

SHAH



hazte el rey del tablero

Memoria proyecto final



Desarrollo de Aplicaciones Web 2025



Jorge Carretero



ENLACE GIT

<https://github.com/JCasuario/ProyectoFinalShah>

ENLACE PRODUCCIÓN

<https://shah-jcasuarios-projects.vercel.app/>

ENLACE MEMORIA

<https://github.com/JCasuario/ProyectoFinalShah/blob/main/MemoriaJorgeCarretero.pdf>

ENLACE MANUAL DE USUARIO

<https://github.com/JCasuario/ProyectoFinalShah/blob/main/Shah%20manual%20de%20usuario.pdf>

Usuarios: Momo, Lorena, jojo

Pass: ChangeMe

EN CASO DE INICIAR EN LOCAL DEBERAS REGISTRAR USUARIOS, LOS INTRODUCIDOS POR SCRIPT NO TIENEN LA CONTRASEÑA HASHEADA Y NO PODRÁN ACCEDER

Para poder ejecutar el back en local necesitarás tener instalado al menos Java 8, es recomendable hacer uso de IntelliJ o Eclipse para poder ejecutar y poder hacer uso del servidor apache incluido en el propio pack de spring.

Las variables a modificar de la base de datos según donde la instales y demás se localizan todos en [esta configuración](#).



ÍNDICE

INTRODUCCIÓN -pag3

Justificación del proyecto -pag4

1. DESCRIPCIÓN DEL PROYECTO -pag4

1.1. Características generales -pag4

1.2. Objetivos y alcance -pag4

2. ANÁLISIS DEL SECTOR / MERCADO -pag4

2.1. Prospectiva del título en el sector -pag4

2.2. Evolución y tendencia del sector -pag4

2.3. Normativa y documentación técnica específica -pag4

3. PLAN DE EJECUCIÓN -pag4

3.1. Diagrama de flujo de procesos -pag4

3.2. Proceso de desarrollo de software -pag4

 3.2.1. Fase de análisis -pag4

 3.2.2. Fase de desarrollo -pag4

 - Base de datos -pag4

 - Servidor -pag4

 - Diseño -pag4

 3.2.3. Fase de despliegue (despliegue en hosting) -pag4

3.3. Seguimiento del control de incidencias -pag4

3.4. Indicadores de calidad -pag4

4. RECURSOS MATERIALES -pag4

5. RECURSOS HUMANOS -pag4

6. VIABILIDAD TÉCNICA -pag4

7. VIABILIDAD ECONÓMICO-FINANCIERA -pag4

7.1. Inversiones y gastos -pag4

7.2. Financiación -pag4

7.3. Viabilidad económico-financiera -pag4

8. CONCLUSIÓN -pag4

ANEXO I -pag4



INTRODUCCIÓN-JUSTIFICACIÓN DEL PROYECTO.

Esta plataforma/aplicación representa la culminación de todo lo aprendido dentro y fuera del grado a lo largo de estos dos años. Integra tanto tecnologías fundamentales, como HTML, SQL o Java (en mi caso), que fueron la base de nuestro primer curso, como herramientas más modernas utilizadas en proyectos actuales, como Vue (que abarca aspectos de frontend como JavaScript y CSS) y Spring (orientado al backend).

Además, ha supuesto un verdadero desafío en cuanto al diseño de interfaces, ya que al tratarse de un juego, es fundamental ofrecer a los usuarios una navegación rápida, clara e intuitiva. La incorporación de un modo en línea y chats entre usuarios también ha implicado retos adicionales, especialmente en lo relativo al despliegue y al estudio de redes.

Por último, ha sido clave la fase de planificación y diseño del proyecto, en la que se han utilizado herramientas de control y gestión de versiones, fundamentales para el trabajo en equipo y el desarrollo organizado.

Este proyecto también supone una demostración de mi nivel actual como profesional y de los retos logísticos y técnicos a los que ha sido necesario enfrentarse.



DESCRIPCIÓN EL PROYECTO.

1.1 CARACTERÍSTICAS GENERALES

El proyecto consiste en el desarrollo de una aplicación web que permite a los usuarios registrarse y enfrentarse entre ellos a un juego ideado por mí, Shah. En la web también podrán comprar distintos cosméticos, ver los rankings y chatear con otros usuarios.

La página web estará construida mediante el framework Vue para toda la sección del cliente y el framework de Java Spring para la parte del servidor.

El diseño trata de ser orgánico, juvenil pero accesible para todo tipo de jugadores, se tiene en cuenta tanto una versión de ordenador como de móvil.

1.2 Objetivos y alcance

El principal objetivo de esta aplicación es mostrar mi nivel actual, que a pesar de lo acotado del tiempo junto a las prácticas de empresa llevadas a cabo a la par con el desarrollo de este proyecto final puede resultar extenuante.

Para demostrar este nivel del que hablo la aplicación debe tener una serie de funcionalidades obligatorias a pleno rendimiento:

- La primera es el juego, con su sistema de turnos y de control de piezas funcione a la perfección.
- Las conexiones sockets tanto para las partidas como el chat se lleven a cabo de forma eficiente.
- La gestión de los usuarios, tanto a nivel personal (comprar, añadir amigos) como a nivel de administrador (banear, añadir y quitar puntos) funcionen adecuadamente

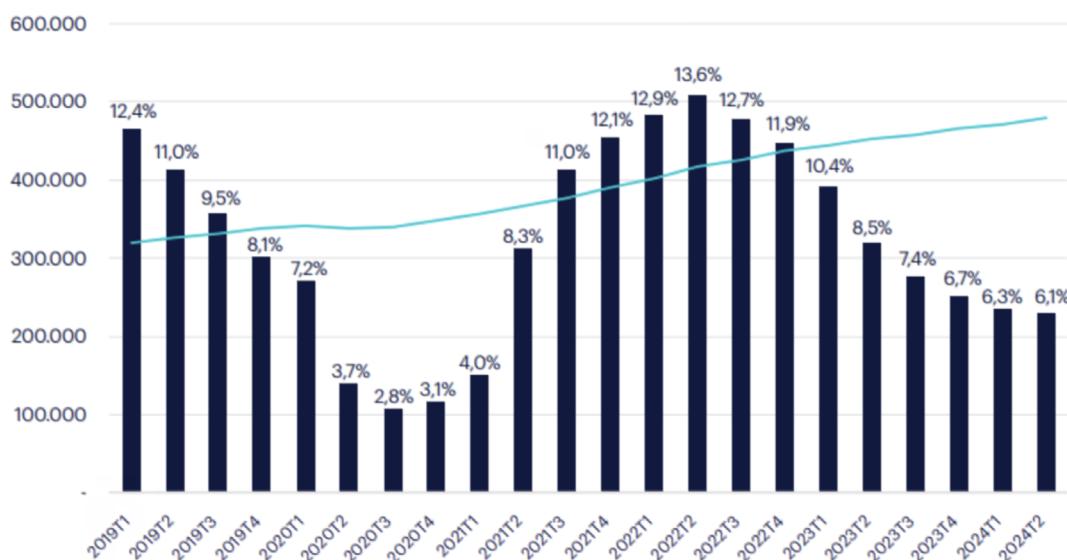
2. ANÁLISIS DEL SECTOR/MERCADO

2.1 PROSPECTIVA DEL TÍTULO EN EL SECTOR

El sector de las Tecnologías de la Información y Comunicación (TIC) en España se ha convertido en uno de [los pilares fundamentales de la economía nacional](#), mostrando un dinamismo excepcional en los últimos años. Actualmente, el sector factura más de 124.000 millones de euros anuales. Sin embargo, este porcentaje sigue siendo inferior al de otros países europeos como Suecia o Finlandia, donde los especialistas digitales superan el 7% de la fuerza laboral.

Afiliación a la Seguridad Social en 62. Actividades informáticas
(% de variación interanual)

Fuente: MISSM



A pesar del crecimiento constante, [el mercado laboral TIC enfrenta importantes desafíos estructurales](#). La escasez de talento cualificado se ha convertido en el principal obstáculo, con un 73% de las empresas tecnológicas reportando dificultades para encontrar perfiles adecuados. Esta situación es particularmente crítica para las pymes, que no pueden competir con los paquetes salariales y beneficios que ofrecen las grandes corporaciones.

Las proyecciones para los próximos años, no obstante, son notablemente optimistas. Los analistas prevén que el empleo en el sector TIC crecerá alrededor del 1,8% en 2025, manteniendo una tendencia alcista que ya acumula más de 40 meses consecutivos de crecimiento interanual positivo. Más de la mitad de los empresarios del sector (55,9%) tienen previsto aumentar sus plantillas en el corto plazo, siendo el sector tecnológico el más optimista en cuanto a generación de empleo, con una previsión neta del 27% para el primer trimestre de 2025 según [Data Center Market](#).

Entre las [áreas que presentan mayores oportunidades laborales](#) destacan especialmente la inteligencia artificial y el big data, donde se está produciendo una reconfiguración profunda de los procesos empresariales. La ciberseguridad se ha convertido en otra prioridad estratégica, con una demanda creciente de profesionales especializados en protección de datos e infraestructuras críticas. El despliegue del 5G y el desarrollo del Internet de las Cosas (IoT) están generando nuevas necesidades de talento.

2.2 EVOLUCIÓN Y TENDENCIA DEL SECTOR

El marco regulatorio y las políticas públicas están jugando un papel clave en esta transformación. Iniciativas como el Plan España Digital 2026 y el programa Década Digital 2030 de la UE están acelerando la transformación digital del país, que ya lidera en la UE en cobertura de redes de alta capacidad, alcanzando al 93% de la población. Estas políticas, combinadas con el impulso privado, están configurando un ecosistema tecnológico cada vez más robusto y competitivo.

Sectores que más empleo crean en España

Noticias TRABAJO



Fuente: INE Gráfico: Noticiastrabajo

En perspectiva, el sector TIC español se encuentra en una fase de expansión sostenida, pero necesita abordar retos críticos para mantener su competitividad. Castilla la Mancha tiene un crecimiento más contenido en el sector, las empresas de la tecnología siguen creciendo y aumentando su contratación, pero es una de las regiones donde lo hace de forma más disimulada.

Principal fuente de información.



2.3 NORMATIVA Y DOCUMENTACIÓN TÉCNICA ESPECÍFICA

A la hora de trabajar y desarrollar productos en el sector hay que tener en cuenta distintos tipos de normativas y reglas como pueden ser la gestión de los datos y la información de los usuarios por parte de Europa y España:

- [Reglamento General de Protección de Datos \(RGPD\)](#): Aplicable a cualquier software que maneje datos personales. Establece requisitos de privacidad, seguridad y transparencia 11.
- [Ley de Servicios de la Sociedad de la Información \(LSSI\)](#): Regula el comercio electrónico y la prestación de servicios digitales en España.
- Desde la perspectiva del diseño y la accesibilidad podemos verlo desde el oficial como el europeo o el español: [Real Decreto 1112/2018](#): Transpone la directiva europea y exige que el software público cumpla con:
 - Diseño perceptible: Alternativas textuales para multimedia, contraste mínimo.
 - [Navegación operable](#): Funcionamiento con teclado, evitando contenido parpadeante.
- Ley 11/2023: Extiende estas obligaciones al sector privado, aplicable a:
 - Interfaces de usuario: Formularios accesibles, etiquetas ARIA para lectores de pantalla.

Y los estándares Técnicos para Diseñadores de agencias y estudios:

- [WCAG 2.1: Guía internacional con 4 principios](#):

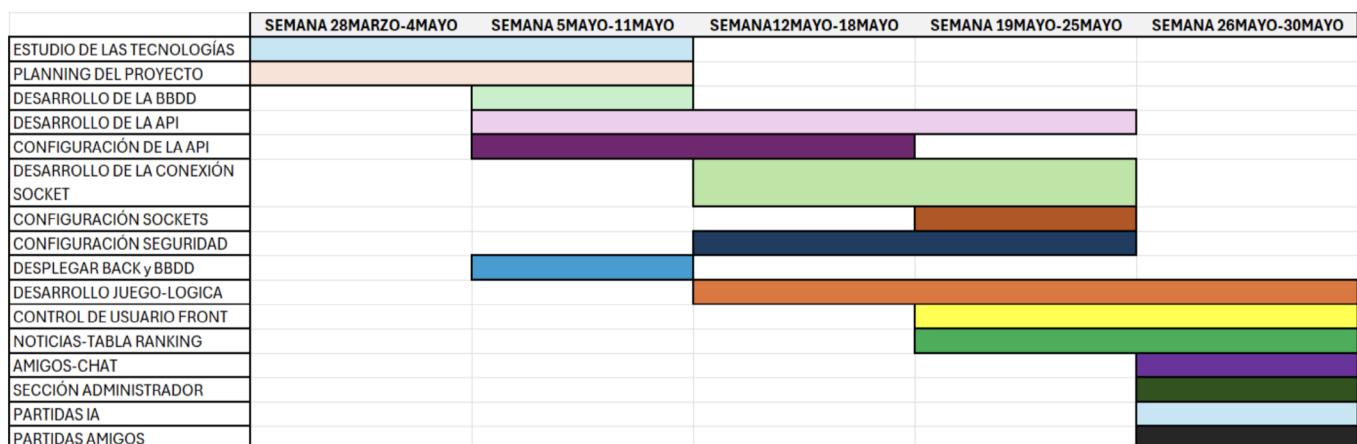
- 1. Perceptible:** Texto alternativo, subtítulos, adaptabilidad visual.
- 2. Operable:** Tiempo suficiente para interacciones, evitando trampas de teclado.
- 3. Comprensible:** Lenguaje claro, predicción de errores en formularios.
- 4. Robusto:** Compatibilidad con navegadores y tecnologías asistivas 811.



