

*Proyecto final de grado*

*Next Move*

Jorge Arcoya López

## **Abstract**

En este proyecto trata en el desarrollo de una página web de ajedrez, orientada tanto en partidas individuales como en partidas de modo multijugador. El objetivo es proporcionar un entorno moderno donde los usuarios pueden jugar partidas en modo online con otros jugadores, partidas contra la máquina o partidas de dos jugadores en el mismo dispositivo, permitiendo además obtener logros, ver rankings de las partidas del modo online y debatir en foros.

Dentro de la aplicación puedes jugar partidas en tiempo real, pudiendo además hablar con el otro jugador mediante un chat en tiempo real y por otra parte pudiendo abandonar la partida. También tiene otras funcionalidades sociales como poder ver tu perfil de usuario, tus estadísticas de partidas online, recibir recompensas tras completar logros por actividad, por jugar partidas online y participar en foros, todos estos datos se manejan utilizando Firebase.

Para realizar esta página web he usado React para el frontend, Node.js junto Express para el backend y Socket.io para la comunicacion en tiempo real

This project deals with the development of a chess web page, oriented in both individual games and multiplayer mode games. The objective is to provide a modern environment where users can play games in online mode with other players, games against the machine or games of two players on the same device, also allowing you to obtain achievements, see rankings of the games of the online mode and discuss in forums.

Within the application you can play games in real time, being able to talk to the other player through a chat in real time and on the other hand and can leave the game. It also has other social functionalities such as seeing your user profile, your online game statistics, receiving rewards after completing activity, by playing online games and participating in forums, all these data are handled using Firebase.

To make this website I have used React for the border, node.js together for the backend and socket.io for real -time communication

# Índice

- Abstrac\_\_\_\_\_ Página 2
- Justificación\_\_\_\_\_ Páginas 4-5
  - Características generales: Página 4
  - Restricciones generales: Página 4
  - Aspectos a cubrir: Página 4
  - Estudio de las prestaciones: Página 5
- Justificación de la tecnología empleada \_\_\_\_\_ Páginas 6-7
- Requerimientos hardware y software \_\_\_\_\_ Páginas 8-9
- Análisis y diseño \_\_\_\_\_ Páginas 10-12
  - Diagrama de casos de uso y diagrama de clases: Página 10
  - Descripción de la base de datos: Páginas 11 y 12
- Implementación\_\_\_\_\_ Página 13-14
- Evaluación y prueba \_\_\_\_\_ Página 15-20
- Manual de estilos \_\_\_\_\_ Página 21-27
  - Sketch: Página 21
  - Criterios de accesibilidad: Páginas 22 y 23
  - Criterios de usabilidad: Páginas 24 y 25
  - Tipografía: Página 25
  - Mapa de color: Páginas 25, 26 y 27
  - Dispositivos/vistas: Página 27
- Software utilizado\_\_\_\_\_ Página 28
- Mejoras posibles y aportaciones \_\_\_\_\_ Página 29

# **Justificación**

## ***a. Características generales***

El proyecto consiste en una página web para jugar al ajedrez, de forma local, online o contra la máquina, añadiendo otras funcionalidades como chats en tiempo real, logros, recompensa y foros.

Se busca crear una experiencia de juego completa donde los usuarios se puedan divertir mientras compiten entre ellos por ver quien es el mejor, además de comunicarse y progresar completando diversos logros.

## ***b. Restricciones generales***

- Este proyecto está pensado como una aplicación web no para móvil ni aplicación de escritorio
- Las partidas contra la máquina usan una Inteligencia Artificial sencilla para realizar movimientos inteligentes, no proporciona información adicional.
- Son partidas de ajedrez de toda la vida, 1 vs 1, no hay otros modos diferentes.
- No cuenta con moderación automática ni avanzada de los chats y foros
- Los recursos de almacenamiento son limitados

## ***c. Aspectos a cubrir, junto con los que no se van a tratar***

A cubrir:

- Registro e inicio de sesión
- Sistema de partidas
- Chat en tiempo real durante partidas online
- Logros y recompensas
- Estadísticas de usuario y ranking
- Conversaciones en los foros
- Sistema de abandono de partidas online

Aspectos que no se cubren:

- Soporte para otros idiomas
- Sistema de ranking y emparejamiento por ELO
- Moderación automática
- Torneos por eliminatorias y partidas simultáneas
- Uso de una IA más compleja para las partidas contra la máquina y el ranking por ELO

***d. Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría***

Plataforma	Partidas online	Partidas contra la IA	Chat en tiempo real	Logros	Ranking	Registro	Foros
Next Move (TFG)	✓	✓	✓	✓	✓	✓	✓
Lichess	✓	✓	✓	✓	✓	✓	✗
Chess.com	✓	✓	✓	✓	✓	✓	✓
Ajedrez online generico	✓	✗	✗	✗	✗	✗	✗

## Justificación de la tecnología empleada

A continuación voy explicar las tecnologías que he utilizado en el proyecto y los motivos por los que he elegido esas tecnologías.

**Frontend:** Para el desarrollo del cliente web, he optado por elegir React como framework principal y Bootstrap como sistema de diseño.

- **React:** Se ha elegido como el framework principal del proyecto por su enfoque basado en componentes y ser un framework fácil de utilizar y el cual tiene una gran comunidad, lo que facilita la resolución de posibles problemas.

Además, al ser el framework que mejor controla, ha permitido centrar los esfuerzos en la lógica y funcionalidad del proyecto, sin tener que dedicar tiempo a aprender nuevas herramientas. React además también proporciona otras funcionalidades que permiten actualizar en tiempo real la interfaz, algo bastante importante en aplicaciones las cuales deben interactuar con los usuarios

- **Bootstrap:** He escogido bootstrap para el diseño de la página web ya que proporciona una gran variedad de componentes predefinidos y estilos responsivos, los cuales solo tenía que personalizar, lo que ha hecho reducir el tiempo necesario para poder conseguir una buena interfaz que sea adaptable a varios dispositivos.

Investigación previa: Durante la fase de análisis se consideró el uso de Vue.js como alternativa a React, así como otras bibliotecas de diseño como Material UI. Sin embargo, al tener más conocimientos de React y Bootstrap se decidió elegir estas dos tecnologías.

**Backend:** Para la parte del desarrollo del servidor he elegido Node.js junto con Express.

- **Node.js con Express:** He elegido utilizar esta tecnología para el servidor por su alto rendimiento a la hora de funcionar con aplicaciones en tiempo real, lo que permite gestionar las partidas y la comunicación de una forma ideal, además de ser uno de los pocos lenguajes de servidor que manejo bastante bien lo que me ha permitido ahorrar tiempo en buscar y aprender otros tipos de lenguajes para el servidor.

Usar express me ha permitido estructurar la lógica de forma clara, facilitando el mantenimiento del código.

Investigación previa: A la hora de elegir la tecnología para el servidor se valoró usar el lenguaje de PHP pero al buscar una página web interactiva y en tiempo real sin mucho tiempo de carga se optó por usar Node.js.

**Comunicación en tiempo real:** Para la comunicación de los usuarios en tiempo real se ha optado por Socket.io.

- Socket.io fue la solución seleccionada para implementar las partidas online en tiempo real y los chats. Permite la comunicación bidireccional eficiente entre cliente y servidor, lo cual es fundamental para que las jugadas se reflejan instantáneamente en ambos jugadores y además se puedan mandar mensajes de forma instantánea entre dos o más usuarios.

