

PROYECTO:

ChessGame



Realizado por: *Miguel Ortega Álvarez*



Galileo

Desarrollo de Aplicaciones Web (DAW)

Valladolid, a 23 de Mayo de 2024

Contenido

1	Introducción.....	4
1.1	Objeto del proyecto.....	6
1.2	Lenguajes empleados	7
1.2.1	React	7
1.2.2	JavaScript	9
1.2.3	PHP	10
1.2.4	MySQL	11
1.2.5	Visual Studio Code	12
1.3	Distribución	14
1.3.1	Requisitos de los clientes	16
1.3.2	Licenciamiento	18
2.	Recursos	19
2.1	Recursos Hardware	19
	Servidor Web:	19
	Almacenamiento:	20
	Conexión Red:	21
2.2	Recursos Software	23
	Sistema Operativo:	23
	Servidor Web:	23
	Motor de Base de Datos:	23
	Herramientas de Desarrollo:	24
2.3	Humanos	25
2.4	Previsión económica del coste del proyecto.....	27
	Costos de Hardware	27
	Costos de Software	27
	Costos de Recursos Humanos	28
	Otros Costos:	29
2.5	Cronograma	30
3	Descripción de la aplicación	31
3.1	Funcionamiento general	31
3.2	Arquitectura.....	38
3.2.1	Diseño de la base de datos.....	39
3.3	Interfaz.....	42

3.3.1	Características generales	42
3.3.2	Adaptación a dispositivos móviles	44
3.3.3	Usabilidad / accesibilidad	45
4	Plan de negocio	46
4.1	Identificación de Clientes o Consumidores Potenciales	46
4.2	Ventajas y Desventajas Competitivas de la Plataforma	46
4.3	Gestión del Proceso y Responsabilidades	47
5	Autoevaluación y conclusiones	48
5.1	Valoración del trabajo y dificultades encontradas.....	48
5.2	Valoración de la herramienta o aplicación desarrollada	49
5.3	Conclusiones finales	49
6	Bibliografía.....	50
Ilustración 1 – Lichess Logo Ilustración 2 - ChessCom Logo		
Ilustración 3-Trofeos de Ajedrez.....		5
Ilustración 4 - React		7
Ilustración 5 - JavaScript		9
Ilustración 6 - PHP		10
Ilustración 7 - MySQL		11
Ilustración 8 - Visual Studio Code		12
Ilustración 9 - 000webhost.....		15
Ilustración 10 - Panel de Control		22
Ilustración 11 - Panel de archivos.....		22
Ilustración 12-Organigrama		26
Ilustración 13-Presupuesto		30
Ilustración 14-Cronograma		30
Ilustración 15 - Diagrama de Secuencia.....		32
Ilustración 16 - Enviar Solicitud		32
Ilustración 17 - Aceptar Solicitud.....		33
Ilustración 18 - Listado de Amigos		33
Ilustración 19 - Solicitud de Partida		35
Ilustración 20 - Listado de Partidas.....		35
Ilustración 21 - Realizar Movimiento		36
Ilustración 22 - Perfil y Estadísticas		36
Ilustración 23 - Jaque		37
Ilustración 24 - Entidad Relación		39
Ilustración 25 - Diagrama Entidad-Relación Extendido.....		41
Ilustración 26 - Menú Responsivo Ilustración 27 - Tablero Responsivo		45
Ilustración 28 - Diagrama de Gantt		49

Comentado [AM1]: El primer punto del índice sobra ChessGame

Comentado [AM2R1]: La introducción sería el 1Recursos el 2 etc.... No tienen que colgar de 1.1, 1.2 lo ves?

1 Introducción

Este proyecto tiene como objetivo crear una plataforma de ajedrez digital que se adapte a la era moderna, ofreciendo una experiencia única para los entusiastas del juego. En un mundo cada vez más conectado, el ajedrez por correspondencia, permite a los jugadores participar en partidas estratégicas a su propio ritmo y en cualquier lugar.

Para aquellos menos familiarizados con esta modalidad, se trata de una forma de jugar al ajedrez en la que los movimientos se realizan a través de medios digitales, como correo electrónico o aplicaciones, permitiendo a los jugadores tomarse el tiempo necesario para analizar sus movimientos y planificar estrategias.

Una de las características distintivas de esta plataforma es la flexibilidad de tiempo que se ofrece a los jugadores. Con plazos de movimiento de 2, 3 y 5 días, los usuarios tienen la libertad de jugar a su propio ritmo, adaptándose a sus horarios y preferencias.

Antes de adentrarnos más en los detalles de esta iniciativa, es crucial comprender el contexto en el que surge. En la era de la tecnología y la conectividad, plataformas como Lichess y Chess.com han revolucionado la forma en que el ajedrez se juega en línea. Lichess, por ejemplo, ha ganado reconocimiento por su enfoque en el software de código abierto y su compromiso con una experiencia de usuario limpia y sin publicidad. Por otro lado, Chess.com ha consolidado su posición como líder del mercado, ofreciendo una amplia gama de características, desde partidas en tiempo real hasta análisis de partidas, torneos y lecciones interactivas.



Ilustración 1 – Lichess Logo



Ilustración 2 - ChessCom Logo

Estas plataformas han demostrado que el ajedrez en línea puede ser tanto una experiencia competitiva como educativa, atrayendo a jugadores de todos los niveles y fomentando una comunidad global de entusiastas del juego. Su éxito es un testimonio de la demanda de experiencias de juego de calidad y la capacidad de la tecnología para conectar a las personas a través de su pasión compartida por el ajedrez.

Este proyecto surge en este contexto y nace de una idea que siempre rondó en mi cabeza: crear mi propia aplicación de ajedrez. Desde que era pequeño, el ajedrez ha sido una parte importante de mi vida, me ha ayudado en la constante toma de decisiones que es la vida. Cada partida, cada movimiento, ha sido un ejercicio de análisis y reflexión, enseñándome la importancia de evaluar situaciones, anticipar consecuencias y asumir las consecuencias de mis elecciones. Por eso, siempre quise llevar esa pasión al mundo digital. La oportunidad de desarrollar una plataforma para jugar al ajedrez es el deseo de crear algo propio, algo que refleje mi amor por el juego y mi visión de cómo debería ser la experiencia ideal para los jugadores de mi aplicación.

Como un tributo a mis inicios en este juego, comparto algunos de los trofeos que gané en mi trayectoria. Estos trofeos no son más que simples premios; representan los hitos y logros que me han llevado a embarcarme en crear una experiencia de ajedrez única y enriquecedora para los jugadores de todo el mundo.



Ilustración 3-Trofeos de Ajedrez

1.1 Objeto del proyecto

El proyecto consiste en el desarrollo de una plataforma de ajedrez por correspondencia, que permite a los usuarios jugar partidas de forma asincrónica a través de la web.

La jugabilidad asincrónica en una plataforma de ajedrez significa que los jugadores no necesitan estar en línea simultáneamente para jugar una partida. En lugar de eso, pueden realizar sus movimientos en su propio tiempo, con la flexibilidad de jugar cuando les resulte conveniente. Esto contrasta con las partidas en tiempo real, donde ambos jugadores deben estar activos al mismo tiempo para realizar sus movimientos.

La decisión de optar por la jugabilidad asincrónica se basa en varios factores. En primer lugar, permite a los jugadores disfrutar del juego sin la presión de tener que estar disponibles en un momento específico. Esto hace que sea más accesible para personas con horarios ocupados o que viven en diferentes franjas horarias. Además, la jugabilidad asincrónica fomenta una experiencia más reflexiva y estratégica, ya que los jugadores pueden tomarse su tiempo para analizar cuidadosamente sus movimientos y planificar su estrategia sin la presión del reloj.

En cuanto al objeto del proyecto, el desarrollo de una plataforma de ajedrez por correspondencia se presenta como una respuesta a la creciente demanda de juegos de mesa en línea y la conveniencia de acceder a ellos desde cualquier lugar y en cualquier momento. Esta plataforma se diferencia por su enfoque en la usabilidad y la experiencia del usuario, buscando ofrecer una interfaz intuitiva y un sistema de emparejamiento avanzado para mejorar la experiencia de juego. Además, la innovación en características como la herramienta de práctica podría destacarse en un mercado donde ya existen herramientas similares.

1.2 Lenguajes empleados

En el desarrollo de la plataforma de ajedrez por correspondencia, se ha realizado un análisis exhaustivo de los lenguajes y tecnologías disponibles para cada componente del proyecto, teniendo en cuenta criterios como rendimiento, escalabilidad, seguridad y mantenibilidad. A continuación, se detallan las razones específicas detrás de la elección de cada uno:

1.2.1 React



Ilustración 4 - React

React es un framework de JavaScript ampliamente utilizado en el desarrollo de interfaces de usuario. Se seleccionó para el desarrollo del frontend de la aplicación debido a las siguientes razones:

- **Componentización:** React se basa en el concepto de componentes reutilizables, lo que facilita la creación y el mantenimiento de una interfaz de usuario modular y escalable. Para conseguir estas características se sigue una metodología de diseño que fomenta la división de la interfaz en componentes pequeños y cohesivos. Cada componente se centra en una funcionalidad específica y puede ser utilizado en múltiples lugares de la aplicación sin necesidad de modificar su código interno. Esto facilita el mantenimiento del código, ya que los cambios realizados en un componente no afectan a otros componentes de la aplicación, lo que resulta en un desarrollo más eficiente y menos propenso a errores.
- **Virtual DOM:** Utiliza un DOM virtual para optimizar las actualizaciones de la interfaz de usuario, lo que resulta en un rendimiento más rápido y una experiencia de usuario más fluida. Esto se logra gracias a la eficiencia en la gestión de actualizaciones de la interfaz de usuario. Cuando ocurren cambios en los datos que afectan a la interfaz, React compara el Virtual

DOM con el DOM real y solo actualiza los elementos que han cambiado, en lugar de renderizar nuevamente toda la interfaz.