3. PLAN DE EJECUCIÓN

3.1 DIAGRAMA DE FLUJO DE PROCESOS

El sector de las Tecnologías de la Información y Comunicación (TIC) en España se ha convertido en uno de los pilares fundamentales de la economía nacional, mostrando un dinamismo excepcional en los últimos años. Actualmente, el sector factura más de 124.000 millones de euros anuales. Sin embargo, este porcentaje sigue siendo inferior al de otros países europeos como Suecia o Finlandia, donde los especialistas digitales superan el 7% de la fuerza laboral.



[Enlace al diagrama en Excel.](#)

Fecha de inicio de análisis del proyecto: **28 de marzo**.

Inicio de codificación: **5 de mayo**.

Fin de la realización del proyecto: **29 de mayo**.



3.2 PROCESO DE DESARROLLO DE SOFTWARE

3.2.1 FASES DE ANÁLISIS

Tipos de usuario

En nuestra aplicación existen dos tipos de usuarios que interactúan con el sistema:

- **Jugadores:** Son usuarios que pueden crear y utilizar una cuenta personal para acceder a las funcionalidades del juego. Entre sus principales acciones se encuentran: jugar partidas, agregar y chatear con amigos u otros jugadores, y enfrentarse tanto a otros usuarios como a la inteligencia artificial. Estas actividades les permiten ganar puntos para escalar posiciones en el ranking —visible desde la propia plataforma—, así como obtener monedas virtuales que pueden utilizar en la tienda para adquirir elementos cosméticos u otros objetos disponibles.
- **Administradores:** Estos usuarios cuentan con privilegios especiales y sus cuentas no pueden ser creadas ni modificadas desde la propia aplicación. Acceden a través de un endpoint exclusivo y tienen la capacidad de visualizar los registros (logs) de las partidas, así como consultar y modificar información relacionada con los jugadores.

Descripción de requisitos

La aplicación contempla una serie de funcionalidades divididas según el tipo de usuario.

Funcionalidades para jugadores. Los jugadores podrán:

- **Registrarse, iniciar sesión y recuperar su contraseña** en caso de haberla olvidado. Una vez autenticados, tendrán acceso completo a su perfil.
- **Visualizar su información personal** desde su página de perfil y acceder a las diferentes secciones de la aplicación.
- **Consultar noticias** relevantes relacionadas con el juego desde la página de inicio.
- **Acceder a la tienda**, donde podrán visualizar la cantidad de monedas virtuales disponibles en su cuenta y explorar los productos cosméticos u otros elementos que pueden adquirir con ellas.
- **Ver el ranking global** de jugadores del servidor, donde podrán consultar las posiciones y calificaciones de todos los usuarios.
- **Agregar amigos** directamente desde el ranking. A través del menú lateral desplegable, también podrán:
 - Iniciar chats con amigos agregados.
 - Cambiar su propia foto de perfil.



Desde **la sección de juego**, los usuarios tendrán cuatro opciones:

- Jugar contra la máquina.
- Jugar contra un amigo.
- Jugar en local.
- Visualizar las instrucciones del juego.

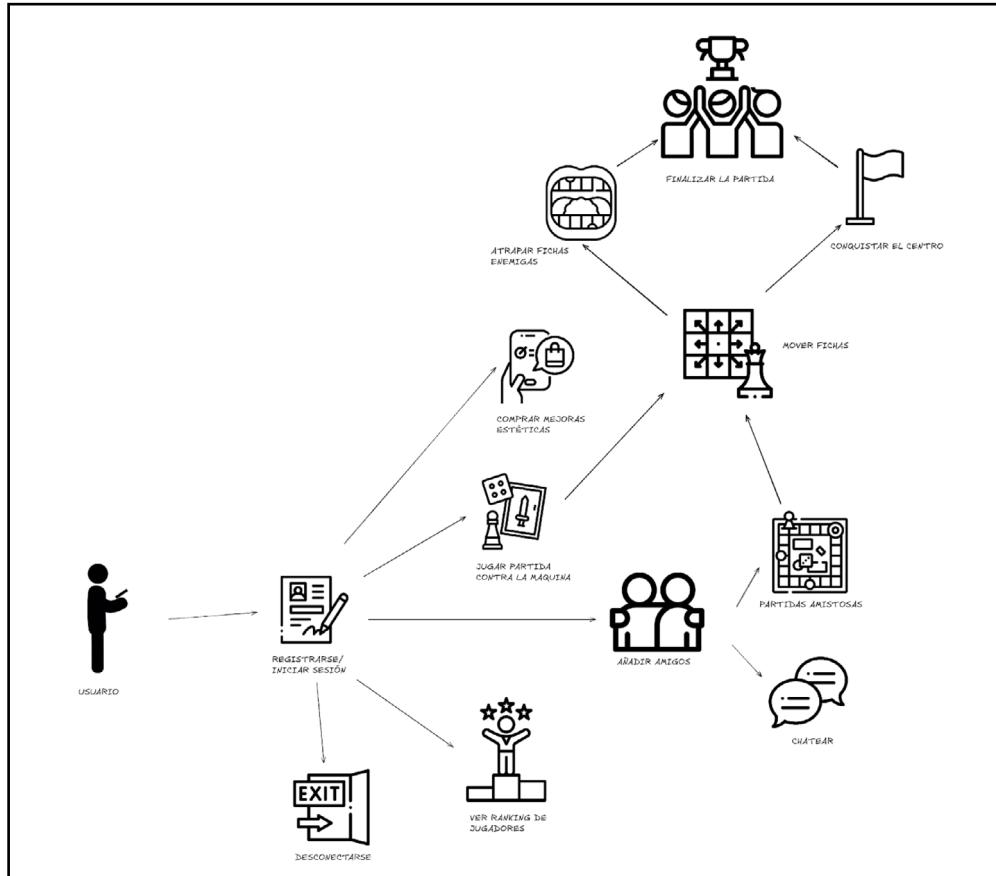
La **mecánica del juego** es común en todos los modos. Durante su turno, cada jugador podrá mover sus piezas siguiendo un conjunto específico de reglas. El objetivo del juego consiste en **obtener dos puntos**, los cuales pueden conseguirse mediante:

- La eliminación de piezas principales del oponente.
- El control de la sección central del tablero.

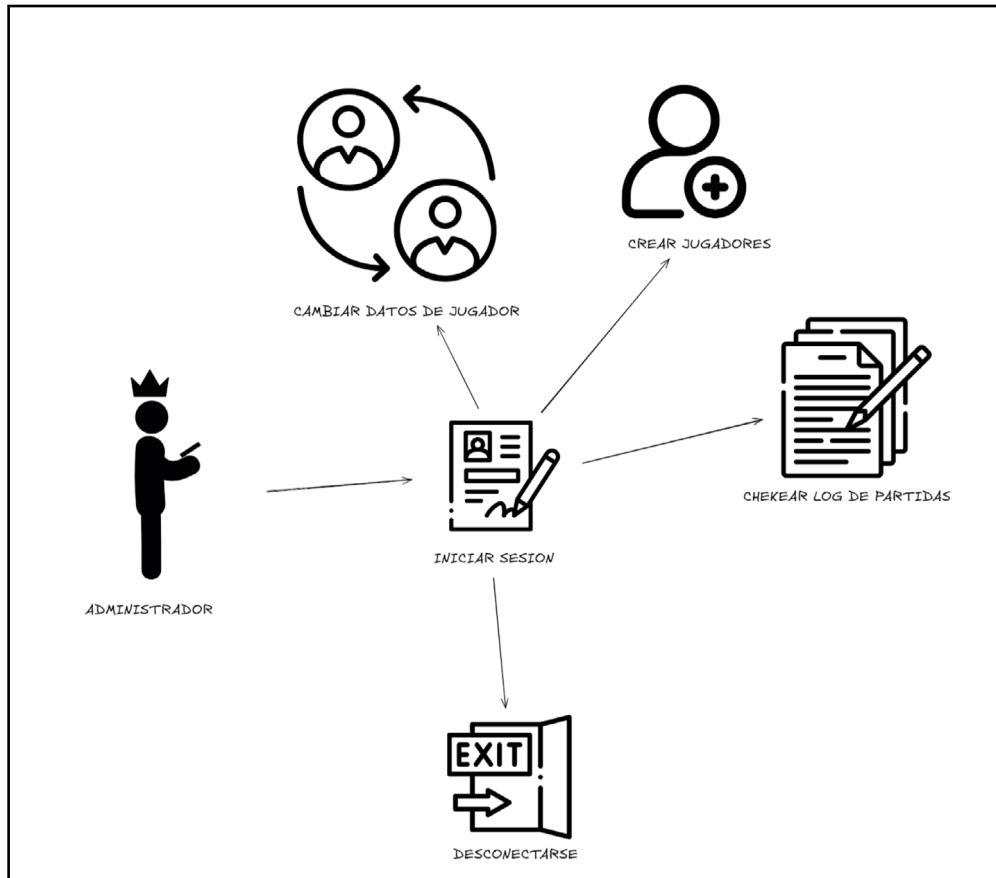
Además, todas las piezas pueden evolucionar si cumplen con uno de los objetivos anteriores, lo que les otorga un **rango de movimiento significativamente mayor** a lo largo del tablero.

Funcionalidades para administradores

- Los administradores contarán con un acceso especial mediante un sistema de inicio de **sesión independiente del de los jugadores**. Desde su panel de administración, podrán:
 - Visualizar y modificar una tabla con determinados datos de los jugadores.
 - Editar las noticias que se muestran en la página de inicio, incluyendo la imagen asociada a cada una.
 - Consultar el último log de juego registrado por cada usuario, con el fin de llevar un control y supervisión más detallada de la actividad dentro del sistema.



Casos de uso de jugador/usuario.



Casos de uso de administrador.



Una vez autenticados, tanto los jugadores como los administradores podrán **navegar por la aplicación** utilizando el menú de navegación principal o el menú lateral desplegable. La experiencia y el flujo de navegación de cada usuario dependerán del **motivo de su visita** y del **tipo de cuenta** con la que haya iniciado sesión.

- Un **administrador** podrá iniciar sesión para realizar tareas específicas como **modificar datos de jugadores, gestionar noticias o consultar los registros** (logs) de partidas. Sin embargo, también puede simplemente acceder para **visualizar información** sin realizar modificaciones.
- Por su parte, un **jugador** puede utilizar su cuenta para diversas actividades: acceder a su perfil, iniciar una partida (ya sea contra otro jugador o contra la máquina), interactuar con sus amigos mediante el chat, o simplemente **explorar las últimas novedades** en la sección de noticias o revisar y adquirir productos en la tienda.

Guia de Estilo

El proceso de definición estética del proyecto fue **largo y complejo**, implicando múltiples pruebas y aproximaciones hasta alcanzar una solución satisfactoria. En las etapas iniciales se experimentó con **amplias gamas de colores**, lo que dificultaba mantener una coherencia visual en la interfaz. Posteriormente, se optó por un enfoque más minimalista, explorando incluso **paletas compuestas por únicamente dos colores**, en busca de simplicidad y claridad.

Finalmente, tras un proceso iterativo de diseño y simplificación, se definió una **paleta final compuesta por cinco colores**:

Tres colores principales destinados al diseño de la interfaz (UI/UX), garantizando una experiencia visual agradable y consistente.

Dos colores adicionales, utilizados específicamente para diferenciar las piezas en el tablero de juego, asegurando una correcta identificación durante las partidas.

Este enfoque permitió equilibrar la **estética con la funcionalidad**, facilitando tanto la navegación como la jugabilidad para los distintos tipos de usuarios.



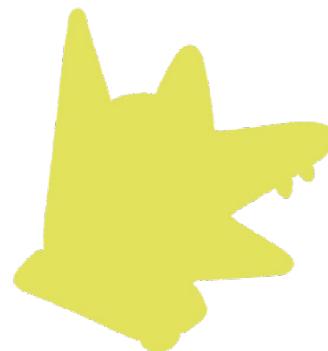
#313240



#10d8a1



#f98397



#e2e271

#FFFFFF White

El “verde” junto al “gris” nos da dos colores con un gran contraste entre sí y el blanco nos permite desplegar distintos ítems y textos corridos sin tener que preocuparnos en ningún momento por la legibilidad de estos. También es una combinación que dependiendo el color al que se le dé prioridad podremos obtener resultados más vibrantes y juveniles junto a otros más sobrios y tranquilos sin tener que preocuparnos de que ningún usuario tenga dificultad para acceder a la página y hacer uso de esta.

Para los colores de las fichas el equipo se decidió por rojo y el verde anteriormente mostrados.

Dos colores contrastados y diferenciados que funcionan a la perfección con los tres anteriores y dan personalidad a ambos lados del tablero.

En el diseño de la aplicación se han empleado **dos tipografías principales**, seleccionadas cuidadosamente para combinar **legibilidad, estética y personalidad** visual:

- **Inter:** Una tipografía sans serif sobria y elegante, ampliamente reconocida por su excelente legibilidad en entornos digitales. Su versatilidad permite utilizarla de forma eficaz en distintos contextos dentro de la interfaz, lo que la convierte en una elección ideal para cuerpos de texto, menús y elementos informativos.
- **Pirata:** Tipografía de inspiración estilográfica que remite a las fuentes góticas clásicas, tradicionalmente creadas con pluma estilográfica. Esta fuente ha sido ligeramente simplificada para mejorar su legibilidad sin perder su carácter distintivo. En la aplicación, se utiliza de forma puntual para añadir personalidad y énfasis visual en elementos destacados, como títulos de secciones o nombres relevantes, por ejemplo, el nombre del usuario con el que se está chateando en ese momento.