Investigación previa: Se consideró usar WebRTC para mandar audios o realizar videollamadas pero por complejidad se optó por descartar finalmente.

**Base de datos y autenticación:** En el almacenamiento de los datos y autenticación de usuarios se acabó optando por elegir Firebase por varios motivos.

- Firebase Authentication: Al permitir almacenar usuarios de una forma sencilla, poder registrarse e iniciar sesión con distintos proveedores y métodos de autenticación.
- Firebase Firestore: Es una base de datos NoSQL la cual permite almacenar datos de forma escalable y eficiente.

Investigación previa: Al investigar se consideró usar MongoDB como base de datos y Passport.js para la autenticación pero acabe eligiendo Firebase por que este ya lo incluye las dos cosas en uno lo que permite ahorrar implementar más tecnologías.

# **Requerimientos hardware y software**

A continuación, voy a detallar los requerimientos necesarios para poder ejecutar de forma correcta el cliente y el servidor del proyecto de la plataforma de ajedrez online.

## **Cliente/Frontend:**

- Software:
  - Navegador compatible con JavaScript (Cualquier navegador moderno lo soporta)
  - Sistema operativo: Windows 10/11, macOS, o cualquier distribución moderna de Linux.
  - Librerías necesarias:
    - React (v18 o superior)
    - Bootstrap (v5)
    - Firebase SDK (para autenticación y Firestore)
    - Socket.IO Client (para conexión en tiempo real)
- Hardware mínimo recomendado:
  - Procesador Dual-core
  - 4 GB de RAM
  - Resolución mínima: 1280x720

## **Servidor/Backend:**

- Software:
  - Sistema operativo: Windows, Linux (Ubuntu 20.04 o superior) o macOS
  - Node.js (v18 o superior)
  - Express.js
  - Socket.IO (para la gestión en tiempo real de partidas y chat)
  - Firebase Admin SDK (para interacción con Firestore)



- Hardware mínimo recomendado:
  - Procesador Quad-core (Intel i5 / Ryzen 5 o superior)
  - 8 GB de RAM
  - Almacenamiento: mínimo 2 GB libres para dependencias y base de datos local (si se usa emulador)
  - Conexión a internet estable para comunicación con servicios de Firebase

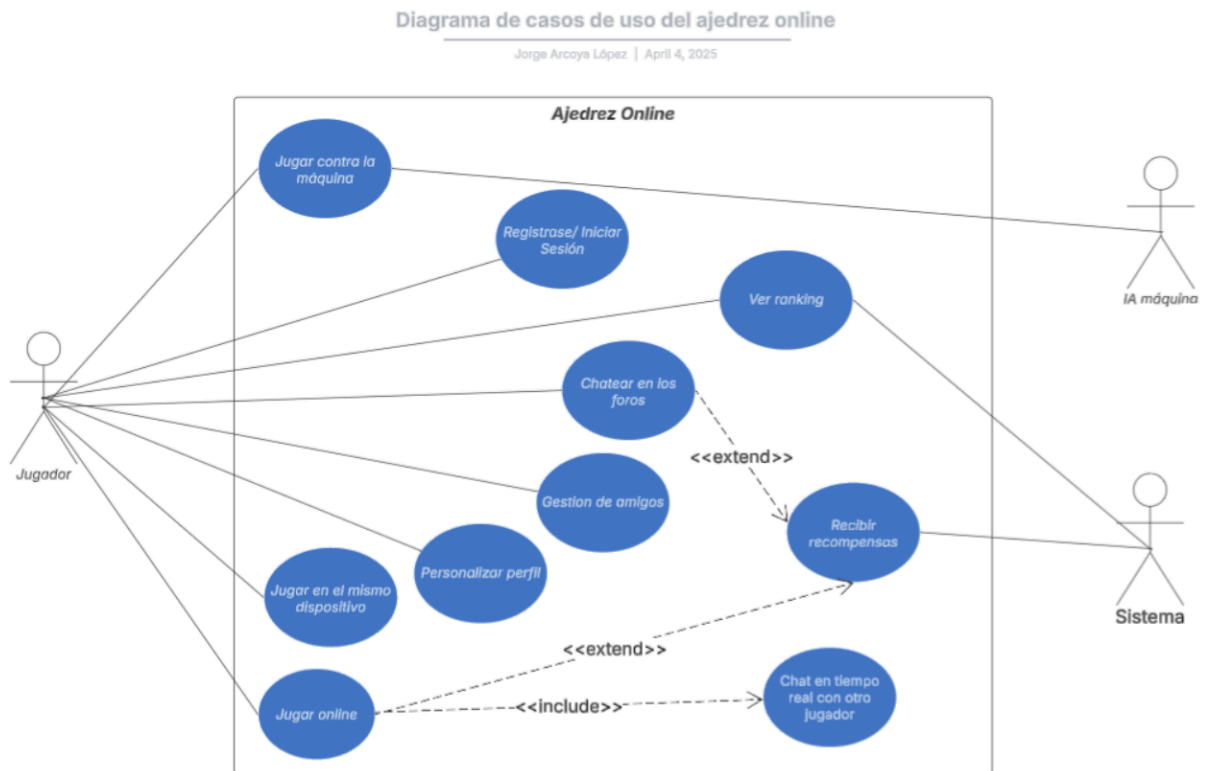
En entorno de producción, se recomienda desplegar el backend en servicios como Render.

Dependencias que tienen en común servidor y cliente: Tanto el cliente como el servidor necesitan acceso a Firebase por lo que se necesita:

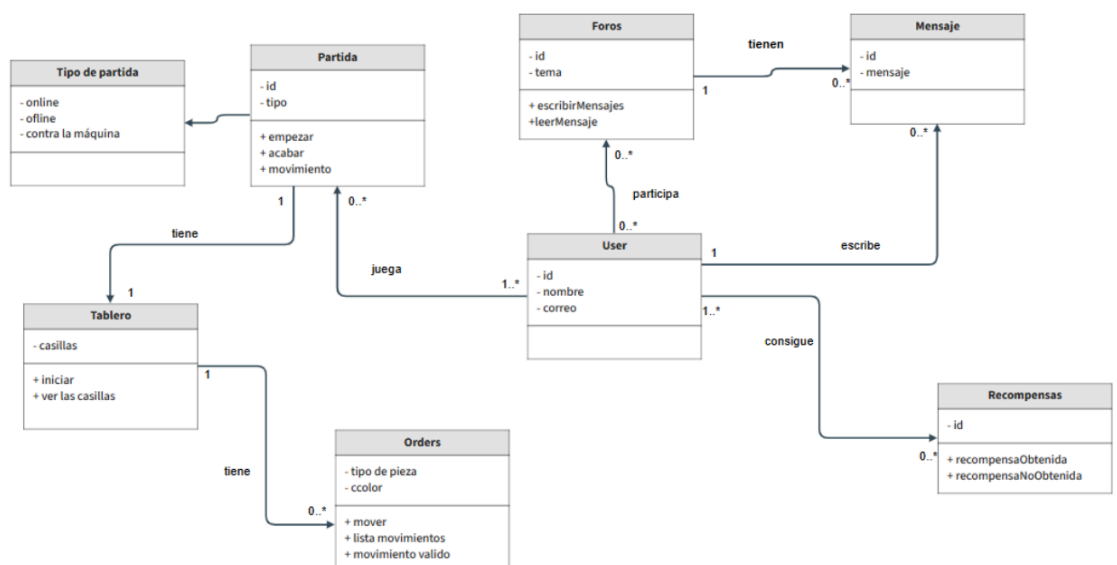
- Tener una cuenta de Google con un proyecto creado en Firebase
- Haber configurado correctamente las reglas de seguridad y los permisos de acceso

