

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERIA INFORMATICA

Curso Académico 2022/2023

Trabajo Fin de Grado

THE ZOMBIE WORLD: DESARROLLO DE UN VIDEOJUEGO INTERACTIVO GRUPAL

**Autor**: Iván del Pino Alonso

**Directores**:

Maximiliano Paredes Velasco

Lorena Rodríguez Calzada

*A mis amigos que he conocido a lo largo de este camino*

*por hacerlo mucho más llevadero.*

*A mi familia por todo el apoyo que me han dado*

*a lo largo de los años.*

*A mis tutores por haberme guiado en esta*

*recta final del camino.*

*Gracias.*

**Resumen**

A lo largo de esta memoria se explicará el proceso de desarrollo de una aplicación web didáctica.

Esta aplicación tiene como objetivo mejorar la enseñanza y el aprendizaje de los alumnos mediante un juego de tablero por equipos.

La mayoría de los alumnos a lo largo de nuestra carrera estudiantil (en el nivel que sea) nos hemos encontrado, alguna vez, con que el proceso de aprendizaje se nos puede hacer cuesta arriba debido a la forma de aprender o falta de motivación por una asignatura. Por este motivo, mediante un juego donde los diferentes alumnos se enfrentan en dos equipo respondiendo preguntas de la materia, se logra una mayor atención y motivación hacia la asignatura.

En este juego, hay tanto preguntas individuales (en las que el alumno tendrá que demostrar los conceptos aprendidos para apoyar a su equipo) como grupales (donde todos los estudiantes de un equipo tendrán que colaborar entre ellos para conseguir avanzar, de esta manera también estimulamos el trabajo en equipo) que tendrán que ir respondiendo para conseguir puntos para ganar.

El desarrollo de la aplicación se ha divido en tres partes principales.

Por un lado, los diferentes menús para que el alumno y el profesor se puedan registrar y buscar la información necesaria acerca de la aplicación.

La segunda parte fue diseñar la interfaz gráfica del tablero, que será donde se desarrolle toda la partida entre los alumnos.

Y por último, el desarrollo del funcionamiento del juego.

Para las dos primeras partes, se ha utilizado, sobre todos, HTML y CSS, aunque algunas funciones han tenido que ser implementadas con Javascript, y para la tercera parte, se ha utilizado por completo Javascript, que da la opción de poder manejar los diferentes elementos de la página y darles la funcionalidad que se requiera.

**Índice**

[Capítulo 1: Introducción 7](#_Toc122618109)

[1.1 Motivación 7](#_Toc122618110)

[1.2 Objetivos 7](#_Toc122618111)

[1.3 Firebase 8](#_Toc122618112)

[Capítulo 2: Fundamentos 9](#_Toc122618113)

[2.1 Tecnologías 9](#_Toc122618114)

[2.1.1 Lenguaje HTML 9](#_Toc122618115)

[2.1.2 Lenguaje CSS 10](#_Toc122618116)

[2.1.3 Lenguaje Javascript 10](#_Toc122618117)

[2.1.4 Visual Studio Code 11](#_Toc122618118)

[2.1.5 Firebase 12](#_Toc122618119)

[2.1.6 Justinmind 13](#_Toc122618120)

[2.1.7 Neocities 14](#_Toc122618121)

[2.1.8 GitHub 15](#_Toc122618122)

[Capítulo 3: Descripción de la aplicación 18](#_Toc122618123)

[3.1 Metodología de trabajo 18](#_Toc122618124)

[3.2 Inicio del proyecto 19](#_Toc122618125)

[3.3 Desarrollo y construcción 19](#_Toc122618126)

[3.3.1 Prototipo 20](#_Toc122618127)

[3.3.2 Firebase 21](#_Toc122618128)

# Capítulo 1: Introducción

## Motivación

La motivación de la que parte esta aplicación es el desarrollo de un apoyo a la enseñanza que sea estimulante para el alumno. Esto se puede conseguir incitando al alumno a sentirse parte de un equipo con el que tiene que estar implicado para obtener la victoria en el juego.

En el sistema actual de enseñanza (entendamos este como el que el profesor explica y los alumnos toman apuntes para evaluarse en un examen), en muchas ocasiones, tanto al profesor que puede sentir que está hablando solo, como para el alumno que no encuentra la motivación necesaria para estar atendiendo, se puede hacer complicado seguir el ritmo de la clase adecuadamente.

Esta aplicación busca solucionar este problema, ya que implica tanto al profesor como a los alumnos, que tienen que competir entre ellos, haciendo que esa competitividad les motive de cara al estudio.

## Objetivos

El desarrollo de la aplicación se ha dividido en varias partes, para que este sea más simple:

* Crear una clase de Javascript para los datos y los métodos que se van a utilizar para el uso de la base de datos.
* Crear la interfaz (HTML y CSS) de los diferentes menús que hay en la aplicación.
* Implementar las diferentes páginas que aportan información a los usuarios.
* Dar funcionalidad a los menús con Javascript.
* Desarrollo de la interfaz del tablero.
* Aportar funcionalidad mediante Javascript al tablero, tanto para el funcionamiento de este como para la conexión con la base de datos.

## Firebase

Por último, se ha usado la aplicación Firebase para la base de datos que se utilizará en el proyecto, más concretamente la herramienta Cloud Firestore que nos ofrece Firebase.

Firebase es una plataforma en la nube que sirve de apoyo para el desarrollo de aplicaciones web y Android.

Esta plataforma nos ofrece los servicios:

* Authentication: autentica y administra usuarios.
* Cloud Firestore: actualizaciones en tiempo real y consultas de una base de datos.
* Crashlytics: prioriza y soluciona problemas de estabilidad.
* Performance: obtiene estadísticas sobre el rendimiento de la app.
* Cloud Messaging: atrae a usuarios.
* A/B Testing: mejora los flujos y las notificaciones clave.

Aunque se ha valorado el uso de otra de las herramientas mencionadas anteriormente (por ejemplo, el uso de mensajes que te ofrece Firebase, para que en las ocasiones que fueran necesarias, se pudiese mandar algún mensaje en específico a los usuarios) para el desarrollo de nuestra aplicación, finalmente solo se ha usado Cloud Firestore para el uso de la base de datos.

# Capítulo 2: Fundamentos

## 2.1 Tecnologías

A lo largo del proyecto, se han utilizado diversas tecnologías que voy a nombrar a continuación.

### 2.1.1 Lenguaje HTML

Es un lenguaje de marcado utilizado para la elaboración de páginas web. Es un estándar que define una estructura básica y un código para la definición del contenido de una página web. Se encuentra a cargo del World Wide Web Consortium (W3C).