- Comunidad activa: Cuenta con una gran comunidad de desarrolladores y una amplia gama de bibliotecas y herramientas disponibles para facilitar el desarrollo. Dos ejemplos de comunidades de react son:
 - Reactiflux: Se trata de una comunidad en línea de desarrolladores de React que se reúnen en un servidor de Discord para discutir sobre temas relacionados con React, compartir conocimientos, hacer preguntas y colaborar en proyectos. Reactiflux es conocido por su ambiente acogedor y su disposición para ayudar a los desarrolladores de todos los niveles de experiencia.
 - Stack Overflow: Esta plataforma de preguntas y respuestas es un recurso invaluable para los desarrolladores de React que buscan soluciones a problemas específicos, consejos de mejores prácticas y orientación en el desarrollo de aplicaciones. La etiqueta [reactjs] en Stack Overflow cuenta con una gran cantidad de preguntas y respuestas útiles.
- Soporte de JSX: JSX, una extensión de sintaxis de JavaScript utilizada en React, permite escribir código HTML dentro de JavaScript, lo que facilita la creación de componentes UI de manera declarativa y legible.

1.2.2 JavaScript



Ilustración 5 - JavaScript

JavaScript es el lenguaje de programación principal utilizado en el desarrollo web y es esencial para la interactividad en el frontend y la comunicación con el backend. Es necesario para el desarrollo de la API del frontend por las siguientes características:

- **Amplio soporte:** JavaScript es compatible con todos los navegadores web modernos y es el lenguaje predeterminado en el desarrollo web. Esta compatibilidad no solo abarca los navegadores en computadoras de escritorio, sino también dispositivos móviles, lo que garantiza una experiencia consistente para los usuarios en una amplia gama de plataformas y dispositivos.
- **Asincronía:** JavaScript es conocido por su naturaleza asíncrona, lo que significa que puede ejecutar múltiples operaciones simultáneamente sin bloquear la ejecución del código. Esto es especialmente útil en el desarrollo web, donde las operaciones de red, como las solicitudes HTTP a un servidor, pueden llevar tiempo y podrían ralentizar la ejecución del código si se manejaran de manera síncrona.
- **JSON:** JavaScript Object Notation (JSON) es un formato de intercambio de datos ampliamente utilizado en las comunicaciones entre el frontend y el backend. JavaScript tiene soporte nativo para trabajar con JSON, lo que facilita el intercambio de datos estructurados entre las diferentes partes de la aplicación.

Comentado [AM3]: Vas a utilizar el framework de react o Javascript a pelo? Cual de los dos?

Comentado [MO4R3]: Utilizo react, pero para gestionar las llamadas del api del front uso javascript

- Librerías y frameworks: JavaScript cuenta con una amplia variedad de bibliotecas y frameworks, como Express.js y Axios, que simplifican el desarrollo de API y la comunicación entre frontend y backend.

1.2.3 PHP



Ilustración 6 - PHP

PHP es un lenguaje de programación de servidor ampliamente utilizado en el desarrollo web, especialmente para el backend de aplicaciones. Se optó por utilizar PHP para el desarrollo de la API del backend por estos motivos:

- Integración con MySQL: PHP tiene una estrecha integración con MySQL, ya que ambos surgieron en la década de 1990 y la combinación de ambas tecnologías se ha optimizado con el paso de los años lo que facilita la interacción con la base de datos y la gestión de datos de usuario, partidas y otros aspectos del juego.
- Escalabilidad: PHP es capaz de adaptarse fácilmente al crecimiento de la plataforma gracias a su flexibilidad y capacidad de escalamiento horizontal y vertical. Esto significa que, a medida que aumenta el número de usuarios y el volumen de datos, PHP puede ajustarse para satisfacer las demandas crecientes sin comprometer el rendimiento ni la estabilidad del sistema. Su arquitectura modular y su capacidad para trabajar con tecnologías de almacenamiento de datos distribuidas facilitan la expansión de la plataforma de manera eficiente y rentable.
- Rendimiento: PHP es conocido por su rendimiento confiable y su capacidad para manejar cargas de trabajo pesadas, lo que lo hace adecuado para aplicaciones web de alto tráfico como una plataforma de ajedrez. Su

Comentado [AM5]: No puedes repetir en cada una de las herramientas siempre esta coletilla cambia en una utiliza una en otra otra... etc

Comentado [AM6]: Si?? Cuenta mas

eficiencia en el procesamiento de solicitudes y su optimización para entornos web garantizan una experiencia de usuario fluida incluso en momentos de alta demanda, lo que contribuye a la satisfacción y retención de los usuarios.

1.2.4 MySQL



Ilustración 7 - MySQL

MySQL se escogió como la base de datos principal para la plataforma de ajedrez debido a su amplia adopción y reconocimiento en el ámbito del desarrollo web. MySQL es un sistema de gestión de bases de datos relacional, lo que significa que organiza los datos en tablas relacionadas entre sí, facilitando la consulta y manipulación de la información. Esta elección se basó en su fiabilidad, escalabilidad y compatibilidad con una variedad de plataformas y lenguajes de programación, lo que garantiza un almacenamiento seguro y eficiente de los datos de usuario, partidas de ajedrez y otros aspectos del juego.

- **Fiabilidad:** MySQL es conocido por su fiabilidad y estabilidad, lo que garantiza que los datos de los usuarios y las partidas de ajedrez se almacenen de forma segura y se puedan acceder de manera rápida y eficiente. Además, su robusto sistema de respaldo y recuperación de datos minimiza el riesgo de pérdida de información, asegurando la integridad de los datos a lo largo del tiempo.
- **Escalabilidad:** MySQL es altamente escalable, lo que significa que puede manejar grandes volúmenes de datos y un alto número de usuarios concurrentes sin sacrificar el rendimiento. Con capacidades de escalado

vertical y horizontal, MySQL puede adaptarse fácilmente al crecimiento de la plataforma, garantizando un servicio rápido y confiable incluso en momentos de alta demanda.

- **Compatibilidad:** MySQL es compatible con una amplia gama de plataformas y lenguajes de programación, incluidos PHP y JavaScript, lo que facilita su integración en el backend de la aplicación de ajedrez.

1.2.5 Visual Studio Code



Ilustración 8 - Visual Studio Code

Visual Studio Code se seleccionó como el entorno de desarrollo integrado (IDE) principal para el desarrollo de la aplicación debido a las siguientes particularidades:

- **Versatilidad:** Visual Studio Code es un IDE altamente versátil que admite una amplia gama de lenguajes de programación y tecnologías web, lo que lo hace adecuado para el desarrollo frontend y backend de la plataforma de ajedrez.
- **Extensibilidad:** Visual Studio Code cuenta con una amplia gama de extensiones disponibles que permiten personalizar y ampliar sus funcionalidades según las necesidades específicas del proyecto. Estas extensiones abarcan desde temas visuales hasta herramientas de productividad y soporte para diferentes lenguajes de programación, lo que ofrece a los desarrolladores una mayor flexibilidad para adaptar el entorno de desarrollo a sus preferencias y requisitos particulares.

- **Comunidad activa:** Actualmente, Visual Studio Code garantiza una comunidad activa de desarrolladores que contribuyen constantemente con nuevas extensiones y mejoras al ecosistema de desarrollo. Esta comunidad colaborativa hace que Visual Studio Code se mantenga en constante evolución, ofreciendo un entorno de desarrollo dinámico y adaptado a las necesidades cambiantes de los desarrolladores de software.
- **Colaboración:** Visual Studio Code ofrece herramientas integradas para la colaboración en equipo, como control de versiones integrado y soporte para servicios de colaboración en la nube como GitHub, lo que facilita el trabajo en equipo en el desarrollo de la aplicación de ajedrez.

1.3 Distribución

El despliegue de la aplicación web de ajedrez por correspondencia implica la distribución adecuada de los componentes necesarios para su funcionamiento. A continuación, se detallan las decisiones tomadas para configurar el entorno de distribución y garantizar su disponibilidad y accesibilidad:

Servidor web:

Se ha elegido Apache como servidor web principal debido a su amplia adopción, estabilidad y compatibilidad con PHP. Apache es capaz de gestionar eficientemente las solicitudes HTTP y servir el contenido estático y dinámico de la aplicación a los usuarios finales.

Versiones de PHP:

La elección de PHP versión 8.2 se basa en sus mejoras significativas en rendimiento, seguridad y nuevas características como soporte mejorado para tipos de datos inmutables, atributos de método no estáticos o la introducción del operador de tubería para ejecutar operaciones de forma secuencial en valores. Esta versión optimizada de PHP garantiza una experiencia de usuario más rápida y fluida, lo que contribuye a una distribución eficiente de la aplicación.

Gestión de la base de datos:

MySQL se ha seleccionado como sistema de gestión de bases de datos debido a su fiabilidad, escalabilidad y amplio soporte en la comunidad de desarrollo web. MySQL es capaz de manejar grandes volúmenes de datos de manera eficiente, lo que asegura una distribución fluida de la información entre los usuarios.

Ventajas de XAMPP y MySQL Workbench:

XAMPP ofrece una solución integral para el desarrollo y despliegue de aplicaciones web al incluir Apache, MySQL, PHP y Perl en un solo paquete de

Comentado [AM7]: No me queda claro para que utilizas xampp la verdad

Comentado [AM8R7]: ¿estás seguro que lo usas?

software. La facilidad de instalación y configuración de XAMPP simplifica el proceso de distribución de la aplicación, permitiendo a los desarrolladores implementar rápidamente un entorno de desarrollo o producción.

MySQL Workbench proporciona una interfaz gráfica intuitiva y potente para administrar bases de datos MySQL. Con características avanzadas para el diseño, modelado y monitoreo de bases de datos, MySQL Workbench facilita la gestión eficiente de la base de datos durante la distribución de la aplicación.