# Análisis y diseño

a. Diagrama de casos de uso y diagrama de clases:



## Diagrama de clases Proyecto final Ajedrez



## b. Descripción de la base de datos:

**Arbol general de la base de datos:**

## Colección usuarios

- o Id de usuario
- Nombre\_email: string
- Avatar: string
- Estadísticas: array
  - Partidas Ganadas: number
  - Partidas Perdidas: number
  - Partidas Empatadas: number

## Colección partidas

## - id de partida

- Jugadores: array
  - Id primer jugador
  - Id segundo jugador
- Duración: number (en segundos o minutos)
- Ganador: string (id del usuario del ganador)
- Fecha: timestamp
- Movimientos: array con la lista de los movimientos

## Colección foros

## - Id del foro

- Tema: string
- Autor: string
- Autor id: string (usuarioID)
- Fecha: timestamp
- Descripción: string
- Mensajes: Subcoleccion
  - Id del mensaje
    - Contenido: string
    - Autor: string
    - Autor id: string (usuarioID)
    - Fecha: timestamp

## Colección logros

## -id del logro

- Descripción: string
- Nombre: nombre del logro
- Recompensa: array
  - ■String con la recompensa

## Colección logros completados

## -id del documento del logro completado

- Fecha: timestamp
- Logro id: string (id del logro que se completa)
- Usuario id: string (usuarioID)

## Descripción de los datos de los documentos

**Colección usuarios:** Esta colección tendrá todos los usuarios registrados en la plataforma. Cada documento dentro de esta colección representará a un usuario y tendrá sus datos de perfil, estadísticas y amigos

- Id del usuario: Documento para identificar al usuario
- Nombre: El nombre del usuario.
- Avatar: URL o referencia a la imagen del avatar del usuario (la cual estará en el servidor node).
- Estadísticas: Array que contiene las estadísticas del usuario (partidas ganadas/perdidas/empatadas).

**Colección partidas:** Colección que tiene los detalles de las distintas partidas que se juegan

- Id de la partida: Documento para identificar las partidas.
  - Jugadores: Lista con los id de los jugadores que participan en la partida
  - Duración: Tiempo que duró la partida.
  - Ganador: id del jugador que ganó la partida.
  - Fecha: Fecha en que termino la partida
  - Movimientos: Lista con los movimientos que se han realizado en la partida

**Colección foros:** Colección que tiene los detalles y mensajes de los distintos foros de la página web

- Id del foro: Documento para identificar los foros.
- Tema: El título o nombre del tema del foro.
- Autor id: El id del usuario que ha creado el foro.
- Autor: Nombre del autor que ha creado el foro
- Fecha: Timestamp de la creación del foro.
- Mensajes: Subcolección con la lista de mensajes dentro del foro, con cada mensaje identificado por un id del mensaje y contiene el contenido, la fecha y el autor.

**Colección logros:** Colección que tiene los logros que los usuarios pueden obtener en la página web

- Id del logro: Documento para identificar los logros
  - Descripción: Descripción de lo que trata el logro
  - Nombre: Nombre del logro
  - Recompensas: Lista con las recompensas de estos logros

**Colección logros completados:** Colección que tiene los logros que han sido completado por cada usuario

- Id del logro completado: Documento para identificar los logros completado
  - Fecha: Timestamp de la fecha en que se completó el logro
  - Logro id: Id con el logro que se ha completado
  - Usuario id: Id del usuario que ha completado el logro

# **Implementación**

Durante el desarrollo del proyecto se han utilizado varios elementos para construir la plataforma y que funcione correctamente. A continuación se describen:

- Hojas de estilo: Se ha utilizado principalmente Bootstrap junto con estilos propios y personalizados mediante archivos CSS.
  - Bootstrap proporciona componentes que son responsivos y clases reutilizables para otros elementos
  - También se han creado archivos de estilos propios donde se definen colores personalizados y clases propias y que complementan o personalizan clases ya definidas por Bootstrap.
- Plantillas: En react no se usan plantillas sino que se usan componentes reutilizables, se han creado varios componentes para la aplicación:
  - Header: Header con barra de navegación
  - Footer: Con datos principales del proyecto
  - Landing page: La página de inicio de la aplicación
  - Página de información: Información sobre lo que se puede hacer en la página web
  - Jugar: Página que redirige a los diferentes modos de juego
  - Modo Online/IA/Offline: Componentes para los distintos modos de juego
  - Ranking: Página para ver el ranking de los mejores jugadores del modo online
  - Foros: Página para entrar a los foros y hablar con otros usuarios
  - Login/Registro: Página para iniciar sesión o registrarse en la página web
  - Perfil: Página para ver tus estadísticas y logros
- Formularios: Se han utilizado formularios tanto para el registro e inicio de sesión como para editar tu perfil:
  - Estos formularios están conectados con Firebase Auth para autenticar al usuario.
  - Se validan campos básicos (email, contraseña, nombre) y se manejan errores de autenticación.
  - Los datos ingresados se guardan o consultan desde Firestore en función del tipo de operación.
- Funciones de envío de datos: El envío de los datos en la aplicación se gestiona con:
  - Firebase Firestore para datos estructurados (usuarios, estadísticas, logros, premios, etc).
  - Firebase Auth para la autenticación de usuarios.
  - Socket.IO para enviar eventos en tiempo real como movimientos del ajedrez o mensajes del chat.

- Conexión y consultas a la base de datos
  - Para acceder a la base de datos se usan funciones como `getDocs`, `setDoc`, `updateDoc`, `query`, `where`, etc.
  - Ejemplo: al iniciar sesión, se realiza una consulta a la colección usuarios para obtener los datos relacionados con el uid del usuario autenticado.
- Uso de fichero de configuración: Se ha utilizado un fichero `firebase.js` para centralizar toda la configuración de Firebase en el cliente y un archivo `FirebaseAdmin.js` para centralizar toda la configuración de Firebase en el servidor.

Este archivo contiene:

- Claves de conexión (API Key, Auth Domain, Project ID...).
- Inicialización de `firebaseApp`, `auth`, `db` (FireStore) y `storage`.

Además para poder implementar la librería de Stockfish ha sido necesario añadir un archivo `worker` y crear otro archivo de utilidad para enlazar el worker con el código React.