Esta combinación tipográfica equilibra funcionalidad y estilo, aportando claridad a la experiencia de usuario sin renunciar a un diseño con identidad propia.

Para los elementos visuales de la aplicación —como ilustraciones, fichas y otros componentes gráficos— se ha optado por un estilo intermedio entre un **aspecto toon (caricaturesco)** y los grabados **medievales clásicos**. Esta combinación busca transmitir un **tono juvenil, atractivo y accesible, pero al mismo tiempo dotado de personalidad y carácter propio**.

El objetivo principal era encontrar un **estilo visual que reflejara la identidad y filosofía del proyecto**, equilibrando lo lúdico con lo temático. El resultado es una estética distintiva que encaja de forma coherente con el resto de los elementos de la interfaz, reforzando tanto la marca visual como la **experiencia general del usuario**.





Prototipado

Aquí mostraré las principales páginas que se encontrarán en la web:

Login

The image shows two prototypes of the login page. The left prototype, titled 'LOGIN', is for a desktop or laptop screen. It features a yellow header with the Shah logo and two input fields labeled 'USUARIO' and 'CONTRASEÑA'. Below these fields are links for password recovery ('¿Has olvidado de la contraseña?') and account creation ('CREAR CUENTA'). A large blue button labeled 'INICIAR' is at the bottom. The right prototype, titled 'LOGIN MOVIL', is for a mobile device. It has a similar layout but is oriented vertically. The 'INICIAR' button is also blue.

Noticias/home

The image shows two prototypes of the news/home page. The left prototype, titled 'NOTICIAS ORDENADOR', is for a desktop. It features a top navigation bar with four tabs: 'NOTICIAS' (highlighted in yellow), 'TIENDA', 'RANKING', and 'JUGAR'. Below the tabs is a grid of six dark green rectangular boxes. To the right is a sidebar with sections for 'PERFIL' (showing 'AMIGOS' with three entries: 'amigo1', 'amigo2', and 'amigo3'), 'CHAT' (listing four messages from users 'Jefij prever', 'Tal y tal', and 'Tal y tal'), and a 'GALERIA' section. The right prototype, titled 'NOTICIAS MOVIL', is for a mobile device. It has a similar layout but is oriented vertically. The 'JUGAR' tab is highlighted in yellow at the bottom of the sidebar.



Tienda

NOTICIAS

TIENDA

RANKING

JUGAR

FICHAS

TABLEROS

EFECTOS VISUALES

PERFIL

AMIGOS

amigo1

amigo2

amigo3

CHAT

Jaja y yo

Bueno y mal

Jaja y yo

Bueno y mal

Tod y yo

Bueno y mal

Tod y yo

Bueno y mal

80~

75~

90~

Ranking

The image shows a mobile application interface with a dark green header bar. The header contains five rounded rectangular buttons labeled "NOTICIAS", "TIENDA", "RANKING" (highlighted in yellow), "JUGAR", and "PERFIL". Below the header is a large yellow rectangular area containing three horizontal rows of buttons: "JUGADOR", "POSICIONES", and "PUNTOS". To the right of this yellow area is a vertical sidebar with a dark green background. The sidebar features a circular "PERFIL" button at the top, followed by a section titled "AMIGOS" listing "amigo1", "amigo2", and "amigo3" each with a small icon. There is a "Agregar Amigo" button with a magnifying glass icon below this. A section titled "CHAT" follows, showing four entries: "Jajaj jajaja", "Jajaj jajaja", "Tal y cas", and "Tal y cas", each with a "Bueno y tal" status indicator. At the bottom of the sidebar is a "Desconectar" button. To the right of the sidebar is a large white rectangular area with a grid of 12 empty cells. The top row of the grid has labels "JUGADORES", "POSICIÓN", and "PUNTOS" above it. The bottom right corner of this area contains a "JUGAR" button with a rounded rectangle outline.

Desarrollo de Aplicaciones Web, 2025, Jorge Carretero



Jugando/Partida

TIEMPO 10:02 **RIVAL : XxxDrakNomiaxxX** **JUGAR**

RENDIRSE

JUGAR MOVIL

JUGAR
Tiempo 2:50 Rival: xxxDrakNomiaxxX

RENDIRSE

Vista administrador

JUGADORES
CHANGE
CREAR JUGADOR
VER LOGS

JUGADOR	LOGS	CAMBIAR ESTADO BAN	ELIMINAR
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	
		<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

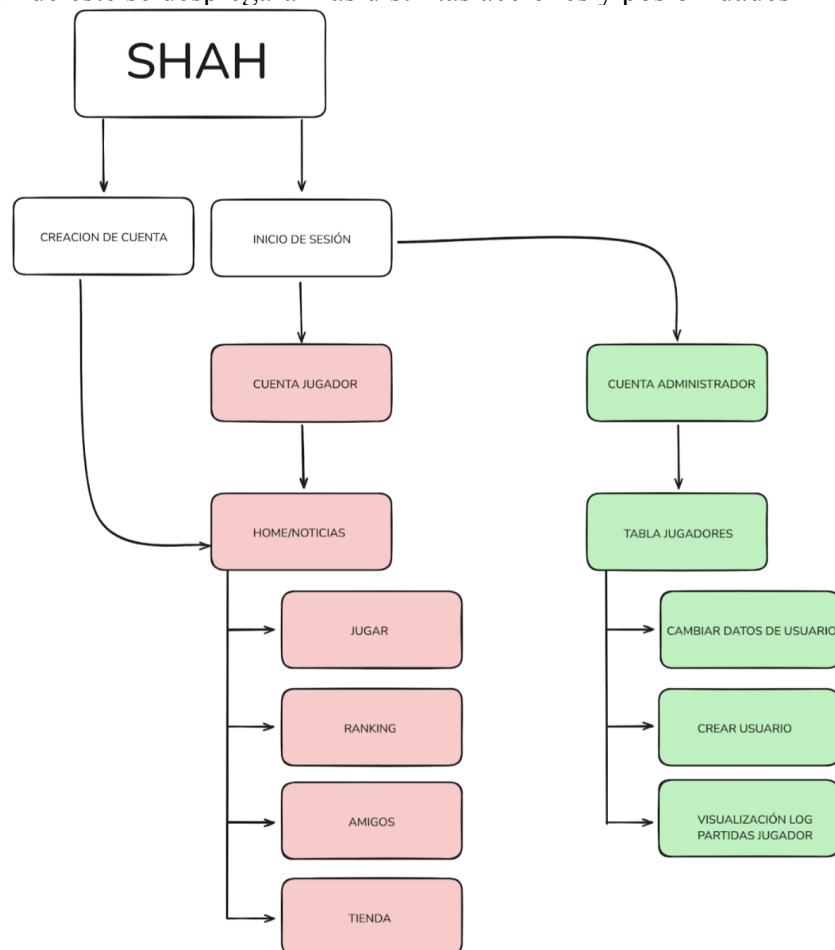
Uno de los objetivos fundamentales en el diseño de la aplicación fue **asegurar que cada sección tuviera una función específica, clara y diferenciada**. Desde el principio se planteó una arquitectura de información donde cada página cumple un propósito concreto, evitando la sobrecarga de contenido y facilitando la navegación del usuario.

La premisa principal fue garantizar la simplicidad y claridad en la experiencia de uso, permitiendo que cualquier usuario —ya sea jugador o administrador— pudiera comprender rápidamente la utilidad de cada sección y acceder de forma intuitiva a las funcionalidades disponibles.

Este enfoque contribuye directamente a una interfaz más usable, donde la jerarquía visual, la organización de contenidos y la disposición de los elementos cumplen un papel esencial en la **eficiencia y comodidad de la interacción**.

Mapa de navegación

Como ya se ha mencionado anteriormente el mapa de navegación de la página empieza con el login/registro y a partir de este se desplegarán las distintas acciones y posibilidades



Parece un mapa de navegación muy sencillo o con pocas opciones, pero hay que tener en cuenta que cada una de las páginas tendrán una gran cantidad de cargas y trabajos.



3.2.2 FASES DE DESARROLLO

3.2.1 BASE DE DATOS

Análisis de requisitos de la base de datos

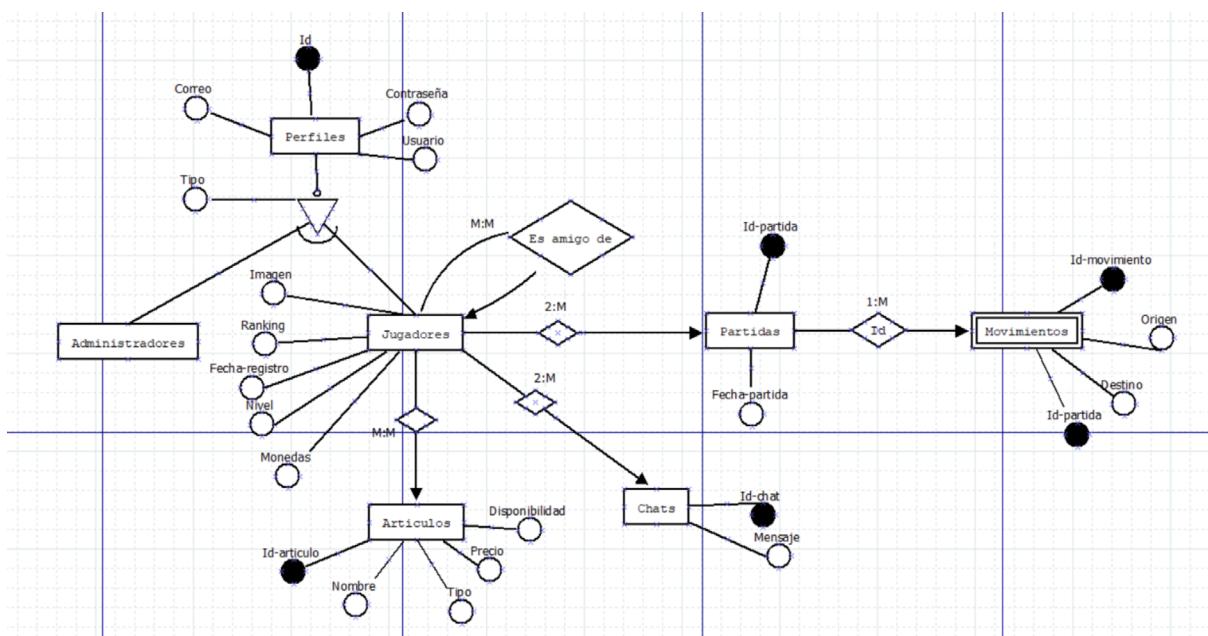
Desde el inicio del desarrollo se estableció una **distinción entre dos tipos de usuarios** principales: jugadores y administradores. Aunque ambos comparten ciertas características comunes —como el uso de un correo electrónico, nombre de usuario y contraseña—, sus roles dentro de la aplicación son claramente diferenciados. Esta similitud permitió partir de un **modelo base común de usuario**, sobre el cual se especializan las funciones y relaciones de cada tipo.

En el caso del **administrador**, su papel es más limitado en cuanto a relaciones dentro del sistema, ya que su funcionalidad se centra en la gestión y supervisión. Por lo tanto, su estructura en la base de datos es más sencilla.

Por otro lado, el **jugador** constituye el núcleo funcional de la aplicación, lo que se refleja en la complejidad de sus relaciones dentro del modelo de datos. Para poder interactuar con el sistema y cumplir con los distintos objetivos de la aplicación, el jugador requiere conexión con múltiples tablas, como:

- **Chat:** para gestionar las conversaciones entre usuarios.
- **Partidas:** para almacenar información sobre los encuentros jugados, sus participantes, resultados, y logs asociados.
- **Artículos:** elementos disponibles en la tienda, como cosméticos y otros objetos, que el jugador puede adquirir con monedas virtuales.

Tras un análisis detallado de las necesidades de la aplicación y sus funcionalidades, se definió el siguiente modelo de base de datos, que refleja las relaciones clave y dependencias entre entidades:





Paso a tablas

-Administradores:

Correo

- Usuario
- Contraseña

-Jugadores:

·Id-jugador

- Usuario
- Contraseña
- e-mail
- Fecha-registro
- Ranking
- Nivel
- Monedas
- Imagen

-Amigos:

·JUGADOR-ID

·AMIGO-ID

-Partidas:

·Id-partida

- ID-JUGADOR1
- ID-JUGADOR2
- Fecha-partida

-Movimientos:

·Id-movimiento

- ID-PARTIDA
- JUGADOR-ID
- Origen
- Destino

-Articulos:

·Id-articulo

- Tipo
- Nombre
- Precio
- Disponibilidad

-Artículos-adquiridas:

·ID-JUGADOR

·ID-ARTICULO

- Fecha-compra

-Chats:

- Id-chat
- ID-REMITENTE
- ID-RECEPTOR
- Mensaje



Normalización de la base de datos

El modelo de base de datos diseñado para la aplicación se encuentra normalizado, cumpliendo con las tres primeras formas normales. A continuación, se detallan los criterios que se han seguido para garantizar dicha normalización:

- **Primera Forma Normal (1FN):**

- Todos los atributos contienen **valores atómicos**, es decir, no existen campos con listas, conjuntos o múltiples valores en una misma celda.
- Se ha evitado la presencia de **grupos repetitivos**, asegurando una estructura plana y coherente en todas las tablas.

