursos para ayudar a las personas afectadas por temas de género y desigualdades, donde al mismo tiempo se pudiera concienciar a los niños y niñas de estos temas mediante un videojuego interactivo dentro de la ciudad de Valencia.

Durante el proceso de elaboración de la aplicación web se ha podido comprobar de primera mano el proceso de crear una aplicación web y un videojuego por mi cuenta donde se ha podido utilizar conocimientos propios y adquiridos durante el grado de Ingeniería Multimedia, como son el análisis de requisitos y funcionalidades, estimación temporal y de costes, análisis de riesgos de un proyecto, planificación de un sistema informático por medio de diagramas de casos de uso, actividad, clases, etc. y el desarrollo de un videojuego mediante Unity, como también se ha mejorado y adquirido nuevos conocimientos como son:

- Mejor entendimiento del funcionamiento y utilidad de la librería de Bootstrap en la aplicación web.
- Mejor entendimiento de los mapas interactivos de Leaflet.
- Implementar una comunicación con ficheros JSON a partir del protocolo HTTP.
- Desarrollar un videojuego de gráficos 2D.
- Aplicar mis conocimientos sociales a una aplicación web.
- Crear documentos a partir de L<sup>A</sup>T<sub>E</sub>X.

### 8.3. Trabajo futuro

Aunque el proyecto se puede considerar acabado, se le pueden hacer diferentes mejoras a cara de futuro, con las que se contará como si se estuviera trabajando con un equipo de profesionales, por lo que se proponen las siguientes mejoras, tanto para el videojuego como para la aplicación web:

- Mejora en gráficos del videojuego, por lo que sería preciso el trabajo de un artista especializado en estilo pixel en dos dimensiones, ya que como se ha mencionado en la memoria, las texturas utilizadas para crear el mapa, como las bases para el personaje son obtenidas de tilesets de dominio público, así dotando al videojuego de características propias, y al mismo tiempo ayudando a la similitud de los gráficos con la vida real.
- Añadir más diversidad en la personalización del personaje en el videojuego, ya que al haber creado todos los sprites una sola persona con el tiempo limitado, hay pocas posibilidades de personalización, por lo que se propone crear más sprites, con diferentes vestimentas, más tonos y tipos de piel, añadir accesorios como sombreros en la personalización y nuevos colores y estilos de peinados así ayudando a la inclusividad de todo el mundo.
- Añadir nuevos niveles, con más localizaciones y nuevos minijuegos, ampliando así el tiempo de jugabilidad.
- Ampliar la sección de filtros de recursos dentro de la aplicación web, ayudando así a los usuarios poder encontrar recursos de forma más precisa con su ubicación, o por otros datos obtenidos de los recursos.