## Evaluación y prueba

### *Testeo de la aplicación web:*

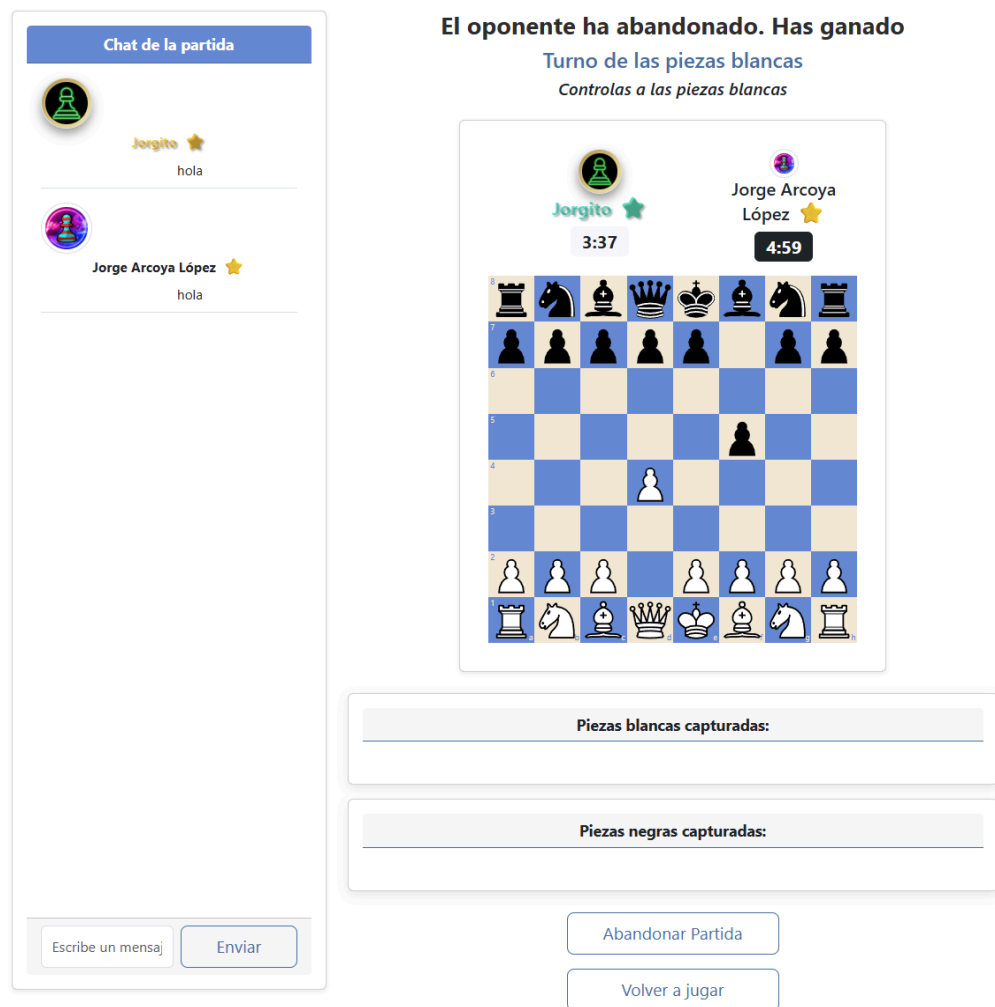
Durante el desarrollo se han realizado pruebas para verificar que las funcionalidades de la página web funcionen correctamente, se han ido realizando pruebas a varios apartados:

- Validación de formularios y realimentación al usuario: Se ha comprobado que no se permite introducir datos de registro e inicio de sesión que no sean correctos. También se revisó la realimentación al usuario mediante mensajes de error o confirmación

The screenshot shows a registration form with the following elements:

- Registrarse** (Title)
- Buttons for social login: "Inicia con Github" and "Inicia con Google".
- Form fields:
  - Correo**: Input field containing "jorge@345".
  - Contraseña**: Password input field with masked characters "\*\*\*\*\*".
  - Confirmar Contraseña**: Confirmation password input field with masked characters "\*\*\*\*\*".
- Registrarse** (Submit button)
- Error message: "Las contraseñas no coinciden" (The passwords do not match).
- Link: "¿Ya tienes una cuenta? [Iniciar sesión](#)" (Do you already have an account? [Log in](#)).


- Comprobación de flujo de partida: Se ha robado que el flujo de inicio, desarrollo y finalización de partidas, incluyendo la sincronización entre varios jugadores online funcionan correctamente



- Vista del perfil y actualización en tiempo real: Se ha testado los cambios en tiempo real del perfil tras acabar las partidas, se han actualizan las estadísticas y se desbloquean los logros.



## Antes



Jorge Arcoya López ★

Editar Perfil

**Estadísticas**

Partidas Jugadas	Partidas Ganadas	Partidas Perdidas	Partidas Empatadas	Porcentaje de victorias
121	62	59	0	51%

**Logros**

**Creador de Conversaciones**

Crear 10 foros

Recompensa: Dar relieve al nombre

**Mensajero Novato**

Manda 100 mensaje diferentes en los foros

Recompensa: Cambiar el color del nombre a colores aleatorios

**Forero Legendario**

Mandar 500 mensajes diferentes en los foros

Recompensa: Circulo alrededor de la imagen de perfil

**Ganador Novato**

Gana 10 partidas de ajedrez en el modo online

Recompensa: Poner una estrella al lado del nombre

Completado


**Maestro en Progreso**

Gana 50 partidas de ajedrez en el modo online

Recompensa: Desbloquear otra imagen de perfil

Completado

## Después



Jorge Arcoya López ★

Editar Perfil

**Estadísticas**

Partidas Jugadas	Partidas Ganadas	Partidas Perdidas	Partidas Empatadas	Porcentaje de victorias
122	63	59	0	52%

**Logros**

**Creador de Conversaciones**

Crear 10 foros

Recompensa: Dar relieve al nombre

**Mensajero Novato**

Manda 100 mensaje diferentes en los foros

Recompensa: Cambiar el color del nombre a colores aleatorios

**Forero Legendario**

Mandar 500 mensajes diferentes en los foros

Recompensa: Circulo alrededor de la imagen de perfil

**Ganador Novato**

Gana 10 partidas de ajedrez en el modo online

Recompensa: Poner una estrella al lado del nombre

Completado


**Maestro en Progreso**

Gana 50 partidas de ajedrez en el modo online

Recompensa: Desbloquear otra imagen de perfil

Completado

Logro desbloqueado al cumplir metas:



Jorgito ★

Editar Perfil

**Estadísticas**

Partidas Jugadas	Partidas Ganadas	Partidas Perdidas	Partidas Empatadas	Porcentaje de victorias
119	50	69	0	42%