Tim Berners-Lee definió HTML en 1990.

Basa su filosofía de desarrollo en la diferenciación, es decir, para añadir un elemento externo a la página, hay que hacer una referencia a la ubicación del elemento mediante texto.

Este lenguaje puede ser creado y modificado por cualquier editor de textos básicos, pero en este caso se ha usado el entorno de desarrollo de Visual Studio, como en el resto de lenguajes utilizados.



Ilustración 1 : Fragmento de código HTML

### 2.1.2 Lenguaje CSS

Es un lenguaje de diseño gráfico utilizado para definir y crear la presentación de un documento estructurado escrito en un lenguaje marcado (en nuestro caso, HTML). Los documentos con este lenguaje se denominan hojas de estilos.

Este lenguaje fue lanzado el 17 de diciembre de 1996.

Su función principal es la separación del contenido del documento de la interfaz que va a tener este. Con esta separación se busca mejorar la accesibilidad, proveer más flexibilidad y control. Además, este permite que varios documentos HTML diferentes puedan usar una misma hoja de estilos, evitando así duplicar código.

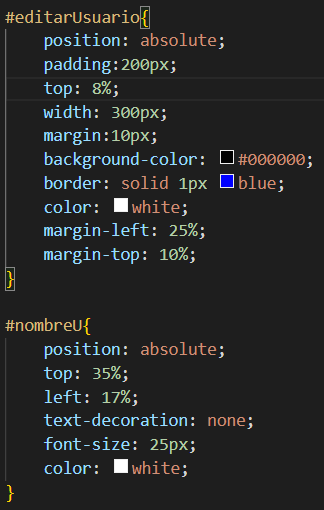


Ilustración 2 : Fragmento de código CSS

### 2.1.3 Lenguaje Javascript

Es un lenguaje de programación interpretado. Es definido como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Fue diseñado por Netscape Communications y la Fundación Mozilla, y lanzado el 4 de diciembre de 1995.

Es utilizado, principalmente, desde el cliente, y permite mejoras en la interfaz de usuario, así como la elaboración de página dinámicas.

Su sintaxis en muy parecida a la de los lenguajes de programación Java y C++.

El uso más común es para escribir funciones embebidas o incluidas en la página HTML, que interactúan con el DOM de la página.

En la aplicación se ha utilizado la librería JQuery, que aporta una forma simplificada de interactuar con los elementos HTML y manejar el DOM, entre otras funcionas.

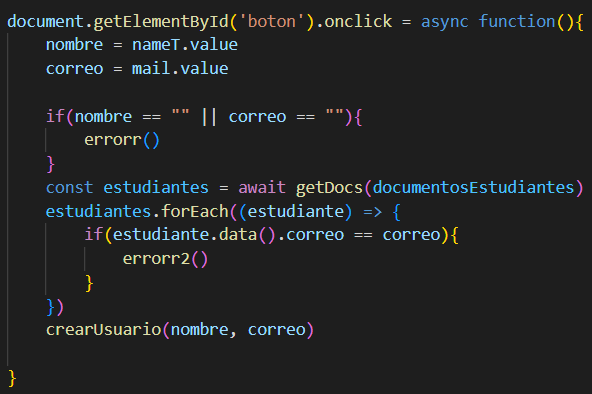


Ilustración 3 : Fragmento de código Javascript

### 2.1.4 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft y lanzado el 18 de noviembre de 2015. Es un software libre y multiplataforma que permite una gran colaboración con Git.

Cuanto con un depurador de código y permite la integración de numerosas extensiones, que entre otras cosas, te permiten poder desarrollar en cualquier lenguaje de programación.

Visual Studio fue construido sobre el framework Electron.

Tiene muchas características que no se pueden acceder desde sus menús. Para poder utilizarlas, este entorno ofrece una paleta de comandos.

Aunque recientemente se ha lanzado una versión web, en mi caso he utilizado la versión de escritorio.

En este entorno se ha instalado la extensión LiveServer, que permitía ejecutar un servidor localmente en el equipo.