En resumen, la combinación de Apache, PHP, MySQL, XAMPP y MySQL Workbench ofrece un entorno de distribución robusto y eficiente para la aplicación web propuesta durante las etapas de desarrollo y pruebas. Aunque el despliegue final se haya realizado en un host externo, estas aplicaciones siguen siendo relevantes para garantizar el adecuado funcionamiento en el entorno local. Además, permitieron simular condiciones reales de producción para detectar posibles errores y optimizar el rendimiento.

Para la distribución global de la aplicación, se utiliza 000webhost, aprovechando sus servicios para alojar y ofrecer acceso a la aplicación desde cualquier parte del mundo.

Esta plataforma se seleccionó por su capacidad para alojar aplicaciones web de manera gratuita y ofrecer un acceso fácil y seguro desde cualquier parte del mundo. Además, 000webhost proporciona la posibilidad de subir y gestionar la base de datos asociada a la aplicación, lo que facilita el mantenimiento y la administración de los datos de los usuarios y las partidas de ajedrez.



Ilustración 9 - 000webhost

1.3.1 Requisitos de los clientes

Para acceder a la herramienta de ajedrez por correspondencia, los usuarios deben cumplir con ciertos requisitos mínimos tanto en sus dispositivos de escritorio como en sus dispositivos móviles. A continuación, se detallan estos requisitos:

Requisitos mínimos para dispositivos de escritorio:

1. Procesador y memoria RAM:

- Se recomienda un procesador Intel Core i3 o equivalente, o superior, para garantizar un rendimiento fluido de la aplicación.
- Se sugiere disponer de al menos 4GB de RAM para una experiencia de usuario óptima. Esto permite una ejecución eficiente de la aplicación y el sistema operativo sin ralentizaciones significativas.

2. Navegador web:

- Se requiere un navegador web moderno y actualizado que admita tecnologías web avanzadas como HTML5, CSS3 y JavaScript.
- Se recomienda utilizar preferiblemente Google Chrome, Mozilla Firefox o Microsoft Edge para garantizar la compatibilidad y el rendimiento óptimo de la aplicación.

3. Conexión a Internet:

- Es imprescindible disponer de una conexión a Internet estable y de buena calidad para acceder a la aplicación web de ajedrez por correspondencia.
- Se recomienda una conexión de banda ancha para evitar retrasos en la carga de la aplicación y en la comunicación con el servidor.

4. Habilitar cookies y JavaScript:

- El navegador web debe tener habilitadas las cookies y JavaScript para el correcto funcionamiento de la aplicación. Estas tecnologías son necesarias para almacenar datos de sesión y ejecutar el código JavaScript que controla la interactividad de la aplicación.

Requisitos mínimos para dispositivos móviles:

1. Sistema operativo:
 - Se requiere un dispositivo móvil con un sistema operativo actualizado, como Android o iOS, para garantizar la compatibilidad con la aplicación web.
2. Navegador web:
 - Se recomienda utilizar un navegador web móvil moderno y actualizado, como Google Chrome o Safari, que admita tecnologías web avanzadas para una experiencia de usuario óptima.
3. Conexión a Internet:
 - Al igual que en los dispositivos de escritorio, se necesita una conexión a Internet estable y de buena calidad para acceder a la aplicación web desde dispositivos móviles.
4. Pantalla táctil:
 - La aplicación debe ser compatible con dispositivos móviles con pantalla táctil para permitir una interacción intuitiva con los elementos de la interfaz de usuario.

Al cumplir con estos requisitos mínimos, los usuarios podrán acceder y disfrutar plenamente de la experiencia de juego del ajedrez por correspondencia en la aplicación web, ya sea desde sus dispositivos de escritorio o móviles.

Comentado [AM9]: Por tanto, si se necesita conexión a internet vas a publicarla! Interensante ver el dominio que utilizas y como gestionas la parte del back!

1.3.2 **Licenciamiento**

La herramienta se distribuirá bajo una licencia de código abierto, como la Licencia MIT, que permite a los usuarios utilizar, modificar y redistribuir el software de forma gratuita, siempre que se incluya el aviso de copyright y la renuncia de garantía. Por lo tanto, se planea publicar el proyecto en GitHub, ya que la licencia MIT va alineada con los principios de código abierto que tiene esta plataforma.

La Licencia MIT es una de las licencias de código abierto más populares y permisivas utilizadas en el mundo del desarrollo de software. Esta licencia permite a los usuarios utilizar, modificar y redistribuir el software de forma gratuita, siempre y cuando se incluyan ciertas condiciones. A continuación, se detallan las características principales de la Licencia MIT:

- **Libertad de uso:** La Licencia MIT permite a cualquier persona utilizar el software con cualquier propósito, ya sea comercial o no comercial. Los usuarios tienen la libertad de utilizar el software de la manera que consideren adecuada, sin restricciones adicionales.
- **Modificación y redistribución:** Los usuarios tienen la libertad de modificar el software y redistribuir las versiones modificadas de forma gratuita o comercial. Esta flexibilidad fomenta la colaboración y el intercambio de conocimientos en la comunidad de desarrollo de software.
- **Inclusión del aviso de copyright:** La única condición impuesta por la Licencia MIT es que se incluya el aviso de copyright y la renuncia de garantía en todas las copias o distribuciones del software. Esto asegura que los usuarios reconozcan la autoría del software y respeten los derechos de los desarrolladores originales.
- **Renuncia de garantía:** La Licencia MIT incluye una renuncia de garantía, que establece que el software se proporciona "tal cual", sin garantías de ningún tipo, ya sean expresas o implícitas. Los usuarios utilizan el software bajo su propio riesgo y los desarrolladores no asumen responsabilidad por posibles daños o errores.

Comentado [AM10]: Indica si el proyecto se subirá a GitHub de manera pública

Comentado [AM11R10]: Pero se planea o se ha subido???

Comentado [MO12R10]: Se planea subirlo después de la entrega

Comentado [AM13]: Te vuelve a pasar como en el punto: 1.1 **Lenguajes empleados**

- **Compatibilidad:** La Licencia MIT es compatible con otras licencias de código abierto y de software libre, lo que facilita la combinación y la integración del software con otros proyectos. Esta compatibilidad promueve la interoperabilidad y la colaboración entre diferentes proyectos de software.

2. Recursos

Los recursos son elementos fundamentales para garantizar el funcionamiento óptimo de la aplicación web de ajedrez por correspondencia. Desde el hardware necesario para alojar el servidor web hasta el almacenamiento y la capacidad de red. En esta sección, se detallan los recursos hardware necesarios, así como su configuración y distribución para satisfacer las demandas del proyecto. Además, se exploran las consideraciones clave para el despliegue de la aplicación en un entorno accesible para usuarios de todo el mundo.

2.1 Recursos Hardware

A continuación, se mostrará un listado con los recursos hardware necesarios para el correcto funcionamiento de la aplicación.

Servidor Web:

Para alojar la aplicación y gestionar el tráfico de usuarios concurrentes de manera eficiente, se requiere un servidor web con una capacidad adecuada. Esto implica considerar la capacidad de procesamiento, la memoria RAM y el ancho de banda necesario para manejar las solicitudes entrantes, ejecutar el código de la aplicación y enviar las respuestas a los clientes. Es esencial asegurar la escalabilidad del servidor para aumentar su capacidad según sea necesario con el crecimiento del número de usuarios y el tráfico.

- **Memoria RAM:** Se requerirá un mínimo de 8 GB de RAM para garantizar un rendimiento fluido incluso en momentos de alta carga.
- **Procesador:** Un procesador de al menos 4 núcleos y 3.0 GHz asegurará una rápida respuesta del servidor ante múltiples solicitudes simultáneas.

- Ancho de banda: Se deberá contar con una conexión de al menos 100 Mbps de ancho de banda para soportar el tráfico entrante y saliente de la aplicación.

Además, la configuración del servidor web debe optimizarse para garantizar un rendimiento óptimo y una alta disponibilidad de la aplicación. Esto incluye ajustes de seguridad, optimización de caché y compresión y configuración de respaldo y recuperación ante fallos.

- Ajustes de seguridad: Se aplicarán configuraciones de firewall y reglas de acceso para proteger el servidor contra posibles amenazas externas.
- Optimización de caché y compresión: Se configurarán técnicas de caché para mejorar el rendimiento y la velocidad de carga de la aplicación. Asimismo, se implementará la compresión de archivos para reducir el tamaño de los recursos enviados al navegador del usuario.
- Configuración de respaldo y recuperación ante fallos: Se establecerán rutinas de respaldo periódicas para garantizar la integridad de los datos. Además, se configurarán sistemas de recuperación ante fallos para minimizar el tiempo de inactividad en caso de incidentes inesperados.

Almacenamiento:

Se necesita un almacenamiento adecuado para la base de datos y los archivos estáticos de la aplicación. Aunque inicialmente el almacenamiento puede ser de un tamaño mínimo, se debe planificar su escalabilidad para agregar espacio a medida que crezca la cantidad de datos generados por los usuarios. En este caso, se optará por el almacenamiento de tipo SSD (Unidades de Estado Sólido) debido a sus tiempos de acceso más rápidos y su rendimiento generalmente mejorado en comparación con los HDD (Unidades de Disco Duro). Esta elección se hace teniendo en cuenta los requisitos de rendimiento y costo del proyecto, así como la necesidad de mantener una experiencia fluida para los usuarios.

- Se utilizarán unidades de estado sólido (SSD) para un acceso rápido a los datos y una alta disponibilidad.

- Se reservará un mínimo de 500 GB de espacio de almacenamiento inicial, con la posibilidad de escalar según sea necesario conforme crezca la base de usuarios y la cantidad de datos generados.