**Logros**

**Creador de Conversaciones**

Crear 10 foros

Recompensa: Dar relieve al nombre

Completado

**Mensajero Novato**

Manda 100 mensaje diferentes en los foros

Recompensa: Cambiar el color del nombre a colores aleatorios

Completado

**Forero Legendario**

Mandar 500 mensajes diferentes en los foros

Recompensa: Circulo alrededor de la imagen de perfil

Completado

**Ganador Novato**

Gana 10 partidas de ajedrez en el modo online

Recompensa: Poner una estrella al lado del nombre

Completado

**Maestro en Progreso**

Gana 50 partidas de ajedrez en el modo online

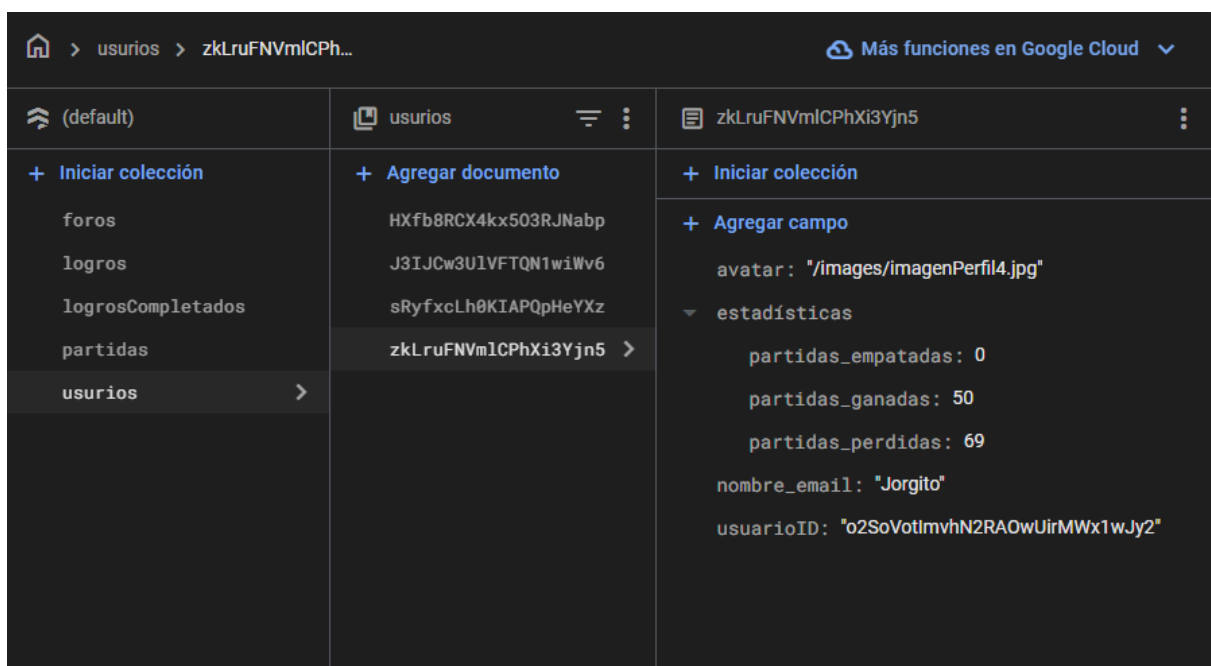
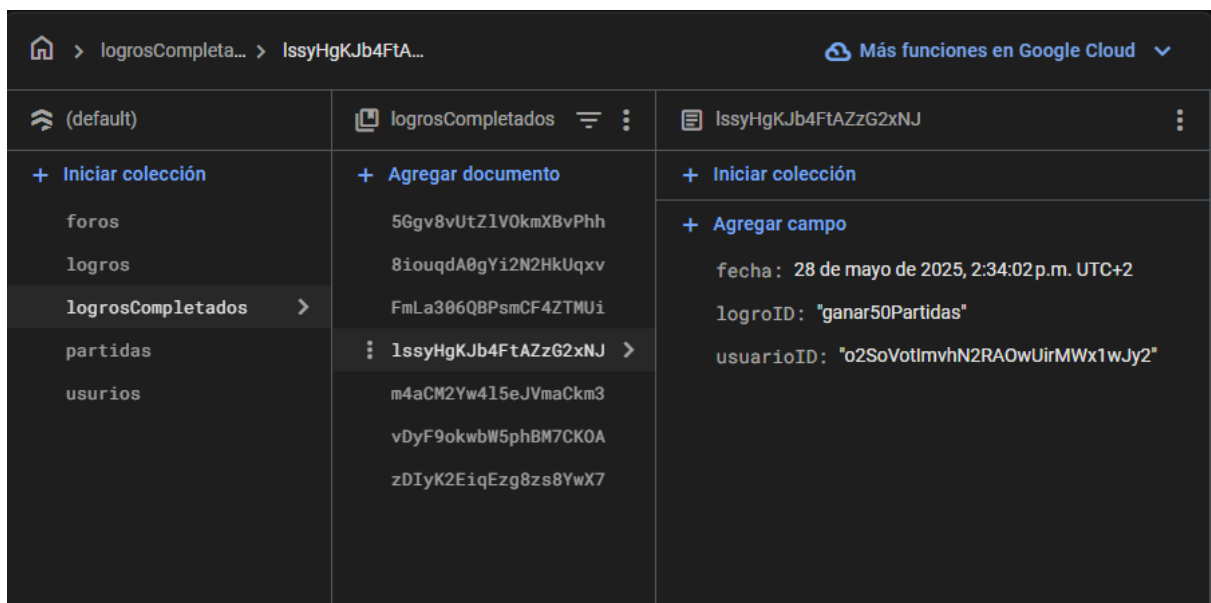
Recompensa: Desbloquear otra imagen de perfil

Completado

**Testeo de la base de datos**

También se han realizado pruebas para validar que se almacenen correctamente los datos en la base de datos de Firebase:

- Los usuarios se guardan correctamente en la colección usuarios, incluyendo sus estadísticas y sus datos personales
- Los logros y los logros completados se almacenan indicando que usuario ha completado ese logro y cual es el logro que ha completado.
- Se ha revisado la consistencia y que coinciden los datos que se muestran en la pantalla y los que hay almacenados en la base de datos



**Batería de ejemplos:**

- Dejar campos vacíos muestra un mensaje de error:

## Registrarse

Inicia con Github

Inicia con Google

---

**Correo**

ejemplo@gmail.com

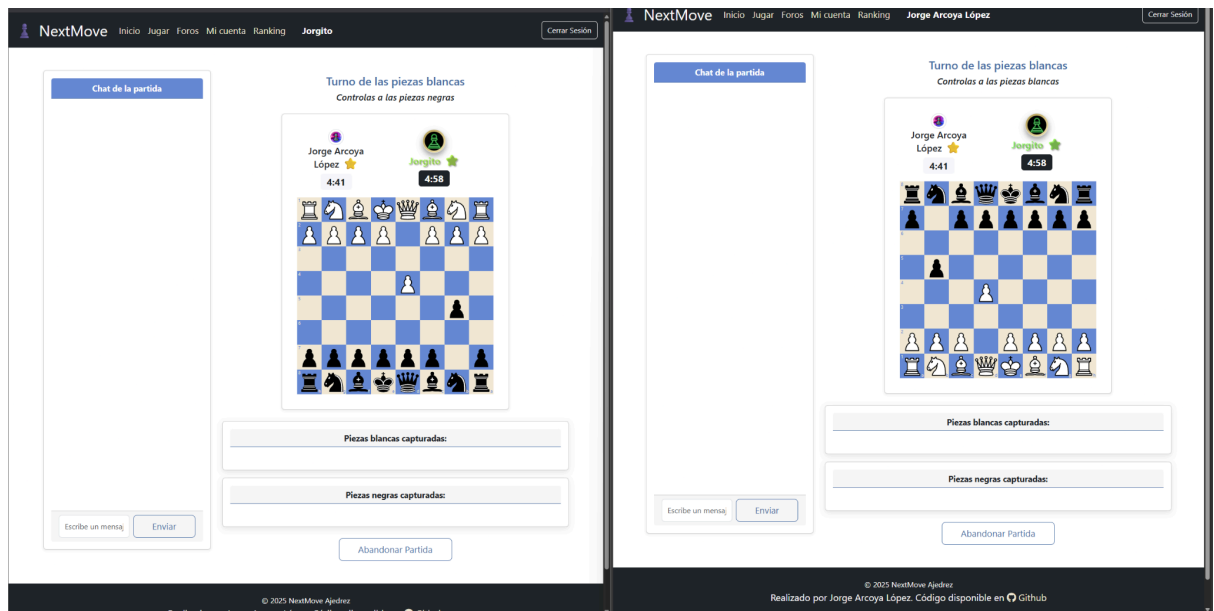
**Contraseña** Completa este campo

**Confirmar Contraseña**

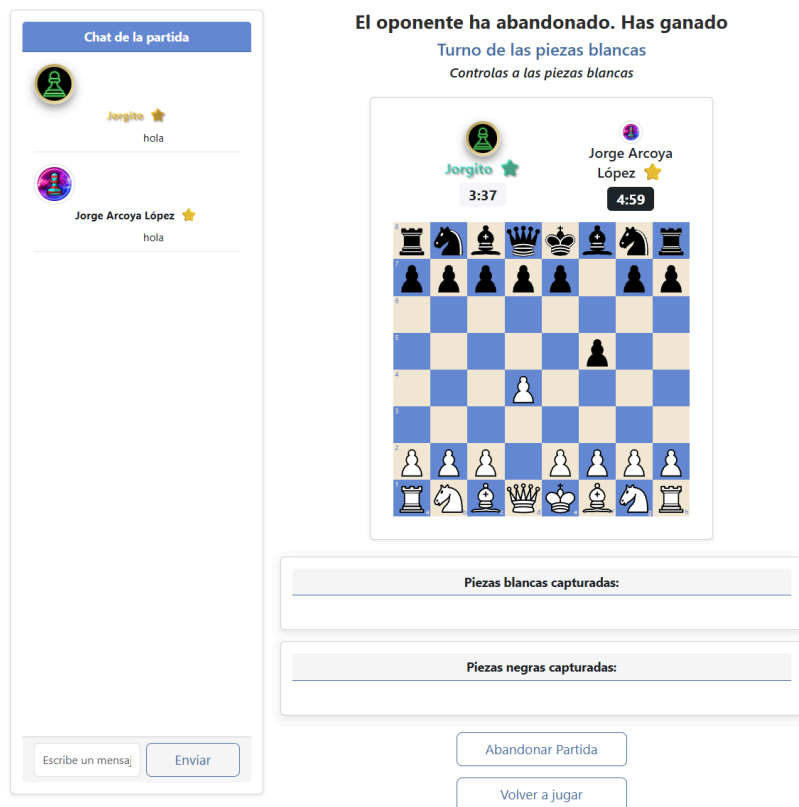
Registrarse

¿Ya tienes una cuenta? [Iniciar sesión](#)

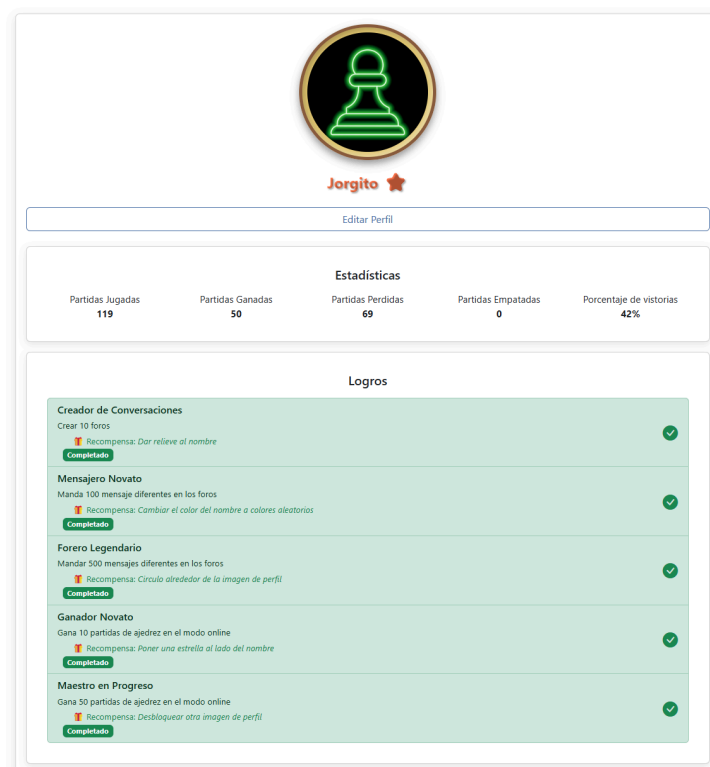
- Ver tablero sincronizado con otro jugador:



- Abandonar partida da la partida por perdida:



- Ganar 10 partidas, u otros logros, te completa el logro y da la recompensa:

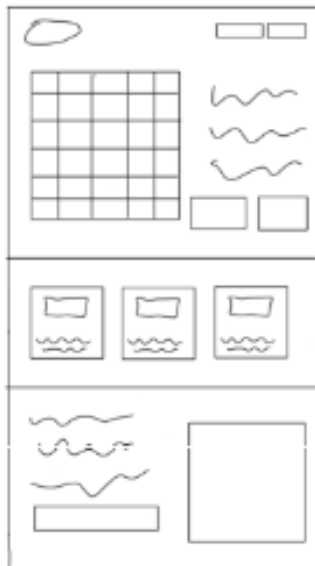


# Manual de estilos

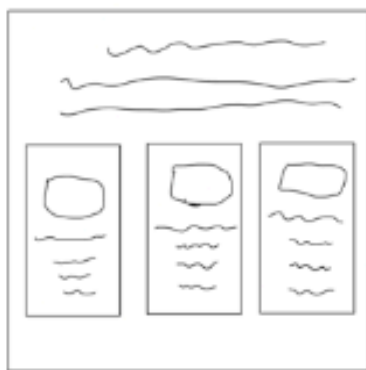
- Sketch:

SKETCH Proyecto final

Landing Page



Página información



Página partidas (Offline, IA, online)



Página foros



Conversación en foro



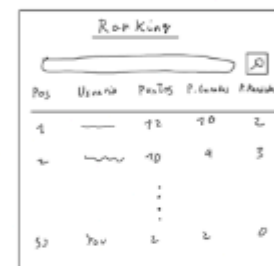
Página inicio sesión / registrarse



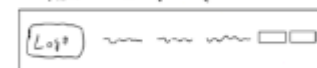
Página perfil



Página ranking



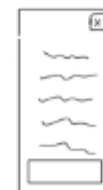
Mano de navegación (Header)