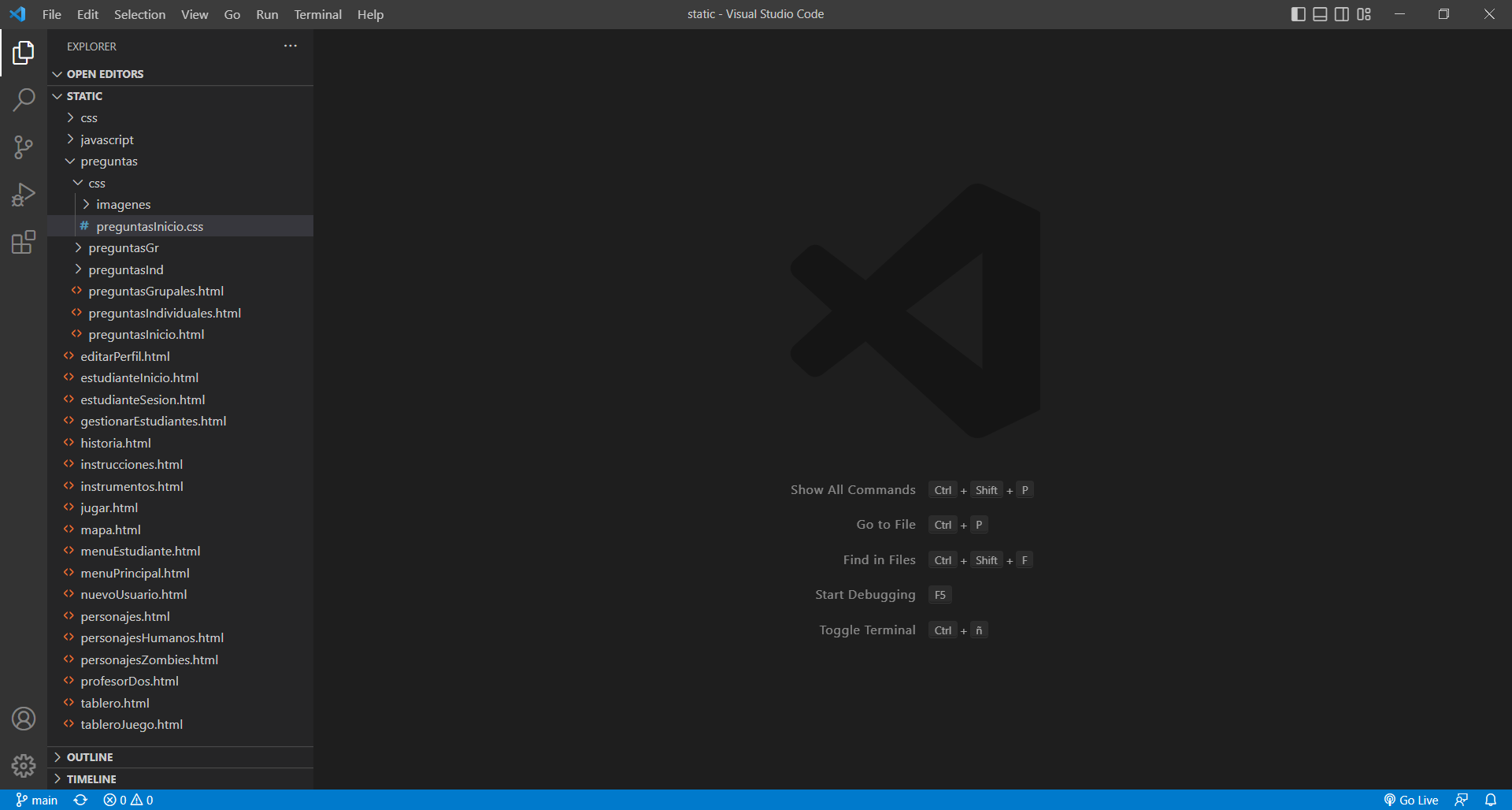


Ilustración 4 : Entorno de desarrollo Visual Studio Code

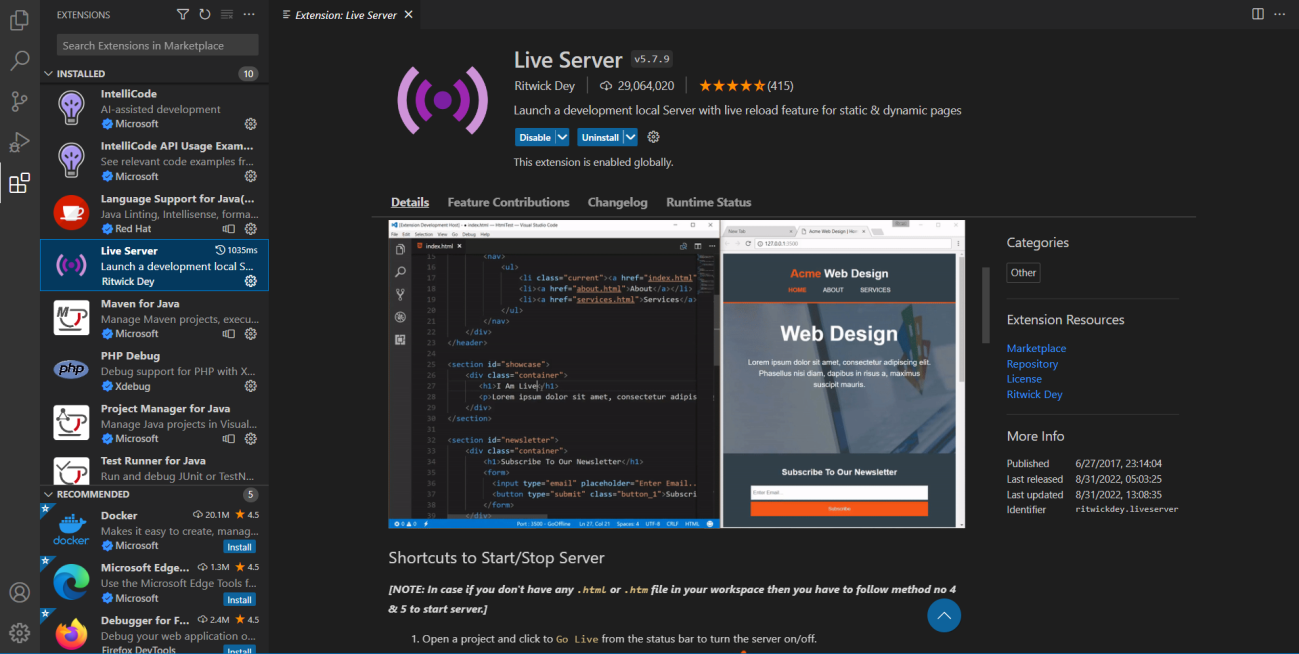


Ilustración 5 : Extensión LiveServer de Visual Studio Code

### 2.1.5 Firebase

Se trata de una plataforma en la nube para el desarrollo de aplicaciones.

Fue creada en 2011, y adquirida por Google en 2015.

Tiene diversas herramientas con usos variados, que al estar integradas en la misma plataforma, facilitan las tareas de gestión. Estas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Permite que los desarrolladores no tengan que fijarse tanto en el backend, tanto para el desarrollo como para el mantenimiento.

Entre sus ventajas están que permite compartir datos en tiempo real, envío de mensajes, tiene un único panel, soporte gratuito, una gran escalabilidad (en mi versión es gratuita y está limitado el uso diario de sus diferentes herramientas, pero se puede acceder a una versión de pago adaptado a las diferentes necesidades que tenga el desarrollador de una manera rápida y sencilla), entre otras.