Conexión Red:

Además de la capacidad del servidor web, es crucial considerar el ancho de banda de red disponible para garantizar una conexión rápida y confiable entre el servidor y los clientes. Esto es especialmente importante para aplicaciones web en tiempo real, como los juegos de ajedrez por correspondencia, donde la comunicación entre el servidor y los clientes debe ser fluida y sin interrupciones. Se debe garantizar un ancho de banda suficiente para manejar el tráfico entrante y saliente de la aplicación, así como para soportar picos de tráfico durante períodos de alta actividad.

- Se asegurará un ancho de banda suficiente para manejar un promedio de 1000 peticiones por segundo, con capacidad de escalabilidad para manejar picos de tráfico durante períodos de alta actividad.

Para llevar a cabo la implementación de la aplicación y hacerla accesible a nivel global, se realiza el despliegue del servidor web en ooowebhost. Esto implica configurar y alojar la aplicación en sus servidores, lo que permite que la aplicación sea accesible a través de Internet. ooowebhost es un proveedor de servicios de hosting gratuito que ofrece recursos suficientes para alojar sitios web y aplicaciones de tamaño mediano sin costo alguno.

Para llevar a cabo el despliegue en ooowebhost, primero se debe crear una cuenta en su plataforma. Luego, se proporciona acceso a un panel de control donde se pueden gestionar los archivos y configuraciones de la aplicación. Se pueden cargar los archivos de la aplicación, incluidos los archivos HTML, CSS, JavaScript y PHP, en el caso de mi aplicación, tuve que compilar todo el proyecto de react y subir la carpeta Dist mediante el administrador de archivos en línea proporcionado por ooowebhost.

Además, para asegurar el funcionamiento completo de la aplicación, es necesario configurar y migrar la base de datos al servidor de ooowebhost. Esto implica exportar la base de datos desde el entorno de desarrollo local o de pruebas y luego importarla al servidor de ooowebhost. Una vez que la base de datos está disponible en el servidor, se deben actualizar las conexiones del API PHP para que apunten a la nueva ubicación de la base de datos y puedan interactuar

Comentado [AM14]: Esto me gusta que vas a utilizar??

Comentado [AM15R14]: Fijate que si utilizas esto, puedes indicar que el servidor web apache que vas a utilizar es para el propio desarrollo

Comentado [AM16R14]: Puedes mirarte el dominio a través de Namecheap

Comentado [AM17R14]: Banahosting y Cpanel
Para la parte del servidor la app

Comentado [AM18]: Ni una sola foto del panel o de como se configura, ni de la base de datos a donde la subes!

correctamente con ella. Este paso es crucial para garantizar que la aplicación pueda acceder y gestionar los datos de manera adecuada en el entorno de producción.

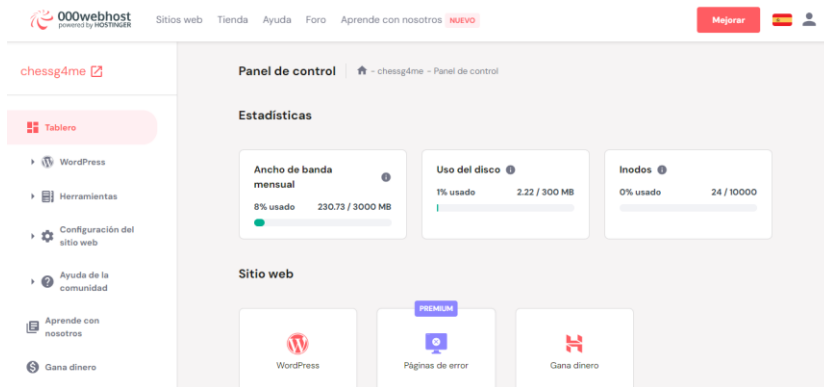


Ilustración 10 - Panel de Control

Una vez que los archivos de la aplicación están cargados en el servidor, se pueden configurar los detalles del dominio y DNS para asignar un nombre de dominio personalizado a la dirección IP del servidor. Esto se realiza mediante la modificación de los registros DNS en el panel de control de 000webhost o a través del proveedor de dominio externo.

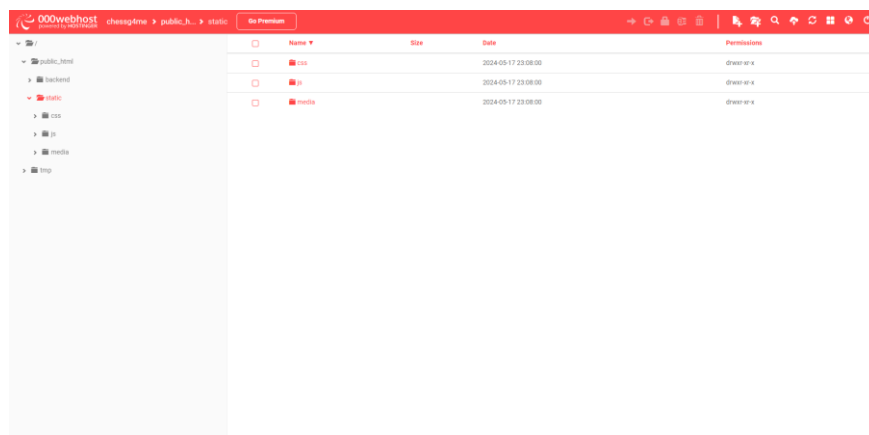


Ilustración 11 - Panel de archivos

También, para garantizar conexiones seguras mediante HTTPS, se puede gestionar la instalación de certificados SSL/TLS en el servidor. ooowebhost ofrece opciones para habilitar HTTPS de forma gratuita mediante la instalación de certificados SSL gratuitos, lo que proporciona seguridad y cifrado para las comunicaciones entre los usuarios y el servidor.

En resumen, la correcta planificación y configuración de los recursos hardware garantizará que el entorno de servidor esté correctamente dimensionado. Además, el despliegue en un dominio público permitirá que la aplicación sea accesible para todos los usuarios a nivel global.

2.2 Recursos Software

En este apartado, se verán los recursos software necesarios para el proceso de desarrollo de la aplicación.

Sistema Operativo:

Para el sistema operativo se considera el uso de Windows Server, dada la familiaridad y preferencia del equipo de desarrollo. Aunque el uso de un sistema basado en Linux como Ubuntu Server, CentOS o Debian también es viable, se prefiere Windows Server por su compatibilidad con las herramientas de desarrollo utilizadas y la comodidad del equipo con este entorno. Se reconoce que el uso de Windows Server puede implicar costos adicionales de licencia, los cuales se detallarán en otro apartado de la memoria.

Servidor Web:

Se opta por Apache como servidor web para alojar la aplicación PHP. Apache es ampliamente utilizado en entornos de desarrollo web debido a su flexibilidad, estabilidad y robustez. Además, cuenta con una gran comunidad de usuarios que proporciona soporte y recursos ampliamente disponibles. Su capacidad para manejar múltiples solicitudes simultáneas y su escalabilidad lo convierten en una opción sólida para el alojamiento de aplicaciones web.

Motor de Base de Datos:

Motor de base de datos MySQL: MySQL es un sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado en aplicaciones web.

Ofrece rendimiento, confiabilidad y facilidad de uso, lo que lo convierte en una opción popular para proyectos de desarrollo web. Se debe instalar y configurar MySQL en el servidor para almacenar los datos de la aplicación, como la información de los usuarios, las partidas de ajedrez y otras entidades relacionadas.

Herramientas de Desarrollo:

- **React:** React es una biblioteca de JavaScript de código abierto utilizada para construir interfaces de usuario interactivas y dinámicas. Se utiliza en el desarrollo del frontend de la aplicación para crear componentes reutilizables y gestionar el estado de la interfaz de usuario de manera eficiente.
- **JavaScript:** JavaScript es un lenguaje de programación ampliamente utilizado en el desarrollo web para agregar interactividad y funcionalidad a las páginas web. Se utiliza tanto en el frontend como en el backend de la aplicación para implementar la lógica del juego de ajedrez, la comunicación con el servidor y otras funcionalidades.
- **PHP:** PHP es un lenguaje de programación de servidor de código abierto utilizado para el desarrollo del backend de la aplicación. Se utiliza para interactuar con la base de datos, gestionar la autenticación de usuarios y proporcionar servicios web para el frontend de la aplicación.
- **MySQL Workbench:** MySQL Workbench es una herramienta de administración y desarrollo visual para MySQL. Se utiliza para diseñar, modelar, monitorear y administrar bases de datos MySQL. Facilita tareas como la creación de tablas, consultas SQL, gestión de usuarios y copias de seguridad de bases de datos.

Al tener en cuenta estos recursos software necesarios, se garantiza que el entorno de desarrollo y producción esté correctamente configurado para ejecutar la aplicación de manera eficiente y segura.

2.3 Humanos

En el contexto de un proyecto desarrollado por un único individuo, es importante reconocer las distintas responsabilidades y roles que normalmente serían ocupados por un equipo. Aunque en este caso específico todas estas funciones hayan sido desempeñadas por una sola persona, es útil tener en cuenta la estructura y las responsabilidades típicas de un equipo de desarrollo completo.

En este sentido, se podría indicar que la versión actual de la aplicación representa una versión beta o prototipo inicial, desarrollada por mi y que ha asumido múltiples roles, incluyendo el diseño frontend, desarrollo backend, diseño visual y gestión de la infraestructura. Sin embargo, para futuras iteraciones y mejoras de la aplicación, sería beneficioso contar con un equipo completo que incluya al menos:

- Un desarrollador frontend con experiencia en React, responsable de la interfaz visual, maquetación y manejo de llamadas del ApiService.
- Un desarrollador backend con experiencia en JavaScript, PHP y SQL encargado de desarrollar las APIs y establecer la estructura de la base de datos.
- Un diseñador encargado de desarrollar el aspecto estético de la aplicación, asegurando una experiencia visualmente atractiva y coherente tanto para ordenadores como dispositivos móviles.
- Un administrador de sistemas encargado de gestionar el servidor y la infraestructura de la herramienta, asegurando su escalabilidad y funcionalidad.

