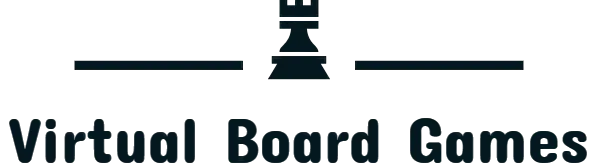
**ç**

José Guillermo Bejar Ccama

2º Desarrollo de Aplicaciones Web

Virtual board games



**Enlace Github del proyecto:**

<https://github.com/jose-guillermo/JoseGuillermoDAW>

**Enlace de la aplicación:**

[https://virtual-board-games.web.app](https://virtual-board-games.web.app/)

**Enlace donde descargar la memoria:**

<https://github.com/jose-guillermo/JoseGuillermoDAW/blob/main/memoria_proyecto.docx>

**Enlace concreto del manual de usuario:**

<https://github.com/jose-guillermo/JoseGuillermoDAW/blob/main/manual_usuario.docx>

**Usuarios registrados:**

**Usuario Administrador:**

Correo electrónico: [joseguille.jbc@gmail.com](mailto:joseguille.jbc@gmail.com)

Contraseña: Jose02

**Usuario normal:**

Correo electrónico: [jose100@gmail.com](mailto:jose100@gmail.com)

Contraseña: Jose02

**Datos para ejecutar la aplicación en local:**

Como tal la aplicación no está diseñada para ejecutarse de manera local, ya que las opciones de las cookies no permiten que se cree en un entorno que no sea https y el cors del backend está configurado para que solo se pueda utilizar a través de [https://virtual-board-games.web.app](https://virtual-board-games.web.app/).

El archivo php para crear la base de datos se encuentra en el backend en **db/inidb.php**, este archivo lee y ejecuta las instrucciones dentro de **db/inidb.sql**.

Índice

[Introducción justificada del proyecto 4](#_Toc191604555)

[1. Descripción del proyecto 5](#_Toc191604556)

[1.1 Características generales. 5](#_Toc191604557)

[1.2 Objetivos y alcance. 5](#_Toc191604558)

[2. Análisis del sector/mercado. 5](#_Toc191604559)

[2.1 Prospectiva del Título en el sector 5](#_Toc191604560)

[2.2 Evolución y tendencias del sector 6](#_Toc191604561)

[2.3 Normativa y documentación técnica específica 6](#_Toc191604562)

[3. Plan de ejecución 7](#_Toc191604563)

[3.1 Diagrama/cronograma de flujo de procesos (Diagrama de Gantt) 7](#_Toc191604564)

[3.2 Proceso de desarrollo software 7](#_Toc191604565)

[3.2.1 Fase de análisis 7](#_Toc191604566)

[3.2.1.1 Tipo de usuario 7](#_Toc191604567)

[3.2.1.2 Descripción de requisitos 7](#_Toc191604568)

[3.2.1.3 Casos de Uso 9](#_Toc191604569)

[3.2.1.4 Guía de estilo 10](#_Toc191604570)

[3.2.1.5 Prototipo del sitio web 11](#_Toc191604571)

[3.2.1.6 Mapa de navegación 17](#_Toc191604572)

[3.2.2 Fase de desarrollo 18](#_Toc191604573)

[3.2.2.1 Base de datos 18](#_Toc191604574)

[3.2.2.1.1 Análisis de requisitos de datos de la aplicación 18](#_Toc191604575)

[3.2.2.1.2 Diseño lógico de datos 19](#_Toc191604576)

[3.2.2.1.3 Paso del modelo lógico (E/R) al modelo relacional (tablas) 20](#_Toc191604577)

[3.2.2.1.4 Aplicación de reglas de normalización al modelo relacional 20](#_Toc191604578)

[3.2.2.1.5 Tipos de datos para el sistema gestor seleccionado 20](#_Toc191604579)

[3.2.2.1.6 Scripts de creación de tablas e inserciones iniciales 21](#_Toc191604580)

[3.2.2.2 Servidor 21](#_Toc191604581)

[3.2.2.2.1 Lista de funciones en php 21](#_Toc191604582)

[3.2.2.3 Cliente 22](#_Toc191604583)

[3.2.2.3.1 Diseño de la interfaz 22](#_Toc191604584)

[3.2.2.3.2 Accesibilidad 22](#_Toc191604585)

[3.2.2.3.3 Usabilidad 27](#_Toc191604586)

[3.2.2.3.4 Desarrollo web entorno cliente 28](#_Toc191604587)

[3.2.3 Fase de despliegue 30](#_Toc191604588)

[3.2.3.1 Despliegue utilizando un hosting 30](#_Toc191604589)

[3.3 Seguimiento y control de incidencias 32](#_Toc191604590)

[3.4 Indicadores de calidad de procesos 32](#_Toc191604591)

[4. Recursos materiales 33](#_Toc191604592)

[4.1 Inventario, valorado, de medios 33](#_Toc191604593)

[4.2 Presupuesto económico 33](#_Toc191604594)

[5. Recursos humanos 34](#_Toc191604595)

[5.1 Organización 34](#_Toc191604596)

[5.2 Contratación 35](#_Toc191604597)

[5.3 Prevención de riesgos laborales 37](#_Toc191604598)

[6. Viabilidad técnica 38](#_Toc191604599)

[6.1 Estudio de viabilidad técnica 38](#_Toc191604600)

[7. Viabilidad económico-financiera. 38](#_Toc191604601)

[7.1 Inversiones y gastos 38](#_Toc191604602)

[7.2 Financiación 38](#_Toc191604603)

[7.3 Viabilidad económica-financiera 39](#_Toc191604604)

[8. Conclusión 40](#_Toc191604605)

[Bibliografía/Webgrafía 41](#_Toc191604606)

[Anexos 41](#_Toc191604607)

[ANEXO 1: Tecnologías y herramientas de desarrollo 41](#_Toc191604608)

# Introducción justificada del proyecto

La razón por la que decidí hacer este proyecto es porque aplica muchos de los conocimientos que he adquirido en este grado superior. Por ejemplo:

* Manejo de bases de datos.
* Utilización de lenguajes como PHP, TypeScript, Html, CSS
* Manejo de usuarios.
* Manejo de formularios.
* Modificación del DOM.
* Desarrollo de una interfaz intuitiva.

Conocimientos que he adquirido durante mis prácticas. Por ejemplo:

* Utilización del framework Ionic basado en Angular.
* Desarrollo de una PWA (Progressive Web Application) que se pueda descargar y utilizar en un dispositivo móvil sin necesidad de entrar a la web.
* Utilización de JWT (Json Web Tokens).

Así como nuevos conocimientos que he tenido que aprender mientras desarrollaba el proyecto. Por ejemplo:

* Creación de una aplicación con soporte para varios idiomas.
* Creación de una aplicación que permita la interacción de varios usuarios en tiempo real.
* Utilización de PostgreSQL en la base de datos.
* Utilización de una aplicación de almacenamiento .de archivos externa para almacenar imágenes como es Cloudinary.

Por eso me interesa principalmente este proyecto, porque me permite utilizar conocimientos ya aprendidos para reforzarlos así como aprender nuevos conocimientos para seguir desarrollándome como programador web.

# Descripción del proyecto

## Características generales.

Virtual Board Game es una plataforma web diseñada para permitir a los usuarios disfrutar de juegos de mesa en línea. El proyecto ofrece una experiencia interactiva donde los usuarios pueden crear cuentas, conectarse con jugadores aleatorios a través de un sistema de emparejamiento en tiempo real.

Además, el sistema de logros recompensa la participación y el rendimiento de los jugadores. La aplicación incluye juegos populares como por ejemplo el ajedrez o las damas, con tableros y piezas personalizables.

Los usuarios también tienen acceso a una tienda virtual donde pueden personalizar sus juegos adquiriendo nuevos estilos de tableros, piezas. La plataforma estará disponible en dos idiomas, español e inglés, y ofrecerá un modo de visualización claro y oscuro para adaptarse a las preferencias de los usuarios.

## Objetivos y alcance.

El principal objetivo de Virtual Board Game es desarrollar una plataforma accesible y funcional para que los usuarios puedan disfrutar de juegos de mesa en línea de manera sencilla y atractiva. El proyecto busca implementar un sistema completo de gestión de usuarios que permita la creación de cuentas y el seguimiento de partidas. Además, se integrará un sistema de logros que motive la participación activa de los usuarios.

En cuanto a la personalización, la plataforma ofrecerá opciones de idioma y tema visual, con la posibilidad de comprar estilos de tableros y piezas dentro de la tienda.

En cuanto a las tecnologías que la aplicación utilizará el framework **Ionic basado en Angular** para el frontend, **PHP** para el backend, **PostgreSQL** para la base de datos, utilizará **Cloudinary** para almacenar las imágenes que se suban a mi aplicación de manera dinámica, también utilizará **Realtime Database de Firebase** para gestionar la interacción en tiempo real de los usuarios como por ejemplo a la hora de buscar partida o en el juego online.

# Análisis del sector/mercado.

## Prospectiva del Título en el sector

El Ciclo Formativo de Grado Superior en Desarrollo de Aplicaciones Web (DAW) se enmarca en el sector de las Tecnologías de la Información y la Comunicación (TIC), un ámbito en constante evolución y con una demanda creciente de profesionales cualificados.

En cuanto a la prospectiva para el sector de desarrollo web, las previsiones apuntan a un continuo crecimiento impulsado por la adopción de tecnologías emergentes. A partir de 2025, la inteligencia artificial (IA) se convertirá en un componente esencial en el desarrollo de aplicaciones, desde la automatización de tareas repetitivas hasta la creación de experiencias personalizadas. Además, la experiencia de usuario (UX) seguirá siendo una prioridad, con un enfoque en interfaces accesibles e intuitivas que mejoren la interacción con los usuarios​​

Otro aspecto importante es el auge de la computación en la nube y el edge computing, que permitirá una mayor eficiencia y reducción de costos en el desarrollo de aplicaciones web. A medida que más empresas adoptan soluciones basadas en la nube, la necesidad de mejorar la seguridad digital también será crucial, debido al aumento de las amenazas cibernéticas​

En cuanto a un desarrollador web, las habilidades relacionadas con la IA, la ciberseguridad y el diseño centrado en el usuario serán clave. Además, la capacidad para adaptarse a tecnologías como la computación en la nube y la colaboración interdisciplinaria será altamente valorada. La formación continua y la actualización en estas áreas serán esenciales para mantenerse competitivo en un entorno que evoluciona rápidamente

## Evolución y tendencias del sector

El V Estudio Empleabilidad y Talento Digital, presentado por la Fundación VASS y la Fundación de la Universidad Autónoma de Madrid (FUAM), destaca la evolución del sector tecnológico en España. El informe, elaborado con aportes de 60 expertos de 47 empresas, 29 universidades y más de 700 estudiantes, revela que el sector TIC español creció un 19.8% entre 2020 y 2023, alcanzando los 653.500 profesionales empleados a principios de 2023.

El estudio subraya que, pese a la incertidumbre global, el sector digital mantiene un papel central en la economía española, con una proyección de crecimiento del 57% en inversiones vinculadas a la transformación digital entre 2023 y 2026. Sin embargo, persiste un desajuste entre las competencias de los graduados universitarios y las demandas del mercado laboral, especialmente en áreas como inteligencia artificial, ciberseguridad y desarrollo de software.

Durante la presentación, Carme Artigas, exsecretaria de Estado de Digitalización e Inteligencia Artificial (2020-2023), enfatizó que el éxito de la digitalización en España depende de cerrar la brecha de talento y anticipar las tendencias tecnológicas. Representantes de empresas como KPMG, Vodafone y EY coincidieron en la necesidad de fortalecer colaboraciones público-privadas y políticas orientadas a la formación especializada.

## Normativa y documentación técnica específica

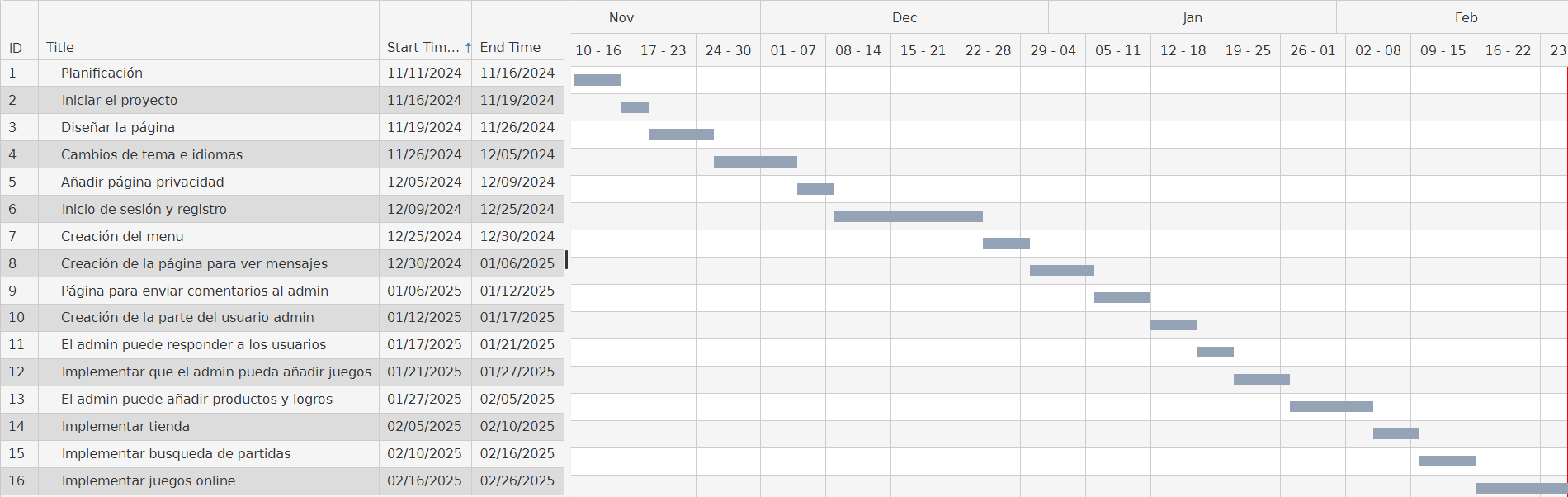
En cuanto a la normativa y documentación técnica específica relacionada con la accesibilidad web, una de las pautas clave es la WCAG (Web Content Accessibility Guidelines), que establece una serie de directrices para hacer que los sitios web sean accesibles para todas las personas, incluidas aquellas con discapacidades. Estas directrices son desarrolladas por el W3C y se dividen en tres niveles de conformidad: A, AA y AAA. El nivel AA es el más utilizado y se considera el estándar adecuado para la mayoría de los sitios web, proporcionando una experiencia accesible para una amplia variedad de usuarios​.

* Nivel A: Esto incluye aspectos como proporcionar texto alternativo para las imágenes no decorativas, asegurando que todos los elementos de navegación sean operables mediante el teclado, y garantizar que los videos tengan subtítulos.
* Nivel AA: Este nivel está diseñado para ser el estándar mínimo para proyectos que buscan alcanzar una buena accesibilidad web. Incluye todos los del A y también incluye otros criterios como el contraste de color más estricto entre el texto y sus fondos, la navegación más clara y predecible, y la compatibilidad con lectores de pantalla.
* Nivel AAA: Incluye todos los anteriores y además puede incluir criterios como la interpretación en lengua de señas para contenido en video, una guía más detallada de pronunciación para contenido textual y un mayor contraste para elementos visuales y de texto. Sin embargo, alcanzar el nivel AAA en su totalidad no es siempre práctico para todos los contenidos web, y no es considerado un requisito para la mayoría de los sitios web.

También existe la norma EN 301 549 que establece los requisitos de accesibilidad para productos y servicios de Tecnologías de la Información y la Comunicación (TIC) en la Unión Europea. Esta norma, que se aplica especialmente a entidades del sector público, busca garantizar que las plataformas digitales, como sitios web y aplicaciones, sean accesibles para personas con discapacidades. En ella se incluyen directrices relacionadas con la compatibilidad con tecnologías asistivas (como lectores de pantalla o magnificadores), la accesibilidad de documentos (como PDFs), y el cumplimiento de las WCAG 2.1 (Directrices de Accesibilidad para el Contenido Web). La norma cubre también hardware y software, y aboga por la creación de productos digitales robustos y operables para todas las personas, independientemente de sus habilidades o discapacidad.

# Plan de ejecución

## Diagrama/cronograma de flujo de procesos (Diagrama de Gantt)

****

## Proceso de desarrollo software

## Fase de análisis

## Tipo de usuario

**Usuario normal:**

Un usuario normal en la aplicación puede registrarse con su correo y una contraseña e iniciar sesión para acceder a todas las funciones del juego. Tiene la posibilidad de buscar partidas para encontrar oponentes automáticamente. Una vez dentro de una partida, puede jugar en línea interactuando con el juego, moviendo piezas o realizando acciones según las reglas de cada juego. Además, puede personalizar su experiencia comprando estilos de tableros y piezas en la tienda, así como desbloquear logros al completar ciertos desafíos dentro de los juegos.

**Usuario administrador:**

El usuario administrador no se puede registrar en la aplicación, es un usuario que se crea directamente en la base de datos y solo hay uno. El usuario administrador tiene funciones especiales para gestionar y mejorar la plataforma. Puede recibir y responder los mensajes enviados por los jugadores, permitiendo una comunicación directa para resolver dudas o recibir sugerencias. Además, tiene la capacidad de añadir nuevos juegos a la base de datos. También puede gestionar la tienda, añadiendo nuevos productos como tableros o piezas personalizadas y estableciendo sus precios. Otro de sus roles es la creación de logros para motivar a los jugadores y enriquecer la experiencia dentro de la aplicación.

## Descripción de requisitos

**Gestión de usuario:**

Los usuarios normales pueden registrarse en la plataforma utilizando su correo electrónico y una contraseña además de poder elegir una imagen de perfíl. Una vez registrados, pueden iniciar sesión para acceder a todas las funcionalidades disponibles. El administrador solo será un usuario que se crea al inicializar la base de datos, por su parte, tiene credenciales especiales que le permite acceder a herramientas exclusivas para gestionar la aplicación.

**Gestión de juegos:**

Los usuarios normales pueden buscar partidas para un juego en específico. Una vez dentro de una partida, pueden jugar en tiempo real con otros usuarios. Los administradores tienen la capacidad de añadir juegos a la base de datos, para que un juego sea jugable el desarrollador debe añadir el juego al código antes que el administrador lo añada a la base de datos.

**Personalización de los juegos:**

Los usuarios normales pueden personalizar su experiencia de juego modificando el diseño visual de los tableros y las piezas. Esto incluye cambiar colores, texturas y estilos que hayan adquirido en la tienda virtual, lo que les permite destacar y disfrutar de una experiencia única.