- **Segunda Forma Normal (2FN):**

- El modelo está en 1FN.
- No existen **dependencias parciales**: todos los atributos no clave dependen **de la clave primaria completa**, especialmente en aquellas tablas donde la clave primaria es compuesta (como en las relaciones de amistad o artículos adquiridos).

- **Tercera Forma Normal (3FN):**

- El modelo está en 2FN.
- No se presentan **dependencias transitivas**, es decir, ningún atributo no clave depende de otro atributo no clave. Todos los datos dependen **únicamente de la clave primaria** correspondiente.

Tipos de datos

Este proceso de normalización garantiza una estructura eficiente y coherente, reduciendo la redundancia y facilitando el mantenimiento, integridad y escalabilidad del sistema de base de datos.

Durante el diseño de la base de datos, se seleccionaron los tipos de datos más adecuados para representar de forma eficiente la información y garantizar un buen rendimiento. A continuación, se describen los principales tipos utilizados y su propósito:



- **VARCHAR:** Se utilizó para almacenar valores alfanuméricos de longitud variable, como nombres de usuario, direcciones de correo electrónico o contraseñas. En el caso específico de las contraseñas, al estar hasheadas, requieren mayor longitud, por lo que se les asignó el máximo número de caracteres permitido dentro del estándar razonable.
- **TINYINT:** Este tipo se empleó para representar valores booleanos (por ejemplo, verdadero o falso). Aunque BOOLEAN es una opción válida, en MySQL este tipo se implementa internamente como un TINYINT(1), por lo que se decidió usar directamente TINYINT, aprovechando así su comportamiento subyacente y dejando explícito su uso.
- **TEXT:** Se utilizó para campos que podían contener textos extensos, como mensajes en los chats o contenido de las noticias, donde un VARCHAR(255) resultaba insuficiente.
- **INT:** Se reservó para valores numéricos enteros, especialmente aquellos que podían participar en operaciones aritméticas, como el ranking de los jugadores, la cantidad de monedas, los identificadores numéricos y otros campos relacionados con cálculos.
- **TIMESTAMP:** Se usó para almacenar fechas y horas correspondientes a eventos como el registro de un usuario, el inicio de una partida o la compra de un artículo.

Cabe señalar que durante la implementación, MySQL Workbench realizó algunos ajustes automáticos en ciertos tipos de datos (por ejemplo, transformando algunos campos TEXT en TINYTEXT). No obstante, estos cambios no afectaron al script original de creación de tablas, ni al funcionamiento general del sistema.



3.2.2.2 SERVIDOR

Análisis de requisitos de la base de datos

Como se ha mencionado previamente a lo largo de esta memoria, el desarrollo del lado servidor se ha llevado a cabo utilizando el framework Java Spring. Esta herramienta proporciona una potente infraestructura que permite trabajar con bases de datos de manera eficiente, adoptando una perspectiva muy cercana a la programación orientada a objetos.

Existen diversas formas de estructurar un proyecto en Spring, pero en este caso se optó por un enfoque tradicional basado en el patrón Modelo – Repositorio – Servicio – Controlador (Model–Repository–Service–Controller). Este patrón favorece una arquitectura limpia, modular y escalable, donde cada capa cumple una función específica:

- **Modelo (Model):**

Cada tabla de la base de datos cuenta con una clase que la representa como un objeto en Java. Estas clases incluyen atributos que reflejan directamente las columnas de sus respectivas tablas en MySQL. Por ejemplo, un campo VARCHAR usuario en la base de datos se traduce en un atributo String usuario en el modelo. Además, estas clases incorporan los métodos típicos de acceso (getters y setters), constructores y anotaciones necesarias para mapear correctamente la entidad a la base de datos.

- **Repositorio (Repository):**

Es la capa encargada de interactuar directamente con la base de datos. En ella se definen las consultas SQL, ya sea mediante el lenguaje propio de Spring Data JPA, a través de anotaciones con SQL tradicional, o bien utilizando las consultas automáticas que Spring genera a partir de los nombres de los métodos.

- **Servicio (Service):**

Aquí se implementa toda la lógica de negocio. Esta capa se encarga de coordinar las operaciones entre el repositorio y el controlador, gestionando las peticiones, validaciones, transformaciones y cualquier otra operación relacionada con los datos.

- **Controlador (Controller):**

Su función es exponer los servicios mediante endpoints accesibles a través de URLs. Los controladores se encargan de recibir las solicitudes HTTP, procesarlas (delegando en los servicios) y devolver la respuesta correspondiente, funcionando así como la puerta de entrada a la API del servidor.

Esta estructura modular permite una separación clara de responsabilidades, facilita el mantenimiento del código y mejora su escalabilidad y reutilización.



FUNCIONES PRINCIPALES DEL SERVIDOR

Todas las entidades del proyecto disponen del conjunto básico de operaciones CRUD (Create, Read, Update, Delete), organizadas a través de métodos genéricos que permiten interactuar con la base de datos de forma eficaz. A continuación, se describen estas funciones comunes:

- **Funciones CRUD genéricas**

- **getAll()**

Devuelve todas las entradas presentes en la tabla correspondiente.

- **getById(id)**

Retorna la entrada asociada al identificador proporcionado. Permite consultar un elemento concreto de la tabla a partir de su ID.

- **save(entrada)**

Guarda o actualiza una entrada en la base de datos.

Si el objeto proporcionado ya existe (es decir, si el ID coincide con uno presente en la tabla), los campos modificados se actualizan.

Si el objeto no existe (porque el ID no está presente o es nulo), se crea una nueva entrada.

Estas funciones han sido implementadas en todas las entidades del sistema, proporcionando una interfaz coherente y reutilizable para las operaciones básicas.

Funciones específicas: Envío de mensajes socket

En estas funciones se ha incluido sus anotaciones java para poder comprender su uso y profundidad. Con message mapping se indica el lugar desde el cual se hará la llamada de la función y send to a donde se deben mandar dichos mensajes controlados por el método.

```
@MessageMapping("/chat/{username}")
@SendTo("/chat/{username}")
```

- **Mensaje sendMessage(@DestinationVariable String username, Mensaje message)**

Función destinada a mandar mensajes a un canal privado de cada usuario, utilizado para los chats privados.

```
@MessageMapping("/game/{username}")
@SendTo("/game/{username}")
```

- **Tablero sendTablero(@DestinationVariable String username, Tablero tablero)**

Función empleada para pasar la matriz del tablero tras cada turno.



Funciones específicas: Gestión de autenticación y tokens

Además de las funciones CRUD estándar, el sistema incorpora una serie de funciones especializadas relacionadas con la gestión de autenticación y el manejo de tokens JWT. Estas funciones permiten implementar seguridad de forma robusta en la aplicación. A continuación, se detallan:

- **public String extractUsername(String token)**
Extrae y devuelve el nombre de usuario contenido en el token JWT proporcionado.
- **public <T> T extractClaim(String token, Function<Claims, T> claimsResolver)**
Permite acceder a un “claim” (información codificada) específico dentro del token JWT, mediante una función de resolución personalizada.
- **public String generateToken(UserDetails userDetails)**
Genera un nuevo token JWT a partir de los datos del usuario.
- **public String generateToken(Map<String, Object> extraClaims, UserDetails userDetails)**
Genera un nuevo token JWT incluyendo información adicional personalizada (extraClaims).
- **public long getExpirationTime()**
Retorna el tiempo de expiración predeterminado del token en milisegundos.
- **private String buildToken(...)**
Método interno encargado de construir el token JWT completo.
- **public boolean isTokenValid(String token, UserDetails userDetails)**
Verifica si el token proporcionado es válido y pertenece al usuario especificado.
- **private boolean isTokenExpired(String token)**
Comprueba si el token ha caducado.
- **private Date extractExpiration(String token)**
Devuelve la fecha exacta de expiración del token.
- **private Claims extractAllClaims(String token)**
Extrae todos los claims del token JWT, es decir, la información contenida en él.
- **private Key getSignInKey()**
Recupera la clave utilizada para firmar y verificar el token JWT.



Funciones específicas: Accesos y modificaciones específicas

Además de las funciones CRUD y las operaciones relacionadas con la autenticación, el sistema cuenta con una serie de funciones específicas destinadas a realizar operaciones concretas sobre entidades como jugadores, amigos y administradores. Estas funciones aportan flexibilidad y permiten cubrir casos de uso particulares de la lógica del sistema.

Funciones implementadas

- **public Jugadores modificarRanking(Integer id, Integer modificacion)**

Permite sumar o restar puntos de ranking y monedas al jugador indicado por su identificador. Esta función se utiliza, por ejemplo, tras el término de una partida para actualizar el progreso del jugador.

- **public List<Amigos> findAllByUsuario(String usuario)**

Devuelve una lista con todos los amigos del jugador identificado por su nombre de usuario. Dado que el campo usuario es único (unique), esta consulta puede realizarse sin necesidad de exponer el ID interno del jugador, facilitando la integración con otras partes del sistema.

- **public Boolean signup(JugadorRegistroDTO input)**

Gestiona el registro de nuevos jugadores en la aplicación.

Recibe como parámetro un objeto JugadorRegistroDTO, que contiene la información necesaria para crear un nuevo usuario (nombre, correo, contraseña, etc.).

Retorna un valor booleano que indica si el registro fue exitoso.

- **public Jugadores authenticate(JugadorLogDTO input)**

Realiza la autenticación de jugadores.

A partir del DTO JugadorLogDTO (que contiene el usuario y contraseña), se comprueba si las credenciales son válidas y se devuelve el objeto Jugador correspondiente en caso afirmativo.

- **public Jugadores authenticateAdmin(AdminLogDTO input)**

Función destinada a la autenticación de administradores.

Opera de forma similar a la función anterior, pero valida las credenciales dentro del contexto del acceso administrativo.



3.2.2.3 DISEÑO

Al igual que para el backend se empleó Spring como pilar central, en el desarrollo del frontend se optó por utilizar Vue 3, un framework progresivo muy enfocado a la construcción de interfaces reactivas y componentes reutilizables. Este se complementó con una serie de librerías como:

- **Vuetify**, un framework de componentes basado en Material Design que, además, incluye compatibilidad con Bootstrap.
- **SweetAlert2**, utilizado para generar ventanas modales y mensajes emergentes con estilos atractivos y personalizables.

Filosofía de trabajo en Vue

Vue 3 se basa en un modelo de desarrollo orientado a **componentes independientes**. Cada componente encapsula tres secciones principales:

- **HTML**: estructura del componente.
- **CSS**: estilos propios (usualmente mediante scoped styles, que limitan el alcance al componente).
- **JavaScript**: lógica y comportamiento del componente.

Esto permite que cada parte de la interfaz se desarrolle, mantenga y escale de forma independiente, favoreciendo la reutilización de código y la separación de responsabilidades.

Vistas y composición

Los componentes se agrupan en lo que Vue denomina vistas. Estas **vistas** representan secciones completas de la aplicación (como el home, la tienda o el perfil) y son las que se muestran dinámicamente en función de la navegación dentro de la aplicación, la cual sigue un enfoque **SPA** (Single Page Application).

Estilos globales y específicos

Aunque cada componente puede tener su propio estilo local, también existen **hojas de estilo** globales que afectan a toda la aplicación. Estas se encuentran en la ruta:

[/src/assets/](#)



Dentro de dicha carpeta se distinguen dos archivos principales:

- **base.css:** contiene estilos generales que afectan a elementos globales como el <body>, la tipografía, los colores principales, o variables CSS utilizadas por todo el proyecto. Por ejemplo, todos los body comparten un fondo animado y las fuentes empleadas a lo largo de la aplicación.

```
color: var(--color-text);
background-image: url("./src/assets/Decoración/Fondo.png");
background-size: 20w;
background-repeat: repeat, no-repeat, repeat;
background-color: #E2E271;
transition:
  color 0.5s,
  background-color 0.5s;
line-height: 1.6;
font-family:
  Inter,
  -apple-system,
  BlinkMacSystemFont,
  'Segoe UI',
  Roboto,
  Oxygen,
  Ubuntu,
  Cantarell,
  'Fira Sans',
  'Droid Sans',
  'Helvetica Neue',
  sans-serif;
font-size: 15px;
text-rendering: optimizeLegibility;
-webkit-font-smoothing: antialiased;
-moz-osx-font-smoothing: grayscale;

animation: moveBackground 600s linear infinite;
}

@keyframes moveBackground {
from {
  background-position: 0 0;
}
to {
  background-position: 18600px 18600px;
}
```

- **main.css:** en este archivo se definen estilos más específicos, destinados a clases concretas, pequeñas etiquetas o personalizaciones que no se engloban en el archivo anterior.

Un ejemplo de esta diferenciación se puede ver en el componente del tablero de juego, que posee su propio fondo y conjunto de reglas visuales, independientes del resto del sitio.

```
.blanca{
  animation: colorBlanco 0.5s linear forwards;
  background-image: url("/public/Fondos/FondoNegro.png");
  background-size: 60%;
  background-repeat: repeat;
}

.negra{
  animation: colorNegro 0.5s linear forwards;
  background-image: url("/public/Fondos/FondoBlanco.png");
  background-size: 60%;
  background-repeat: repeat;
}
```

A la hora de gestionar los contenedores y su posicionamiento la inmensa mayoría ha sido gestionado mediante flex-box



ACCESIBILIDAD

The screenshot shows the WAVE accessibility evaluation tool interface. On the left, there's a summary of findings:

- Errors: 0
- Contrast Errors: 0
- Alerts: 6
- Features: 3
- Structural Elements: 9
- ARIA: 17

A message at the bottom says: "Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility."