Además de estos roles, se incluyen los siguientes miembros en el equipo:

- Un manager o jefe de proyectos para supervisar y coordinar todas las actividades del equipo, asegurando el cumplimiento de los plazos y la calidad del producto final.
- Un consultor para brindar orientación experta sobre los requisitos del proyecto y garantizar su alineación con las necesidades del usuario.

Comentado [AM19]: •Planificación: en horas

- Diseño:
- Desarrollo:
- Pruebas:
- Despliegue y Mantenimiento:

Comentado [AM20]: Esto nooo piensa que tu eres el único que vas a desarrollar así que cumplirás tu todos esos papeles

Comentado [AM21R20]: Indica de alguna forma que es una versión beta y que en un futuro sería interesante contar con todo eso

Comentado [MO22R20]: Se agregan párrafos explicando el contexto

Comentado [AM23]: Bueeeeno venga si vas a crear un organigrama entonces sí, puedes decir que es una idea que va a ser desarrollada a pequeño nivel pero que alguna empresa podría "Comprar" la idea y entonces si que necesitas un equipo detrás

- Un analista de calidad responsable de realizar pruebas exhaustivas para garantizar que la aplicación cumpla con los estándares de calidad y funcione correctamente en diferentes escenarios.

Este equipo multidisciplinario garantiza una cobertura completa de las áreas clave del proyecto y promueve la colaboración entre diferentes especialidades para lograr un resultado exitoso.

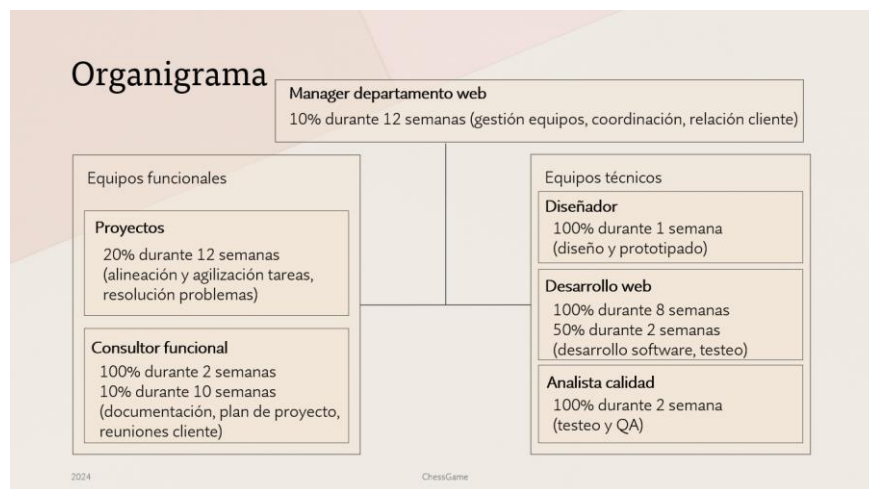


Ilustración 12-Organigrama

2.4 Previsión económica del coste del proyecto

Comentado [AM24]: Agrega listas de viñetas para servidor web, licencias de SÍ, herramientas de desarrollo...

En esta sección, se ofrece una visión detallada de los gastos anticipados asociados con el desarrollo, despliegue y mantenimiento de la aplicación. Desde los costos iniciales de desarrollo hasta los gastos recurrentes de alojamiento y gestión. En este apartado se proporciona una estimación económica y financiera.

Costos de Hardware

Estos costos están vinculados a la adquisición y mantenimiento de los componentes físicos necesarios para el desarrollo, despliegue y operación de la aplicación. Esto incluye la compra de servidores, computadoras, dispositivos de red, almacenamiento de datos, así como cualquier otro equipo necesario para respaldar la infraestructura tecnológica requerida.

- **Servidor Web:** Utilizando el servidor en la nube de 00owebhost, actualmente es 0 pero podría variar en un futuro dependiendo de la capacidad y recursos requeridos. Dentro de la versión gratuita de 00owebhost, se permite un máximo de 2000 peticiones HTTPS diarias. Es importante contar con un servidor web confiable y escalable para garantizar el funcionamiento óptimo de la aplicación en línea. Además del costo mensual del servidor en la nube, se deben considerar posibles costos adicionales relacionados con el almacenamiento de datos y el tráfico de red.

Comentado [AM25]: En que sentido? Menciona el número de conexiones simultáneas antes de que se caiga el server

Comentado [AM26R25]: Si es que se cae

Costos de Software

Estos costos están directamente relacionados con la adquisición, licenciamiento y mantenimiento de las herramientas y programas informáticos necesarios para el desarrollo, despliegue y operación de la aplicación.

- **Licencias de Software:** Dado que se asume el uso de software de código abierto, no hay costos directos asociados con las licencias de software. Esto permite reducir los costos operativos del proyecto y aprovechar herramientas de desarrollo de alta calidad sin incurrir en gastos adicionales.

- **Herramientas de Desarrollo:** Aunque muchas herramientas de desarrollo de código abierto son gratuitas, algunas pueden requerir suscripciones premium o complementos adicionales que pueden tener un costo mensual. En el caso de este proyecto, no se requerirá de ningún complemento adicional.


Costos de Recursos Humanos

Los costos humanos están relacionados con el personal necesario para llevar a cabo el desarrollo, despliegue y mantenimiento de la aplicación. Esto incluye los salarios y beneficios de los trabajadores.

Aunque actualmente la aplicación la he hecho yo solo, estos podrían ser los salarios estimados de un equipo de trabajo que llevaría un proyecto como este.

- **Manager:** Con 48 horas de trabajo a un precio de 60€ por hora, el total sería de 2880€. El manager desempeña un papel crucial en la coordinación y supervisión del equipo de desarrollo, asegurando que el proyecto avance según lo planeado y se alcancen los objetivos establecidos.
- **Responsable del Proyecto:** Con 96 horas de trabajo a un precio de 40€ por hora, el total sería de 3840€. El responsable del proyecto se encarga de gestionar los aspectos operativos y administrativos del proyecto, asegurando que se cumplan los plazos y se mantenga la comunicación efectiva entre los miembros del equipo.
- **Consultor:** Con 120 horas de trabajo a un precio de 30€ por hora, el total sería de 3600€. El consultor aporta experiencia y conocimientos especializados al proyecto.
- **Diseñador:** Con 40 horas de trabajo a un precio de 30€ por hora, el total sería de 1200€. El diseñador se encarga de crear una experiencia de usuario intuitiva y atractiva, asegurando que la interfaz de la aplicación sea fácil de usar y visualmente atractiva.
- **Desarrollador Senior Backend:** Con 360 horas de trabajo a un precio de 30€ por hora, el total sería de 10800€. El desarrollador senior backend se encarga de la implementación y optimización del backend de la aplicación, asegurando su funcionalidad y rendimiento.

Comentado [AM27]: Piensa que actualmente no dispones de un equipo y que te encuentras tu sólo

Comentado [AM28R27]: Y cuidadín con la IA.... 
Probability AI generated

Comentado [AM29R27]: Naaa no aportas nada dalo una vuelta veo mucha gente involucrada pero lo has hecho tu

Comentado [AM30R27]: Tendrá's que buscar de donde sacas estos precios? Porque 30€ la hora de donde sale? Alguna fuente?

Comentado [MO31R27]: Son sueldos aproximados, se añade el último párrafo para aclararlo.

- Desarrollador Junior Frontend: Con 180 horas de trabajo a un precio de 20€ por hora, el total sería de 7200€. El desarrollador junior colabora en el desarrollo del frontend de la aplicación, aprendiendo y adquiriendo experiencia bajo la supervisión del equipo.
- Analista de Calidad: Con 80 horas de trabajo a un precio de 20€ por hora, el total sería de 1600€. El analista de calidad se encarga de realizar pruebas exhaustivas para identificar y corregir posibles errores o problemas en la aplicación, garantizando su calidad y fiabilidad.

Los sueldos y tarifas mencionados en el ejemplo anterior son hipotéticos y pueden variar según diversos factores, como la ubicación geográfica, la experiencia y las habilidades del profesional, el tamaño y la complejidad del proyecto, entre otros.

Otros Costos:

Estos costos pueden incluir una variedad de gastos necesarios para el desarrollo y funcionamiento de la aplicación que no están directamente relacionados con hardware, software o personal. Estos pueden abarcar una amplia gama de áreas, como licencias, costos de alojamiento, servicios de seguridad, marketing o publicidad, así como cualquier otro gasto operativo necesario para mantener la aplicación en funcionamiento.

Costes Fijos (servidores, redes, licencias): Supondría aproximadamente un total de 2000€. Este costo cubre los gastos asociados con el mantenimiento y la operación de la infraestructura tecnológica necesaria para ejecutar la aplicación de manera eficiente y segura.

Analizando los costes totales del proyecto, se podría estimar un total 33120€ en el caso, nuevamente, de contar con un equipo completo. Este presupuesto incluiría todos los costos asociados con el desarrollo, implementación y mantenimiento del proyecto.