**Tienda virtual**:

Los usuarios normales pueden acceder a una tienda virtual donde pueden comprar productos como diseños exclusivos de tableros y piezas personalizadas. Una vez realizadas las compras, pueden gestionar su inventario y aplicar las personalizaciones adquiridas. Los administradores son responsables de publicar, editar o eliminar productos en la tienda.

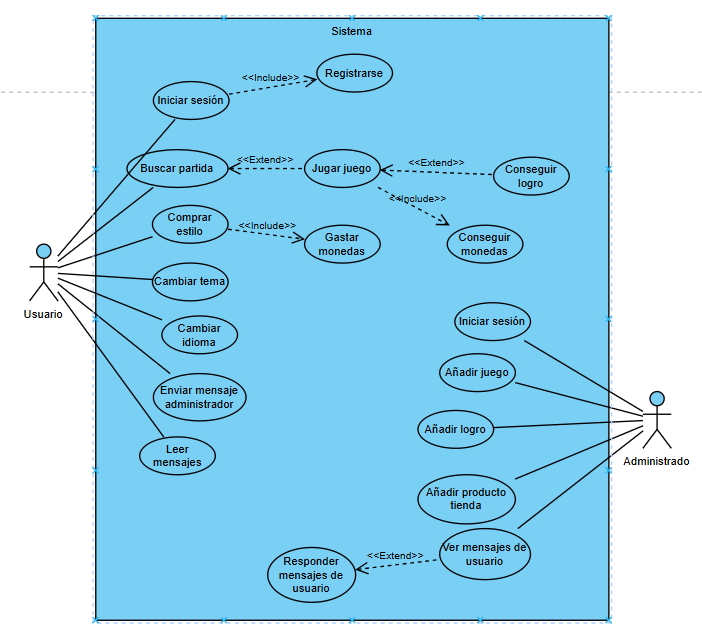
**Sistema de logros:**

Los usuarios normales pueden desbloquear logros al completar objetivos específicos, como ganar un número determinado de partidas o alcanzar ciertos hitos dentro de los juegos. Los administradores tienen la capacidad de añadir nuevos logros a la base de datos, el desarrollador debe incluir el logro en el código antes que el administrador lo añada a la base de datos.

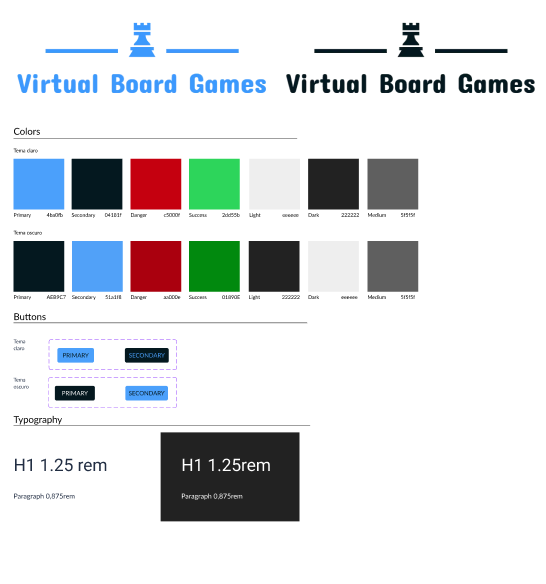
**Sistema de feedback:**

Los usuarios normales pueden enviar mensajes a los administradores para reportar errores, hacer sugerencias o simplemente dar su opinión sobre la aplicación. Los administradores, pueden responder a estos mensajes y gestionar las solicitudes de los usuarios, asegurando una comunicación fluida y efectiva.

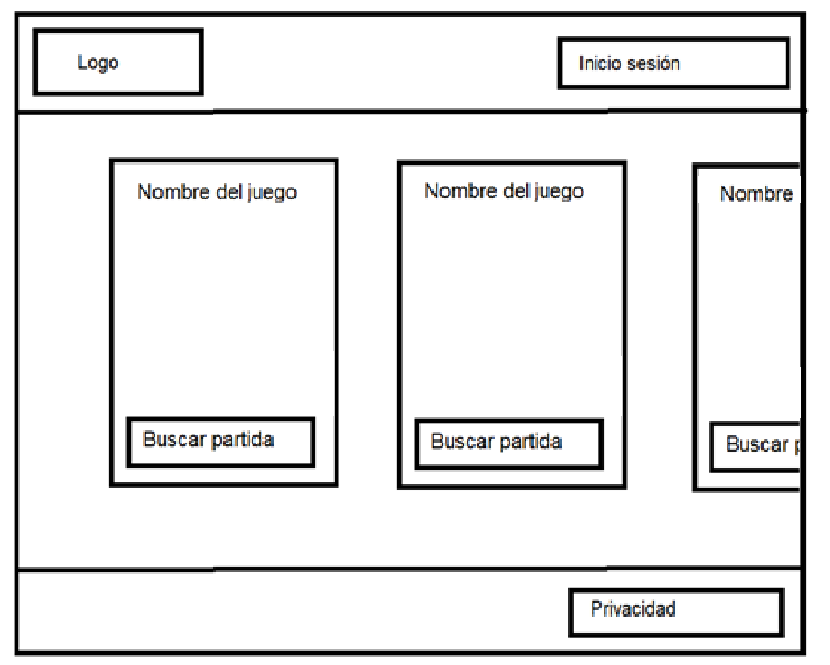
## Casos de Uso



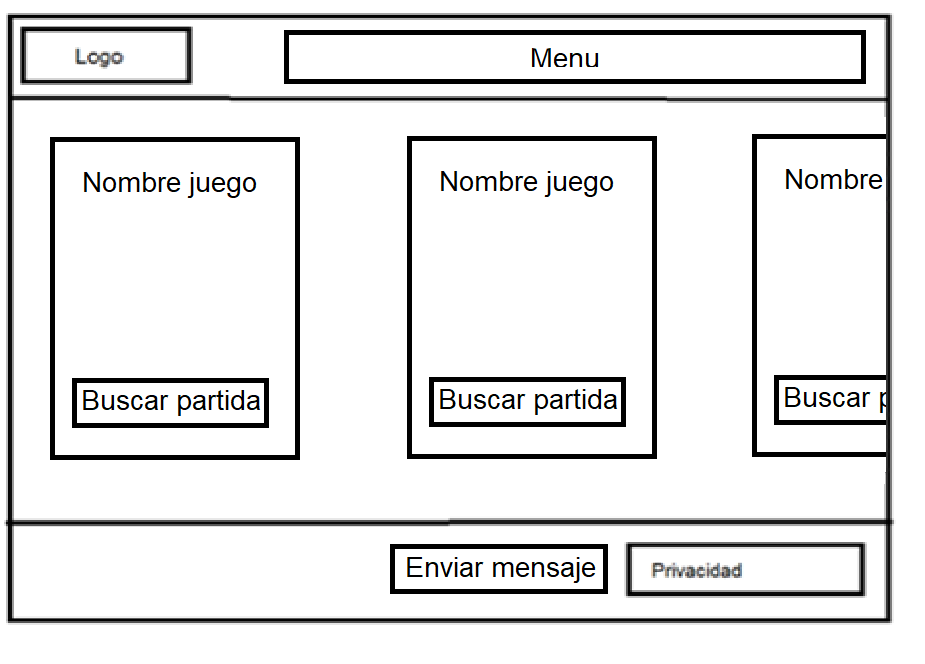
## Guía de estilo



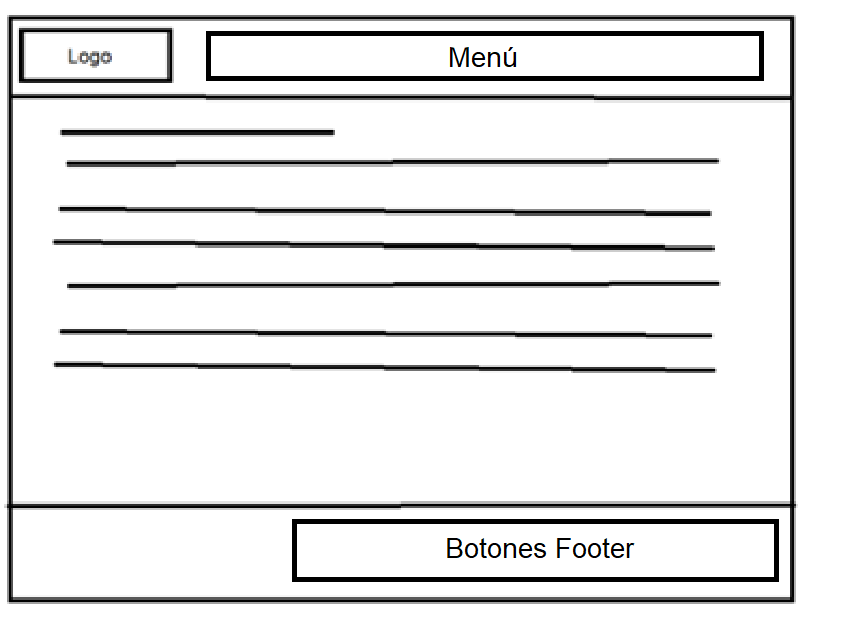
## Prototipo del sitio web



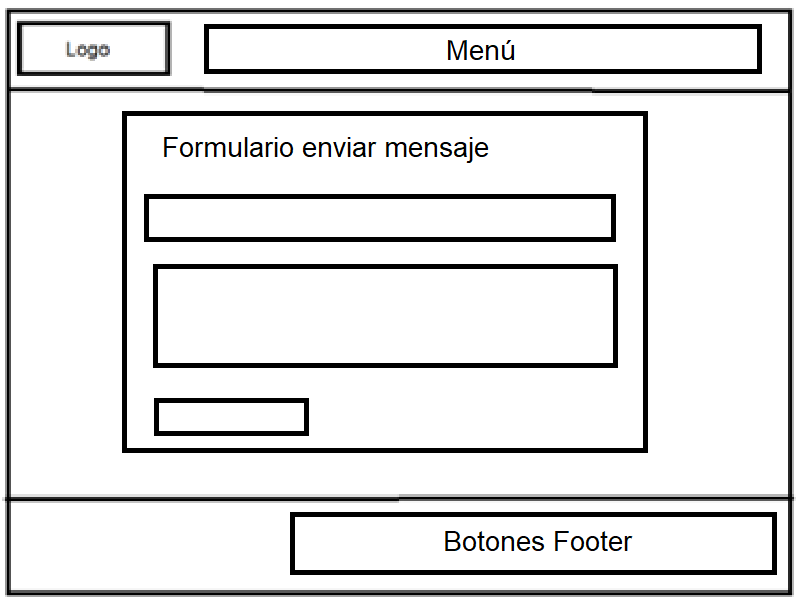
Esta será **la página de inicio que verán todos** antes de iniciar sesión en caso de que le den a otro botón que no sea iniciar sesión, les saldrá una notificación que les pedirá que inicien sesión



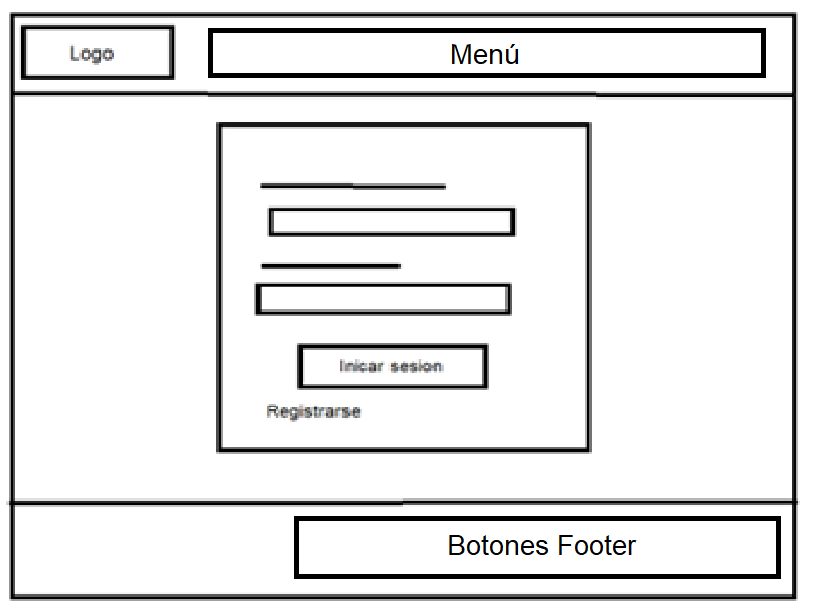
**Página de inicio de la aplicación**, cada usuario verá solo las páginas a las que pueda acceder, el logo te llevará siempre a inicio, el botón tema y el botón idioma son para que el usuario personalice como se verá la página, podrá elegir entre tema claro u oscuro y entre inglés y español.



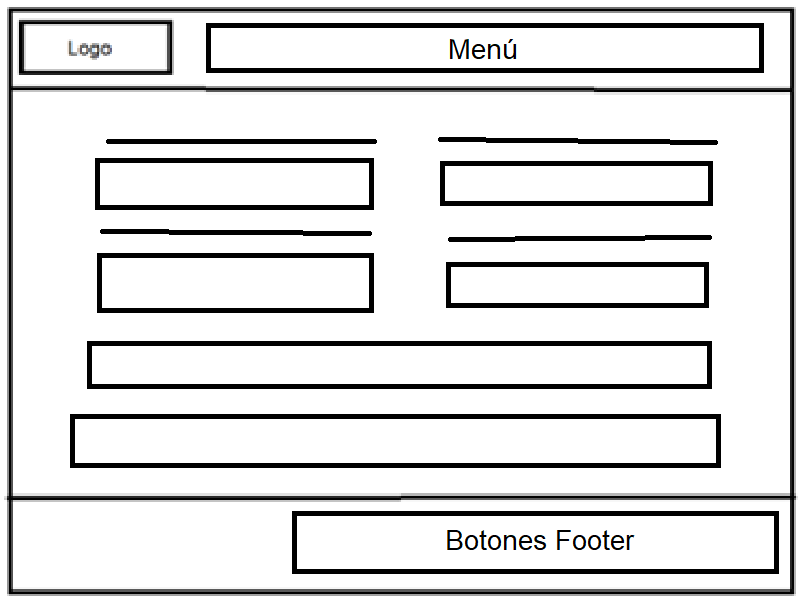
**Página de privacidad:** Explicará los términos de privacidad de la página, se accede mediante los botones del footer.



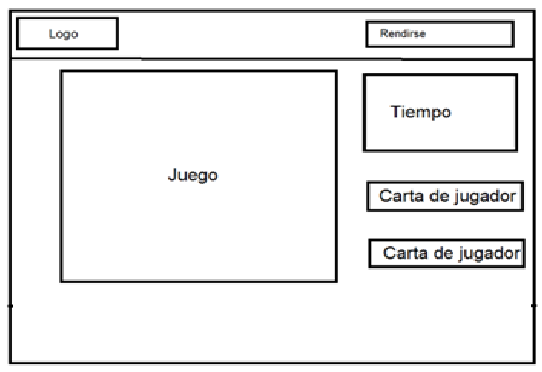
**Pagina para enviar mensaje al administrador:** Página donde los usuarios podrán enviar mensajes para reportar bugs o dar consejos para mejorar la página, se accederá mediante los botones en el footer.



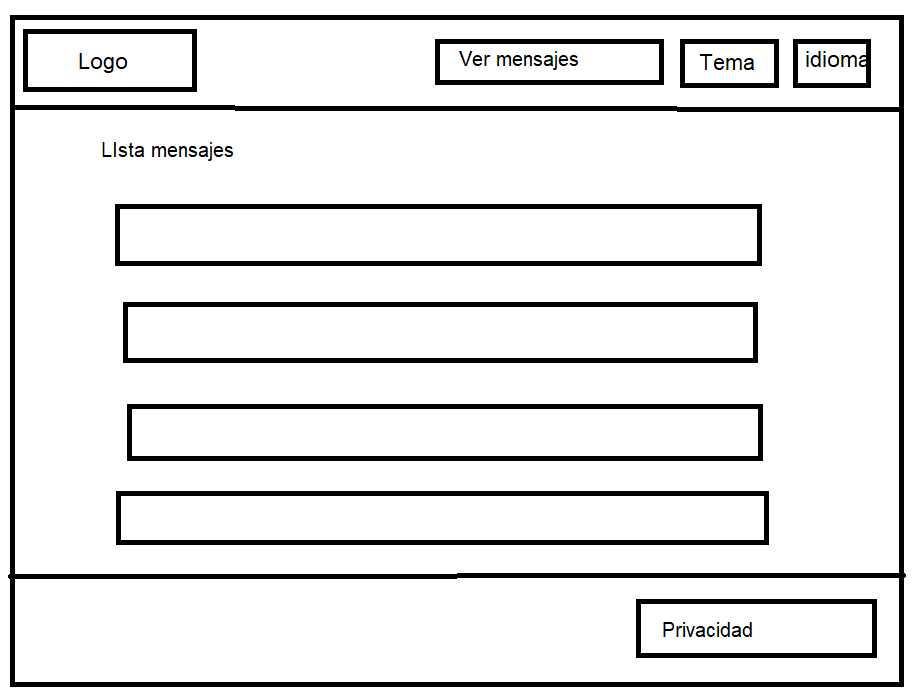
**Página de inicio sesión:** Página donde los usuarios podrán iniciar sesión y en caso de que no tengan cuenta podrán ir a la página de registro



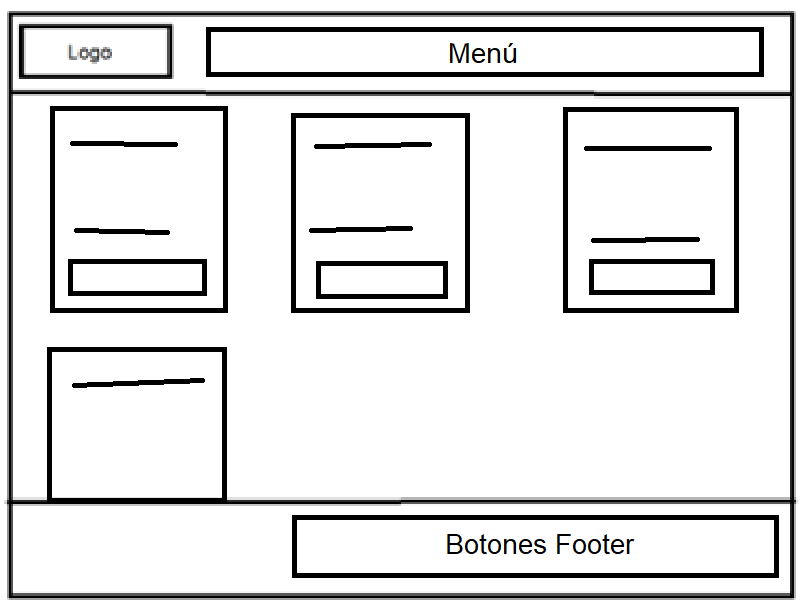
**Página de registro:** Página donde los usuarios podrán registrarse en la aplicación que se accede a través del menú de login