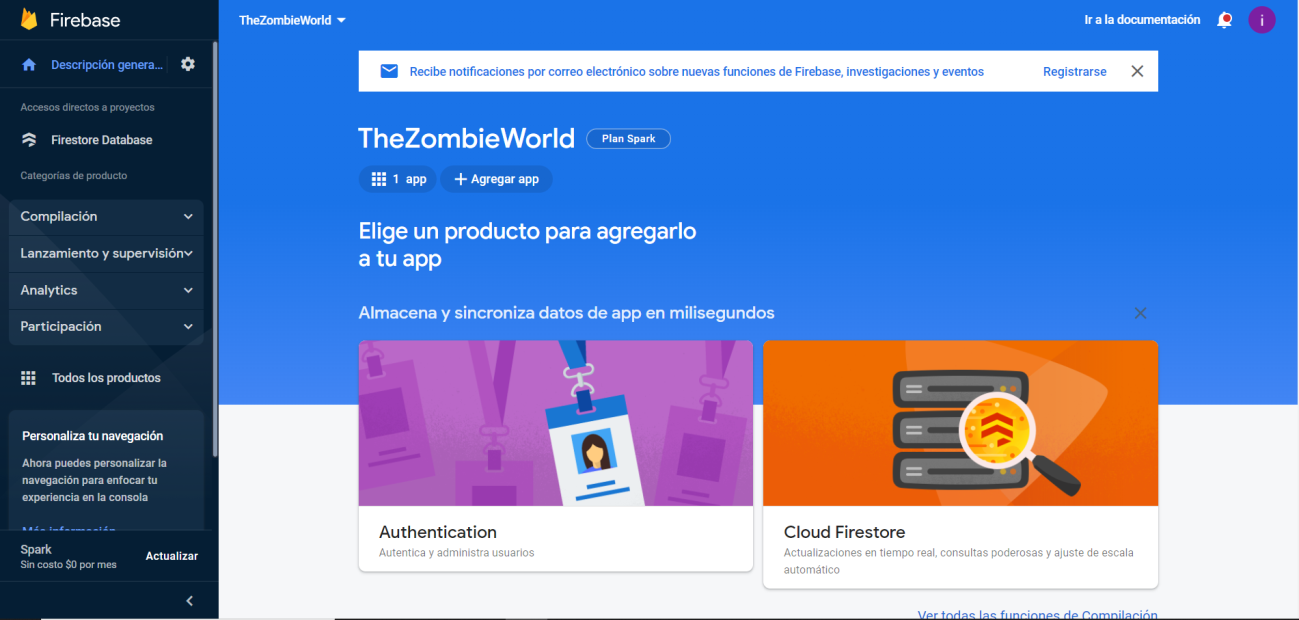


Ilustración 6 : Herramienta Firebase

### 2.1.6 Justinmind

Justinmind es una herramienta de prototipado de sitios web, aplicaciones software y aplicaciones móviles que puede trabajar en Windows, Mac, iOS y Android.

Uno de sus principales beneficios es que permite reducir en gran medida el coste de desarrollo de los proyectos, al permitir su simulación y solucionar fallos antes de entregarle el producto al cliente.

En mi caso, he utilizado la versión gratuita que ofrece este programa, y he utilizado un prototipo de navegador que nos ofrece.

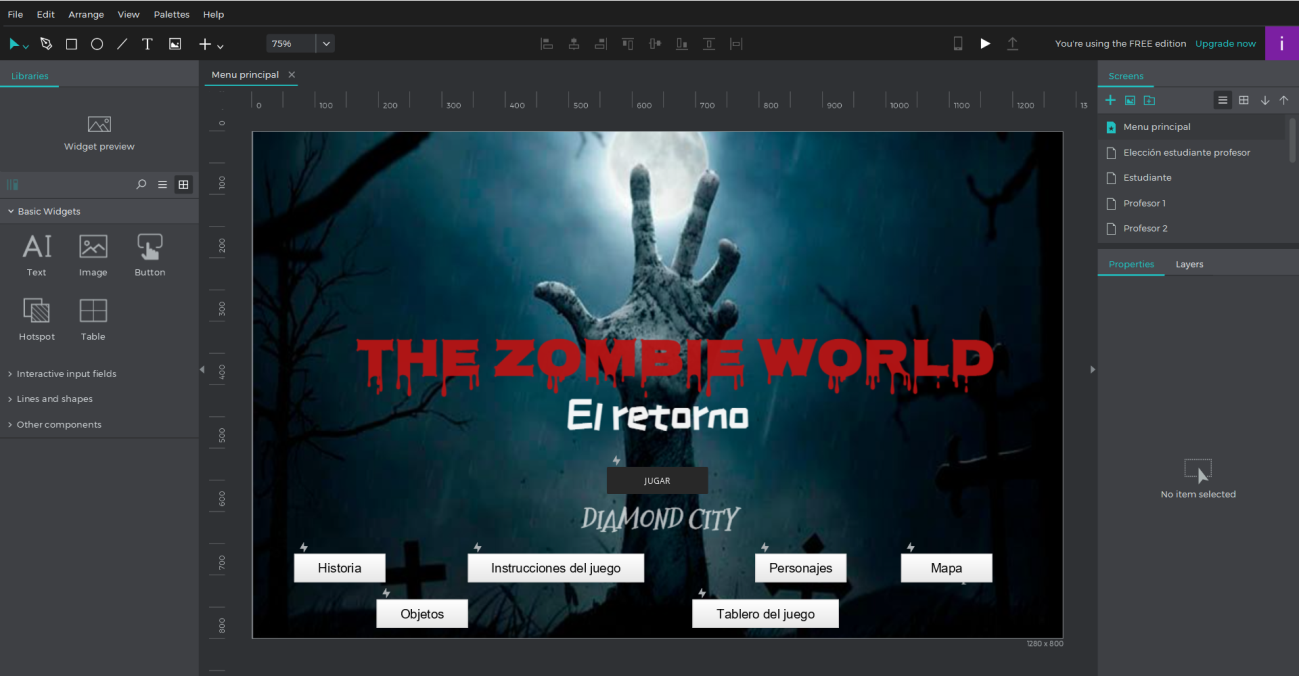


Ilustración 7 : Herramienta de prototipado Justinmind

### 2.1.7 Neocities

Es una web fundada en 2013 que permite crear webs gratis y alojarlas en sus servidores.

Tiene un plan gratuito y otros de pago, pero en mi caso he utilizado el plan gratuito que permite 1GB de almacenamiento y 200GB de ancho de banda.

Esta web nos da un dominio una vez hemos escogido un nombre de usuario y nos ofrece cursos gratuitos, pero en mi caso no han sido necesarios.

Permite programar dentro de la propia página, pero como ya estaba usando el entorno Visual Studio Code que me ofrece muchas facilidades que Neocities no me ofrece, solo lo he utilizado para poder tener la página web alojado en un servidor web y que sea su uso mucho más sencillo que al estar alojado de manera local.