The main area displays a news article from 'NOTICIAS' with the heading "agnus Carlsen lanza su videojuego de ajedrez". Below the heading is a detailed description of the game. To the right of the article, there are several cards with icons and text, such as "h3 "SHAH" supera los mil usuarios" and "h3 Nueva i Stockfish en p DeepMind prese inteligencia artifici motores tradicio más "humano" e".

Debido a las características propias de Vue y la forma en que genera el DOM dinámicamente, algunas herramientas automatizadas de análisis de accesibilidad, como WAVE, pueden reportar “respuestas fantasma” o falsos positivos. Por ejemplo, ciertos elementos `<input>` pueden aparecer sin etiquetas `<label>` asociadas correctamente o con etiquetas duplicadas, lo cual no siempre refleja un problema real sino una limitación de estas herramientas frente a frameworks reactivos.

A pesar de estas limitaciones, se han realizado esfuerzos conscientes para garantizar una buena accesibilidad en la aplicación, destacando:

La medición y corrección de **errores de contraste** para asegurar que los textos y elementos gráficos cumplen con los estándares recomendados.

La inclusión de atributos `alt` en todas las imágenes para proporcionar descripciones adecuadas y mejorar la experiencia para usuarios con lectores de pantalla.

La revisión manual de las principales interacciones para minimizar barreras de accesibilidad dentro de lo posible.

Aunque existen áreas con limitaciones técnicas inherentes al framework y al nivel de experiencia, se ha puesto especial atención en cubrir los aspectos fundamentales de accesibilidad para ofrecer una experiencia usable a la mayoría de usuarios.

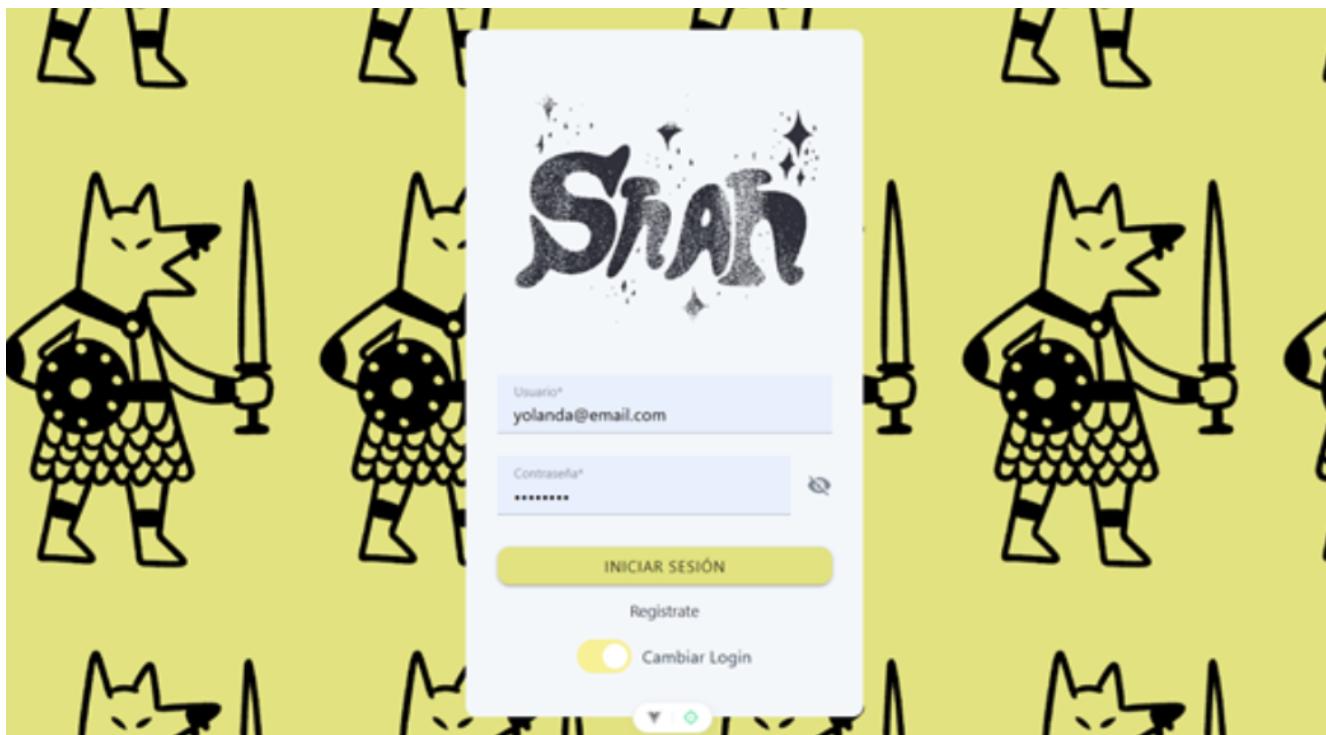
La página cumple los requisitos de usabilidad.

Desarrollo de Aplicaciones Web, 2025, Jorge Carretero



RESULTADO FINAL

Login



Noticias

NOTICIAS

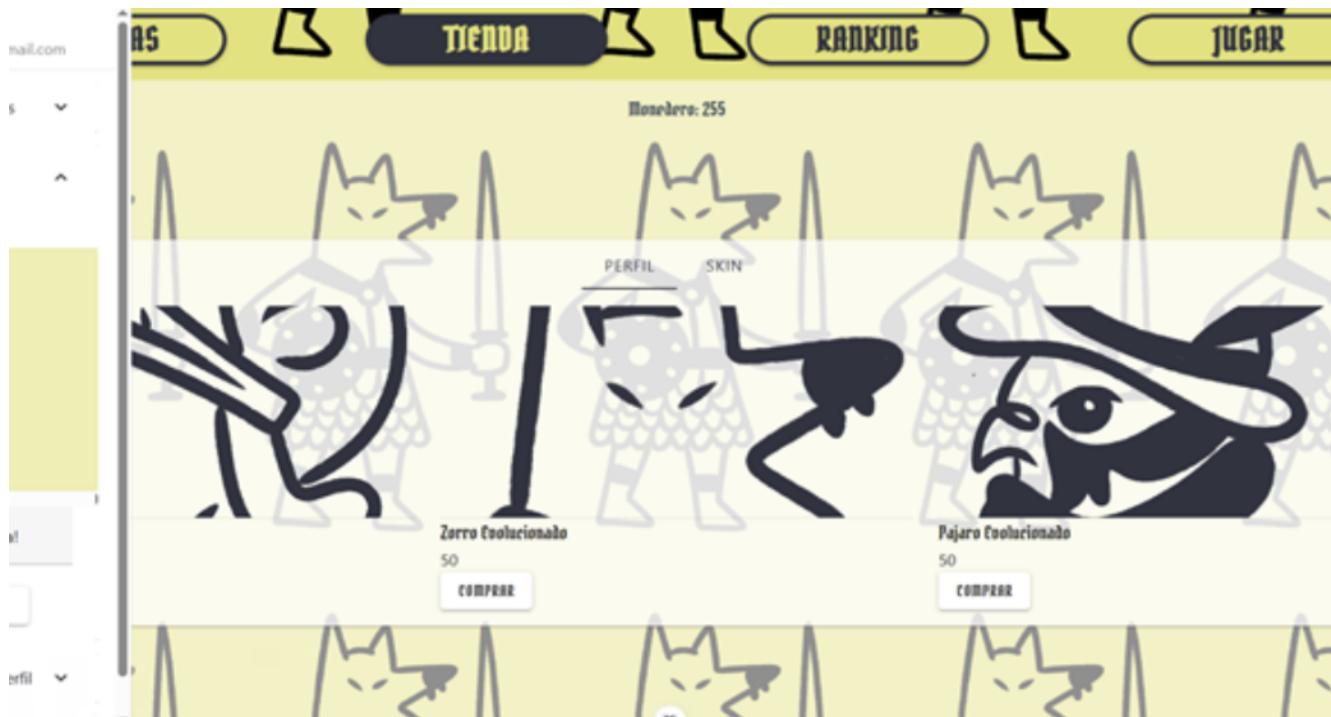
- Carlsen lanza su propio videojuego**
El ajedrecista mundial Magnus Carlsen presentó "Play Magnus", un videojuego que permite a los usuarios enfrentarse a la IA del campeón mundial en partidas de ajedrez.
- TIENDA**
- RANKING**
- JUGAR**

"SHAH" supera los mil usuarios
La plataforma de juego alcanzó este hito gracias al boom durante la pandemia y su integración con streamers y creadores de contenido. Las competiciones están en boca de todos!

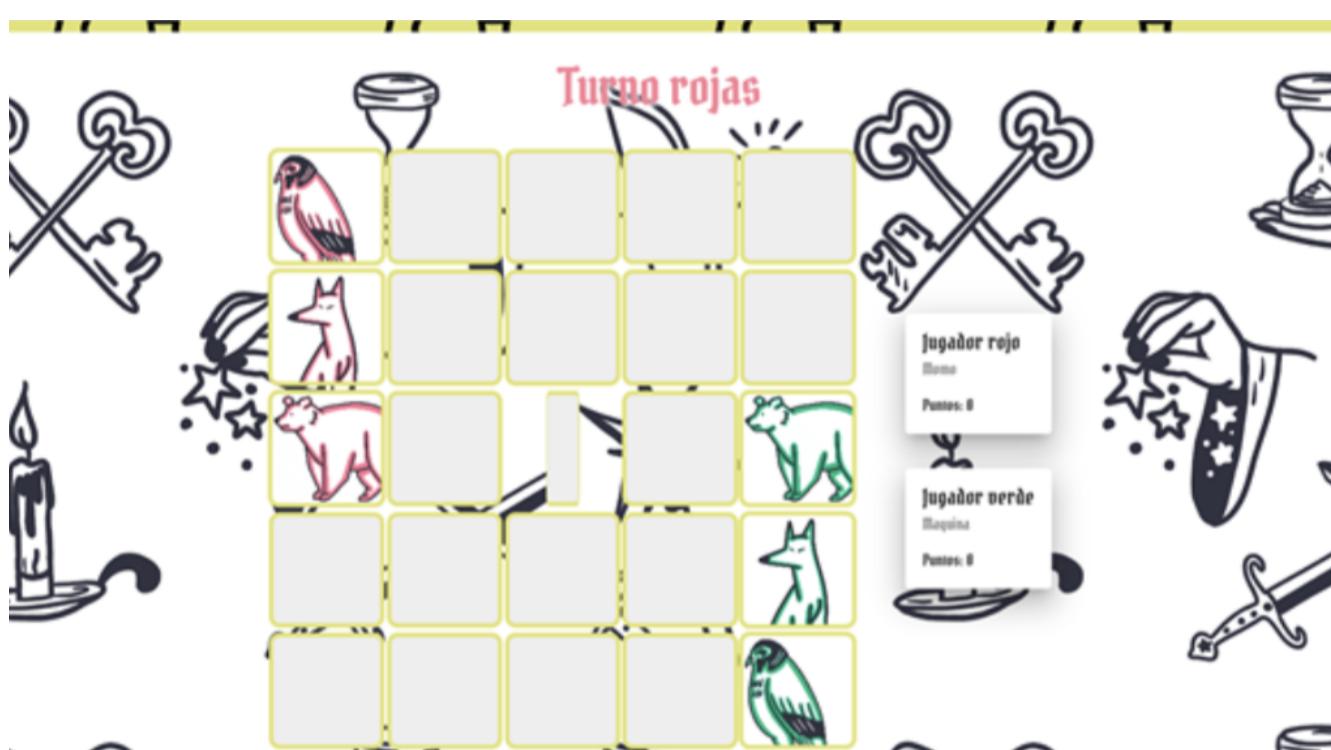
Nueva IA de Google vence a Stockfish en partidas rápidas
DeepMind presentó una nueva versión de su inteligencia artificial, que supera a los motores tradicionales usando un enfoque más "humano" en lugar de fuerza bruta.



Chat en la tienda



Partida en curso





Desarrollo en el entorno cliente

- **Formularios y su validación**

Todos los formularios implementados en la aplicación cuentan con validaciones para garantizar la integridad y corrección de los datos introducidos antes de ser enviados al servidor. Se impide el envío de formularios incompletos o con información inválida en secciones como el login, registro, creación de usuarios (por parte de administradores) o edición de perfiles. Además, se aplican reglas específicas, como la longitud mínima para el nombre de usuario y la validación de estructura en los correos electrónicos introducidos.

- **Gestión de eventos**

La aplicación hace un uso intensivo de la gestión de eventos, aprovechando la sintaxis propia de Vue. Entre los más destacados se encuentran los eventos `@blur` y `@input` en la barra de búsqueda, que permiten filtrar dinámicamente los resultados según el texto introducido. También se utilizan numerosos eventos `@click` en el sistema de juego, por ejemplo, para calcular y mostrar los movimientos posibles al seleccionar una ficha. En Vue, estos eventos se declaran con `@nombreDelEvento` y se les asocia una función manejadora como parámetro.

- **Gestión y almacenamiento de la información del cliente**

La autenticación se gestiona mediante JSON Web Tokens (JWT), cuya validez es esencial para mantener la sesión activa y permitir las peticiones al servidor. Al iniciar sesión, se almacena información del usuario en una store gestionada con Pinia, una librería similar a localStorage, pero reactiva y más integrada con Vue. Esta store centraliza el estado del usuario y permite acceder a sus datos en distintas partes de la aplicación.