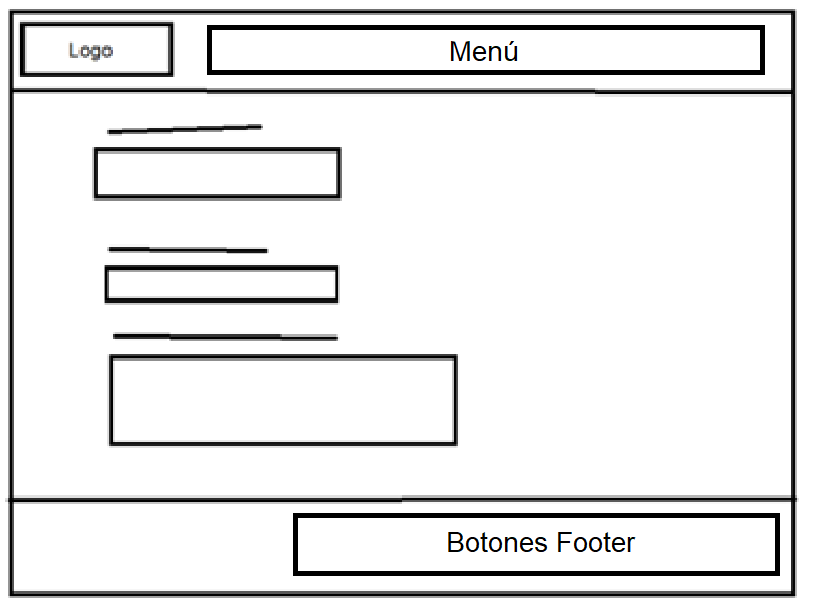
**Página de juego:** Página donde los jugadores podrán jugar, después de encontrar un oponente



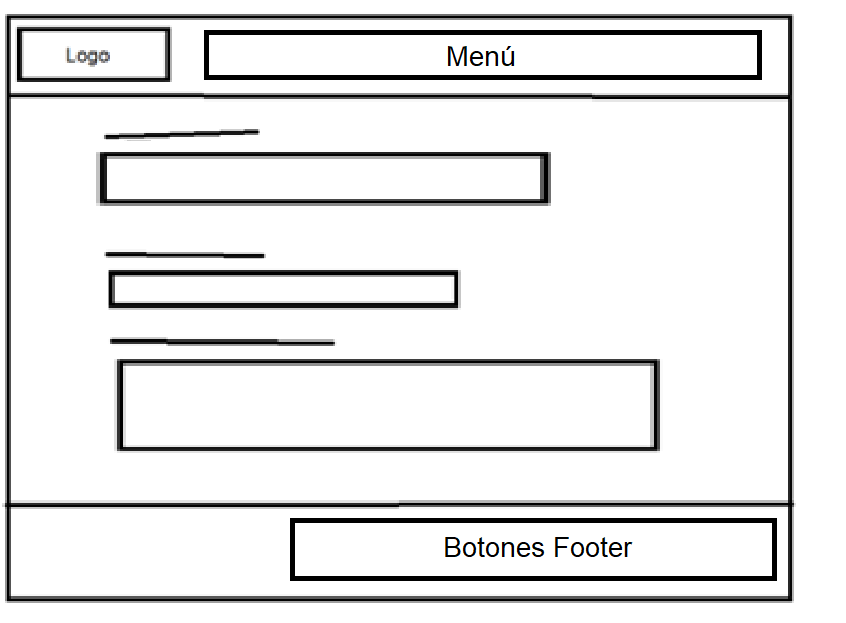
**Página del para ver mensajes:** Página que verán ambos usuarios cuando quieran ver los mensajes que se le han enviado, se accede a través del menú.



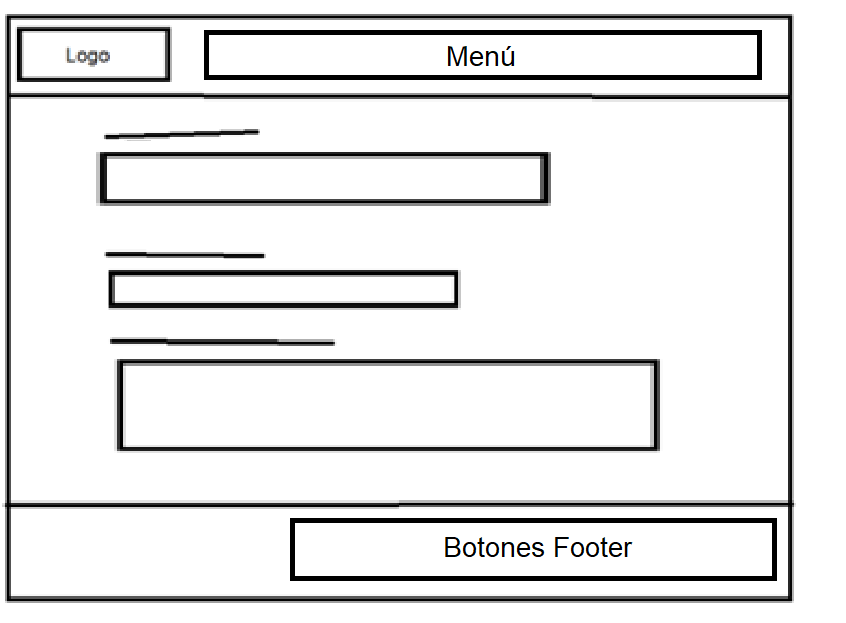
**Página de la tienda:** Página donde los usuario podrán gastar sus monedas para comprar aspectos para sus juegos



**Página agregar producto del administrador:** Página que verá el administrador cuando quiera añadir un nuevo producto a la tienda



**Página agregar logro del administrador:** Página donde el administrador podrá añadir un logro para los juegos.



**Página agregar juego del administrador:** Página donde el administrador podrá añadir un juego a la base de datos

## Mapa de navegación

Mapa de navegación del administrador:

Mapa de navegación del usuario:

## Fase de desarrollo

## Base de datos

## Análisis de requisitos de datos de la aplicación

**Usuario:**

Representa a cada jugador y al administrador dentro de la plataforma

Se relaciona con Partida a través de la relación Juega lo que indica que un usuario puede participar en múltiples partidas Se relaciona con Mensajes mediante Envía y tiene que permite la comunicación entre usuarios. Se relaciona con Logro a través de la relación Consigue lo que significa que un usuario puede obtener logros en los distintos juegos. También se relaciona con Producto mediante Compra lo que indica que los usuarios pueden adquirir artículos en la tienda

En cuanto a sus campos, tendrá: id de clave principal, el nombre de usuario que será único, el rol que indicará el rol en la aplicación, contraseña, email que también será único y la fecha de creación del usuario.

**Producto:**

Representa los artículos que los usuarios pueden comprar dentro de la tienda

Se relaciona con Usuario mediante Compra lo que indica que un usuario puede adquirir productos para personalizar su experiencia de juego y se relaciona con Juego mediante tiene ya que un producto pertenecerá a un juego

El producto tendrá: un id como clave primaria, el precio y el nombre del producto.

**Partida:**

Representa cada sesión de juego en la que participan los usuarios

Se relaciona con Usuario mediante Juega lo que significa que una partida debe tener exactamente dos jugadores Se relaciona con Juego mediante Es de que indica que cada partida pertenece a un juego específico.

De campos tendrá: un id como clave primaria y la fecha y hora en la que se celebró la partida.

**Juego:**

Representa los diferentes juegos de mesa disponibles en la plataforma

Se relaciona con Partida a través de Es de lo que indica que cada partida está basada en un juego. Se relaciona con Logro mediante Pertenece lo que significa que un juego puede contar con varios logros que los jugadores pueden conseguir y se relaciona con producto mediante Tiene ya que los productos pertenecen a un juego

De campos tendrá: un id como clave primaria, el nombre y las piezas que tiene el juego de mesa.

**Logro:**

Representa los logros que los jugadores pueden obtener al alcanzar ciertos objetivos

Se relaciona con Usuario mediante Consigue lo que indica que un usuario puede desbloquear varios logros. Se relaciona con Juego a través de Pertenece lo que significa que cada logro está asociado a un juego específico

Los campos de logro son: un id como clave primaria, un nombre, una recompensa que serán las monedas que el usuario obtendrá al conseguir el logro y el nivel que será la dificultad del logro.

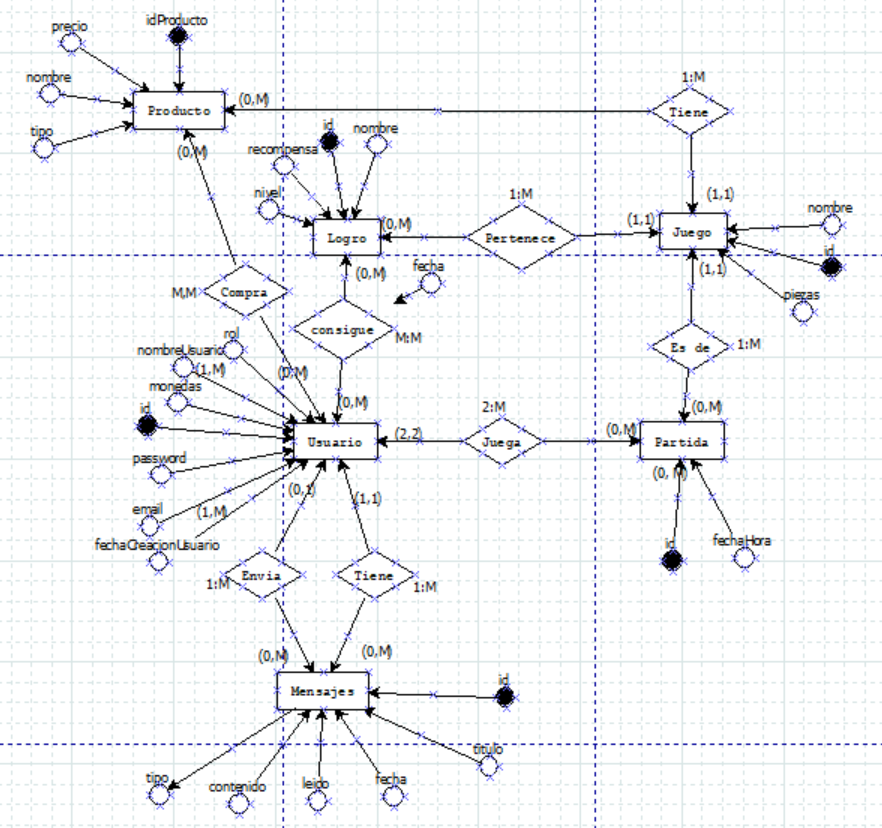
**Mensajes:**

Representa los mensajes enviados entre usuarios y administradores dentro de la plataforma

Se relaciona con Usuario a través de Envía lo que indica que un usuario puede mandar mensajes. También se relaciona con Usuario mediante Tiene lo que significa que un usuario puede recibir mensajes en su bandeja de entrada

Los campos de mensaje son: id como clave primaria, el título, la fecha, si se ha leído o no, el contenido y el tipo.

## Diseño lógico de datos



## Paso del modelo lógico (E/R) al modelo relacional (tablas)

**Usuarios:** (id, email, password, email, fechaCreacionUsuario, monedas, rol, nombreUsuario)

**Juegos:** (id, nombre, piezas)

**Productos:** (id, precio, nombre, tipo, ID\_JUEGO)

**Compras:** (ID\_USUARIO, ID\_PRODUCTO)

**Logros:** (id, nombre, recompensa, nivel, ID\_JUEGO)

**Logro\_Usuario:** (ID\_USUARIO, ID\_LOGRO, fechaObtención)

**Mensajes:** (id, titulo, fecha, leído, contenido, tipo, ID\_REMITENTE, ID\_DESTINATARIO)

**Partidas:** (id, fechaHora, ID\_JUGADOR1, ID\_JUGADOR2, ID\_JUEGO)

## Aplicación de reglas de normalización al modelo relacional

FN1

Todos los campos son atómicos, aunque puede parecer que piezas es un campo no atómico, ya que PostgreSQL permite arrays, piezas será un array de texto, por lo que será atómico

FN2

Si, todos los atributos dependen completamente de la clave primaria.

FN3

Si, no hay transitividades.

## Tipos de datos para el sistema gestor seleccionado

**Usuarios:**

Id(TEXT PRIMARY KEY), email(TEXT UNIQUE NOT NULL), password(TEXT NOT NULL), email(TEXT), fechaCreacionUsuario (BIGINT), monedas(INI), rol(TEXT que solo puede ser “user” o “admin”), nombreUsuario (TEXT UNIQUE NOT NULL)

**Juegos:**

id(TEXT PRIMARY KEY), nombre(TEXT UNIQUE NOT NULL), piezas(TEXT[] NOT NULL)

**Productos:**

Id(TEXT PRIMARY KEY), precio(INT NOT NULL), nombre (TEXT NOT NULL), tipo (TEXT NOT NULL), ID\_JUEGO(TEXT NOT NULL),

**Compras:**

ID\_USUARIO(TEXT PRIMARY KEY), ID\_PRODUCTO(TEXT PRIMARY KEY),

**Logros:**

id(TEXT PRIMARY KEY), nombre(TEXT NOT NULL), recompensa(INT), nivel(TEXT que solo puede ser ‘bronze’, ‘plata’, ‘oro’ o ‘diamante’), ID\_JUEGO(TEXT NOT NULL)

**Logro\_Usuario:**

ID\_USUARIO(TEXT PRIMARY KEY), ID\_LOGRO(TEXT PRIMARY KEY), fechaObtención (BIGINT)

**Mensajes:**

id(TEXT PRIMARY KEY), titulo(TEXT NOT NULL), fecha(BIGINT), leído(BOOLEAN), contenido(TEXT NOT NULL), tipo(TEXT NOT NULL), ID\_REMITENTE(TEXT NOT NULL), ID\_DESTINATARIO(TEXT NOT NULL)

**Partidas:**

id(TEXT PRIMARY KEY), fechaHora(BIGiNT), ID\_JUGADOR1(TEXT NOT NULL), ID\_JUGADOR2(TEXT NOT NULL), ID\_JUEGO(TEXT NOT NULL)

## Scripts de creación de tablas e inserciones iniciales

El archivo para inicializar la base de datos es **db/inidb.php** que lee el archivo **db/inidb.sql**  y lo ejecuta.

## Servidor

## Lista de funciones en php

**cors.php:** Establece los cors para que mi backend permita el acceso a mi frontend.

**cloudinary/conf.php:** Establece la conexión con cloudinary.

**cloudinary/upload\_img.php:** permite subir archivos de imagen a cloudinary.

**cloudinary/upload\_lang.php:** permite subir archivos json para las traducciones a cloudinary.

**session/createSession.php:** Crea la sesión en para un usuario.

**session/destroySession.php:** Destruye la sesión para que un usuario haga logout

**session/verify.php:** Verifica si un usuario tiene la sesión iniciada.

**db/conn.php:** Conecta a mi servidor con la base de datos.

**db/inidb.php:** Inicializa la base de datos leyendo inidb.sql.

**db/inidb.sql:** El script que tiene las instrucciones sql para inicializar la base de datos.

**db/achievement/achievements.php:** Busca y devuelve todos los logros de la base de datos.

**db/achievement/add.php:** Añade un logro nuevo a la base de datos.

**db/game/create.php:** Añade un nuevo juego a la base de datos.

**db/game/games.php:** Busca y devuelve todos los juegos de la base de datos.

**db/message/delete.php:** Elimina un mensaje en la base de datos.

**db/message/markAsRead.php:** Marca como leído un mensaje.

**db/message/messages.php:** Busca y devuelve todos los mensajes enviados a un usuario en específico dentro de la base de datos.

**db/message/send.php:** Envía un mensaje a un destinatario de parde de el remitente.

**db/product/add.php:** Añade un nuevo producto a la base de datos.

**db/product/defaultProduct.php:** Añade un producto por defecto a la base de datos, los productos por defecto son los aspectos que tienen los juegos por defecto por lo que todos los jugadores empiezan con este aspecto en sus juegos.