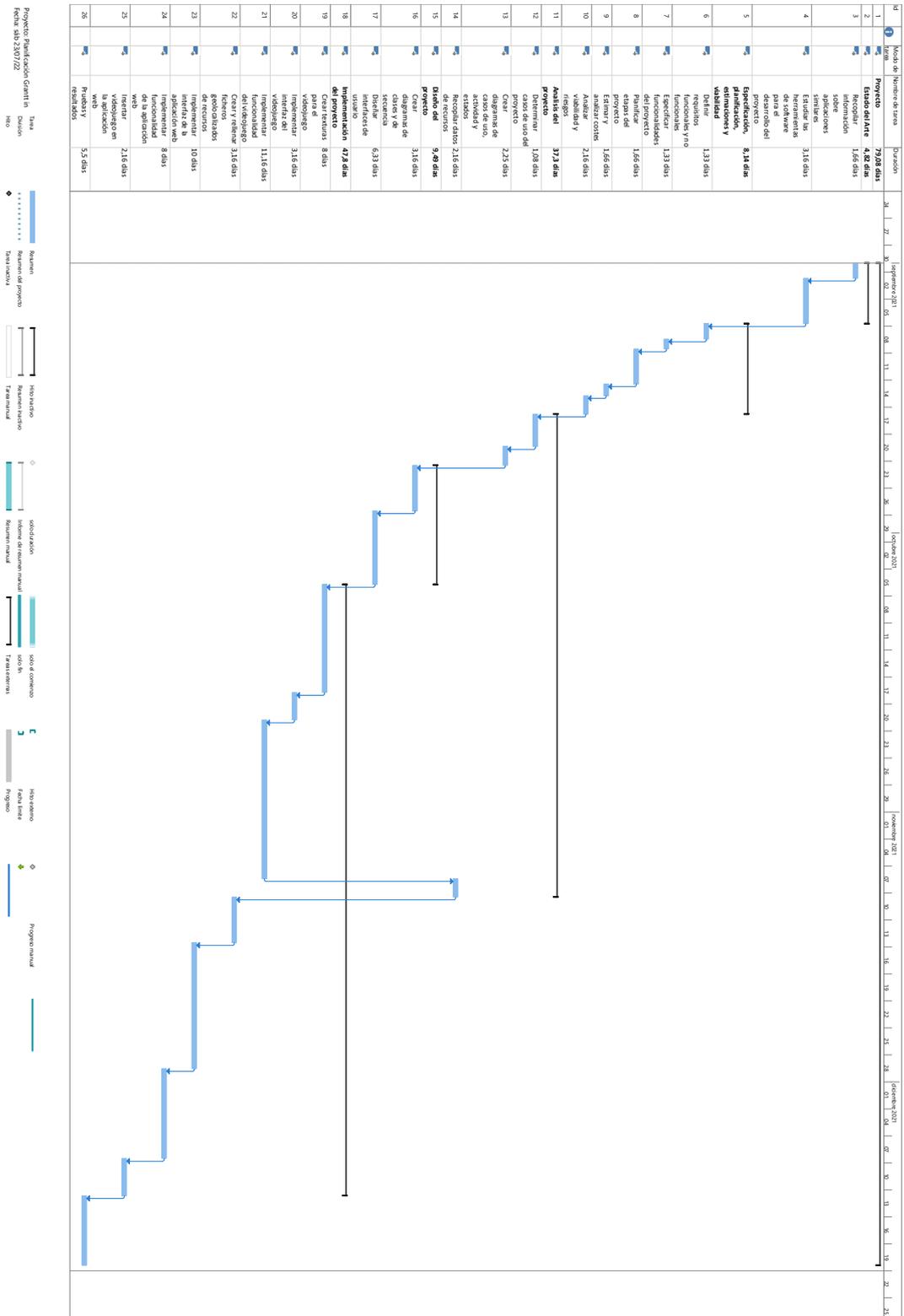
Como se ha mencionado antes, este proyecto ha sido centrado en la ciudad de Valencia, por lo que uno de los posibles proyectos que podrían mejorar la aplicación web en el futuro es el poder ampliarla al resto del país, añadiendo nuevas funciones como filtrado por comunidad autónoma.