- **Modificación del DOM**

El DOM se actualiza de forma dinámica a lo largo de toda la experiencia de usuario. Ejemplos de esto son:

- La lista de amigos se actualiza automáticamente cuando se añade un nuevo contacto.
- En el tablero de juego, las fichas se mueven en tiempo real y los cambios visuales reflejan los turnos o las evoluciones de piezas.
- En el perfil, la imagen de usuario se actualiza inmediatamente al seleccionar una nueva.



• Animaciones

Se ha prestado especial atención a la experiencia visual del usuario mediante animaciones y efectos. En el contexto del juego, se incluyen:

- Cambios de color y estilo entre turnos.
- Efectos visuales al mover piezas o al evolucionarlas.
- Animaciones suaves en transiciones y menús para dinamizar la interfaz.

• Comunicación AJAX

La aplicación realiza numerosas peticiones AJAX (GET, POST, PUT) desde las stores de Pinia, donde reside la lógica asociada a cada módulo. Estas peticiones permiten, por ejemplo:

- Cargar los artículos disponibles en la tienda
- Obtener las noticias mostradas en la sección principal.
- Solicitar los datos de los jugadores tanto en la vista de administración como en la del ranking.

• Comunicación asíncrona con el servidor

El modelo de comunicación del frontend con el backend se basa en llamadas asíncronas, que permiten al cliente obtener y actualizar datos en segundo plano sin recargar la página. Este enfoque es fundamental para mantener la fluidez de la aplicación y soportar operaciones en tiempo real como:

- La actualización de estados en partidas en curso.
- El envío y recepción de mensajes en el chat.
- La sincronización del estado de usuario y su información en tiempo real.



3.2.3 FASE DE DESPLIEGUE DESPLIEGUE EN HOSTING

La aplicación ha pasado por dos fases principales de despliegue: una primera centrada en el backend (API) a través de Railway, y una segunda fase enfocada al frontend, el cual fue desplegado utilizando Vercel.

Despliegue del backend (Railway)

Desde fases tempranas del desarrollo se optó por trabajar en un entorno de producción real mediante Railway. Esta decisión se tomó de forma premeditada, anticipando que el despliegue de una API desarrollada con Java Spring y con soporte para sockets iba a implicar una configuración compleja, especialmente dadas las limitaciones de tiempo y los requisitos técnicos de la aplicación.

El backend, desarrollado en Java con Spring Boot, presentaba ciertas restricciones en cuanto a hosting gratuito debido al uso de WebSockets y a la necesidad de configuración avanzada. Inicialmente, se utilizó la versión gratuita de Railway, la cual fue suficiente hasta etapas avanzadas del desarrollo. En ese punto, las limitaciones impuestas por la plataforma (tiempo de ejecución, memoria, etc.) comenzaron a ser incompatibles con las necesidades del proyecto.

El proceso de despliegue en Railway no fue trivial. Se enfrentaron diversos problemas tanto en la fase de compilación (build) como en la configuración de las variables de entorno. Este proceso consumió varios días, debido también a la necesidad de implementar múltiples configuraciones de seguridad.

Uno de los mayores avances fue la dockerización del proyecto. A través de un Dockerfile, se consiguió encapsular todo el entorno de ejecución en un contenedor, facilitando así la compilación y ejecución de la aplicación dentro de Railway. Las variables de entorno fueron configuradas en el propio entorno de Railway, integrándose profundamente con la configuración interna de Spring Boot.

The screenshot shows a list of five build logs from the Railway CI/CD pipeline. Each log entry has a red 'FAILED' status box, a GitHub icon, and a timestamp indicating it was triggered via GitHub. The logs are:

- Cambio en el docker y acentos - 24 hours ago via GitHub
- Cambio en el docker - 24 hours ago via GitHub
- Cambio de espacios - 24 hours ago via GitHub
- Cambio de espacios - 46 hours ago via GitHub
- Subiendo backend



Una vez solucionados los problemas iniciales, el flujo de trabajo se optimizó: cualquier modificación del código fuente en GitHub puede ahora desplegarse automáticamente, siempre que se sigan las instrucciones y convenciones establecidas en el [Dockerfile](#).

The screenshot shows the Railway app deployment interface. On the left, there's a sidebar with icons for creating new services and managing existing ones. Two services are visible: 'ShahMySQL' and 'SHAHBackend'. 'ShahMySQL' was deployed 2 weeks ago via Docker Image. 'SHAHBackend' was deployed 83 minutes ago via GitHub. A dashed line connects the two services, indicating they are connected via a database volume named 'scene-volume'. On the right, the 'SHAHBackend' service details page is shown. It shows the deployment status as 'ACTIVE' with a green icon, last updated 83 minutes ago via GitHub. Below it, a green bar indicates a 'Deployment successful'. Under the 'HISTORY' section, two previous deployments are listed under the 'REMOVED' status, both also triggered by GitHub. The interface includes tabs for Deployments, Variables, Metrics, and Settings, and shows EU West as the region with 1 replica.

Despliegue del frontend (Vercel)

El despliegue del frontend fue considerablemente más sencillo. Utilizando Vue 3 como framework y Vercel como plataforma de hosting, bastó con iniciar sesión en Vercel y conectar el repositorio de GitHub del proyecto. Tras realizar algunos ajustes menores en las configuraciones de seguridad (como permitir las peticiones al dominio del backend), la aplicación fue desplegada exitosamente.

Vercel ofreció una solución ágil, estable y con tiempos de despliegue muy rápidos, ideal para un frontend de tipo SPA (Single Page Application).

The screenshot shows the Vercel production deployment interface for a project named 'snan'. At the top, there are tabs for Repository, Usage, Domains, and Visit. The 'Visit' tab is selected. Below that, the 'Production Deployment' section is shown. It features a cartoon illustration of a group of characters. The deployment status is 'Created' and 'Ready' (green), updated 10m ago by JCasuario. The domains listed are 'shah-rose.vercel.app' and 'shah-jcasuarios-projects.vercel.app'. The source code branch is 'main' with commit hash '8157621' and the message 'Corrección imágenes'. There are buttons for Deployment Configuration, Fluid Compute, Deployment Protection, Skew Protection, Instant Rollback, Build Logs, Runtime Logs, and Deployments. A note at the bottom says 'To update your Production Deployment, push to the main branch.'



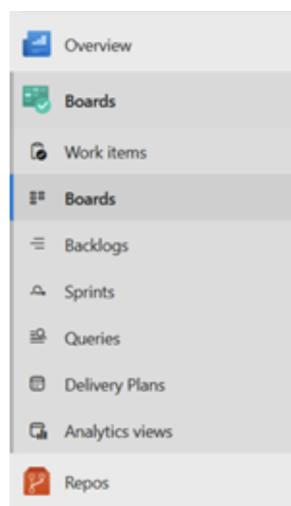
3.3 SEGUIMIENTO DEL CONTROL DE INCIDENCIAS

El control de incidencias durante el desarrollo del proyecto se gestionó principalmente mediante una tabla de Excel personalizada. En esta tabla se registraban todas las incidencias detectadas, incluyendo información clave como:

- **Descripción del problema**
- **Fecha de aparición**
- **Origen estimado del error**
- **Posibles soluciones planteadas**
- **Solución definitiva (en caso de haber sido resuelta)**

Esta metodología nos permitió tener un seguimiento detallado y organizado de cada error o anomalía que surgió durante las distintas fases del desarrollo, facilitando la identificación de patrones repetitivos y la mejora continua del código y la estructura del proyecto.

Además, se llevó a cabo un pequeño experimento de uso de herramientas profesionales de gestión, concretamente Azure DevOps y Azure Boards. Esta exploración se realizó aprovechando el acceso proporcionado a través del entorno de prácticas. Sin embargo, su uso fue puntual y limitado, ya que se optó por mantener el flujo principal de control de errores en el sistema ya establecido con Excel, más ágil y adaptado al tamaño del equipo y a la dinámica del proyecto.





3.4 INDICADORES DE CALIDAD

Más allá del uso constante de nuestro log de desarrollo y de la realización de múltiples pruebas manuales, el control de calidad ha supuesto uno de los mayores retos del proyecto. A pesar de ello, se ha logrado alcanzar un funcionamiento estable y fiable en todos los módulos principales de la aplicación.

El seguimiento de la calidad se ha sustentado en los siguientes elementos:

- **Pruebas funcionales manuales:** realizadas tras cada nueva implementación o modificación, permitiendo detectar errores de comportamiento y corregirlos de forma inmediata.
- **Registro de errores y comportamiento** mediante logs, tanto en el cliente como en el servidor, lo que ha facilitado la identificación de fallos durante la ejecución.
- **Control de tokens y validaciones:** en todo momento se ha vigilado la validez de los tokens JWT, asegurando un control seguro y correcto de las sesiones.
- **Validación de formularios en el cliente,** garantizando la integridad y estructura adecuada de los datos enviados al servidor.

Aunque no se han implementado pruebas automatizadas (unitarias o de integración), el resultado global tras las diferentes fases de desarrollo y corrección ha sido una aplicación funcional, robusta y sin errores críticos conocidos al cierre del proyecto.

The screenshot shows the Railway platform interface for the ShahBackend deployment. The top bar displays the project name 'SHAHBackend / 5040be2' and the date 'May 29, 2025 7:02 PM'. The main area is titled 'Deploy Logs' and shows a list of log entries from May 29, 2025, at 20:36:08. The logs include messages such as DEBUG 2 levels, DispatcherServlet processing requests, RequestResponseBodyMethodProcessor handling JSON, and Hibernate select queries for retrieving player data. The logs are filtered using the pattern '*-*'. The interface includes tabs for 'Details', 'Build Logs', 'Deploy Logs', and 'HTTP Logs', along with a search bar and filter options.



4 RECURSOS MATERIALES INVENTARIO VALORADO DE MEDIOS

BIEN DE INVERSIÓN	FOTO	PRECIO/UND	Nº UNDS	SUBTOTAL
PcCom Ready		969€	2	1.938€
GitHub Copilot		228€	2	456€
IMPORTE TOTAL				2394€

Es un presupuesto inicial muy positivo, en el futuro sería necesario apuntar los múltiples requisitos y gastos que en el futuro van a venir del despliegue tanto de la aplicación del front y el back.



5 RECURSOS HUMANOS

Se determina el personal necesario para llevar a cabo la actividad.

ESTRUCTURA ORGANIZATIVA DE LA EMPRESA

Aquí delimitamos los distintos departamentos o secciones que compondrán nuestro negocio.

DEPARTAMENTO / ÁREA FUNCIONAL	FUNCIÓN/TAREA
Recursos Humanos	Gestionar todo el apartado relacionado con el apartado de trabajadores y su relación entre ellos y la empresa. Selección, contratación (junto administración y finanzas) y demás tareas.
Administración y Finanzas	Se encargará de todo aquello relacionado con el control monetario y la búsqueda de financiación. También se aseguran de los presupuestos y los pagos.
Producción y mantenimiento	Departamento central donde nuestro proyecto será primero desarrollado y por último controlado y parcheado. Aquí estará la columna central de trabajo de la empresa, apoyado, controlado y potenciado por el resto de departamentos.
Ventas y atención al cliente	Harán llegar el producto a nuevos clientes, también trabajarán las redes sociales y se encargarán de atender las distintas quejas y problemas que puedan tener nuestros jugadores con el servicio. Área de gran importancia.

Organigrama:





DISEÑO DE LOS PUESTOS DE TRABAJO

Aquí trataremos de determinar los puestos de trabajo que necesitamos para desarrollar el proyecto. Definimos las tareas a realizar, aquí mostraremos dos profesiogramas el del propio autónomo como jefe y las tareas que deberá enfrentar y la de un segundo que le ayude con las tareas de producción y mantenimiento.

PROFESIOGRAMA	
Puesto Trabajo	DESCRIPCIÓN DEL PUESTO
Puesto Trabajo	Jefe de áreas y servicios
Categoría Profesional	<p>Empleados y empleadas: es el personal que por sus conocimientos y/o experiencia realiza tareas administrativas, comerciales, organizativas, de informática, de laboratorio y, en general, las específicas de puestos de oficina, que permiten informar de la gestión, de la actividad económico-contable, coordinar labores productivas o realizar tareas auxiliares que comporten atención a las personas.</p>
Tareas del puesto de trabajo	<ol style="list-style-type: none"> 1. Tareas técnicas que consisten en el ejercicio del mando directo al frente de un conjunto de operarios de oficio o de procesos productivos en instalaciones principales (siderurgia, electrónica, automación, instrumentación, montaje o soldadura, albañilería, carpintería, electricidad, etc.). 2. Tareas técnicas de codificación de programas de ordenador en el lenguaje apropiado, verificando su correcta ejecución y documentados adecuadamente. 3. Tareas técnicas que consisten en la ordenación de tareas y de puestos de trabajo de una unidad completa de producción. 4. Actividades que impliquen la responsabilidad de un turno o de una unidad de producción que puedan ser secundadas por una o varias personas trabajadoras del Grupo profesional inferior. 5. Tareas técnicas de inspección, supervisión o gestión de la red de ventas. 6. Tareas técnicas de dirección y supervisión en el área de contabilidad, consistentes en reunir los elementos suministrados por los ayudantes, confeccionar estados, balances, costos, provisionales de tesorería y otros trabajos análogos en base al plan contable de la empresa. 7. Tareas técnicas consistentes en contribuir al desarrollo de un proyecto que redacta un técnico (ingeniero, aparejador, etc.) aplicando la normalización, realizando el cálculo de detalle, confeccionando planos a partir de datos facilitados por un mando superior. 8. Tareas técnicas administrativas de organización o de laboratorio de ejecución práctica, que suponen la supervisión según normas recibidas de un mando superior. 9. Tareas técnicas administrativas o de organización de gestión de compra de aprovisionamiento y bienes convencionales de pequeña complejidad o de aprovisionamiento de bienes complejos. 10. Tareas técnicas de gestión comercial con responsabilidad sobre un sector geográfico delimitado y/o una gama específica de productos.