**db/product/products.php:** Devuelve todos los productos dentro de la base de datos.

**db/purchase/purchase.php:** Devuelve todos los ids de los productos que un usuario en específico ha comprado.

**db/user/check.php:** Comprueba que un usuario existe.

**db/user/create.php:** Registra un usuario en la base de datos.

**db/user/getUser.php:** Devulve todos los datos de un id de usuarios pasado.

**db/user/login.php:** Hace el login de un usuario en mi aplicación

**db/user/user.php:** Devulve el usuario que tiene la sesión iniciada actualmente.

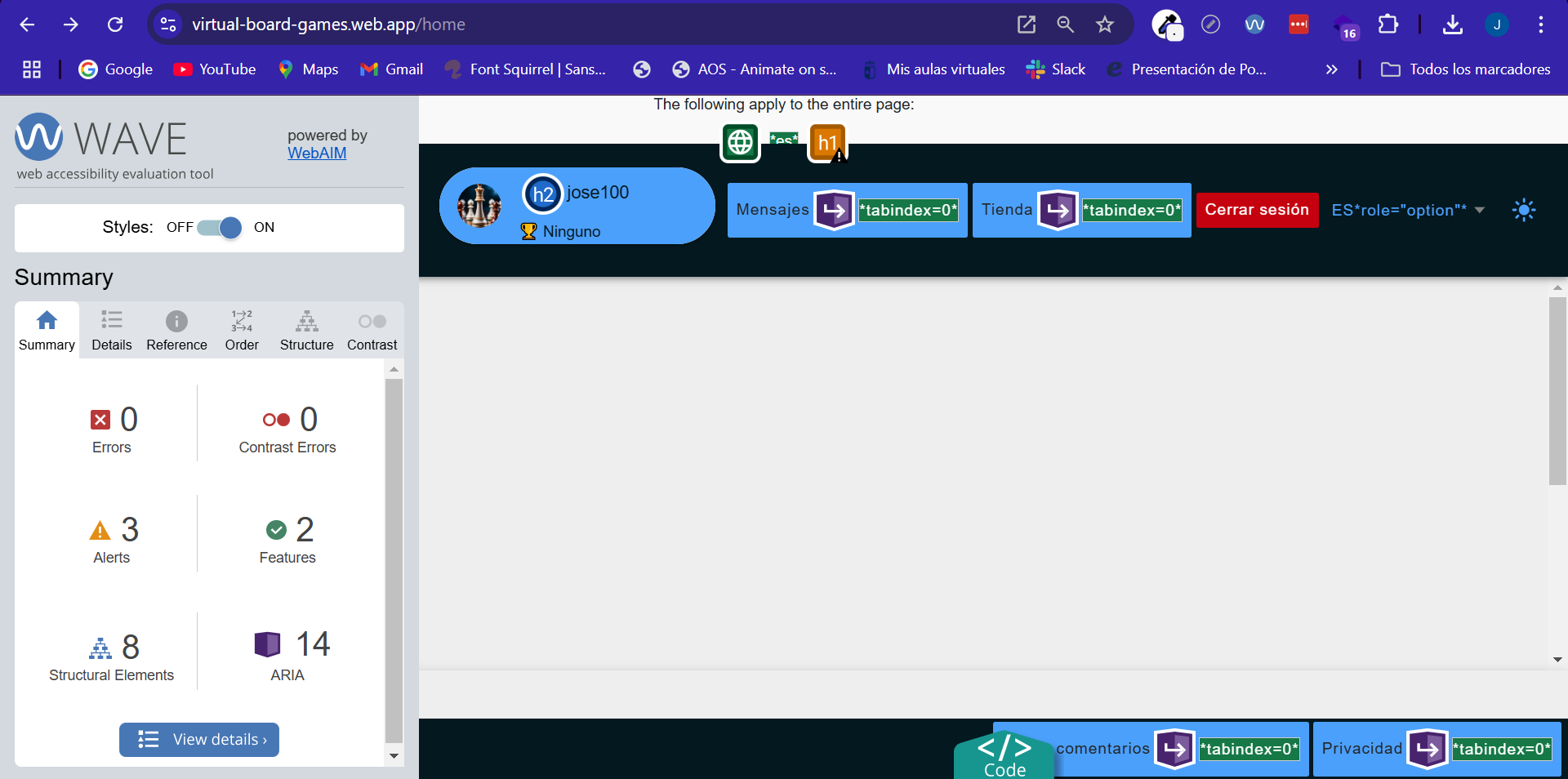
## Cliente

## Diseño de la interfaz

Una de las cosas más potentes que tiene Angular es que gracias a la división por componentes, en estos componentes los estilos solo afectan dentro de este componente por lo que es mucho más fácil dar estilos, no hay un archivo concreto que de estilo a toda la página ya que cada componente tiene el suyo propio, aunque hay un solo archivo que afecta de manera global aunque lo he utilizado poco que está en el **src/global.scss**, también está el archivo **src/theme/variables.scss** donde guardo las variables para establecer los temas claro y oscuro en mi página.

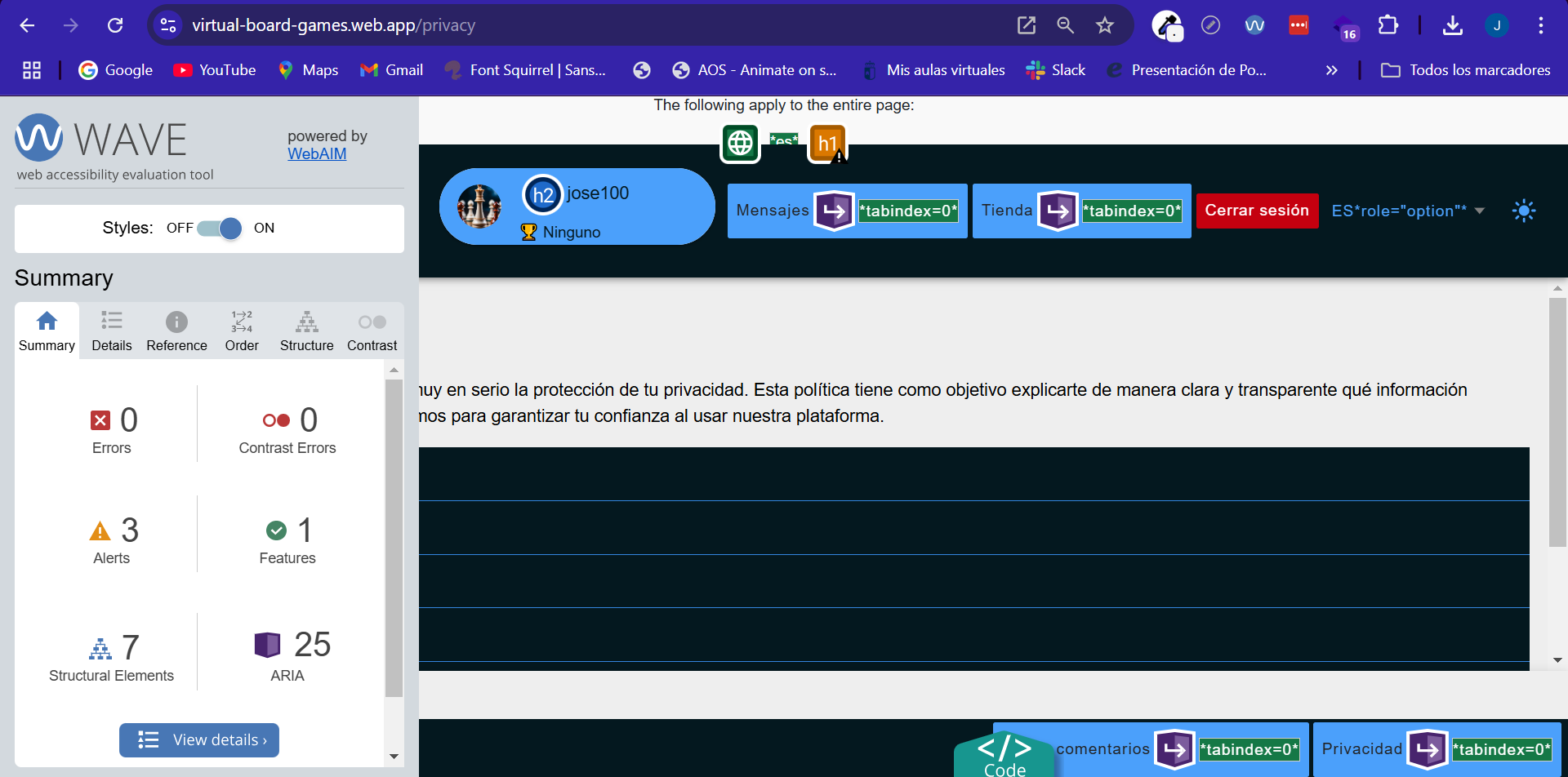
## Accesibilidad

**/home:**



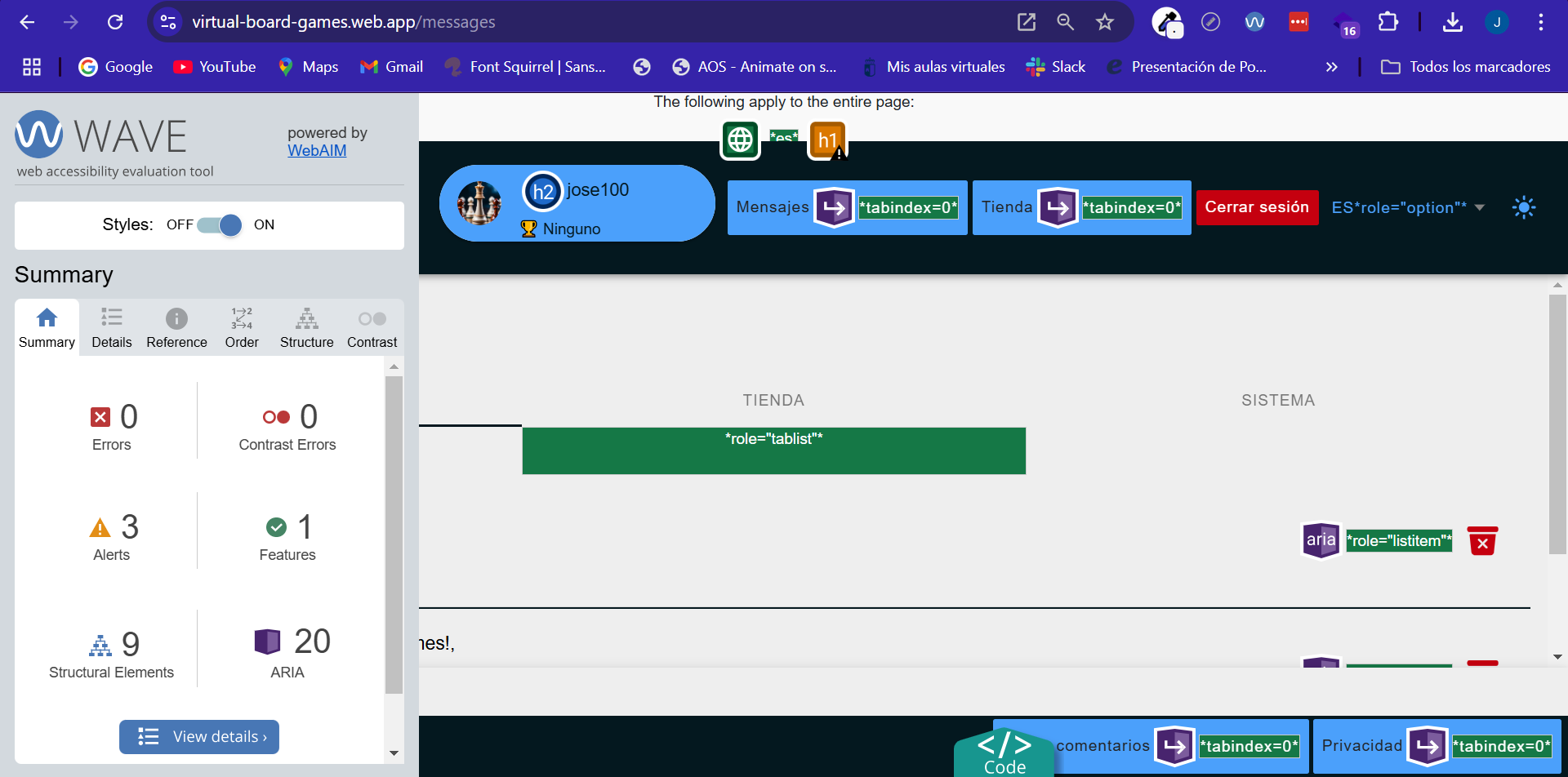
Los alerts son 1 por no tener un h1 pero no me pega que en está página haya un h1 y los otros 2 es por **Noscript element** que es cosa de angular y no lo puedo cambiar.

**/privacy**



Los alerts son exactamente los mismos, pero en este caso si tengo un título, lo que pasa es que utilizo un <ion-title> que es una etiqueta de Ionic para poner títulos y este no lo detecta como h1, casí todas las páginas tendrán los mismo alert.