Comentado [AM32]: De donde sale este número

Presupuesto

	Horas trabajo	Precio/hora (€)	Total (€)
Manager	48	60	2880
Proyectos*	96	40	3840
Consultor funcional	120	30	3600
Diseñador	40	30	1200
Equipo desarrollo			
Desarrollador senior	360	30	10800
Desarrollador junior	180	20	7200
Analista calidad	80	20	1600
Costes fijos (servidores, redes)			2000
Total			33120

*Para un caso más realista se considera un solo responsable de proyectos

2024

ChessGame

Ilustración 13-Presupuesto

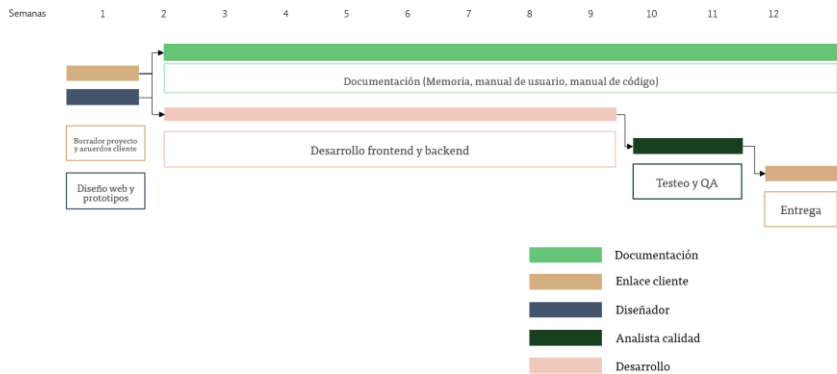
2.5 Cronograma

Esta diapositiva detalla la gestión del tiempo a grandes rasgos si el proyecto tuviese un cliente final. Para ver más detallada la planificación de este proyecto puede consultar el punto 5.1 Valoración del trabajo y dificultades encontradas

Comentado [AM33]: Este diagrama esta escaso de contenido, parte back parte front pero que haces en esas partes subdividelo en tareas más pequeñas

Comentado [AM34R33]: Además para cada uno de los entregables que hacías, realizabas pruebas! Y luego me lo enseñabas por tanto las pruebas no lo hacías al final del desarrllo

Cronograma



2024

ChessGame

Ilustración 14-Cronograma

3 Descripción de la aplicación

La aplicación es una plataforma diseñada para jugar partidas o mejorar las habilidades en el mundo del ajedrez. Los usuarios podrán registrarse, buscar o invitar a otros jugadores a partidas, y participar en partidas estratégicas a su propio ritmo. La aplicación permitirá a los jugadores gestionar múltiples partidas simultáneas, recibir notificaciones de los movimientos de sus oponentes. Además, se incluirán funcionalidades como la herramienta de prácticas, para mejorar el rendimiento del jugador o incluso poder jugar dos personas desde el mismo ordenador. Para una visualización más clara de cómo interactúan los usuarios con la aplicación y cómo se desarrollan las diferentes actividades, se utilizarán diagramas de casos de uso y diagramas de secuencia.

Comentado [AM35]: Esto no lo has dicho en la introducción??? Te repetess

3.1 Funcionamiento general

Comentado [AM36]: Buen punto en general

Para comenzar a explicar el funcionamiento general de la aplicación, empezaré comentando la sección de registro donde los usuarios podrán crear una cuenta en la plataforma para acceder a todas las funcionalidades disponibles. Este proceso de registro suele ser el primer paso para comenzar a utilizar la aplicación y permite a los usuarios personalizar su experiencia según sus preferencias y necesidades.

Durante el proceso de registro, los usuarios serán solicitados a proporcionar ciertos datos básicos, tales como nombre de usuario, dirección de correo electrónico y contraseña. Estos datos serán utilizados para crear y administrar la cuenta del usuario en la plataforma. Además de estos datos esenciales, se requerirá a los usuarios que suban una foto de avatar, que será utilizada para personalizar su perfil y mejorar la interacción dentro de la aplicación. Dependiendo de la aplicación, también se pueden solicitar otros detalles opcionales, como información de perfil adicional o preferencias de notificación.

Completado el proceso de registro, los usuarios podrán acceder a su cuenta utilizando las credenciales proporcionadas y comenzar a utilizar todas las funcionalidades de la aplicación. Es importante que el proceso de registro sea intuitivo, seguro y eficiente para garantizar una buena experiencia de usuario y fomentar la participación en la plataforma.

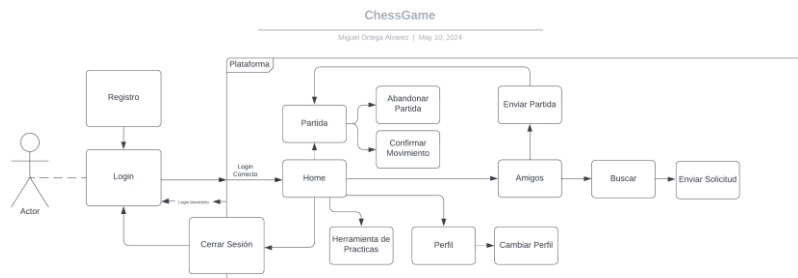


Ilustración 15 - Diagrama de Secuencia

Una vez que el usuario ha terminado exitosamente el proceso de registro y ha iniciado sesión en la aplicación, será dirigido automáticamente a la página principal, representada por la ruta `"/home"`. Aquí, se encontrará con una interfaz intuitiva y funcional que le permitirá navegar por las diferentes secciones y funcionalidades de la aplicación.

Antes de sumergirse en la emoción de las partidas de ajedrez, el usuario tendrá la oportunidad de explorar y expandir su red de contactos a través del apartado de amigos, accesible mediante la ruta `"/friends"`. Aquí, podrá buscar el nombre del usuario y agregarle para establecer una relación. Para hacerlo, simplemente seleccionará el botón correspondiente para enviar una solicitud de amistad. Esta función fomenta la interacción social dentro de la comunidad de la aplicación y facilita la organización de partidas con otros usuarios.



Ilustración 16 - Enviar Solicitud

Una vez que el usuario ha enviado una solicitud de amistad, el destinatario tendrá la opción de aceptarla o rechazarla. En caso de aceptación, ambos usuarios se agregarán mutuamente a sus listas de amigos, lo que les permitirá disfrutar de una comunicación más fluida y una experiencia de juego más enriquecedora.



Ilustración 17 - Aceptar Solicitud



Ilustración 18 - Listado de Amigos

Con la lista de amigos configurada, el usuario estará listo para sumergirse en el emocionante mundo de las partidas de ajedrez.

Una característica fundamental de la experiencia de usuario en la aplicación es la capacidad de los usuarios para desafiar y participar en partidas de ajedrez entre ellos. Esta funcionalidad se facilita a través de un sistema de solicitud de partida, que permite a los usuarios enviar invitaciones personalizadas a otros jugadores para iniciar una nueva partida.

Al acceder al apartado de amigos, los usuarios podrán seleccionar a quién desean desafiar y configurar los detalles específicos de la partida. Entre las opciones disponibles, los usuarios podrán establecer el número de días por turno, permitiendo una mayor flexibilidad en la velocidad del juego y adaptándose así a diferentes horarios y preferencias de los jugadores.

Además, los usuarios podrán elegir el color de sus piezas, lo que les permite personalizar aún más su experiencia de juego y estrategia. Esta elección puede influir en la dinámica del juego y agregar un elemento adicional de emoción y desafío a la partida.

Otro aspecto importante al enviar una solicitud de partida es la opción de clasificar la partida o mantenerla como no clasificada. Las partidas clasificadas pueden afectar el ranking y la puntuación del jugador en la aplicación, lo que añade un elemento competitivo y motivador a la experiencia. Por otro lado, las partidas no clasificadas son ideales para prácticas informales, aprendizaje o simplemente disfrutar del juego sin preocupaciones de puntuación.

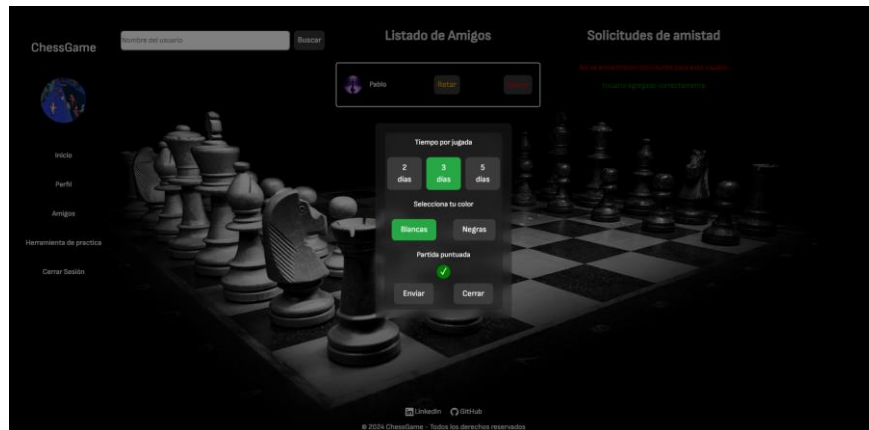


Ilustración 19 - Solicitud de Partida

Al regresar a la página principal ("/home"), podrá ver el listado de partidas activas, incluidas aquellas iniciadas por amigos aceptados. Esto le proporcionará una visión rápida y conveniente de las partidas en curso, lo que le permitirá seleccionar fácilmente la que desea jugar, seguir o abandonar.



Ilustración 20 - Listado de Partidas

Una vez accedemos a una partida, podremos realizar un movimiento si es nuestro turno y posteriormente confirmarlo. También podremos abandonar la partida en caso de que no se quiera jugar y esta no contara como derrota si la partida no alcanza mas de los dos movimientos.

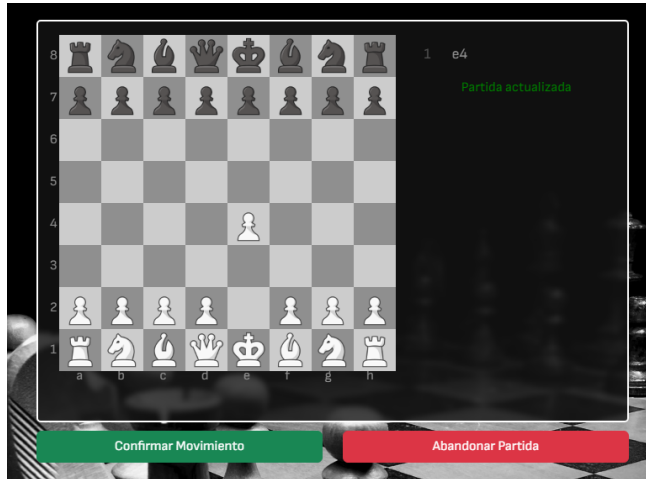


Ilustración 21 - Realizar Movimiento

Además de la función de desafiar a otros jugadores, la aplicación ofrece a los usuarios la posibilidad de gestionar su perfil y explorar estadísticas detalladas sobre su rendimiento en el ajedrez. Este apartado, accesible desde el apartado del perfil, permite a los usuarios personalizar y actualizar la información de su perfil, como nombre, correo o contraseña, además de estar al día de las estadísticas personales.