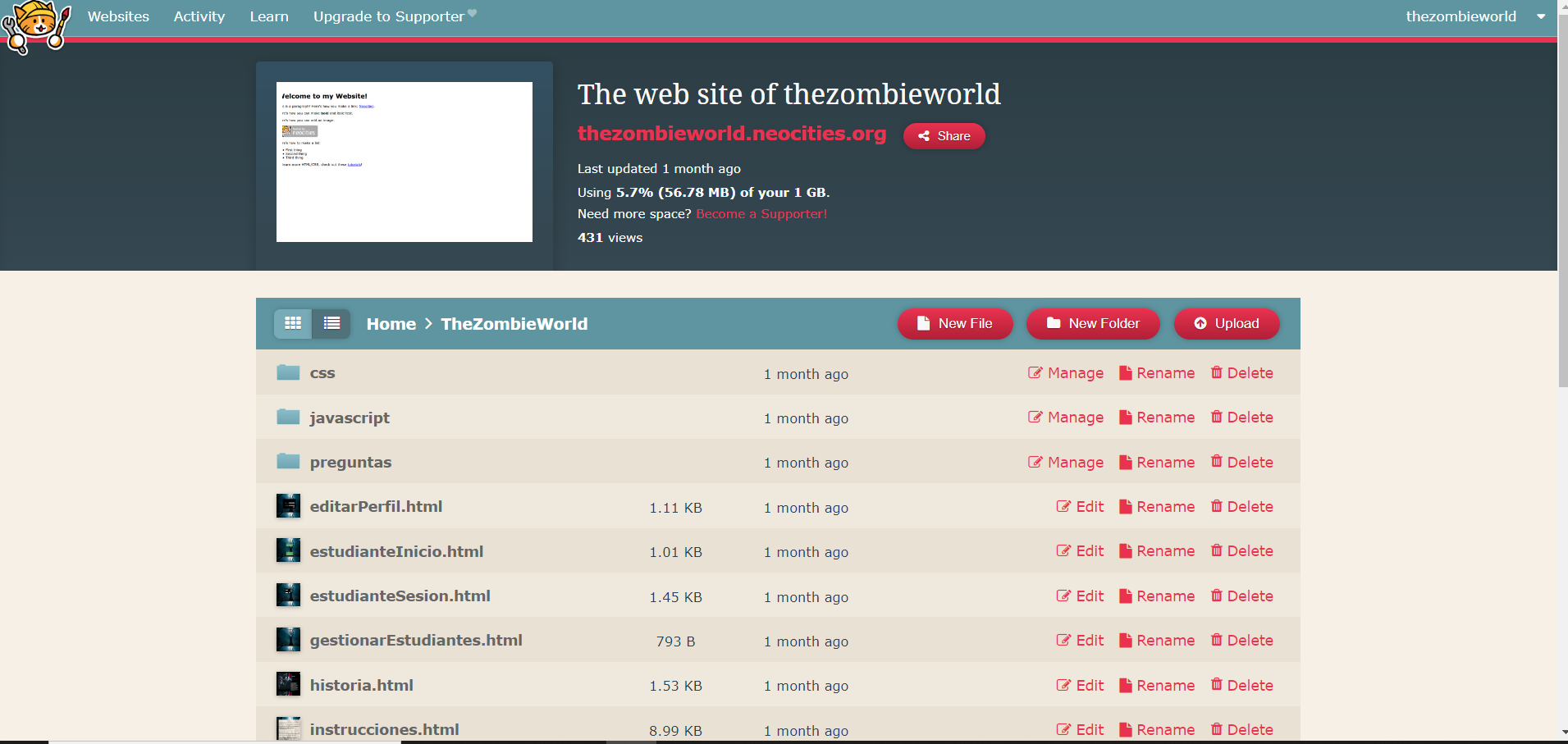


Ilustración 8 : Web Neocities

### 2.1.8 GitHub

GitHub es una plataforma web que sirve para crear proyectos abiertos de herramientas y aplicaciones. Se caracteriza por su capacidad colaborativa entre diferentes desarrolladores.

El código de un proyecto abierto puede ser descargado, lo que ayuda a crear diferentes ramificaciones.

Fue fundado el 8 de diciembre de 2008 y adquirido por Microsoft en junio de 2018.

Esta web utilizado el sistema de versiones Git diseñado por Linus Torvalds. Un sistema de gestión de versiones es ese con el que los desarrolladores pueden administrar su proyecto, ordenando el código de cada una de las nuevas versiones que de las aplicaciones. De esta manera, no se perderán los estados anteriores cuando se va a actualizar.

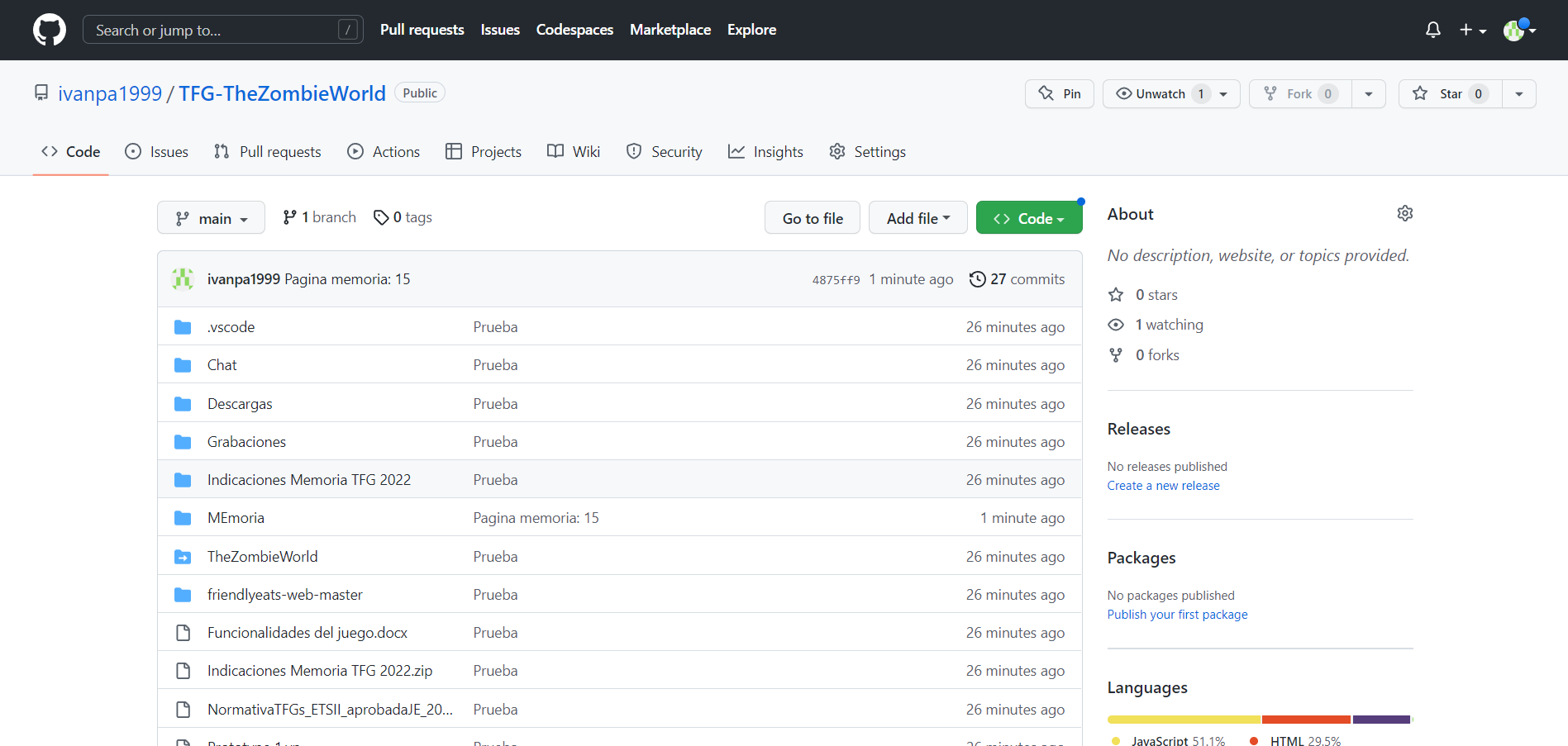


Ilustración 9 : Servicio Web GitHub

# Capítulo 3: Descripción de la aplicación

En este capítulo se hablará sobre el desarrollo de la aplicación, desde la metodología que se ha decidido seguir hasta el final del desarrollo del proyecto.