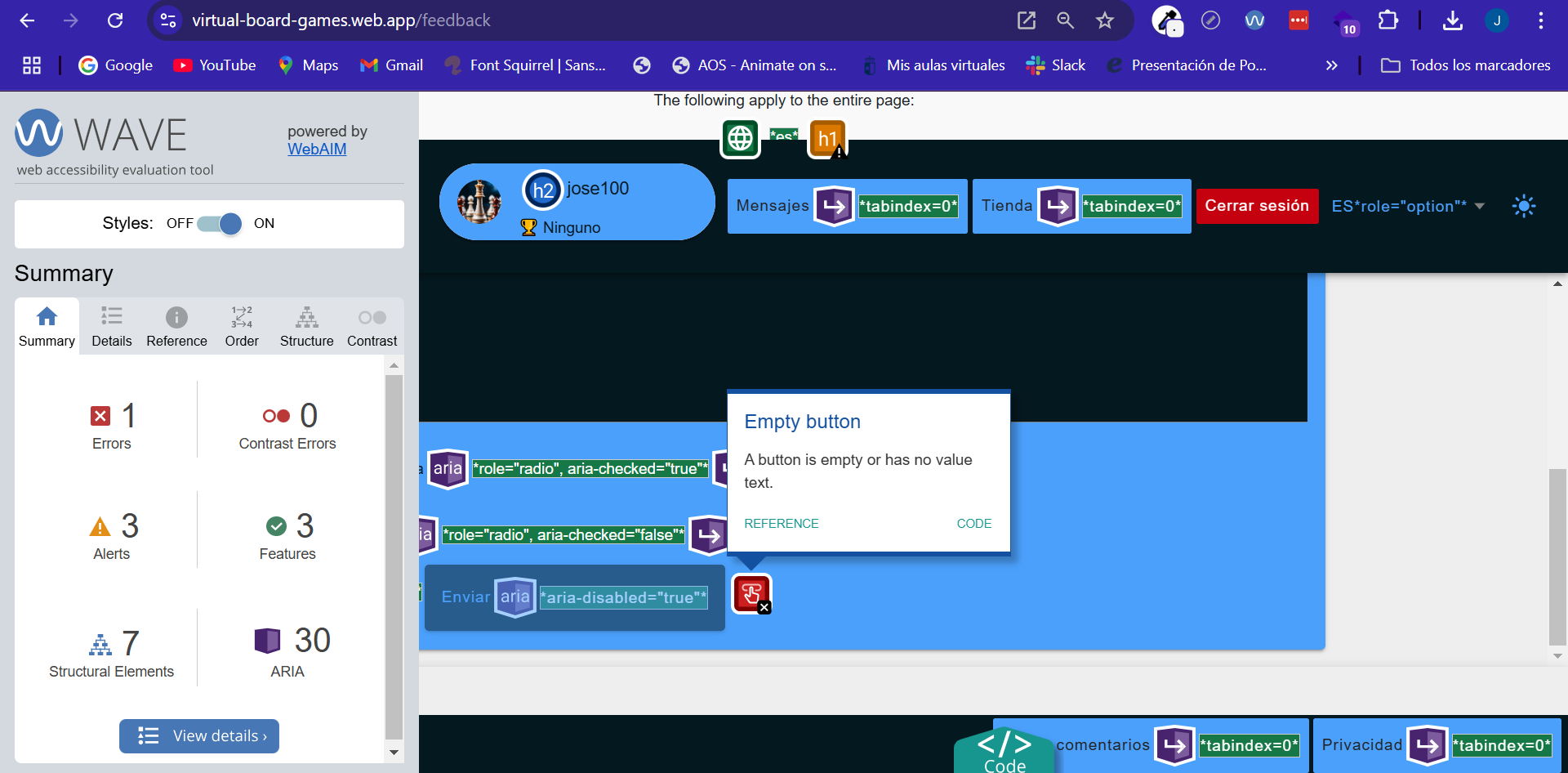
**/messages**



**/shop**

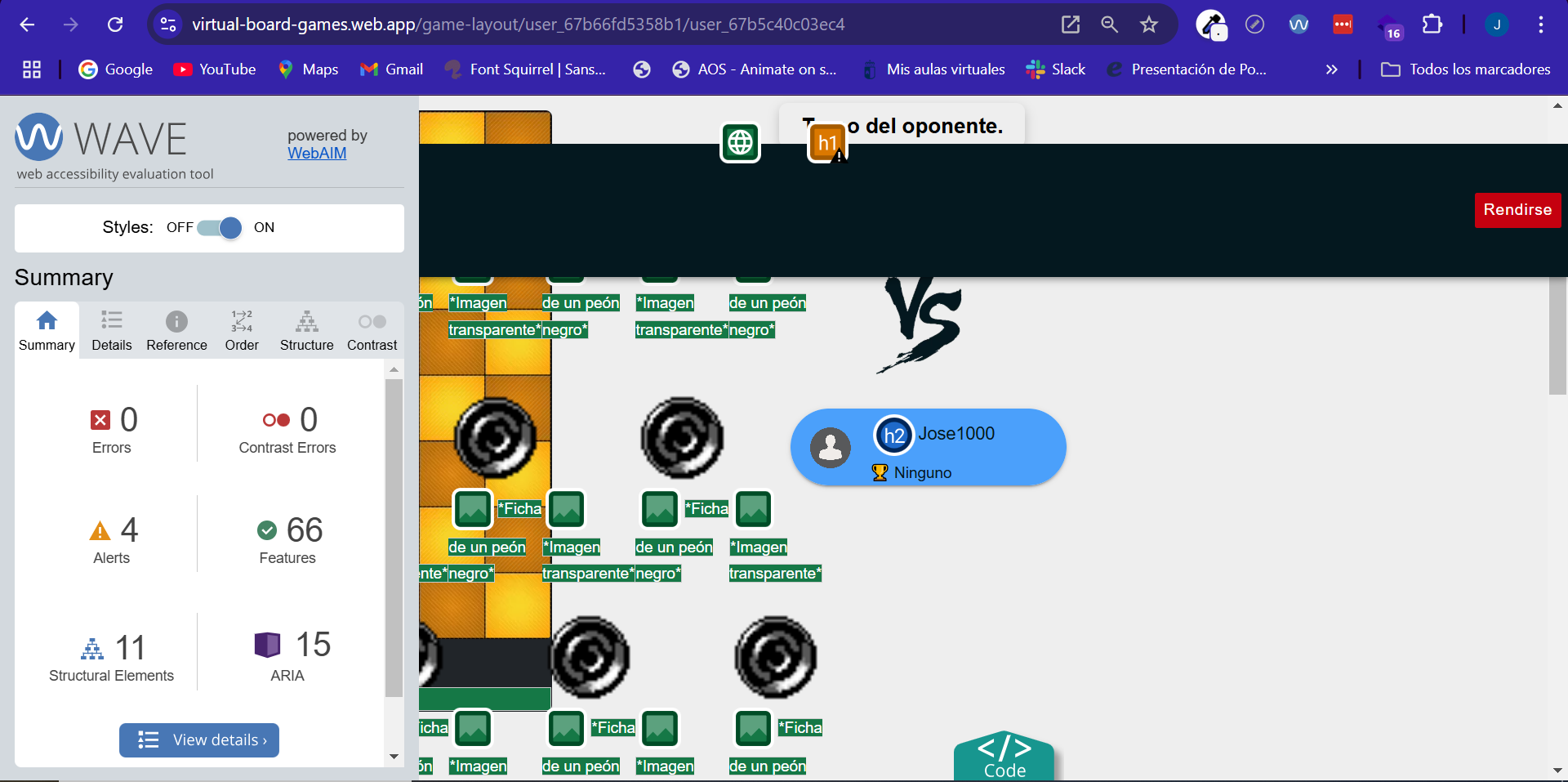


**/feedback**

****

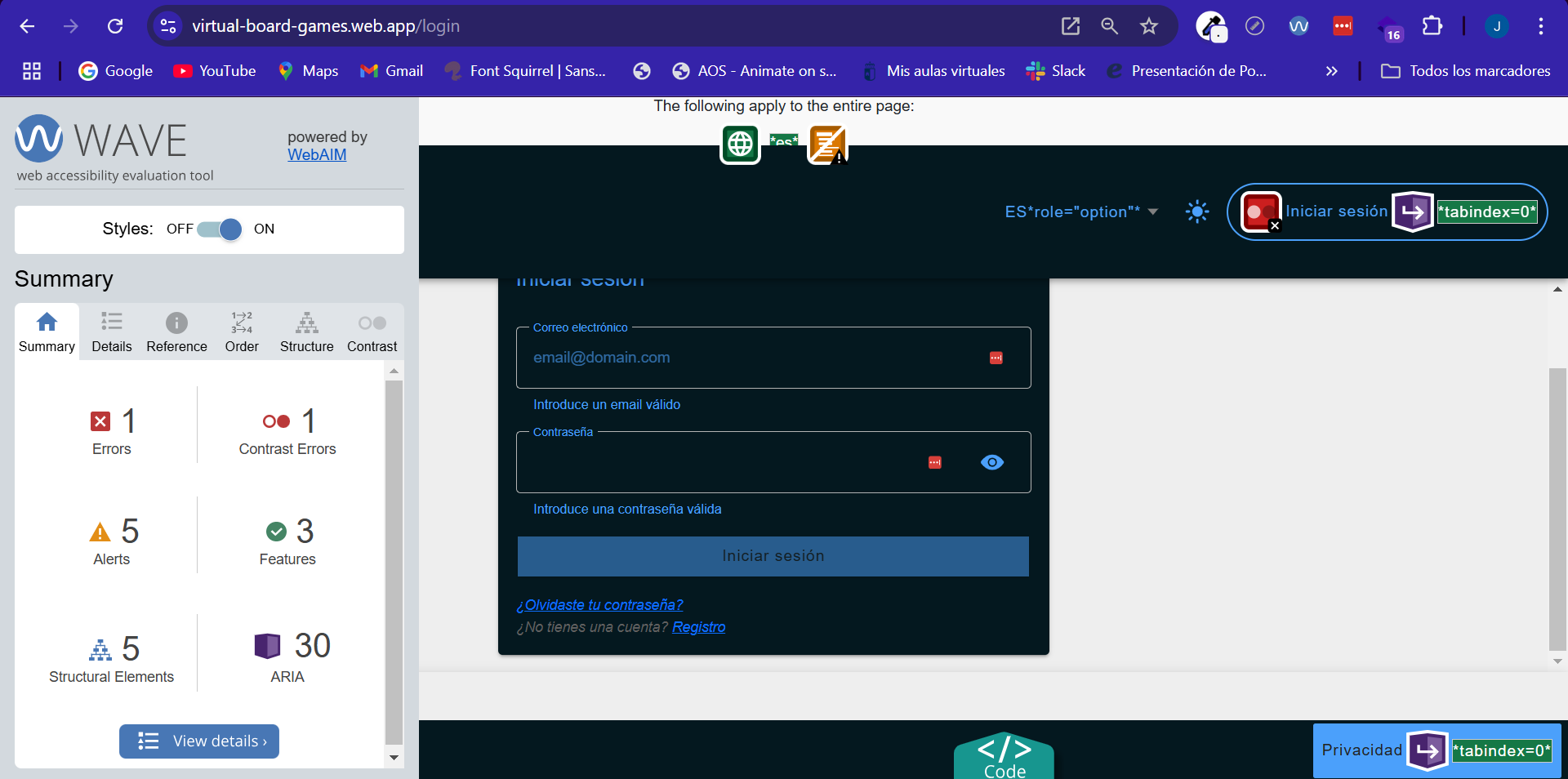
También tiene los mismos alerts pero este tiene un error que dice que tiene un botón vacío pero yo lo tengo bien, así que debe ser por algo de los elementos de Ionic.

**/game-layout**

****

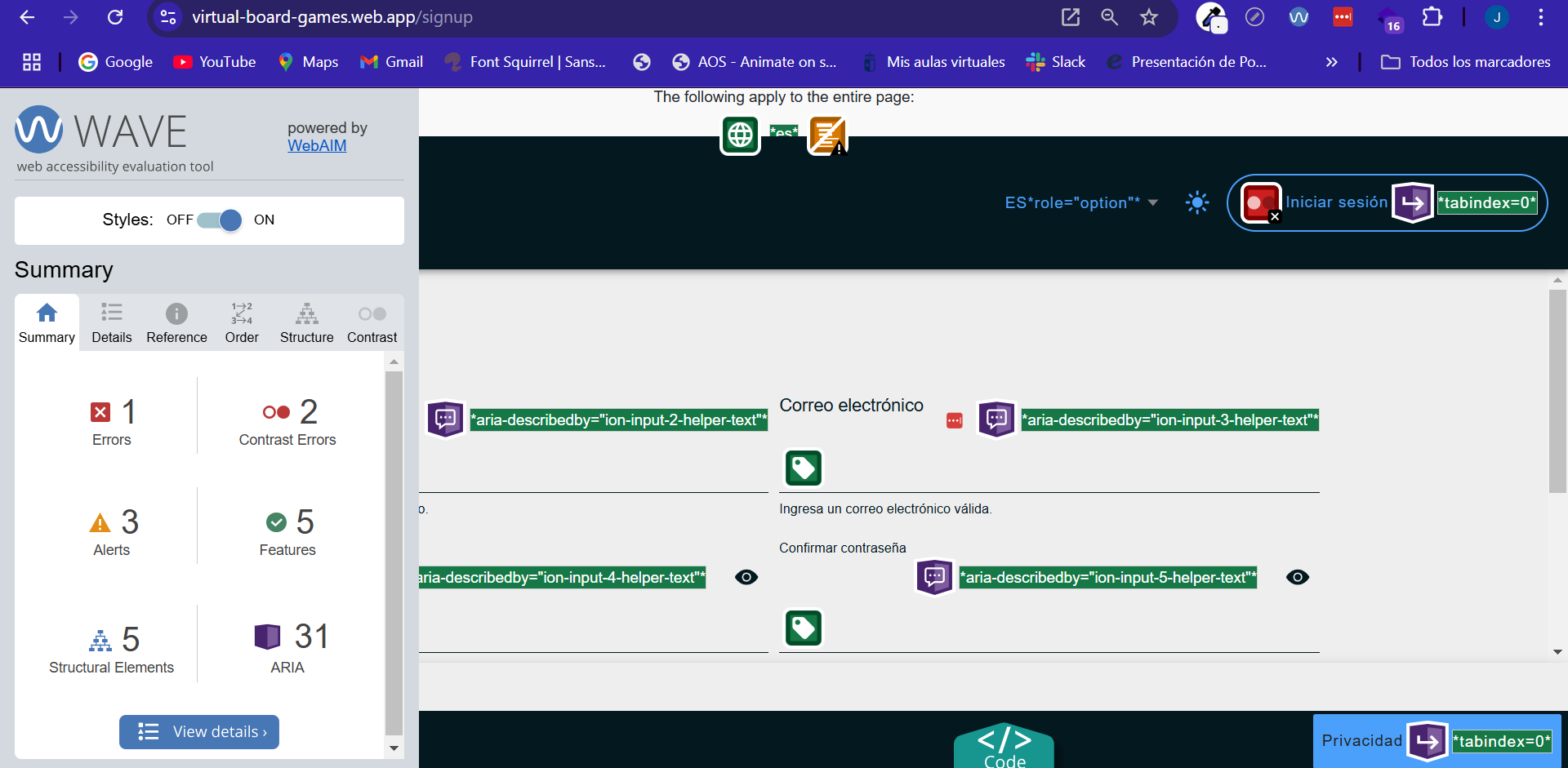
Aquí tiene los mismo alert que los otros pero también tiene otro, este alert dice que no utilice los <table> para diseñar sino que las utilice para mostrar datos modulares, pero como sigo necesitándola vale con ponerle a la tabla el rol=”presentation” para que sea ignorado por los lectores de pantalla.

**/login**



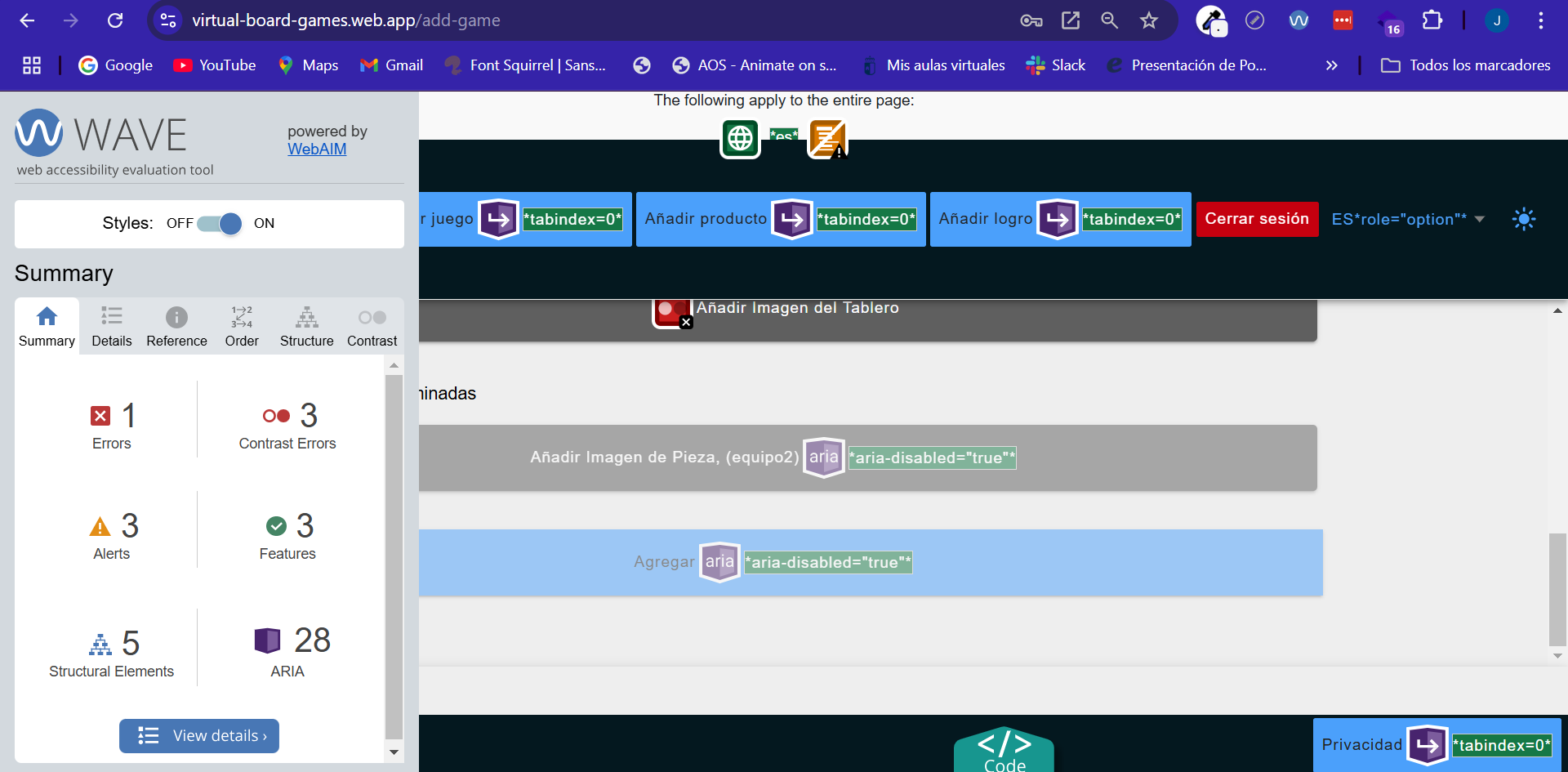
En el login tengo el mismo error que tuve en el otro formulario que es por culpar de Ionic, en cuanto a los alerts me dan otros 2 por que el texto es muy pequeño.

**/signup**



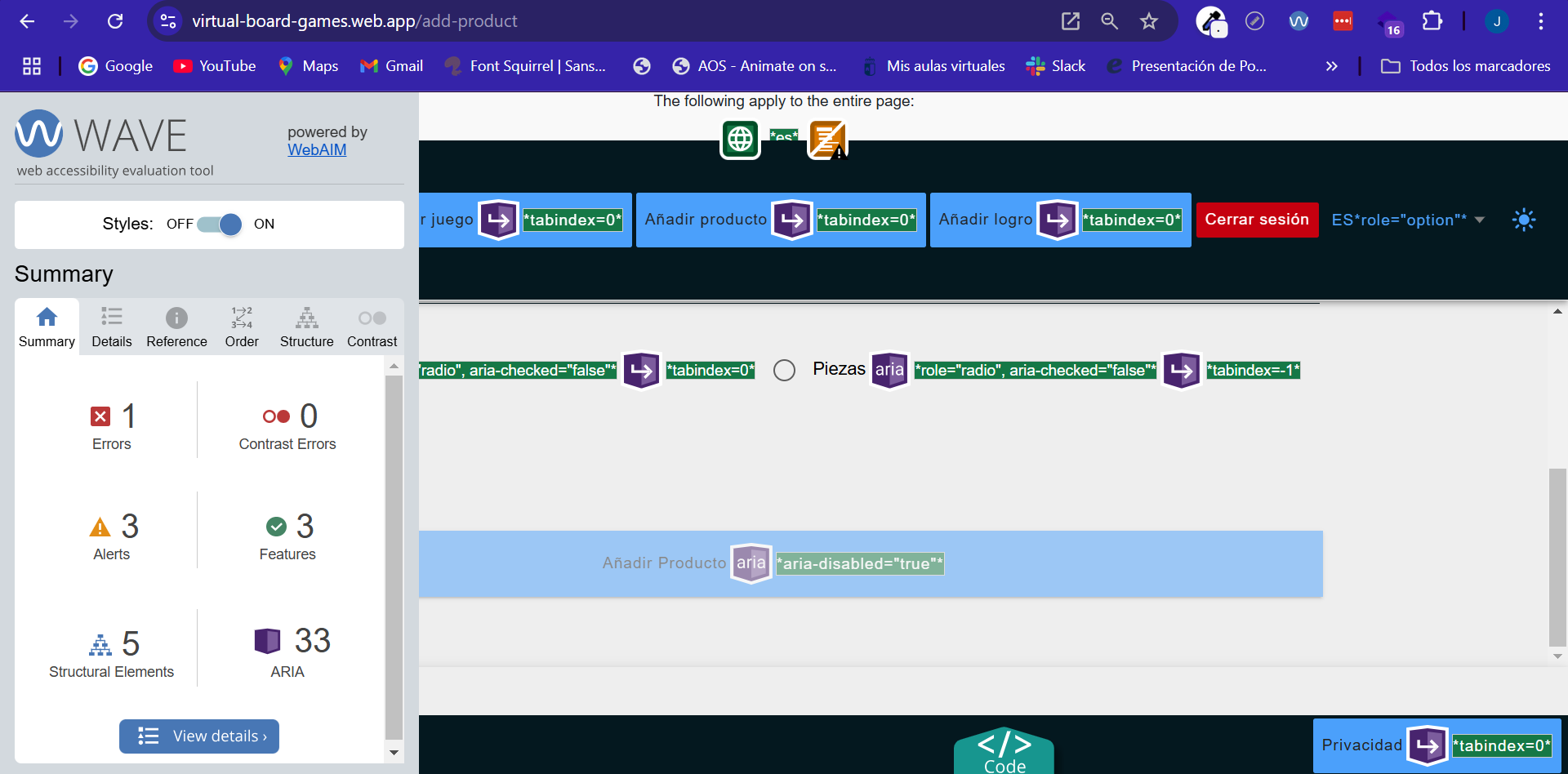
Me da los alerts de siempre y el error que me da en todos los formularios.

**/add-game**

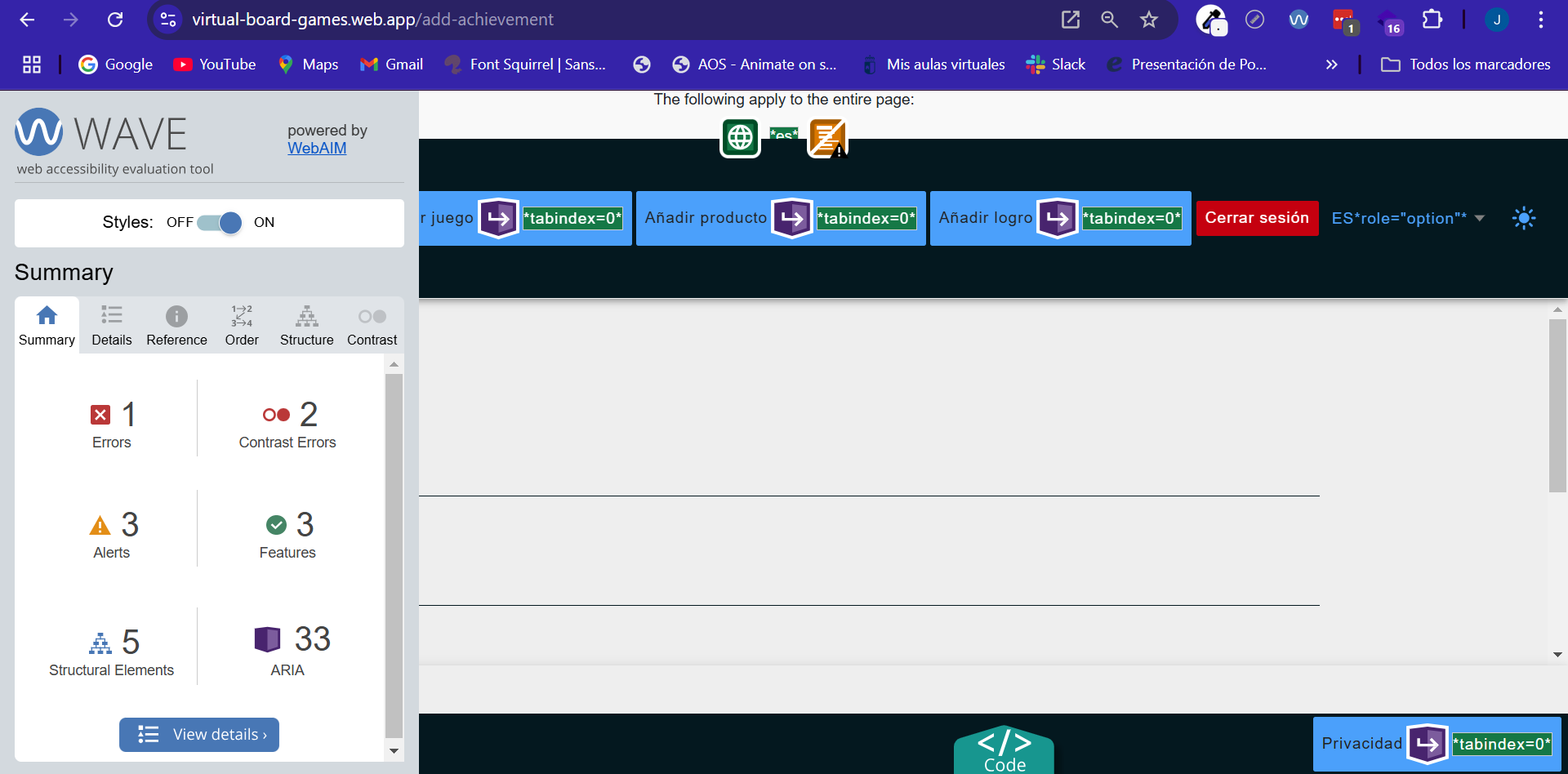


Mismo error en todos los formularios.