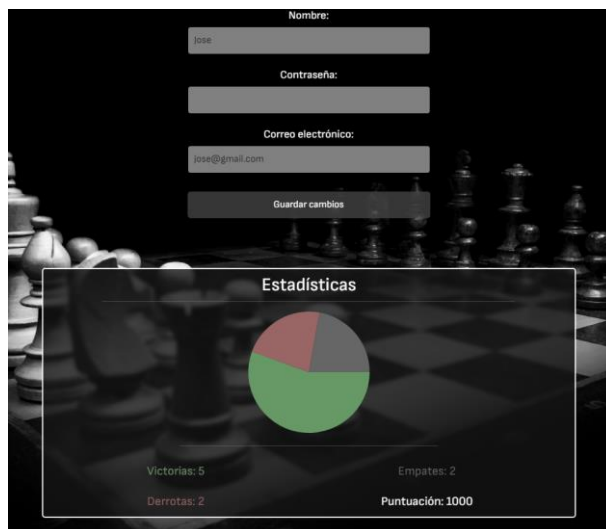


Ilustración 22 - Perfil y Estadísticas

Comentado [AM37]: Esta foto en una página sin ningún texto mira a ver como lo haces pero no puede quedar aquí una foto sin naaa

Por último, el usuario dispondrá de una herramienta de practicas para mejorar sus aptitudes, probar estrategias, mejorar sus movimientos o incluso poder jugar en local con otro jugador. Además, en esta herramienta se encuentran facilidades como la posibilidad de retroceder los movimientos, los movimientos disponibles de cada pieza o la casilla marcada al hacer jaque.

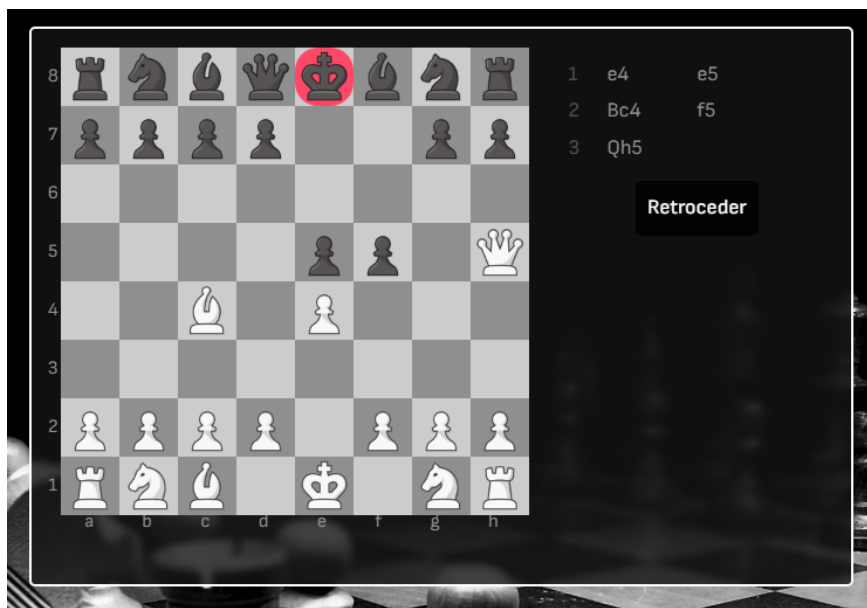


Ilustración 23 - Jaque

En resumen, la experiencia de usuario en la aplicación se ve enriquecida por la integración de funciones sociales, funciones de accesibilidad o la capacidad de agregar amigos y organizar partidas con ellos. Esta dinámica y fluida interacción entre usuarios crea un ambiente comunitario y promueve la participación en la aplicación.

3.2 Arquitectura

La arquitectura de la aplicación se basa en una estructura de cliente-servidor, donde el cliente, representado por la interfaz de usuario desarrollada en React, interactúa con un servidor backend que gestiona la lógica de la aplicación y la comunicación con la base de datos.

- **Interfaz de Usuario (Cliente):** La interfaz de usuario está desarrollada utilizando React, lo que permite crear una experiencia de usuario dinámica e interactiva. Está compuesta por una serie de componentes modulares que representan diferentes partes de la aplicación, como la página de inicio, el perfil del usuario o la lista de amigos.
- **APIServices:** Para manejar las solicitudes del frontend al backend, se implementa una API del frontend que actúa como intermediario entre la interfaz de usuario y el servidor backend. Esta API se encarga de enviar solicitudes HTTP al servidor backend y procesar las respuestas recibidas para actualizar la interfaz de usuario en consecuencia.
- **Servidor Backend:** El servidor backend está diseñado para manejar las solicitudes del cliente, procesar la lógica empresarial y comunicarse con la base de datos. Está construido utilizando un marco como Express.js en Node.js para proporcionar una arquitectura de servidor escalable y eficiente.
- **API del Backend:** Para gestionar las solicitudes del cliente y dirigir las adecuadamente, se implementa una API del backend que define endpoints para cada operación requerida por la aplicación. Esta API se encarga de recibir las solicitudes del cliente, procesarlas según el endpoint y devolver las respuestas correspondientes.
- **Base de Datos:** La base de datos almacena todos los datos relevantes para la aplicación, como la información del usuario, las partidas de ajedrez y las solicitudes de amistad. Se utiliza un sistema de gestión de bases de datos relacional como MySQL para garantizar la integridad y la consistencia de los datos.

En conjunto, esta arquitectura permite una separación clara de responsabilidades entre el cliente y el servidor, lo que facilita el desarrollo, la escalabilidad y el mantenimiento de la aplicación. Los componentes modulares y la comunicación fluida entre el frontend y el backend garantizan una experiencia de usuario receptiva y eficiente.

3.2.1 Diseño de la base de datos

El diseño de las bases de datos es un componente crucial en el desarrollo de cualquier aplicación, ya que define la estructura y la organización de los datos que la aplicación manejará. En el caso de nuestra aplicación, hemos diseñado cuidadosamente las tablas de la base de datos para garantizar la eficiencia, integridad y escalabilidad del sistema.

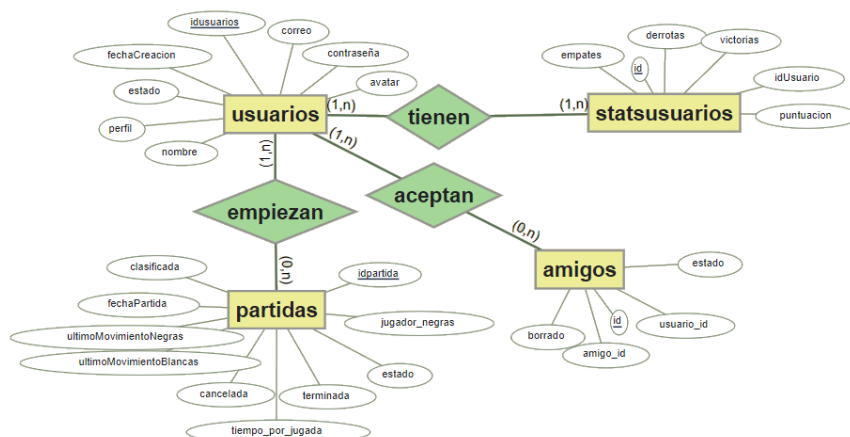


Ilustración 24 - Entidad Relación

A continuación, presento las entidades principales de la base de datos:

- **Usuarios:** Esta tabla almacena la información de cada usuario registrado en la aplicación, incluidos detalles como nombre de usuario, dirección de correo electrónico, contraseña (o hash de contraseña) y cualquier otra información relevante del perfil.

- **StatsUsuarios:** En esta tabla se registran las estadísticas y métricas asociadas a cada usuario, como su puntuación, número de victorias, empates, derrotas y cualquier otra información relacionada con el rendimiento y la actividad del usuario en la aplicación.
- **Amigos:** La tabla de amigos registra las relaciones de amistad entre los usuarios de la aplicación. Cada entrada en esta tabla representa una conexión entre dos usuarios que se han agregado mutuamente como amigos.
- **Partidas:** Finalmente, la tabla de partidas almacena información sobre las partidas de ajedrez jugadas en la aplicación. Esto puede incluir detalles como los jugadores involucrados, el estado de la partida, el historial de movimientos y cualquier otro dato relevante para la gestión y seguimiento de las partidas.

En la base de datos de la aplicación, existen varias relaciones que reflejan la interacción entre diferentes entidades.

1. Relación entre Usuarios y Partidas:

- Un usuario puede iniciar cero o varias partidas.
- Una partida es iniciada por un usuario.

2. Relación entre Usuarios y Amigos:

- Un amigo puede tener a uno o varios usuarios como amigos.
- Un usuario puede tener cero o varios amigos.

3. Relación entre Usuarios y Estadísticas:

- Un usuario puede tener cero o varias estadísticas asociadas.
- Una estadística puede estar asociada a uno o varios usuarios.

Estas relaciones reflejan cómo los usuarios interactúan entre sí y con otros elementos de la aplicación, como las partidas de ajedrez o las estadísticas. Al establecer y gestionar estas relaciones de manera adecuada en la base de datos, podemos garantizar un funcionamiento coherente y eficiente de nuestra aplicación.

Comentado [AM38]: Agrega las claves primarias que tienes y foráneas

Un Diagrama Entidad-Relación Extendido (EER) es una herramienta fundamental en el diseño de bases de datos, ya que permite representar de manera gráfica y detallada las entidades, sus atributos y las relaciones entre ellas. A continuación, se describen las principales entidades de nuestra plataforma de ajedrez por correspondencia, junto con sus claves primarias y foráneas.