## 3.1 Metodología de trabajo

El desarrollo del proyecto se ha dividido en seis partes, empezando por un aprendizaje del uso de la plataforma Firebase, así como la creación de una clase Javascript para facilitar el uso de esta.

A continuación, se desarrollo el prototipo de la interfaz con la aplicación Justinmind, dotándole de cierta funcionalidad para ir avanzando entre las diferentes pestañas y poder encontrar de esta manera fallos antes de realizar la implementación.

En tercer lugar, se desarrollo la interfaz de los menús y demás páginas (exceptuando al tablero), aportándoles únicamente la funcionalidad de poder ir avanzando entre las diferentes páginas.

Después, se implementó la interfaz del tablero, sin ningún tipo de funcionalidad.

La quinta fase fue dotar de funcionalidades a los menús, ya que estos tenían que ser capaces de registrar usuarios, permitir iniciar sesión a aquellos usuarios ya registrados, dotar de cierta seguridad al apartado del profesor para que ningún alumno pudisese acceder como profesor, aparte de cargar ciertos datos que se iban a utilizar en el tablero.

Por último, se desarrolló la funcionalidad del tablero, que es la responsable de que el juego pueda desarrollarse en sí, acatando las normas y dando visibilidad a todo lo que va ocurriendo en el juego.

Se eligió esta metodología debido a, exceptuando la cuarta y la quinta fase que podían alternar el orden entre ellas, el resto de fases dependían de las anteriores para poder estar operativas, o por lo menos parte de ellas, dando facilidad a la hora de realizar ciertas pruebas para comprobar el correcto funcionamiento de la parte que estaba siendo implementada.

## 3.2 Inicio del proyecto

Al principio del proyecto, se estuvieron planteando varias opciones para realizar el backend de este, pero finalmente se decantó por Firebase, debido a las múltiples funcionalidades que daba y la facilidad de uso que esta plataforma tenía. A la hora de trabajar con Firebase, se decidió que lo más óptimo y legible para utilizarlo era crear una clase en Javascript dedicada a tener los datos necesarios de esta plataforma para realizar la conexión y la implementación de las distintas funciones que iban a ser requeridas para acceder a los datos. De esta manera, en el resto de clases solo teníamos que importar las funciones que necesitáramos, sin necesidad de crear código duplicado ni tener que acceder a los datos dentro de estas.

A la hora de realizar el prototipo, también hubo varias alternativas, donde las más importantes fueron Justinmind y Origami Studio, eligiendo finalmente por Justinmind por la simpleza a la hora de trabajar que este ofrecía y los buenos resultados que daba.

En cuanto a Javascript, su tuvieron dudas sobre que bibliotecas incluir en el proyecto, siendo finalmente JQuery la única en ser introducida debido a las funcionalidades y facilidades que ofrece, sobre todo, a la hora de hacer la página más dinámica.

Por último, se escogió Neocities ya que era una web fácil de usar para el alojamiento de servicios web y ofrecía las características que se necesitaban para el proyecto de forma gratuita.

## 3.3 Desarrollo y construcción

En este capítulo, se hablará de las diferentes partes que se han desarrollado para la implementación total de la aplicación, desde el prototipado hasta la realización del tablero con toda su funcionalidad.

El desarrollo de la aplicación en sí se ha divido en tres archivos por cada página diferente de la web, el archivo CSS, el de Javascript y el de HTML.

Para la parte de CSS se ha podido reutilizar, en varias ocasiones, un mismo archivo para varios HTML. De esta manera, se ha podido reutilizar código, aparte de simplificar el sistema de archivos de la aplicación bastante. Este método también se ha podido utilizar para los archivos Javascript, en menor medida debido a que, aunque por la parte estética que se encargaba CSS las página fueran iguales, en cuanto a la implementación de la funcionalidad, había bastante diferencia entre lo que las diferentes partes realizaban. Por último, los archivos HTML, que aunque seguían una estructura muy parecida muchos de ellos, no se podía reutilizar el código, por lo que para cada página hay un archivo diferente HTML.

Como se ha mencionado en capítulos anteriores, el desarrollo se podía dividir en tres partes principales, una primera en la que se ha preparado el prototipo y el archivo de Javascript que se iba a encargar de la conexión y el uso de Firebase, así como la creación de la base de datos de Firestore de Firebase.

La segunda parte consistiría en la creación de los archivos HTML y CSS de los menús y del tablero, y por último, se ha dotado de funcionalidad mediante archivos Javascript a los menús y al tablero. Hay que darle especial atención a esta parte, ya que era la encargada de controlar toda la dinámica del juego, desde asegurar que se cumpliesen las diferentes normas de este hasta que el turno de los jugadores fuese el correcto. Ha sido donde más llamadas a la base de datos se realizan.

Ahora, procederé a explicar más detalladamente todo el proceso que he tenido que realizar en el proyecto.

### 3.3.1 Prototipo

Al empezar el desarrollo del prototipo, se empezó analizando los diferentes apartados que la aplicación tenía que tener, así como la distribución de estos apartados a lo largo de la página (desde donde se podía acceder a otra parte de la web. Esto se realizó, recogiendo en un esquema sencillo, las diferentes subpáginas de la web con las conexiones y explicando brevemente lo que hacía cada una.

Una vez estaba claro, se realizo un boceto a mano de las diferentes pestañas que iba a tener, de manera simple pero mostrando, sobre todo, la diferente distribución de los elementos en las distintas paginas.

Al comprobar que todo estaba bien y quedaba bien distribuido, se paso al desarrollo del prototipo en la aplicación Justinmind, asegurando que las diferentes conexiones entre las pestañas estuvieran correctas. En esta parte se pudo comprobar mejor como iba a quedar la aplicación final, ya que el prototipo se hizo pensando en una página web y al realizar las pruebas se pudo observar en la pantalla como se vería aunque los detalles todavía no estaban definidos, así se podía apreciar correctamente que la distribución de los elementos fuera la correcta para dar una mayor visibilidad y accesibilidad a esta.