**/add-product**

****

**/add-achievement**

****

## Usabilidad

Se ha aplicado el test WAMMI a un grupo de usuarios, quienes han respondido un cuestionario basado en cinco dimensiones clave:

* **Atracción**: Evaluación de la interfaz visual y la experiencia de usuario.
* **Control**: Facilidad para navegar y realizar acciones dentro de la aplicación.
* **Eficiencia**: Velocidad y respuesta del sistema ante las acciones del usuario.
* **Aprendizaje**: Facilidad con la que un nuevo usuario puede comprender el uso de la aplicación.
* **Satisfacción global**: Percepción general sobre la calidad del servicio.

Cada dimensión ha sido calificada en una escala de 1 a 5, donde 1 representa una experiencia deficiente y 5 una excelente.

A continuación, se presentan los resultados obtenidos en cada dimensión:

|  |  |
| --- | --- |
| Dimensión | Puntuación media |
| Atracción | 4.3 |
| Control | 3.8 |
| Eficiencia | 3.7 |
| Aprendizaje | 4.2 |
| Satisfacción global | 4.0 |

Sobre estas puntuaciones podemos observar lo siguiente

* La interfaz ha sido bien valorada en cuanto a diseño y atractivo visual.
* Algunos usuarios han reportado dificultades menores en la navegación, lo que explica la puntuación de 3.8 en "Control".
* La aplicación ha obtenido un 3.7 en "Eficiencia" por lo que la aplicación ha tenido algunos problemas en cuanto a la velocidad del sistema en algunas partes.
* Los nuevos usuarios consideran que la plataforma es fácil de aprender, con una puntuación alta en "Aprendizaje".
* La satisfacción general es positiva, pero con margen de mejora.

Algunas ideas de mejora para mi aplicación son:

Optimizar la navegación en ciertas secciones para mejorar la fluidez y usabilidad.

Mejorar la organización de los menús.

Los problemas en eficiencia probablemente sean a causa de las partidas online, así que debería mejorarlo de alguna forma.

Los resultados indican que *Virtual Board Games* ofrece una buena experiencia de usuario, con un diseño atractivo y una curva de aprendizaje amigable. Sin embargo, hay oportunidades de mejora en la navegación y estructura de la interfaz para, además de los problemas de eficiencia para optimizar la usabilidad.

## Desarrollo web entorno cliente

En mi aplicación hay demasiadas funciones, cada uno de los componentes tienen funciones específicas que solo funcionan en ese componente, pero hay funciones globales que se pueden utilizar en todos los componentes por lo que será eso lo que explique:

Todas estas están en la carpeta frontend.

**src/app/shared/guards/admin.guard.ts:** Aquí están la función que te impide que un usuario normal acceda a las páginas de administrador

**src/app/shared/guards/auth.guard.ts:** Impide que alguien no logeado pueda acceder a páginas de usuarios

**src/app/shared/guards/game.guard.ts:** Impide que un usuario pueda acceder a la página de juego sin buscar partida.

**src/app/shared/guards/user.guard.ts:** Impide que el administrador acceda a las páginas de un usuario normal.

**src/app/shared/pipes/image.pipe.ts:** Aquí esta el pipe que pasándole los parámetros correctos te devuelve la url de una imagen en cloudinary.

**src/app/shared/validators/register-login.validator.ts:** Aquí esta todo lo necesario para validar los formularios de registro y login.

**src/app/shared/validators/services/animation.service.ts:** En este servicio se encuentran las funciones para hacer animaciones con angular dentro de mi aplicación.

**src/app/shared/validators/services/data-local.service.ts:** Aquí estan las funciones que me permiten guardar, leer y eliminar datos en mi navegador de forma local.

**src/app/shared/validators/services/realtime-database.service.ts:** Permite utilizar la realtime database de firebase para poder leer y escribir en esta base de datos no relacional.

**src/app/shared/validators/services/search-game.service.ts:** Este es el servicio que permite buscar partida online a un usuario.

**src/app/shared/validators/services/theme.service.ts:** Este servicio maneja todo lo relacionado con la gestión del tema claro y el tema oscuro dentro de mi aplicación.

**src/app/shared/validators/services/toast.service.ts:** Este servicio permite mostrar toast informativos a los usuarios.

**src/app/shared/validators/services/transalator.service.ts:** Maneja todo lo relacionado a los idiomas dentro de mi aplicación.

**src/app/shared/validators/services/backend/achievement.service.ts:** En este servicio se almacenan todas las funciones que permite gestionar los logros en mi base de datos conectándose con el backend.

**src/app/shared/validators/services/backend/cloudinary.service.ts:** Permite subir archivos tipo img y json a cloudinary, enviando los archivos al backend.

**src/app/shared/validators/services/backend/game.service.ts:** En este servicio se almacenan todas las funciones que permite gestionar los juegos en mi base de datos conectándose con el backend.

**src/app/shared/validators/services/backend/messages.service.ts:** En este servicio se almacenan todas las funciones que permite gestionar los mensajes en mi base de datos conectándose con el backend.

**src/app/shared/validators/services/backend/products.service.ts:** En este servicio se almacenan todas las funciones que permite gestionar los productos en mi base de datos conectándose con el backend.

**src/app/shared/validators/services/backend/user.service.ts:** En este servicio se almacenan todas las funciones que permite gestionar los usuarios en mi base de datos conectándose con el backend, incluyendo el login y el logout, también los registros de nuevos usuario.

**src/app/shared/validators/services/backend/game.service.ts:** En este servicio se almacenan todas las funciones que permite gestionar los juegos en mi base de datos conectándose con el backend.

**src/app/games/services/start-game.service.ts:** En este servicio están las funciones para almacenar el juego y los usuarios que participan en una partida.

## Fase de despliegue

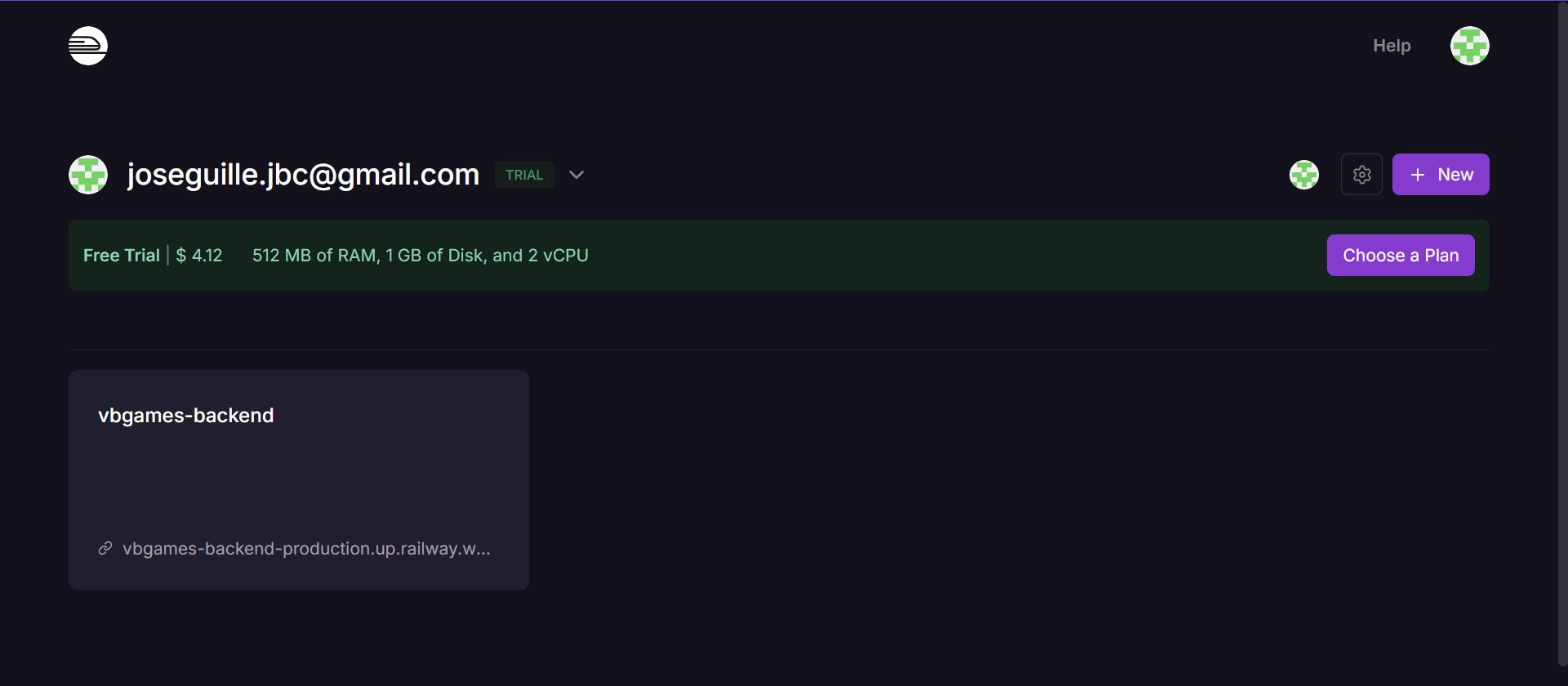
## Despliegue utilizando un hosting

Tengo subido mi backend y frontend en sitios completamente diferentes por lo que explicaré uno por uno como lo he hecho.

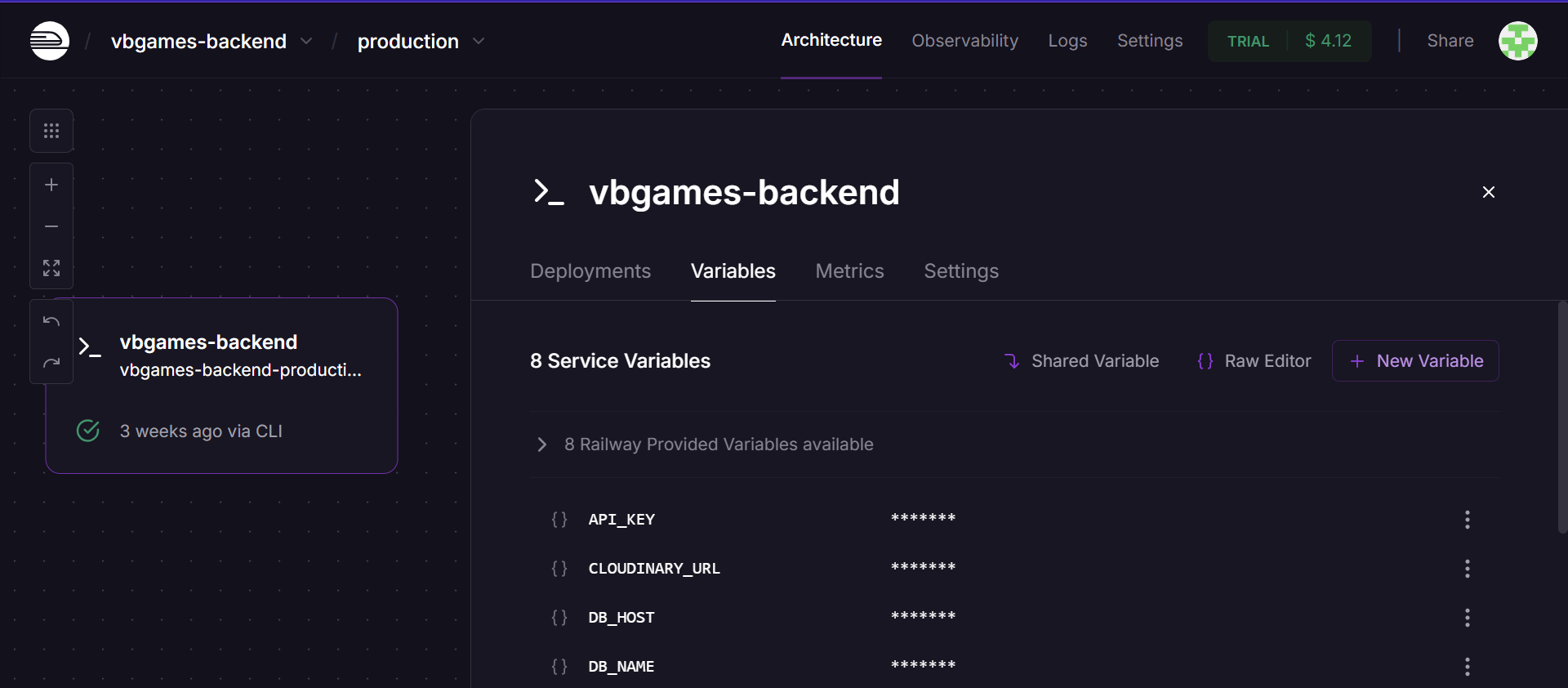
**Backend**

Está subido en railway, para subirlo he seguido los siguientes pasos:

* Me creé una cuenta en Railway, al iniciar sesión voy al dashboard y creo un nuevo proyecto.



* Para poder subir mi proyecto a Railway primero tengo que instalar el cliente de railway en mi ordenador con el comando **npm install -g @railway/cli.**
* Luego para poder subir mi proyecto tengo que logearme en el CLI con el comando **railway login,** que te abrirá el navegador para hacer el login.
* Después de logearme tengo que ir a la carpeta donde está mi proyecto y ejecutar el comando **railway link** para vincular mi proyecto local al proyecto de mi cuenta de railway
* Después de vincular el proyecto, en mi carpeta local ejecuto el comando **railway up** para subir mi proyecto, a partir de ahí cada vez que quiera actualizar mis avances ejecuto railway up.
* Luego voy al proyecto en mi cuenta de railway e ingreso las variables de entorno.

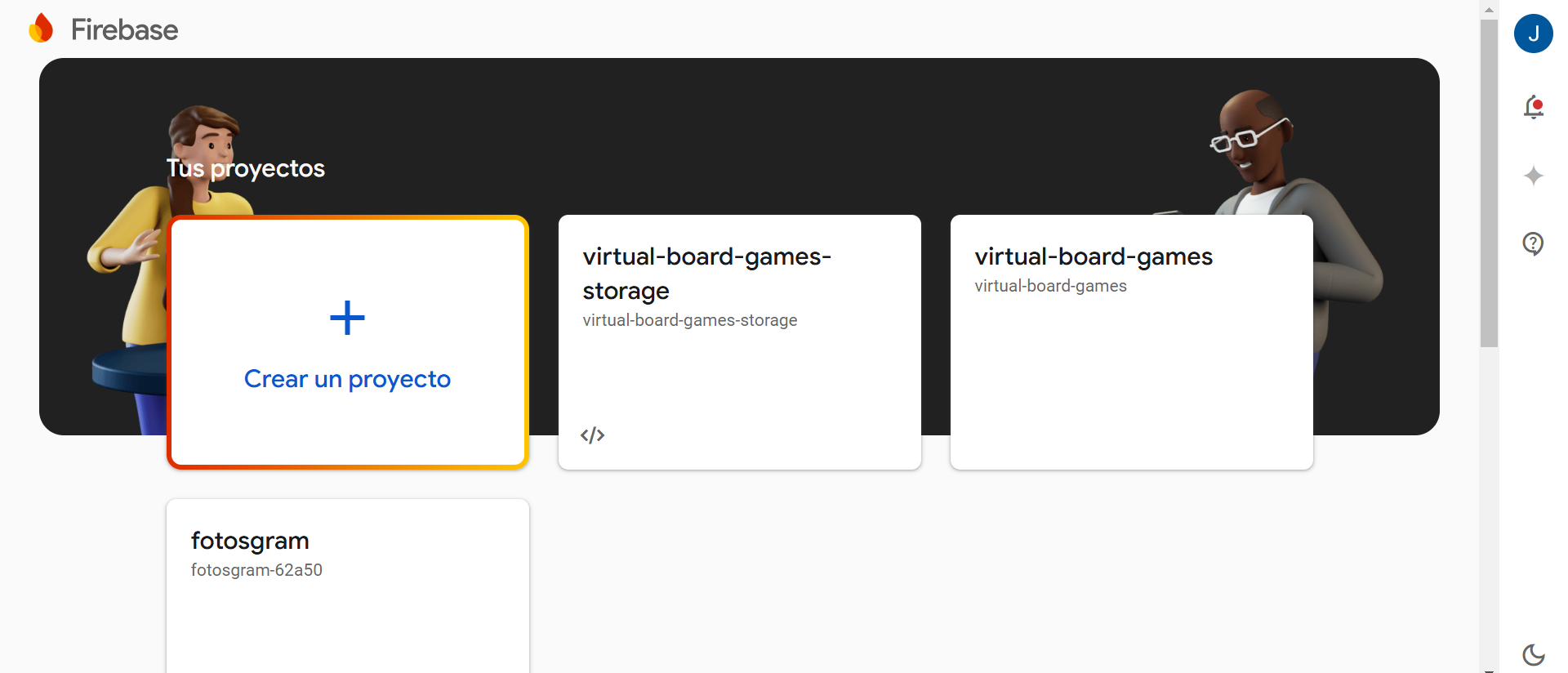


* Para conectar mi base de datos PostgreSQL a mi backend todo lo que tengo que usar la función pg\_connect de la siguiente forma: “pg\_connect(‘user=$user password=$pass host=$host port=$port dbname=$dbname’)” poniendo mis credenciales, no tengo que hacer nada más.