PERFIL PROFESIONAL	
Titulación	Técnico superior en desarrollo de aplicaciones web
Conocimientos	<p>Lenguajes de marcas y sistemas de información Sistemas informáticos Bases de datos Programación Entornos de desarrollo Desarrollo web en entorno cliente Desarrollo web en entorno servidor Despliegue de aplicaciones web Diseño de interfaces web Inglés profesional para ciclos formativos de Grado Superior Digitalización aplicada al sector productivo Sostenibilidad aplicada al sistema productivo. Itinerario personal para la empleabilidad I Itinerario personal para la empleabilidad II Optatividad Proyecto intermodular de desarrollo de aplicaciones Web</p>
Actitudes Personales	<p>Capacidad de observación, de razonamiento y reflexión. Capacidad de organización, rigurosidad y disciplina. Capacidad numérica. Capacidad para el trabajo en equipo, cooperación. Facilidad en el aprendizaje de idiomas. Capacidad para el autoaprendizaje y uso de las nuevas tecnologías.</p>
Experiencia	1 año de experiencia
CONDICIONES LABORALES	
Grupo Profesional	3
Categoría Profesional	Empleado Jefe de Área y Servicio
Grupo Salarial	3A
Salario Base	1093,51
Complemento Salarial (Calidad y cant)	120,02
Contrato	Autónomo con alta inicial en el RETA
Tipo Jornada	Completa

AYUDANTE

PROFESIOGRAMA	
Puesto Trabajo	Profesional Oficial de primera
DESCRIPCIÓN DEL PUESTO	
Categoría Profesional	Personal de operaciones: es el personal que por sus conocimientos y/o experiencia ejecuta operaciones relacionadas con la producción, bien directamente, actuando en el proceso productivo, o en labores de mantenimiento, transporte u otras operaciones auxiliares, pudiendo realizar, a su vez, funciones de supervisión o coordinación.
Tareas del puesto de trabajo	<p>1. Tareas administrativas desarrolladas con utilización de aplicaciones informáticas.</p> <p>2. Tareas de electrónica, siderurgia, automación, instrumentación, montaje o soldadura, albañilería, carpintería, electricidad, pintura, mecánica, etc., con capacitación suficiente para resolver todos los requisitos de su oficio o responsabilidad.</p> <p>3. Tareas de control y regulación de los procesos de producción que generan transformación de producto.</p> <p>4. Tareas de cierta complejidad de preparación de operaciones en máquinas convencionales que conlleve el autocontrol del producto elaborado.</p> <p>5. Tareas de archivo, registro, cálculo, facturación o similares que requieran algún grado de iniciativa.</p> <p>6. Tareas de despacho de pedidos, revisión de mercancías y distribución con registro en libros o mecánicas, al efecto de movimiento diario.</p> <p>7. Tareas de lectura, anotación y control, bajo instrucciones detalladas, de los procesos industriales o el suministro de Servicios generales de fabricación.</p> <p>8. Tareas de mecanografía, con buena presentación de trabajo y ortografía correcta y velocidad adecuada que pueden llevar implícita la redacción de correspondencia según formato e instrucciones específicas, pudiendo utilizar paquetes informáticos como procesadores de textos o similares.</p> <p>9. Tareas elementales de delineación de dibujo, calcos o litografías que otros han preparado, así como cálculos sencillos.</p> <p>10. Realizar agrupaciones de datos, resúmenes, estadísticas, cuadros, seguimientos, histogramas, certificaciones, etc., con datos suministrados por otros que los toman directamente en base a normas generalmente precisas.</p>



PERFIL PROFESIONAL	
Titulación	Técnico superior en desarrollo de aplicaciones web
Conocimientos	Lenguajes de marcas y sistemas de información Sistemas informáticos Bases de datos Programación Entornos de desarrollo Desarrollo web en entorno cliente Desarrollo web en entorno servidor Despliegue de aplicaciones web Diseño de interfaces web Inglés profesional para ciclos formativos de Grado Superior Digitalización aplicada al sector productivo Sostenibilidad aplicada al sistema productivo. Itinerario personal para la empleabilidad I Itinerario personal para la empleabilidad II Optatividad Proyecto intermodular de desarrollo de aplicaciones Web
Actitudes Personales	Capacidad de observación, de razonamiento y reflexión. Capacidad de organización, rigurosidad y disciplina. Capacidad numérica. Capacidad para el trabajo en equipo, cooperación. Facilidad en el aprendizaje de idiomas. Capacidad para el autoaprendizaje y uso de las nuevas tecnologías.
Experiencia	No necesaria

CONDICIONES LABORALES	
Grupo Profesional	5
Categoría Profesional	Profesional Oficial de Primera
Grupo Salarial	5D
Salario Base	979,19
Complemento Salarial (Calidad y cant)	114,52
Contrato	Temporal Contrato para Parados de larga duración
Tipo Jornada	Completa

A destacar que se ha optado por un contrato de parado de larga duración, para aportar socialmente y ayudar a una persona a finalizar el sufrido periodo de búsqueda laboral. Por otro lado, nosotros tendremos una serie de bonificaciones y reducciones a la seguridad social.

COSTE RECURSOS HUMANOS:

Calcularemos una aproximación de cuánto nos costará el salario y seguridad social del trabajador y el propio autónomo que forman el equipo que desarrollará la idea de negocio.

TRABAJADOR DE LA EMPRESA	PUESTO DE TRABAJO	TIPO DE CONTRATO	Nº pax	SALARIO MENSUAL (A x B)	SEGURIDAD SOCIAL MENSUAL (A x C)	SALARIO ANUAL (B x 12)	SEGURIDAD SOCIAL ANUAL (C x 12)	BONIFICACIÓN Y REDUCCIONES A LA SEGURIDAD SOCIAL 12 meses	COSTE TOTAL CON BONIFICACIÓN Autónomo (D +F) Contratados (D + E) - F
PROMOTOR/AUTÓNOMO	Jefe de Área y Servicios	Gerente	1	1.093,51€	273,11€	13122,12	3277,32	960€	14082,12
PERSONAL CONTRATADO	Profesional de Oficio de primero	Contratos parados larga duración	1	1.367,14€	471,39€	16.405,68€	5.656,58€	1.536€	20.526,32€
		TOTAL		2.460,65€	744,50€	29527,8	8933,9	2.496€	34608,44

Modalidad de organización preventiva elegida

En nuestro caso, hemos optado por que el empresario asuma personalmente la actividad preventiva, salvo la vigilancia de la salud, que será gestionada externamente. Esta decisión se ha tomado porque cumplimos con los requisitos legales: somos una empresa de menos de seis trabajadores, no desarrollamos actividades consideradas especialmente peligrosas según el Anexo I del Reglamento de los Servicios de Prevención, y el empresario desarrolla su actividad de forma habitual en el centro de trabajo. Además, cuenta con la formación necesaria para desempeñar estas funciones.



6. VIABILIDAD TÉCNICA

En el desarrollo del proyecto se ha apostado por un stack tecnológico sólido, actual y ampliamente utilizado en entornos profesionales. Todas las herramientas empleadas han sido de uso gratuito, lo cual no ha sido un impedimento para lograr un desarrollo avanzado, eficiente y con garantías de calidad técnica.

Entorno de desarrollo del back-end

Para el desarrollo del lado servidor se ha empleado Java Spring, una de las tecnologías más robustas y versátiles del desarrollo web actual. Se ha trabajado en IntelliJ IDEA Community Edition, un entorno de desarrollo muy completo que ha permitido una gestión cómoda del proyecto y una excelente integración con herramientas clave como Git o Docker. Además, el uso de extensiones específicas ha facilitado tanto la depuración como la gestión de dependencias.

Dentro del ecosistema Spring, se han integrado varios módulos fundamentales:

- **Spring Boot**, como base para la creación de servicios REST de forma estructurada y rápida.
- **Spring JPA**, para facilitar la conexión con la base de datos y trabajar con entidades a través de una capa de abstracción basada en objetos.
- **Spring Security**, para el control de accesos, el login con tokens JWT y el cifrado de contraseñas.
- **Spring Sockets**, para permitir la comunicación en tiempo real entre usuarios (imprescindible para funcionalidades como el chat o las partidas en línea).



Entorno de desarrollo del front-end

El desarrollo del cliente se ha llevado a cabo utilizando Vue 3, un framework moderno que facilita la construcción de interfaces dinámicas a través del uso de componentes. Se ha trabajado con Visual Studio Code como editor principal, acompañado de extensiones como Docker, Vue Devtools y Vue Snippets, que han contribuido a agilizar notablemente el proceso de codificación.

En cuanto a librerías y herramientas complementarias:

- **Vuetify** ha sido la librería de componentes empleada para dotar a la interfaz de un diseño profesional, responsive y accesible.
- **Pinia** se ha utilizado como sistema de gestión de estado, lo que ha permitido almacenar datos persistentes del usuario (como su sesión o información de perfil) y compartirlos entre componentes de forma eficiente.
- **Vue Router** ha gestionado la navegación dentro de la aplicación SPA (Single Page Application), controlando el acceso a determinadas vistas dependiendo del estado de autenticación del usuario.

Control de versiones y despliegue

Para el control de versiones y el trabajo colaborativo, se ha utilizado **Git** junto con **GitHub** como repositorio central. Esto ha permitido mantener un flujo de trabajo ordenado, con seguimiento de cambios y posibilidad de revertir versiones en caso de errores.

En cuanto al despliegue:

- El **back-end** y la base de datos han sido desplegados en **Railway**, plataforma que permite alojar proyectos Java y gestionar servicios como bases de datos MySQL. Debido a las necesidades específicas del proyecto (uso de sockets, configuración de seguridad, variables de entorno), se optó por una solución basada en **Docker**, lo que permitió encapsular toda la lógica del servidor en un contenedor que se puede ejecutar fácilmente en Railway.
- El **front-end** ha sido desplegado en **Vercel**, aprovechando su integración directa con GitHub para desplegar automáticamente cada vez que se actualiza el repositorio.



Conclusión

La viabilidad técnica del proyecto queda plenamente demostrada. No solo se ha hecho uso de tecnologías modernas y bien soportadas, sino que también se ha trabajado con una arquitectura modular y escalable que permite el crecimiento futuro del proyecto. La elección de herramientas ha respondido tanto a criterios de funcionalidad como de disponibilidad, permitiendo un desarrollo sostenible y técnicamente robusto.



7. VIABILIDAD ECONÓMICO FINANCIERA

Aquí se detallarán las distintas inversiones necesarias, la financiación para poder llevarlas a cabo y la viabilidad que el plan de tesorería y demás estudios reflejarán sobre la idea de negocio.

7.1 Inversiones y gastos

COSTE RECURSOS HUMANOS:

Calcularemos una aproximación de cuánto nos costará el salario y seguridad social del trabajador y el propio autónomo que forman el equipo que desarrollará la idea de negocio.

TRABAJADOR DE LA EMPRESA	PUESTO DE TRABAJO	TIPO DE CONTRATO	Nº pax	SALARIO MENSUAL (A x B)	SEGURIDAD SOCIAL MENSUAL (A x C)	SALARIO ANUAL (B x 12)	SEGURIDAD SOCIAL ANUAL (C x 12)	BONIFICACIÓN Y REDUCCIONES A LA SEGURIDAD SOCIAL 12 meses	COSTE TOTAL CON BONIFICACIÓN Autónomo (D +F) Contratados (D + E) - F
PROMOTOR/AUTÓNOMO	Jefe de Área y Servicios	Gerente	1	1.093,51€	273,11€	13122,12	3277,32	960€	14082,12
PERSONAL CONTRATADO	Profesional de Oficio de primero	Contratos parados larga duración	1	1.367,14€	471,39€	16.405,68€	5.656,58€	1.536€	20.526,32€
		TOTAL		2.460,65€	744,50€	29527,8	8933,9	2.496€	34608,44

COSTE RECURSOS MATERIALES:

Con el objetivo de conocer los costes de inversión que enfrentaremos al inicio de la actividad se tratará de hacer un inventario de inversiones que realizaremos en las infraestructuras necesarias.



PLAN DE INVERSIONES

BIENES DE INVERSIÓN	COSTE		COMPAÑIA	FUENTE DE FINANCIACIÓN
	ALQUILER	€/MES	€/AÑO	
Edificio/Local	420€		5.040€	Leasing
Reformas/ Acondicionamiento del local				
Maquinaria/Equipos				
Herramientas/ Utillaje				
Mobiliario				
Equipos Informáticos			2.394€	Ahorros Propios
Vehículos				
TOTAL DE INVERSIÓN	420€	5.040€	2394€	7.434€

PLAN DE GASTOS

GASTOS INICIALES	COSTE	
	€ / mes	€ / año
Constitución de la empresa		800€
Capital social mínimo exigido legalmente		3000€
Reservas (dinero primeros meses)		
Seguros, (local, vehículos, etc.)	13.52€	162.23€
Publicidad	250€	3000€
Servicios externos, (gestoría, etc.)	60€	720€
Suministros, (agua, luz, teléfono, etc.)	88€	1056€
TOTAL GASTOS INICIALES	8738.23€	

7.2 FINANCIACIÓN

Desglose de los planes que resumen la financiación y su obtención.