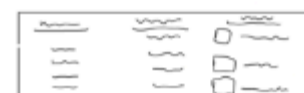
Vista página



Desplegable



Footer



- Criterios de accesibilidad:
  - Uso correcto de etiquetas semánticas

```
return (
  <BrowserRouter>
    <Header />
    <ScrollTop />

    <main>
      <Routes>
        <Route path="/" element={<LandingPage />} />
        <Route path="/informacion" element={<Informacion />} />
        <Route element={<RutasLogin />}>
          <Route path="/login" element={<Login />} />
        </Route>
        <Route path="/jugar" element={<Jugar />} />
        <Route element={<RutasProtegidas />}>
          <Route path="/modo-ia" element={<ModoIA />} />
        </Route>
        <Route element={<RutasProtegidas />}>
          <Route path="/modo-online" element={<ModoOnline />} />
        </Route>
        <Route element={<RutasProtegidas />}>
          <Route path="/perfil" element={<Perfil />} />
        </Route>
        <Route element={<RutasProtegidas />}>
          <Route path="/foros" element={<Foros />} />
          <Route path="/foros/:id" element={<ForosConversacion />} />
        </Route>
        <Route path="/modo-local" element={<ModoLocal />} />
        <Route path="/ranking" element={<Ranking />} />
        <Route path="*" element={<Error404 />} />
      </Routes>
    </main>

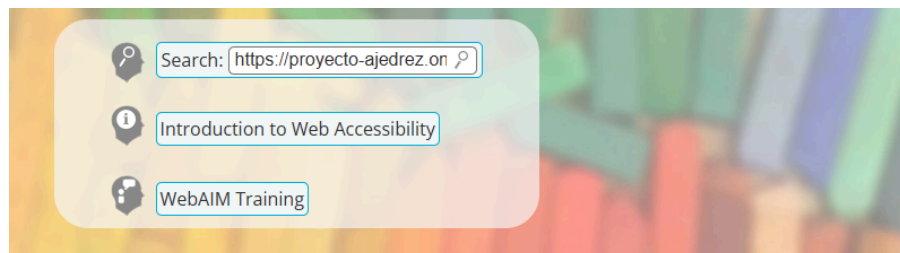
    <Footer />
  </BrowserRouter>
)
```

- Uso de etiquetas de encabezado jerárquico:

```

[/* Hero */]
<div className="row align-items-center mb-5">
  <div className="col-md-6 text-center mb-4 mb-md-0">
    <div className="bg-light border rounded p-4">
      <div className="contenedorImgTablero">
        
      </div>
    </div>
  </div>
  <div className="col-md-6">
    <h1 className="fw-bold">Juega donde, de la forma y con quien quieras al ajedrez</h1>
    <p className="text-muted">Desafia a la IA, juega con amigos tanto online como offline o únete a los distintos foros para disfrutar del
```

- Contraste de colores adecuado:



## Contrast Checker

[Home](#) > [Resources](#) > Contrast Checker

**Foreground**

Hex Value

#0000FF

Color Picker

Alpha

1

Lightness

**Background**

Hex Value

#FFFFFF

Color Picker

Lightness

Contrast Ratio

**8.59:1**

[permalink](#)

- Texto alternativo en imágenes (alt):

```
<div className='d-flex align-items-center gap-2'>
  <img
    src={` ${SERVER_URL}${mensaje.avatar}`}
    alt={`Avatar de ${mensaje.nombre_email} || 'Jugador'`}
    className={`rounded-circle img-thumbnail imagenModoOnline ${mensaje.claseAvatar}`}
  />
</div>
```

- Idioma del documento especificado

```
<!doctype html>
<html Lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">
    <title>NextMove</title>
    <link rel="icon" href="public/chessboard.png" type="image/png">
  </head>
```