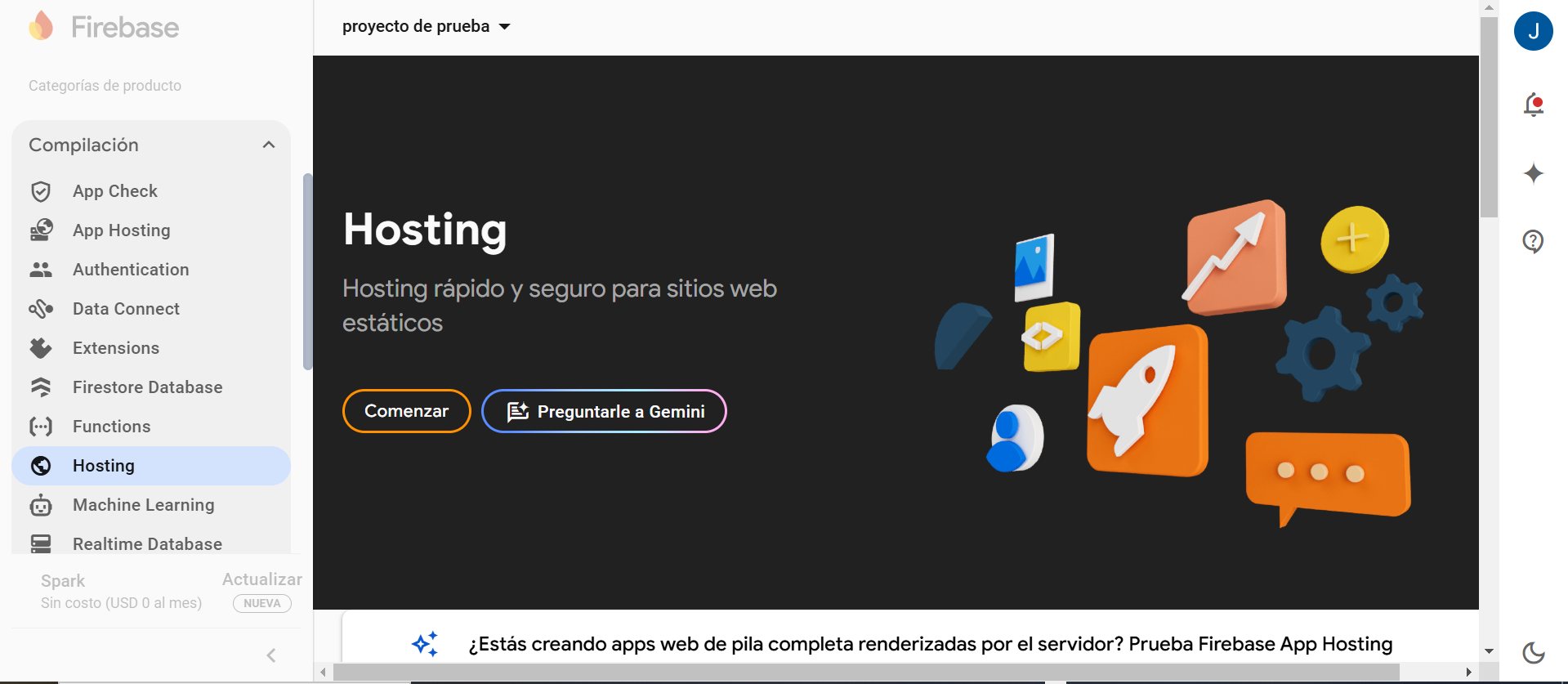
**Frontend**

Está subido a firebase, para subirlo he seguido los siguientes pasos:

* Me creo una cuenta en firebase e inicio sesión, en el dashboard creo un nuevo proyecto



* Después de crear el proyecto y entrar en él en el menú de la izquierda le das a hosting y después a comenzar, luego solo hay que seguir los pasos.



* En el terminal instalamos el firebase-tools en el ordenador con el comando **npm install -g firebase-tools.**
* Luego hay que ejecutar el comando **firebase login** para iniciar sesión
* Ahora hay que ir a la carpeta donde está el proyecto y utilizo el comando **firebase init** que pedirá que indiques la carpeta que quieres subir, en mi caso es la carpeta www.
* Luego para subir el proyecto Ionic antes hay que utilizar el comando **ionic build –prod** para que generé la carpeta www. Después para desplegar el proyecto definitivamente hay que ejecutar el comando **firebase deploy**.

## Seguimiento y control de incidencias

Gracias a la ayuda de algunos conocidos que probaron la aplicación pude encontrar algunos errores como los siguientes.

Uno de los usuarios informó que, tras cerrar sesión, al intentar volver a iniciar sesión, la aplicación no los redirigía correctamente a la pantalla principal. Esto se debía a un problema con la gestión del estado de autenticación, que fue corregido para garantizar una mejor experiencia.

Algunos elementos de la tienda no se mostraban correctamente en dispositivos móviles, lo que dificultaba la compra de estilos de tableros y piezas. Este problema se solucionó mejorando el diseño responsivo de la interfaz.

Se detectó que, en algunos casos, al cancelar una búsqueda de partida y volver a iniciarla, el usuario quedaba bloqueado sin poder encontrar o ser encontrado por otros jugadores. Esto se debía a que el estado de la solicitud anterior no se eliminaba correctamente en la base de datos, lo que fue corregido asegurando que cada nueva búsqueda eliminara la solicitud previa antes de crearse.

## Indicadores de calidad de procesos

Para evaluar la calidad de los procesos en la aplicación, primero realicé pruebas exhaustivas por mi cuenta, verificando el correcto funcionamiento de cada funcionalidad, la fluidez en la navegación y la estabilidad del sistema. Posteriormente, pedí a algunos conocidos que utilizaran la aplicación y reportaran cualquier error o comportamiento inesperado que encontraran. Gracias a sus comentarios, pude identificar y corregir diversos bugs, mejorando así la experiencia del usuario y la estabilidad general del sistema. Esta fase de pruebas permitió asegurar que la aplicación funcionara de manera óptima antes de su despliegue final.

# Recursos materiales

## Inventario, valorado, de medios

Ordenador portatil de desarrollo lenovo con procesador i5, 8 GB de RAM y almacenamiento SSD de 256 GB.

Segunda pantalla para trabajar a dos pantallas, pantalla MSI PRO LED.

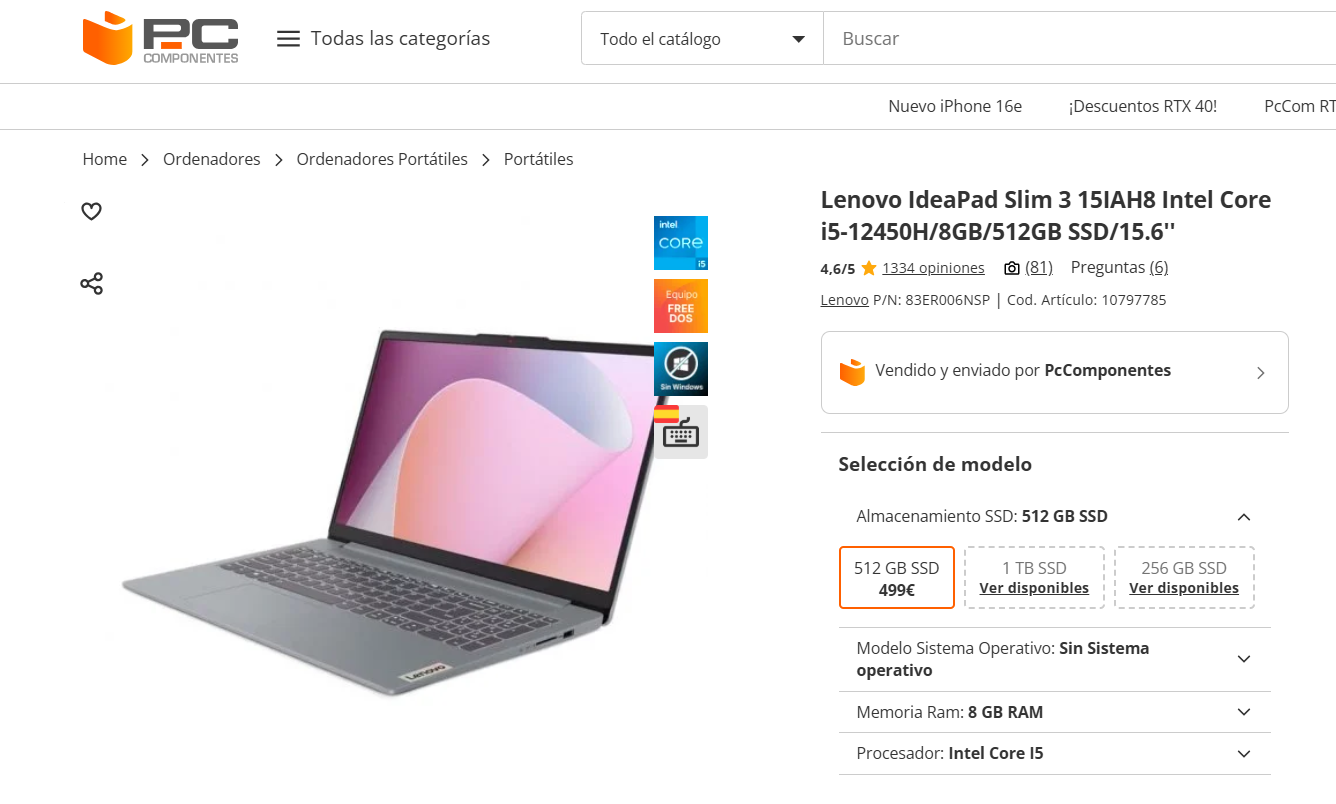
Dispositivos móviles para pruebas Android, un realme X50 Pro 5G 8/256GB.

Servidor de producción, utilizando la versión PRO de railway para el backend, frontend en firebase es gratuito, al igual que la realtime database de firebase y cloudinary también son gratuitos.

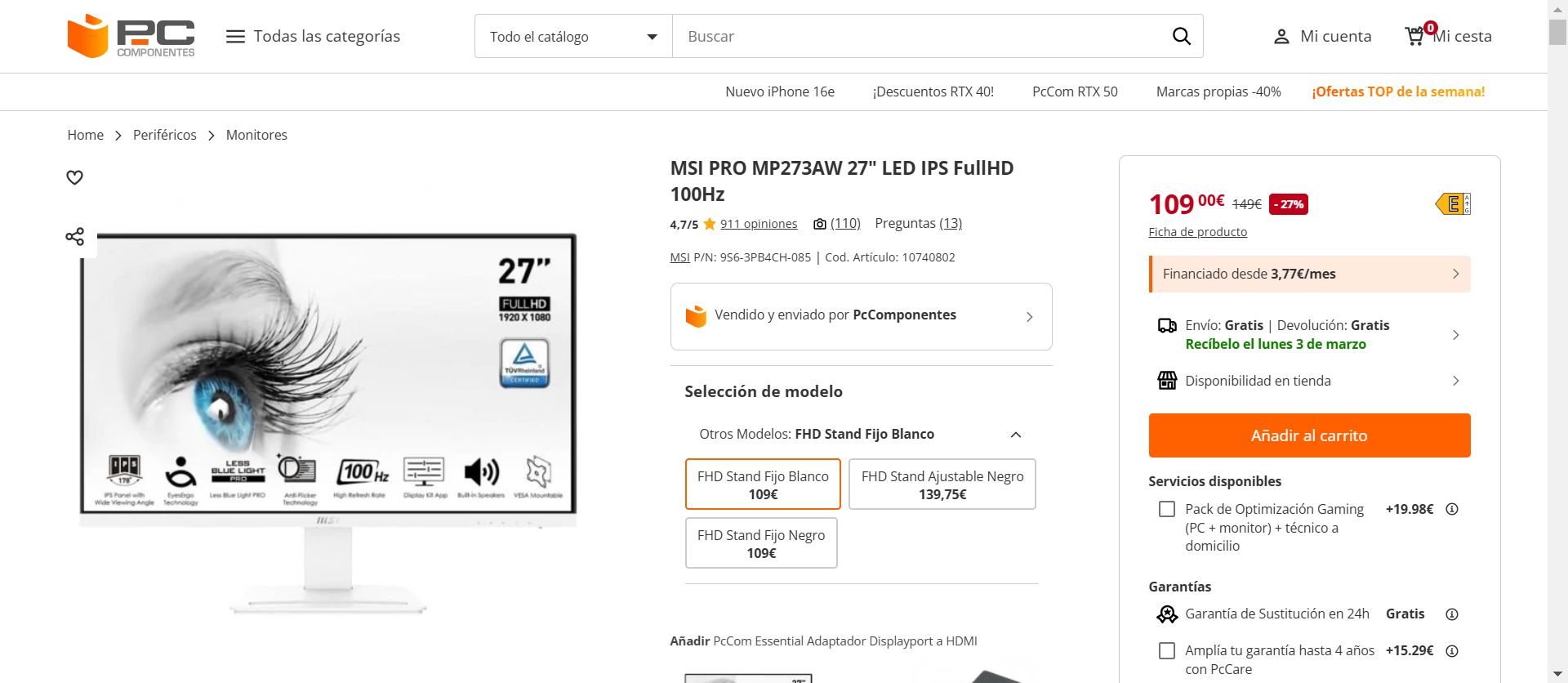
Licencias de software Visual Studio Code gratuito al igual que el Windows.

## Presupuesto económico

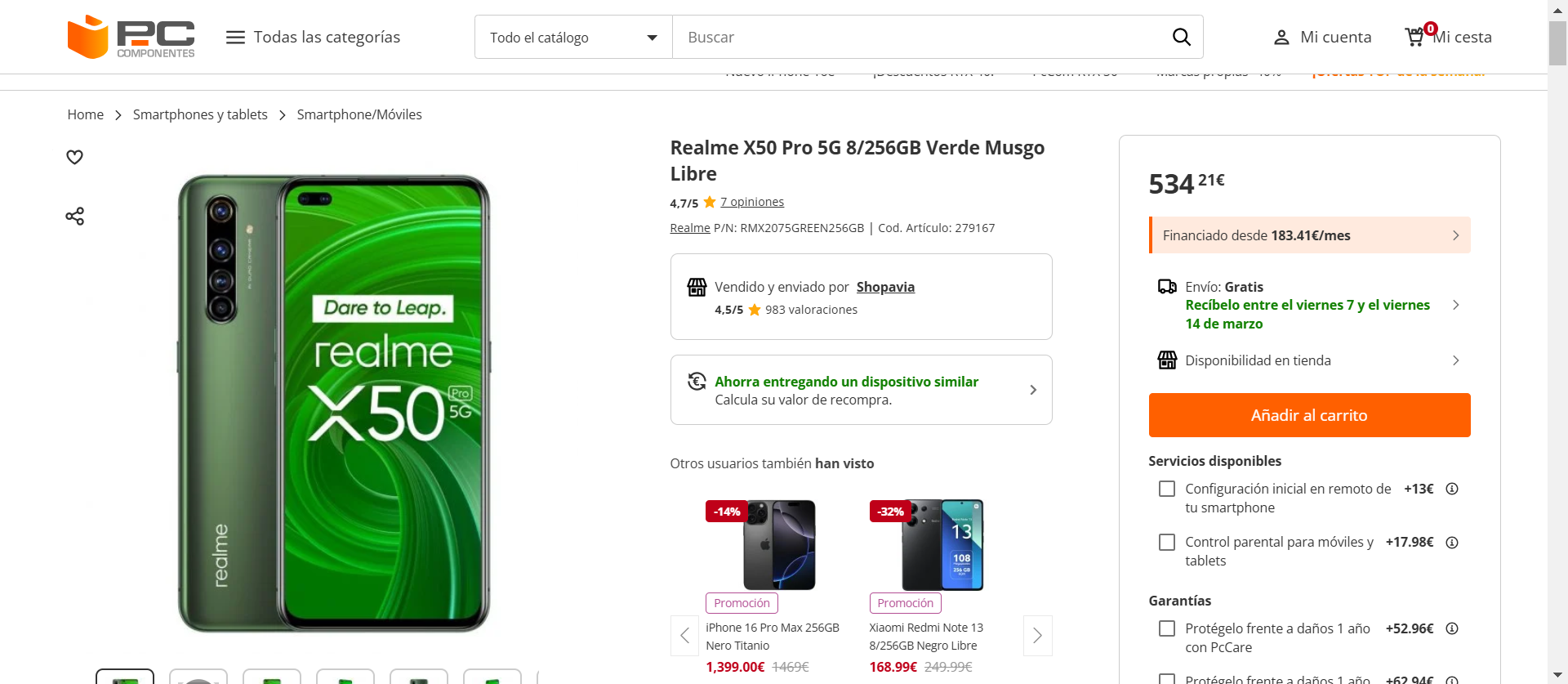
El ordenador portátil lenovo cuesta 499€ en pc componentes.



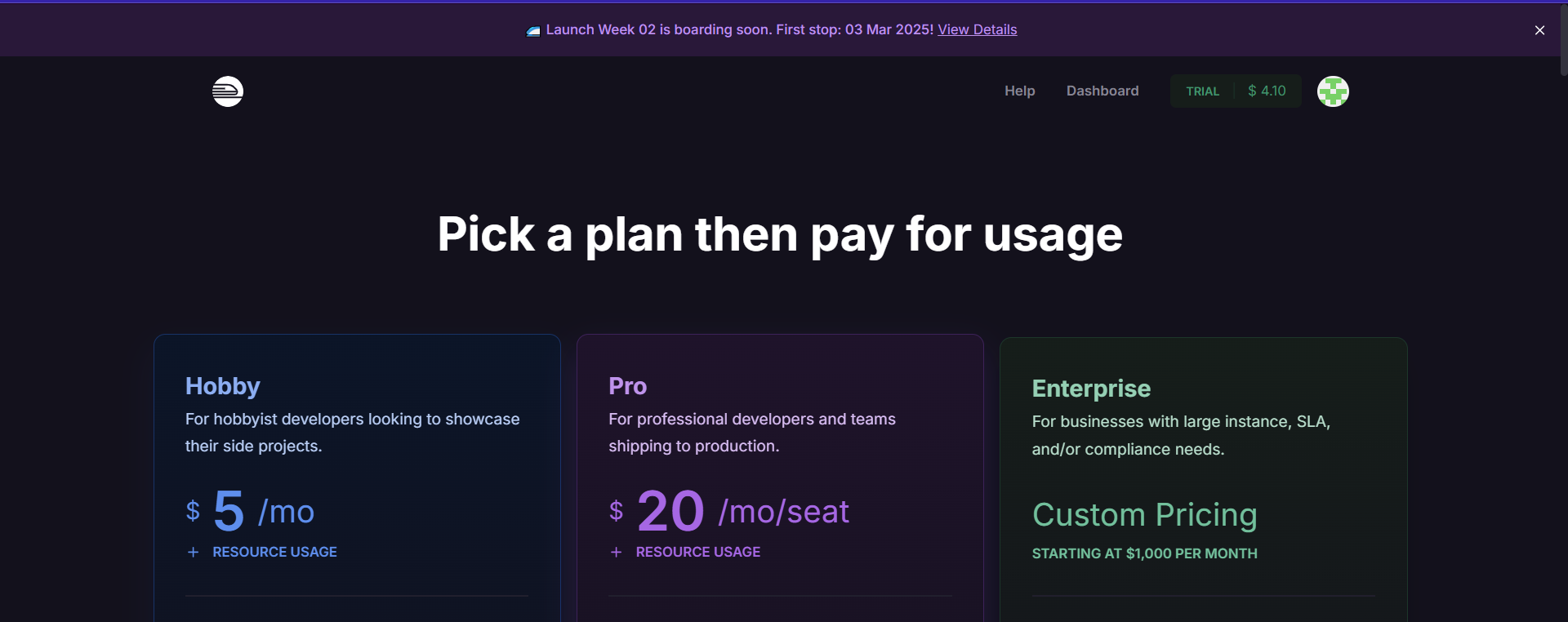
Segunda pantalla MSI a 109€.