PLAN DE FINANCIACIÓN

Aquí detallaremos cómo financiaremos u obtendremos el dinero necesario para pagar las inversiones y gastos previstos.

TOTAL BIENES DE INVERSIÓN	7434€
TOTAL GASTOS INICIALES	8738,23€
INVERSIÓN NECESARIA	16172,23€

FINANCIACIÓN DE INVERSIONES Y GASTOS INICIALES	
FINANCIACIÓN PROPIA	3500,03 €
Ahorros personales.	1500,03 €
Préstamos familiares.	2000 €
FINANCIACIÓN AJENA	718,2 €
AYUDAS Y SUBVENCIONES	718,2 €
Ayuntamiento de Cuenca.	€
Diputación de Cuenca.	€
Junta de comunidades de Castilla-La Mancha.	718,2 €
✓ Creación.	€
✓ Inversión.	718,2 €
✓ Financiación.	€
Gobierno de España.	€
PRÉSTAMOS	11954 €
Préstamo bancario	11954 €
Líneas ICO	€
FINANCIACIÓN NECESARIA	16172,23€



AMORTIZACIÓN DEL PRÉSTAMO

CONDICIONES DEL PRÉSTAMO	
Entidad financiera	Liberbank
Capital solicitado	11954€
Tipo de interés (i) (Euribor + Diferencial)	1.4%
Plazo de devolución	4 años

PERIODO	Cuota Anual	Interés anual D*i	Capital devuelto A-B	Capital pendiente D-C	Capital Amortizado
0				11.954€	
1	3.093,82€	167,36€	2.926,46€	9.027,58€	2.926,46€
2	3.093,82€	126,39€	2.967,43€	6.060,15€	5.893,89€
3	3.093,82€	84,84€	3.008,98€	3.051,17€	8.902,87€
4	3.093,82€	42,72€	3.051,10€	0	11.953,98€

AMORTIZACION DE LOS BIENES DE INVERSION

Vamos a calcular el importe que amortizamos anualmente por cada uno de los bienes de inversión que vamos a necesitar.

Aquí detallaremos cómo financiaremos u obtendremos el dinero necesario para pagar las inversiones y gastos previstos.

BIENES DE INVERSIÓN <i>(Adquiridos en propiedad, los alquilados NO)</i>	Valor de adquisición (A)	Coeficiente lineal máximo (B)	Cuota de amortización anual (A x B)
Edificio / Local		3%	
Reformas / Acondicionamiento del local		10%	
Maquinaria / Equipos		12%	
Herramientas / Utillaje		30%	
Mobiliario		10%	
Equipos informáticos	2394€	26%	622.44€
Vehículos		16%	
TOTAL	2394€		622.44€



7.3 VIABILIDAD ECONÓMICO FINANCIERA

Esta es una sección de gran importancia, ya que en esta mostraremos las previsiones de los resultados que va a tener la empresa; las ventas esperadas, los gastos en los que se incurrirá y los beneficios o las pérdidas.

Para efectuar esta previsión se debe tener en cuenta:

PREVISION DE TESORERIA

La previsión de tesorería ofrece una visión dinámica de los cobros y pagos esperados, permitiendo anticipar la situación de liquidez de la empresa. Si los ingresos superan los pagos, se contará con liquidez suficiente para reducir deuda o depositar fondos en el banco. En caso contrario, será necesario analizar fuentes alternativas de financiamiento, como la apertura de una línea de crédito.

ENTRADAS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE	DICIEMBRE	TOTAL
Ahorro/Personal	3.500€												3.500€
Prestamo	11.954€												11.954€
Ventas de bienes /servicios	5.033€	1.855€	2.782€	2.782€	4.637€	4.637€	4637,5	4.637€	4.637€	4.637€	4.637€	4.637€	49.551€
Ayudas y subvenciones													718,20€
Interes de c/c e inversiones													
TOTAL ENTRADAS	20.487€	1.855€	2.782€	2.782€	4.637€	4.637€	4.638€	4.637€	4.637€	4.637€	4.637€	5.355,52€	65.721€
SALIDAS													
Devolucion del prestamo e interes	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	257,82€	3.093,82€
Alquileres(local,vehiculo,etc...)	420€	420€	420€	420€	420€	420€	420€	420€	420€	420€	420€	420€	5.040€
Maquinarias y herramientas													0€
Vehiculos(propiedad)													0€
Mobiliario(propiedad)													0€
Equipos Informáticos(propiedad)	2.394€												2.394€
Reformas y acondicionamientos													0€
Gastos de constitución	800€												800€
Seguros	162,23€												162,23€
Publicidad	3.000€												3.000€
Compra de mercaderias													0€
Servicios externos	60€	60€	60€	60€	60€	60€	60€	60€	60€	60€	60€	60€	720€
Personal	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	2.884,04€	34.608,44€
Pagos a cuenta del IRPF o del IS							167€				501€		501€
Suministros(agua, telefono, luz)	88€	88€	88€	88€	88€	88€	88€	88€	88€	88€	88€	88€	1.056€
TOTAL SALIDAS	10.066€	3.710€	3.710€	3.710€	3.710€	3.710€	3.877€	3.710€	3.710€	4.211€	3.710€	4.211€	51.243€
RESULTADO(Entradas-Salidas)	10.421€	-1.855€	-927€	-927€	927€	927€	761€	927€	927€	427€	927€	1.144,83€	
SALDO EN CAJA	10.421€	8.566€	7.639€	6.711€	7.639€	8.566€	9.327€	10.254€	11.182€	11.609€	12.536€	13.681€	

Evaluando la previsión de tesorería, se observa que el préstamo inicial permite mantener saldo positivo en caja durante los primeros meses, a pesar de los resultados negativos iniciales. En los meses siguientes, estas pérdidas se compensan gracias al crecimiento de las ventas, lo que permite cerrar el año con un saldo superior al de enero. Este análisis se complementará con un estudio más detallado en la cuenta de resultados.

CUENTA DE RESULTADOS

En la cuenta de resultados se reflejan los datos del ejercicio económico, evaluando si la empresa ha obtenido beneficios o pérdidas al cierre del año. Esto se realiza comparando los ingresos y gastos previstos durante el período.

INGRESOS EXPLOTACIÓN	49.551€	GASTOS EXPLOTACIÓN	47.177,53€
Ventas	49.551€	Amortización de inversiones	622,44€
		Alquileres (local, vehículo...)	5.040€
		Gastos de constitución	800€
		Seguros (local, vehículo...)	162,23€
		Publicidad	3.000€
		Compra de mercadería	0€
		Servicios Externos	720€
		Personal(salario, seg. social...)	34.608,44€
		Pagos a cuenta del IRPF o del IS	1.168€
		Suministros (agua, teléfono..)	1.056€
INGRESOS FINANCIEROS		GASTOS FINANCIEROS	167,35€
Interés de c/c e inversiones	0€	Intereses de préstamos	167,35

Resultado de explotación	2.374€
Resultado Financiero	-167,35€
Resultado antes de impuestos	2.207€
Impuestos sobre sociedades	331€
Pagos a cuenta del IRPF	1.168,60€
Liquidación del I.S	-838€
RESULTADO FINAL	3.044€

Como resultado final, se obtiene un beneficio de 3.044 €, tras incorporar la devolución de Hacienda a los resultados de explotación y financieros. Aunque el beneficio es modesto, se espera un crecimiento en los años siguientes gracias al aumento de los ingresos de explotación. Un resultado nada pesimista para un negocio que está dando sus primeros pasos.



BALANCE PATRIMONIAL

ACTIVO (Bienes y derechos en propiedad)		PASIVO (Deudas a devolver)	
		PATRIMONIO NETO	6.762,20€
		Capital social mínimo exigido	3.000€
		Cuenta de resultados	3.044€
		Ayudas y subvenciones	718,20€
ACTIVO CORRIENTE (Derechos) (permanecen en la empresa menos de 1 año)	14.018,22 €	PASIVO CORRIENTE (Deudas a devolver en menos de un año)	0€
Existencias		Creditos pendientes de devolucion	
Facturas Cleintes	337,22 €	Deudas pendientes a proveedores	
Efectivo en Banco	13.681,00 €	Deudas pendientes a acreedores	
ACTIVO NO CORRIENTE	1.771,56 €	PASIVO NO CORRIENTE	9.027,58€
Bienes de inversion	2.394,00 €	Prestamo pendiente devolución	9.027,58€
Amortización de inversiones	-622,44€		
TOTAL ACTIVO	15.789,78 €	TOTAL PASIVO	15.789,78€

BALANCE FINAL

Al analizar el balance patrimonial, se concluye que el futuro de la empresa es viable económicamente.

a)La empresa se encuentra en una situación de solvencia, ya que el activo total (15.789,78 €) supera el pasivo no corriente + pasivo corriente (9.027,58 €), lo que descarta una situación de quiebra.

b)Además, según lo previsto, la empresa gozará de liquidez, dado que el activo corriente (14.018,22 €) es significativamente superior al pasivo corriente (0 €).

c)Aunque el capital social actual es de 3.000 €, el mínimo exigido, se confía en que los beneficios futuros permitirán incrementarlo. La relación entre los fondos propios y las deudas es adecuada, lo que fortalece la posición financiera.

d)Por último, la proporción entre deuda a corto y largo plazo es favorable. La deuda a largo plazo (9.027,58 €) representa únicamente un 57 % de la deuda total, muy por debajo del límite del 75 %.



8. CONCLUSIÓN

¿Es Shah un proyecto viable? La respuesta es claramente afirmativa, tanto desde el punto de vista técnico como económico. Durante el desarrollo del proyecto se han empleado tecnologías modernas, estables y con gran proyección, lo que garantiza una base técnica sólida, escalable y capaz de adaptarse a futuras necesidades o funcionalidades adicionales. La arquitectura empleada —compuesta por un back-end en Java Spring y un front-end en Vue 3— permite una separación clara de responsabilidades, facilitando tanto el mantenimiento como el crecimiento del sistema.

Desde el plano económico, **Shah** se ha desarrollado íntegramente con herramientas y plataformas gratuitas o de código abierto, optimizando así los recursos y reduciendo de forma considerable los costes asociados al desarrollo inicial. Esta característica, sumada al potencial de monetización del proyecto mediante modelos freemium, venta de artículos virtuales, publicidad o incluso suscripciones premium, lo convierten en una propuesta con gran capacidad para atraer inversión y generar beneficios sostenibles a largo plazo.

Sin embargo, **Shah no solo es un proyecto técnico o económico. También representa una apuesta por el impulso del talento y la innovación en ciudades como Cuenca**, tradicionalmente alejadas de los grandes polos tecnológicos. En este sentido, el proyecto pretende demostrar que es posible generar valor desde entornos menos centralizados, contribuyendo al fortalecimiento del tejido industrial y digital de la región.

Por último, **ha supuesto un auténtico desafío a todos los niveles**: desde el diseño de una arquitectura robusta hasta el despliegue en entornos reales, pasando por la integración de sistemas de seguridad, comunicación en tiempo real, experiencia de usuario y lo apretado del calendario. Este camino ha sido una experiencia enriquecedora tanto en lo técnico como en lo personal, reafirmando el compromiso con el desarrollo de software de calidad y con impacto real.

Shah no es solo un juego. Es un proyecto con visión de futuro.



ANEXO I – RESUMEN DE TECNOLOGÍAS EMPLEADAS

A continuación, se presenta una recopilación de todas las tecnologías utilizadas durante el desarrollo del proyecto, agrupadas por entorno (front-end y back-end), junto con una breve descripción de su uso concreto en la aplicación.

FRONT-END

Vue 3

Framework JavaScript progresivo orientado a la creación de interfaces de usuario dinámicas. Ha sido la base del desarrollo del cliente, organizando la aplicación en componentes reutilizables.

Vuetify

Librería de componentes basada en Material Design. Se ha usado para construir interfaces visualmente atractivas y responsive, facilitando la creación de botones, formularios, modales y más.

Pinia

Sistema de gestión de estado para Vue. Se ha utilizado para almacenar y compartir datos globales como la sesión del usuario, sus credenciales o su información personal, permitiendo persistencia entre vistas.

Vue Router

Manejador de rutas para aplicaciones Vue. Ha gestionado la navegación de la aplicación SPA (Single Page Application), controlando el acceso a páginas según el estado de autenticación del usuario.

BACK-END

Java Spring Boot

Framework principal para el desarrollo del servidor y la creación de una API REST. Ha permitido estructurar el proyecto en capas (modelo, repositorio, servicio y controlador) de forma organizada y escalable.

Spring JPA (Java Persistence API)

Librería de persistencia que facilita la comunicación con la base de datos mediante mapeo objeto-relacional (ORM). Ha simplificado la creación de entidades y consultas personalizadas.

Spring Security

Módulo de seguridad empleado para proteger el acceso al servidor, gestionar autenticaciones y cifrar contraseñas. También se ha utilizado para generar y validar tokens JWT.

Spring WebSockets

Herramienta que permite establecer conexiones en tiempo real entre el servidor y los clientes. Ha sido fundamental para funcionalidades como el chat o la sincronización en partidas multijugador..