- Criterios de usabilidad:
  - Diseño responsivo



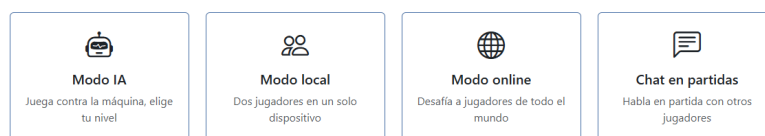
- Menús intuitivos y visibles



- Consistencia visual

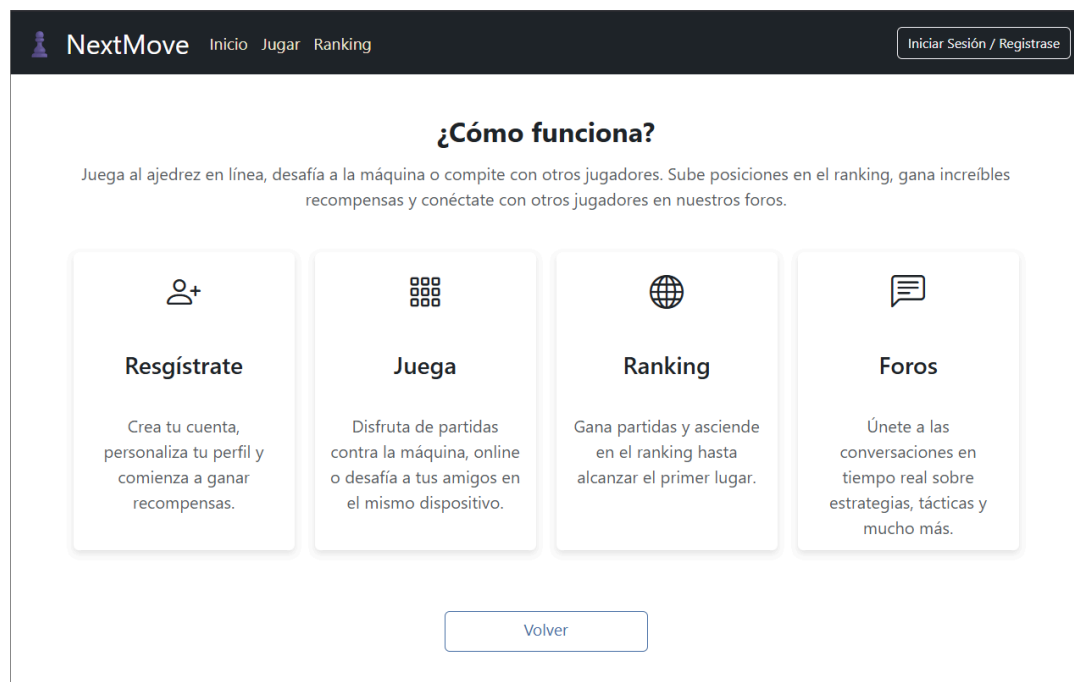


### Modos de juego

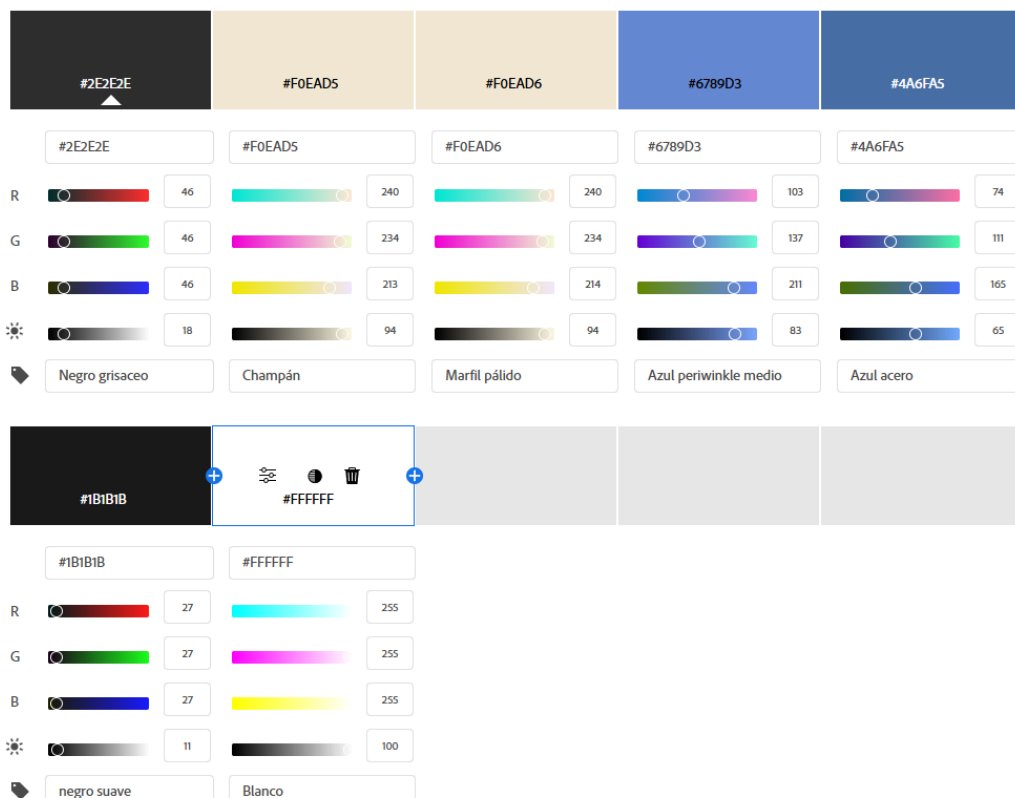




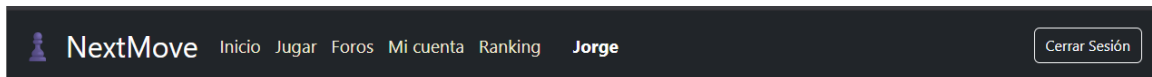
- Accesibilidad desde el primer uso



- Tipografía:  
Fuente: Press Start 2P
- Mapa de colores:

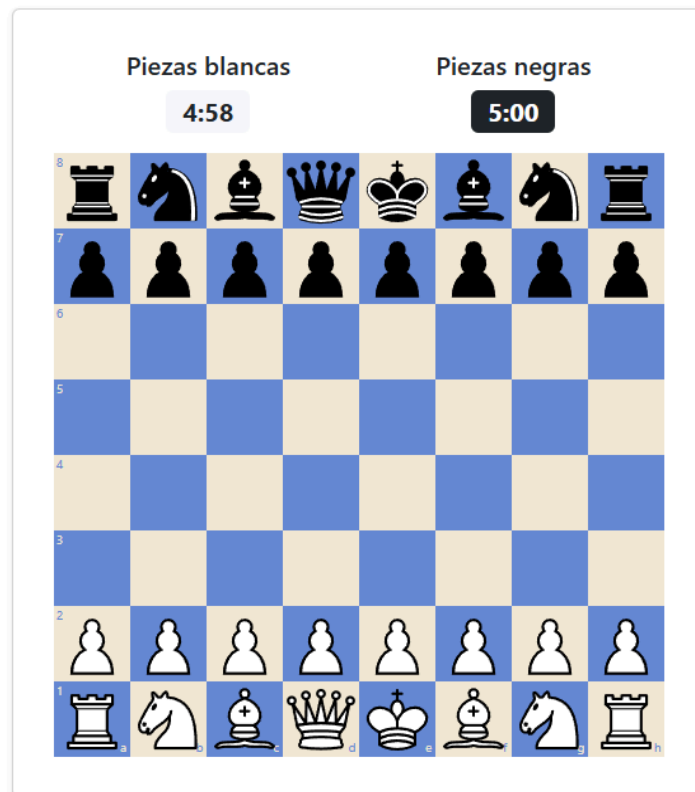


- Negro grisáceo: Se utiliza en el header, y footer
- Champán: Se usa en texto de enlaces y botones (hover y no hover)



- Marfil pálido: Se usa para las casillas blancas del tablero de ajedrez
- Azul periwinkle medio: Se usa para las casillas negras del tablero de ajedrez

### Turno de las piezas blancas



- Azul acero: Se usa como borde de los cards, fondo de algunos botones y borde de otro tipo de botone
- Negro suave: Se usa como color del texto principal de la página
- Blanco: Se usa como color de fondo y color de texto en fondos oscuros



- Dispositivos/vistas:

El proyecto ha sido diseñado para ser responsive y adaptable, es decir que se vea bien en diferentes vistas, las vistas que dispone la página son:

- Vista ordenador: Es la vista principal en la cual los elementos están distribuidos horizontalmente y son visibles en un mismo layout.
- Vista tablet: Es la segunda vista para la que está pensada la página en la cual se adaptan los elementos automáticamente de columnas a filas y se reajustan los botones y menús.
- Vista móvil: Es la última vista para la que está pensada la página, donde el menú de navegación pasa a ser un menú hamburguesa, los elementos se apilan verticalmente y se optimiza la interfaz para poder usarla con una solo mano

Tecnologías utilizadas:

- **React:** Permite dividir la interfaz en componentes reutilizables, lo cual facilita la gestión de diferentes vistas en función del tamaño del dispositivo.
- **Bootstrap:** Se ha utilizado clases responsivas (col-sm-, d-md-none, container-fluid, etc.) para asegurar el correcto ajuste en distintos tamaños de pantalla.

## **Software utilizado**

<b>Software</b>	<b>Uso en el proyecto</b>
Visual Studio Code	Usado para desarrollar el código del proyecto
React	Librería de javascript usada para el código del cliente
Bootstrap	Framework CSS para facilitar el diseño responsive y el uso de componentes visuales predefinidos
Node.js + Express	Usado para crear el código del servidor
Firebase	Usado como backend para la autenticación de usuarios y base de datos en tiempo real
Socket.io	Biblioteca usada para la comunicación en tiempo real en las partidas online y en los foros
Render	Plataforma usada para desplegar la aplicación web
Git	Control de versiones durante el desarrollo del proyecto
Github	Repositorio remoto para alojar el código del proyecto
Lucidchart	Herramienta usada para crear los diagramas de casos de uso, diagrama de clases, etc.
PowerPoint	Usado para crear la presentación del proyecto
Wave / Lighthouse	Herramienta usada para comprobar los criterios de usabilidad y accesibilidad

## **Mejoras posibles y aportaciones**

Durante el desarrollo del proyecto he ido viendo apartados y áreas las cuales se pueden mejorar tanto a nivel funcional como a nivel de organización.

- Mejoras de funcionalidad:
  - Añadir un modo espectador a las partidas del modo online sin tener que participar en la partida
  - Añadir un sistema de notificaciones para cuando se gana un logro o tienes mensajes en los foros
  - Añadir un modo de entrenamiento donde la IA vaya dando consejos
  - Añadir un sistema de amigos a los cuales se pueden agregar y desafiar a partidas online
  - Cambiar el sistema de ranking y emparejamiento a uno basado en el ELO y los movimientos inteligentes de los jugadores
  - Añadir tutoriales para nuevos jugadores
- Posibles mejoras de código:
  - Refactorizar posible código repetido
  - Organizar mejor algunos componentes
- Aportaciones:
  - La mecánica de logros y recompensas ayuda a mantener el interés del usuario
  - Tiene un diseño responsive pensado para usuarios en distintos dispositivos
  - Tener multijugador en tiempo real ayuda a la interacción dinámica de los jugadores.