Dispositivo móvil Realme X50 PRO a 534,21€



Plan pro de railway 20$ que en euros son 19,24€ Mensuales



Siendo un total de **1.142,21€.**

# Recursos humanos

## Organización

La dirección sería yo y me encargaría de dirigir a los demás departamentos.

Producción se encargaría de crear la página web y de hacer el mantenimiento, contrataría a alguien para este departamento, pero yo también trabajaré en este departamento.

Compras se encargarán de comprar los equipos que necesitemos como ordenadores, cables, periféricos, etc. De este departamento también me encargaré yo.

Economía se encargará llevar la contabilidad de la empresa, gestionar pagos, gestiones bancarias, etc. Yo también me encargaré de este departamento.

Administración se encargará de la gestión y archivo de la documentación de la empresa, así como la atención al cliente. Yo seré quién se encargue de este departamento.

Recursos humanos se encargará de la gestión de los recursos humanos como nóminas, contratos, seguridad social, etc. Yo me encargaré de este departamento.

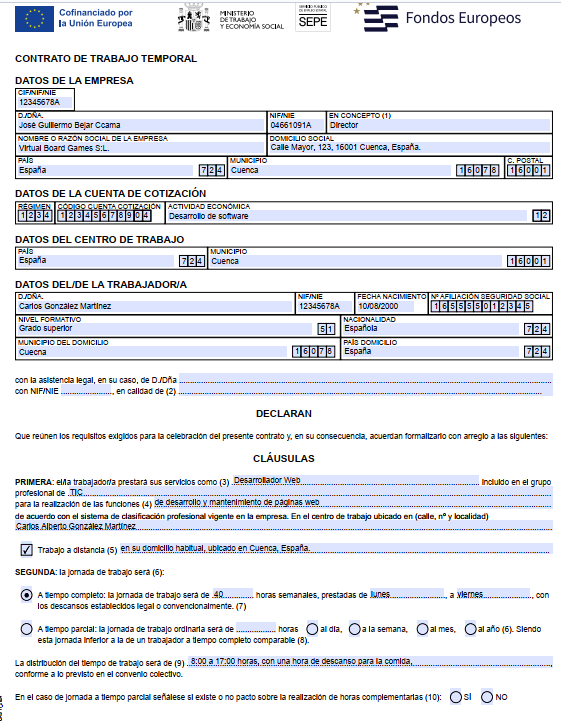
Marketing será quién se encargará de promocionar la aplicación Virtual Board Games, me encargaré yo.

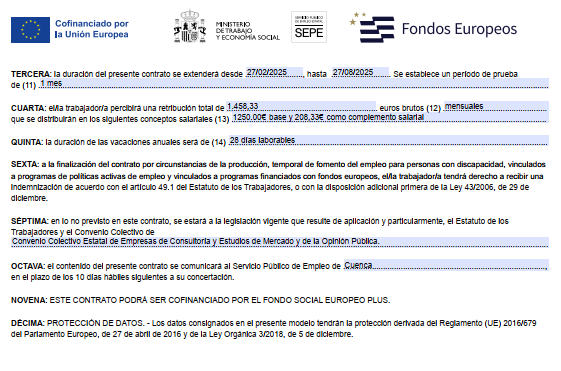
## Contratación

Para poder manejar la el departamento de producción necesitaré a un empleado para llevar el desarrollo y el mantenimiento de la aplicación.

|  |  |  |
| --- | --- | --- |
| Descripción del puesto | Condiciones laborales | Perfíl profesional |
| Puesto:  Desarrollador de páginas web | **Salario:**  1250**€** | **Formación y titulación:**  Técnico grado superior en desarrollo de aplicaciones web |
| Departamento:  Producción | **Horario y jornada:**  De 8 a 16h | **Conocimientos especificados:**  Desarrollo de páginas web. |
| Nivel de responsabilidad:  Desarrollo y mantenimiento de la aplicación web. | **Lugar de trabajo:**  Remoto | **Experiencia profesional:**  No es necesaria |
| Tareas a realizar:  -Desarrollar código limpio y eficiente para la aplicación web.  -Garantizar la compatibilidad multiplataforma y la respuesta de diseño.  -Realizar pruebas y depuración de software. | **Tipo de contrato:**  Temporal 6 meses. | **Habilidades profesionales:**  Conocimiento en html5 así como lenguajes de programación TypeScript y PHP.  Preferiblemente que sepa usar Ionic basado en Angular |
| **Actitudes y habilidades personales:**  Capacidad de resolución de problemas, creatividad, atención al detalle, autonomía. |

|  |  |
| --- | --- |
| **Calculo salario neto trabajador:**  Salario bruto: 1250 + 208,33 = 1458,33€  Seguridad social trabajador = 4,8% + 1,55% + 0,1% de BCCC = 6,45% de 1458,33 = 94,06€.  IRPF del trabajador = 10% de 1458,33 = 145,83  Salario neto = 1458,33 – 94,06 – 145,83= 1222,2€  El trabajador percibe que trabaja por 1218,44€ | **Cuotas a pagar por la empresa:**  CC = 24,10 de 1458,33 = 351,46€  CP = 1,5% de 1458,33 = 21,87€  Desempleo = 5,50% de 1458,33 = 80,20€  FOGASA = 0,2% de 1458,33 = 2,92€  FP = 0,6% de 1458,33 = 8,75€  **TOTAL** seguridad social a cargo de la empresa = 465,20€ |
| **Percepción de la empresa del coste del trabajador:** Salario neto a entregar = 1218,44€ + seguridad social del trabajador = 94,06€ + IRPF del trabajador = 145,83 + seguridad social a cargo de la empresa = 465,2€ =  = **Coste total trabajador = 1923,53€,** al sumar lo que recibe el trabajador más IRPF más seguridad social. | |





El contrato completo se encuentra en:

<https://github.com/jose-guillermo/JoseGuillermoDAW/blob/main/contrato_de_trabajo.pdf>

## Prevención de riesgos laborales

**Riesgos ergonómicos** El trabajo prolongado frente a un ordenador puede causar problemas como fatiga visual, dolor en la espalda, cuello y muñecas.

Medidas preventivas: Uso de sillas ergonómicas, pausas activas cada cierto tiempo, ajustes en la altura de la pantalla y el teclado.

**Riesgos psicosociales** El estrés relacionado con los plazos de entrega, la carga de trabajo y la presión por el rendimiento puede afectar la salud mental.

Medidas preventivas: Organización adecuada del trabajo, descansos regulares y establecimiento de plazos realistas.

**Riesgos asociados a la exposición prolongada a pantallas** La exposición continua a la luz azul de las pantallas puede afectar la vista y causar dolores de cabeza.

Medidas preventivas: Uso de filtros de luz azul en las pantallas, así como tomar descansos regulares de la pantalla.

En este tipo de proyecto, donde mi trabajador trabajará en remoto tendré que organizarlo principalmente de la siguiente forma

**Prevención personal.** Cada trabajador debe ser responsable de aplicar medidas preventivas relacionadas con su bienestar. Esto incluye mantener una postura adecuada, hacer pausas activas y gestionar el estrés de forma adecuada. Por lo que tendré que concienciar a mi trabajador de estas prácticas.

# Viabilidad técnica

## Estudio de viabilidad técnica

En el desarrollo frontend se ha utilizado Angular con Ionic, empleando TypeScript, HTML y CSS. Como entorno de desarrollo se ha elegido Visual Studio Code, y npm para la gestión de dependencias. Se ha incluido un sistema de estilos que permite la personalización de temas claro/oscuro

Para el backend, se ha optado por PHP con una base de datos PostgreSQL Para realizar pruebas de la API se ha utilizado Postman. Además, se ha integrado Firebase Realtime Database para almacenar ciertos datos en tiempo real y Cloudinary para la gestión de imágenes en la plataforma. También se ha utilizado Composer en PHP para la instalación de paquetes adicionales, como por ejemplo, cloudinary/cloudinary\_php para poder utilizar cloudinary desde mi backend y firebase/php-jwt para poder usar las jwt (Json Web Token) en la gestión de sesiones.

En cuanto al control de versiones, el código se aloja en GitHub y se gestiona mediante Sourcetree. El frontend está desplegado en Firebase Hosting y el backend en Railway, con las siguientes direcciones: <https://virtual-board-games.web.app> y <https://vbgames-backend-production.up.railway.app>.

# Viabilidad económico-financiera.

## Inversiones y gastos

Mis gatos serían

**Sueldo de mi empleado:** El gasto que supone mi empleado será 1930,60€.

**Los equipos informáticos y de software:** Entre todos los equipos informáticos y de software son **1.142,21€** y **19,24€** mensuales.

Suministros: El gasto de la luz será alrededor de 40€ e internet unos 60€.

## Financiación

La inversión inicial que tengo que hacer es de **1142,21**€.

Para la primera inversión de mi empresa, al ser esa cantidad puedo aportarlo yo, así evito los intereses o costos adicionales que puede suponer pedir un préstamo, también evito los riesgos que puedo tener.

A partir de esta primera inversión mediante amortizaciones que serán un 26% anual de Equipos informáticos que son (**296,97**€), así me aseguraré de tener el dinero necesario para cubrirlo en el futuro.

Los gastos me salen a **2.030,60€/mes.**

Como no estoy seguro de si mi empresa generará el dinero necesario para cubrir estos gastos los primeros meses, pediré un crédito bancario de 3000€ para asegurarme, porque es a corto plazo y así si no lo gasto tendré que pagar menos intereses.

Con una comisión de entrada del 1%, intereses por las cantidades dispuestas 6% y las cantidades no dispuestas 1%, que planeo hacer en un único pago al final del año.

Lo que dispongo de este crédito según mi plan financiero es:

Enero: 753.53€, Febrero: 733.53€, Marzo: 313.53€, Abril: 153.53€, Mayo: 123.53€.

A partir de mayo ya no planeo tener que disponer del crédito bancario.

En total gastaré: **2077,65€**

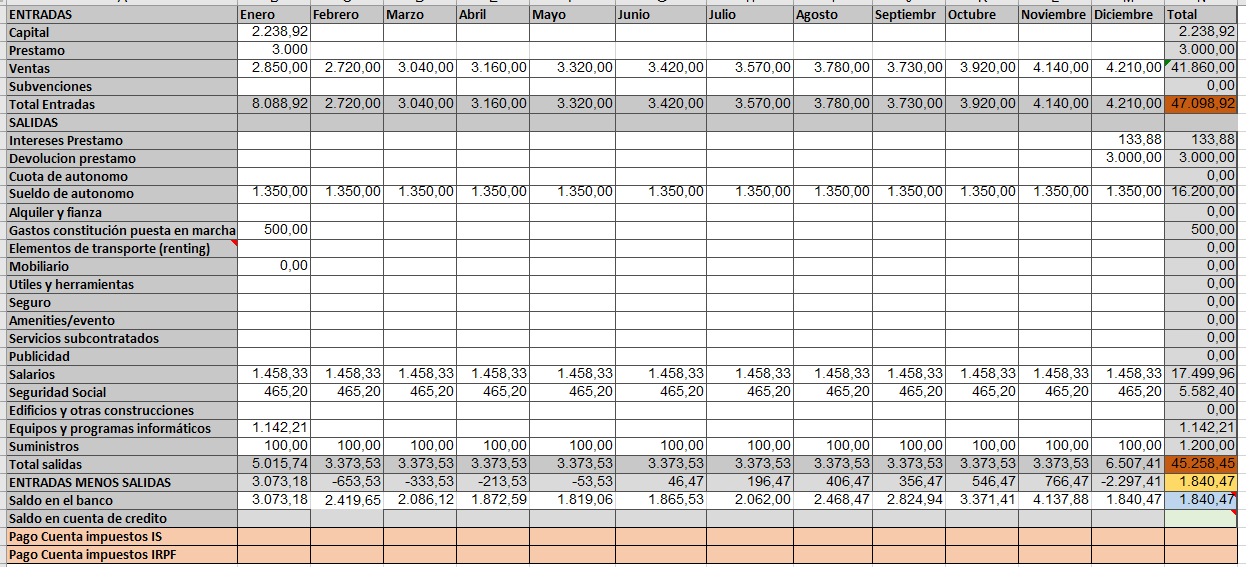
Intereses por cantidades gastadas (6% de 2077,65): **124,66€**

Intereses por cantidades no gastadas (1% de 922,35): **9,22€**

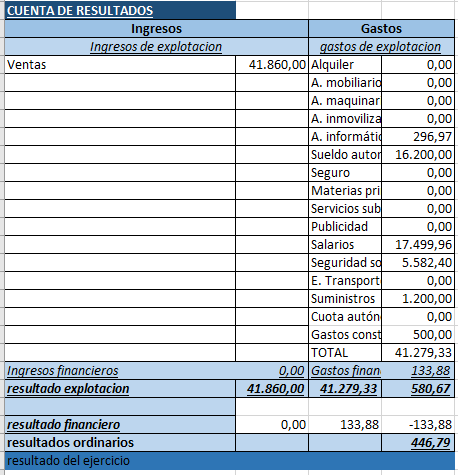
Intereses totales a pagar: **133,88€**

## Viabilidad económica-financiera

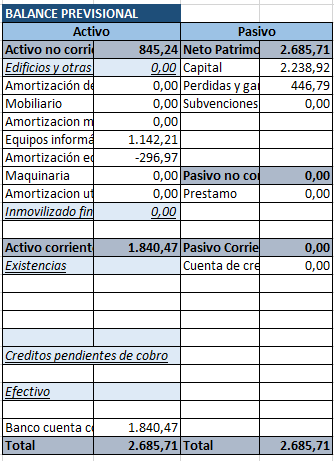
**Previsión de tesorería:**



**Cuenta de resultados**

****

**Balance patrimonial**



El plan financiero completo se encuentra en:

<https://github.com/jose-guillermo/JoseGuillermoDAW/blob/main/plan_financiero.xlsx>

# Conclusión

El desarrollo del proyecto se ha realizado utilizando tecnologías modernas y ampliamente adoptadas en la industria, lo que garantiza su viabilidad técnica y mantenimiento a largo plazo.

* **Frontend:** Se ha desarrollado con Ionic basado en Angular, tecnologías populares y robustas que facilitan la creación de aplicaciones web y móviles híbridas.
* **Backend:** Se ha optado por PHP junto con PostgreSQL como base de datos, tecnologías robustas y muy utilizadas en entornos web. También se ha integrado Firebase Realtime Database para la gestión de datos en tiempo real y Cloudinary para el almacenamiento de imágenes, lo que optimiza el rendimiento y escalabilidad del sistema.
* **Herramientas y Desarrollo:** Se ha utilizado Visual Studio Code como entorno de desarrollo, npm y Composer para la gestión de dependencias, y Postman para la prueba de la API.
* **Control de versiones y despliegue:** El código se gestiona a través de GitHub y Sourcetree, asegurando un historial claro de cambios y colaboración eficiente. El frontend está desplegado en Firebase Hosting, mientras que el backend se encuentra en Railway, proporcionando estabilidad y escalabilidad a la infraestructura del proyecto.

En cuanto a la viabilidad económica,

Se ha optado por una inversión inicial de 1.142,21€ con fondos propios, evitando así el pago de intereses o comisiones bancarias desde el inicio. Esto permite que el negocio comience sin deuda, lo que es ideal para reducir la presión financiera en los primeros meses.

Se ha solicitado un crédito de 3.000€, pero se dispone del dinero de forma escalonada para minimizar los intereses. Los intereses totales a pagar (133,88€) son razonables y permiten cubrir cualquier déficit sin afectar la viabilidad del negocio. A partir de mayo, ya no se planea usar más crédito, por lo qué la empresa empezará a ser rentable.

En la cuenta de resultados se muestra que en el primer año ya se habrá pagado el préstamo y estará en positivo incluso los equipos informáticos se habrán amortizado.

# Bibliografía/Webgrafía

**2.1 Prospectiva del título en el sector:**

<https://scriptcaseblog.net/es/development-es/tendencias-de-ti-y-desarrollo-web-para-2025/>

<https://www.mgpanel.org/post/las-10-tendencias-en-desarrollo-web-que-revolucionaran-el-2025>

**2.2 Evolución y tendencias del sector:**

<https://www.uam.es/uam/noticias/estudio-uam-vass-empleabilidad-talento-digital-2023>

**2.3 Normativa y documentación técnica específica:**

<https://www.accesibilidad-web.info/niveles-conformidad>

<https://olgacarreras.blogspot.com/2014/02/en-301-549-primera-norma-europea-de.html>

# Anexos

## ANEXO 1: Tecnologías y herramientas de desarrollo

A continuación, se detallan las tecnologías y herramientas utilizadas en el desarrollo del proyecto, así como su aplicación específica:

**Frontend:**

* **Angular e Ionic:** Desarrollo de la interfaz de usuario y experiencia en dispositivos móviles y web.
* **TypeScript, HTML y CSS:** Creación y estructuración de componentes, estilos y lógica del frontend.
* **Visual Studio Code:** Entorno de desarrollo utilizado para programar y depurar el código.
* **npm:** Gestión de dependencias y paquetes del frontend.

**Backend:**

* **PHP:** Desarrollo de la lógica del servidor y gestión de la API.
* **PostgreSQL:** Base de datos relacional utilizada para almacenar la información del sistema.
* **Firebase Realtime Database:** Almacenamiento de ciertos datos temporales y no relacionales en tiempo real.
* **Composer:** Gestión de paquetes en PHP, como cloudinary/cloudinary\_php para imágenes y firebase/php-jwt para autenticación con JWT.
* **Postman:** Herramienta para probar y depurar la API.

**Despliegue y almacenamiento:**

* **Firebase Hosting:** Hospedaje del frontend para acceso web.
* **Railway:** Plataforma utilizada para el despliegue del backend.
* **Cloudinary:** Servicio en la nube para gestionar imágenes dentro de la aplicación.

**Control de versiones:**

* **GitHub:** Almacenamiento y control del código fuente del proyecto.
* **Sourcetree:** Herramienta para la gestión visual de repositorios Git.