Un ejemplo del prototipo sería lo que se muestra en la ilustración 10, donde se pueden apreciar dos botones, el de Estudiante y el de Profesor, y dos íconos, el de una casa y la interrogación, aunque los botones como los iconos cambiarían de imagen al llegar al desarrollo final, la distribución y función de estos sería el mismo que en el prototipo. En el botón de Estudiante se avanza a una pestaña donde se permite iniciar sesión o crear un nuevo usuario, y en el botón profesor a la pestaña profesor, donde podría gestionar diferentes aspectos de la partida. Al pulsar la casa, como se utiliza típicamente en aplicaciones de este estilo este icono, se accedería al menú principal, y en la interrogación se abriría un mensaje donde se mostraría información necesaria para esa pestaña.

Todo esto, aparte de estar implementado en la aplicación final de manera más visual, Justinmind ha permitido también representarlo en el prototipo, pudiéndose mover entre las diferentes páginas solo pulsando el botón.

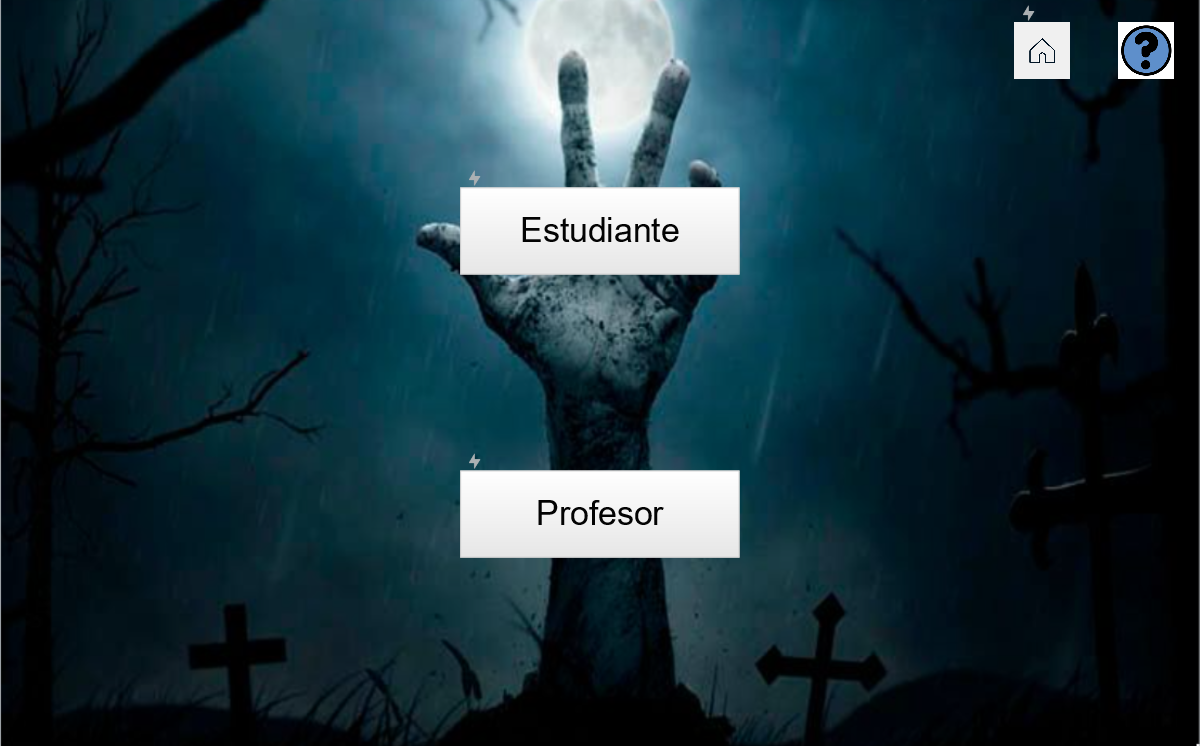


Ilustración 10 : Menú elección estudiante o profesor del prototipo

### 3.3.2 Firebase

Durante este proceso, primero tuve que mirar tutoriales y la documentación de esta plataforma, ya que era algo totalmente nuevo para mí y necesitaba que me pudiera familiarizar con el manejo de sus datos y aprender las diferentes opciones que esta me aportaba para poder decidir como la iba a utilizar dentro de la aplicación.

Una vez aprendido las diferentes aplicaciones que me ofrecía, me decidí por utilizar únicamente la aplicación Firestore, que es una base de datos en tiempo real donde iba a almacenar a los datos de los estudiantes, objetos y aquellos que estuvieran relacionados con el correcto funcionamiento de la partida. También pensé en utilizar una aplicación que ofrecía Firebase que permitía el envío de mensajes a diferentes equipos, pero recapacité al ver que una de las funciones de la base de datos permitía realizar una acción al detectar un cambio en la base, pudiendo así ahorrar el mensaje.

Esto se implementó en la clase Javascript firebase.js, donde se pueden encontrar los datos necesarios para realizar la conexión con Firebase y los diferentes métodos que se usaron a la hora de acceder en la diferentes clases a la base de datos. Para poder usar estos métodos fuera de esta clase, tuve que importar en la clase que se iba a querer utilizar cualquier método, el que quisiéramos usar.

De esta manera ahorre en código debido a que se evitaba la duplicación de código. Además, se evitó que los datos de acceso a la base de datos estuvieran visibles en diferentes clases, aparte de tener el proyecto más estructurado.

Los principales métodos de Firebase que se han usado han sido:

* SaveTasks: era el encargado de crear una colección nueva con los datos que se pasarán por argumentos. Principalmente se usa para la creación de nuevos usuarios.
* GetTasks: su función es la extracción de todos los datos de la base de datos. Este método se usaba cuando necesitabas usar más de uno de los elementos que estaban guardados en la colección.
* GetTask: su función era parecida a la de GetTasks, con la diferencia que este solo extraía uno de los elementos.
* OnGetTasks: este método se encargaba de realizar la acción que nosotros implementásemos al usarlo una vez hubiese un cambio dentro de la base de datos. Se ha usado, sobre todo, para poder avisar a los diferentes clientes que hubiesen conectados, de cambios una vez que la partida estaba iniciada (por ejemplo, del cambio de turno). De esta manera se evitaba tener que usar otro de los servicios que usaba Firebase.
* UpdateTask: se encarga de actualizar los datos de aquellas colecciones que ya estuviesen creadas, como por ejemplo, a la hora de sumar o restar puntos a alguno de los jugadores.