Apéndice A

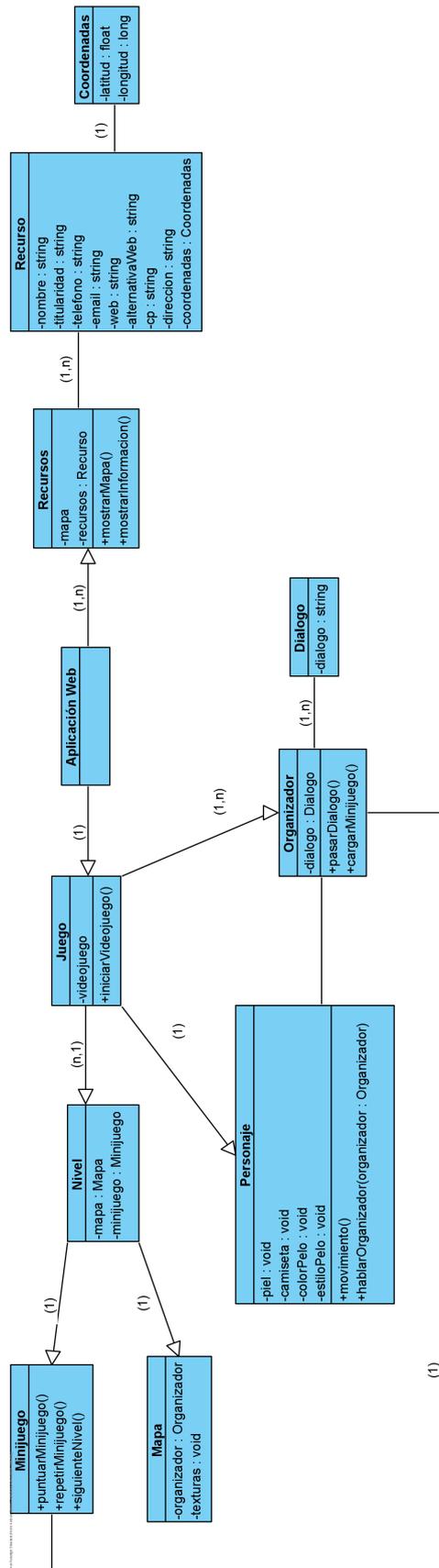
Apéndice

# A.1. Diagrama de Gantt



Página 1

## A.2. Diagrama de clases





# Bibliografía

- [1] Mirtha Moran. La Agenda para el Desarrollo Sostenible. <https://www.un.org/sustainabledevelopment/es/development-agenda/>, 05 2021.
- [2] INFOCIUTAT | Ajuntament de Valencia - València. <https://www.valencia.es/web/guest/val-infociutat/-/categories/37663>.
- [3] Direcciones y teléfonos - Ayuntamiento de Madrid. <https://www.madrid.es/portales/munimadrid/es/Inicio/Igualdad-y-diversidad/Direcciones-y-telefonos/?vgnextfmt=default&vgnnextchannel=ab79aaa26f535510VgnVCM1000008a4a900aRCRD>.
- [4] Mapa de entidades feministas | Mujeres y feminismos | Ayuntamiento de Barcelona. <https://ajuntament.barcelona.cat/dones/es/equidad-de-genero/mapa-de-entidades-feministas>, 10 2021.
- [5] Unity Technologies. Componentes Visuales - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/UIVisualComponents.html>, 2020.
- [6] Unity Technologies. Componentes de Interacción - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/UIInteractionComponents.html>, 2020.
- [7] Unity Technologies. El Asset del Animator Controller - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/Animator.html>, 2020.
- [8] Mark Jacob Thornton Otto. Bootstrap. <https://getbootstrap.com/>.
- [9] Leaflet — an open-source JavaScript library for interactive maps. <https://leafletjs.com/>.
- [10] Mirtha Moran. Igualdad de género y empoderamiento de la mujer. <https://www.un.org/sustainabledevelopment/es/gender-equality/>, 06 2020.
- [11] Instituto de la Mujer y para la Igualdad de Oportunidades - Noticias y novedades. <https://www.inmujeres.gob.es/actualidad/noticias/2021/OCTUBRE/informeEIGE.htm>.
- [12] Instituto de la Mujer y para la Igualdad de Oportunidades - Conócenos. <https://www.inmujeres.gob.es/elInstituto/conocenos/home.htm>.
- [13] Un poco de historia. <https://www.unwomen.org/es/csw/brief-history>.
- [14] Mirtha Moran. Reducir las desigualdades entre países y dentro de ellos. <https://www.un.org/sustainabledevelopment/es/inequality/>, 12 2020.
- [15] L. Ayala and O. Cantó. Radiografía de medio siglo de desigualdad en España. [https://elobservatoriosocial.fundacionlacaixa.org/documents/22890/492074/T01\\_ID\\_ES\\_AyalaCant%C3%B3.pdf/a0746431-109f-e009-6c77-296c378f0438?t=1642072938395](https://elobservatoriosocial.fundacionlacaixa.org/documents/22890/492074/T01_ID_ES_AyalaCant%C3%B3.pdf/a0746431-109f-e009-6c77-296c378f0438?t=1642072938395), 01 2022.