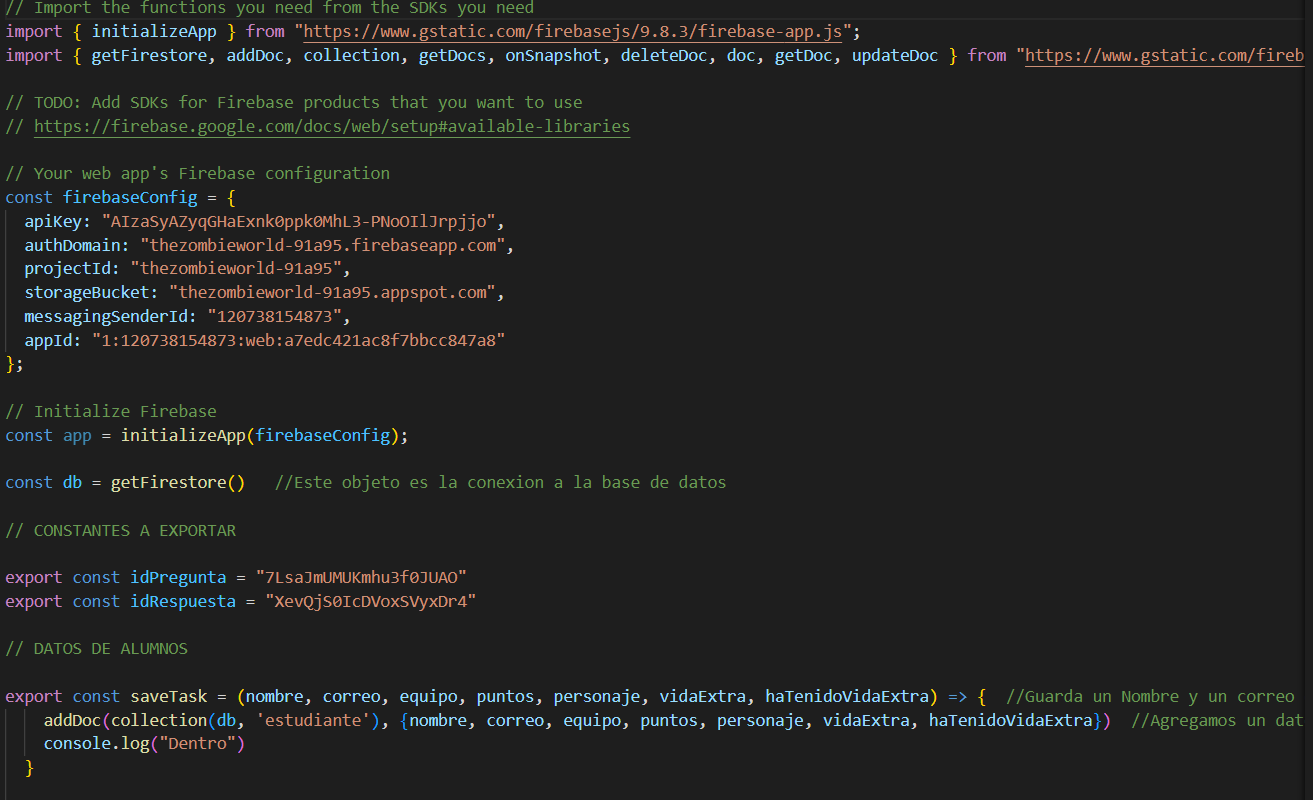


Ilustración 11 : Clase Firebase

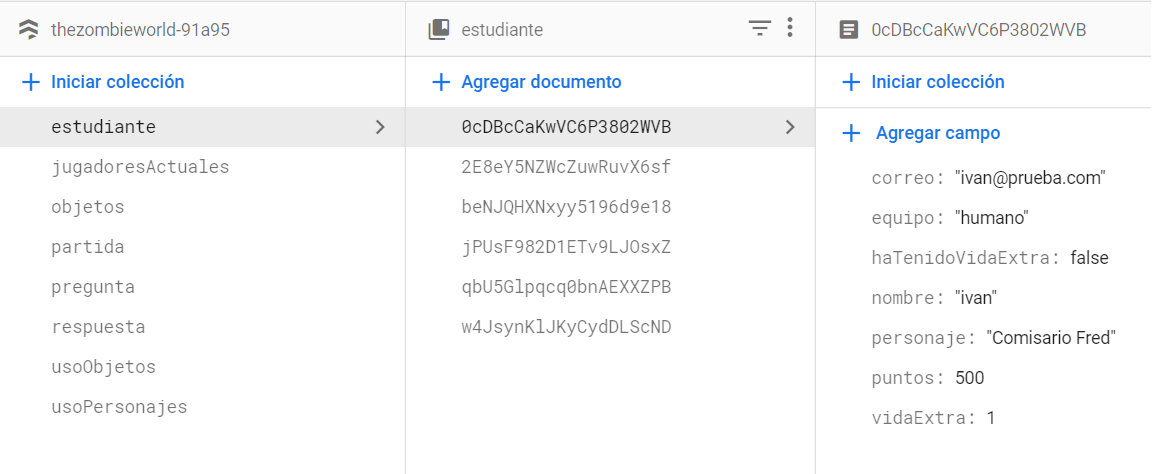


Ilustración 12 : Datos de Firestore

### 3.3.3 Comienzo del desarrollo de la interfaz

Una vez ya estaba elaborado el prototipo y la funcionalidad de Firebase implementada en el proyecto, el siguiente paso era decidir si continuar por la interfaz de los menús o la del tablero. Al ser más compleja la del tablero y parte de su funcionalidad iba a necesitar, para realizar pruebas, ya que algunas funcionalidades de los menús estuviesen ya implementadas, decidí empezar por la interfaz de los menús aunque no incluyese funcionalidad primero, ya que así podía tener una idea de cómo esta iba a ser incluida.

Empecé por el menú principal, ya que esta era la más básica y pretendía seguir un orden que empezase al arrancar la aplicación del juego e ir avanzando poco a poco por los menús.

Como se puede apreciar en la Ilustración 13, la página cuenta con dos apartados diferentes, la parte inferior izquierda, donde se encuentran las pestañas de Historia, Instrucciones del juego, Objetos, Personajes y Mapa, donde, pulsando a cualquiera de estos, se accede a otra pestaña que proporciona información acerca del juego, y la parte inferior derecha, que aparece jugar junto al botón de play. Pulsando aquí accedes a la interfaz encargada de iniciar sesión dependiendo si eres profesor o estudiante.



Ilustración 13 : Menú Principal