- 
- [16] HTML: Lenguaje de etiquetas de hipertexto | MDN. <https://developer.mozilla.org/es/docs/Web/HTML>, 04 2021.
- [17] aNieto2k Nieto. Las principales diferencias entre HTML5 y HTML4 | aNieto2K. <https://www.anieto2k.com/2007/06/16/las-principales-diferencias-entre-html5-y-html4/>, 06 2007.
- [18] CSS básico - Aprende sobre desarrollo web | MDN. [https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/CSS\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/CSS_basics), 12 2020.
- [19] CSS | MDN. <https://developer.mozilla.org/es/docs/Web/CSS>, 07 2021.
- [20] Sintaxis en JavaScript. <https://devcode.la/tutoriales/sintaxis-en-javascript/>, 09 2017.
- [21] JavaScript | MDN. <https://developer.mozilla.org/es/docs/Web/JavaScript>, 05 2022.
- [22] Iván Asensio. Qué es Unity y para qué sirve. <https://www.masterd.es/blog/que-es-unity-3d-tutorial>, 11 2019.
- [23] Unity Technologies. Experiencia de Juego en 2D - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/Overview2D.html>, 2020.
- [24] Unity Technologies. Empezar con el desarrollo de WebGL - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/webgl-gettingstarted.html>, 2020.
- [25] Unity Technologies. Visión General de Scripting - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/ScriptingConcepts.html>, 2020.
- [26] Unity Technologies. EventSystem (Sistema de Eventos) - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/EventSystem.html>, 2020.
- [27] Unity Technologies. Unity UI - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/com.unity.ugui.html>, 2020.
- [28] Unity Technologies. Canvas - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/UICanvas.html>, 2020.
- [29] Unity Technologies. Animator Controllers - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/AnimatorControllers.html>, 2020.
- [30] Unity Technologies. Clips de Animación (Animation Clips) - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/AnimationClips.html>, 2020.
- [31] Unity Technologies. Animación - Unity Manual. <https://docs.unity3d.com/es/2020.2/Manual/AnimationSection.html>, 2020.
- [32] H Butler, M Daly, A Doyle, Hobu Inc., and Cadcorp. RFC 7946 - The GeoJSON Format. <https://datatracker.ietf.org/doc/html/rfc7946>, 08 2016.
- [33] Ediciones El País. Esta es la edad a la que podré jubilarme para cobrar el 100% de la pensión. <https://cincodias.elpais.com/herramientas/calculadora-irpf/>, 07 2022.
- [34] PT Sans - Google Fonts. <https://fonts.google.com/specimen/PT+Sans#standard-styles>.
- [35] Minecraftia Font | dafont.com. <https://www.dafont.com/minecraftia.font>.

- 
- [36] ArMM1998. Zelda-like tilesets and sprites. <https://opengameart.org/content/zelda-like-tilesets-and-sprites>, 07 2017.
- [37] Heartlessdragoon. Game Boy Advance - Pokémon Emerald - Exterior Tileset - The Spriters Resource. [https://www.spriters-resource.com/game\\_boy\\_advance/pokemonemerald/sheet/61816/](https://www.spriters-resource.com/game_boy_advance/pokemonemerald/sheet/61816/).
- [38] PageSpeed Insights. <https://pagespeed.web.dev/>.
- [39] Pedro Antonio Ramos Guardiola. Recursos IGRD. <https://ludificacion.uv.es/wp-content/uploads/recursosIGRD/>, 07 2022.