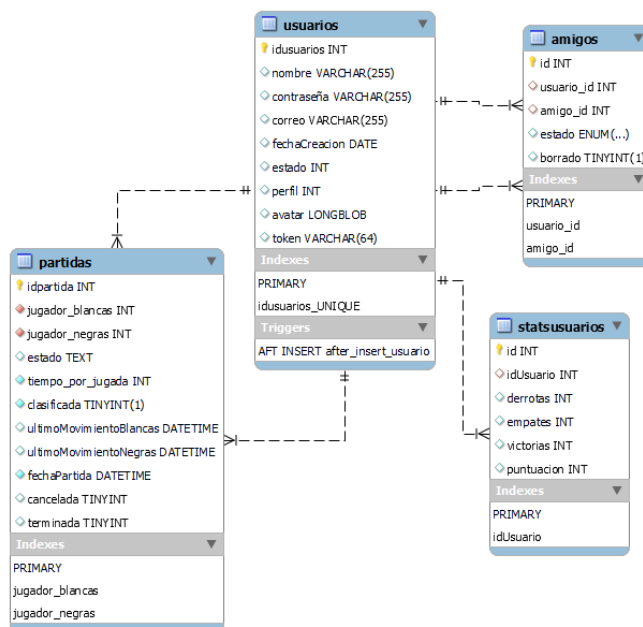


Ilustración 25 - Diagrama Entidad-Relación Extendido

En la tabla usuarios, la clave primaria es idusuarios, este campo es único para cada usuario y sirve como identificador principal en la tabla.

En la tabla partidas, la clave primaria es idpartida y las claves foráneas son jugador_blanco, que referencia al campo idusuarios en la tabla usuarios,

indicando el usuario que juega con las piezas blancas y jugador_negras, que referencia al usuario que juega con las piezas negras.

En la tabla stats_usuarios, la clave primaria es el id, que sirve como identificador único de cada registro de estadísticas y la clave foránea es el idusuario, que referencia al campo idusuarios en la tabla usuarios, vinculando las estadísticas con el usuario correspondiente.

Por último, en la tabla amigos, la clave primaria es el id que hace la función de identificador único de cada registro de amistad y las claves foráneas son el usuario_id que indica el usuario que envía la solicitud de amistad y el amigo_id, que indica el usuario que recibe la solicitud de amistad.

Con este diseño de base de datos, se busca proporcionar una estructura sólida y que respalde todas las funcionalidades de la aplicación y facilite el acceso y la manipulación de los datos por parte del sistema.

Comentado [AM39]: Agrega un diagrama de clases (te lo da el propio workbench)

3.3 Interfaz

En esta sección, nos adentraremos en la interfaz de usuario de la aplicación, centrándonos en sus características generales, diseño y navegación. La interfaz de usuario es fundamental para proporcionar una experiencia atractiva y funcional a los usuarios, por lo que es importante describir sus aspectos principales.

3.3.1 Características generales

Los aspectos claves de la aplicación en cuanto al diseño y a la interfaz, podrían ser los siguientes:

- El diseño se ha enfocado principalmente en tonos oscuros y clásicos, inspirados en la naturaleza atemporal del juego de ajedrez, aunque también se incluyan colores de contraste. Esta paleta de colores no solo evoca la sensación de tradición y elegancia asociada con el ajedrez, sino que también garantiza una experiencia visual agradable y relajante para los usuarios. Los tonos oscuros proporcionan un fondo neutral que resalta los elementos clave del tablero y los controles de la aplicación.

Comentado [AM40]: SI bueno,,, y luego me metes un color rojo y un color verde!!!

- **Navegación fluida:** Se ha prestado especial atención a la navegación, asegurando que los usuarios puedan moverse de manera intuitiva entre las diferentes secciones de la aplicación y acceder rápidamente a las funciones principales. La disposición de los elementos se ha optimizado cuidadosamente para maximizar el espacio disponible y proporcionar una experiencia de usuario cómoda y eficiente.
- **Lenguajes empleados y validación:** La interfaz se ha desarrollado utilizando tecnologías modernas como HTML, CSS y JavaScript, con un enfoque en estándares de codificación limpios y eficientes. Se han implementado transiciones suaves y animaciones sutiles en toda la aplicación para mejorar la interacción del usuario. Además, se han incorporado mecanismos de validación para garantizar la integridad y precisión de los datos ingresados por los usuarios.
- **Se ha puesto un énfasis particular en la legibilidad y la usabilidad de la interfaz.** Los elementos de la interfaz, como los botones y los textos, se han diseñado con tipografías claras y de fácil lectura, lo que garantiza que los usuarios puedan comprender fácilmente la información presentada y realizar acciones sin dificultad. Además, se han utilizado contrastes adecuados para mejorar la visibilidad y la accesibilidad de la aplicación, especialmente en diferentes tamaños de pantalla y condiciones de iluminación.
- **La disposición de los elementos se ha optimizado cuidadosamente para maximizar el espacio disponible y proporcionar una experiencia de usuario cómoda y eficiente.** Se han evitado las interfaces sobrecargadas y se ha priorizado la distribución ordenada de los elementos para evitar confusiones y facilitar la navegación intuitiva. Además, se ha buscado mantener una coherencia visual en toda la aplicación, lo que ayuda a los usuarios a familiarizarse rápidamente con la estructura y el funcionamiento de esta.
- **Para mejorar la interacción del usuario, se han implementado transiciones suaves y animaciones sutiles en toda la aplicación.** Estas animaciones no solo agregan un toque de elegancia y dinamismo al diseño, sino que también ayudan a guiar al usuario a través de diferentes acciones y estados de la aplicación de manera natural y fluida. Desde las transiciones entre pantallas hasta los efectos de desplazamiento y cambio de estado, cada animación se ha diseñado cuidadosamente para mejorar la experiencia del usuario y hacer que la interacción con la aplicación sea más placentera y envolvente.

- La validación de datos se realiza en tiempo real siempre que sea posible, lo que significa que los usuarios reciben retroalimentación inmediata sobre cualquier error o información incorrecta ingresada. Esto ayuda a prevenir errores y garantiza que los usuarios puedan completar sus tareas de manera eficiente y sin frustraciones. Además, se han incorporado mensajes de error claros y descriptivos para guiar a los usuarios sobre cómo corregir los problemas detectados y continuar con su interacción en la aplicación de manera efectiva.

3.3.2 Adaptación a dispositivos móviles

En este apartado, se detalla la adaptación de la interfaz de usuario de la aplicación de ajedrez para dispositivos móviles. La descripción incluye cómo se ha diseñado y optimizado la interfaz para otorgar una experiencia fluida y funcional en pantallas más pequeñas, así como las consideraciones específicas tomadas en cuenta para la navegación y usabilidad en dispositivos móviles.

En respuesta a la creciente prevalencia de dispositivos móviles, la adaptación de la interfaz de usuario se ha convertido en una prioridad para garantizar una experiencia óptima para los usuarios en todas las plataformas. En el caso de la aplicación de ajedrez, he dedicado esfuerzos significativos para diseñar una interfaz que se adapte correctamente a pantallas más pequeñas, como las de los móviles.

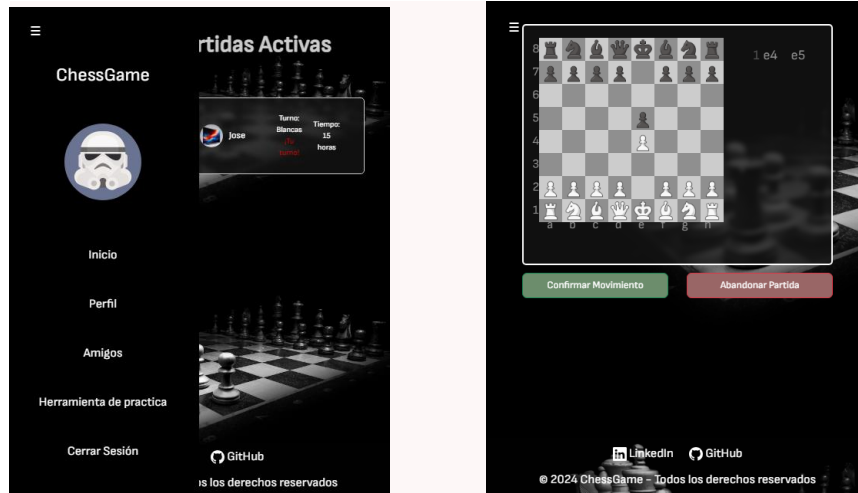


Ilustración 26 - Menú Responsivo

Ilustración 27 - Tablero Responsivo

La adaptación a dispositivos móviles se ha logrado mediante el diseño de una interfaz responsiva que se ajusta dinámicamente al tamaño y orientación de la pantalla del dispositivo. Se han optimizado elementos como botones, menús desplegables y campos de entrada para garantizar que sean fácilmente accesibles y utilizables en pantallas táctiles. Además, se han realizado pruebas exhaustivas en una variedad de dispositivos móviles y tamaños de pantalla para asegurar una experiencia consistente y satisfactoria para todos los usuarios, independientemente del dispositivo que utilicen.

3.3.3 Usabilidad / accesibilidad

En este apartado, exploraremos las características introducidas en la aplicación para mejorar la accesibilidad y la usabilidad en sus diversos aspectos. En un esfuerzo por garantizar que la plataforma sea accesible para todos los usuarios, he implementado una serie de características y prácticas de diseño que facilitan la navegación y la interacción, especialmente para aquellos con necesidades especiales o limitaciones físicas. Además, he priorizado la usabilidad en cada etapa del proceso de diseño, con el objetivo de ofrecer una experiencia fluida e intuitiva para todos los usuarios, independientemente de su nivel de habilidad o experiencia previa en el juego.

Además, en la visualización del tablero, he incorporado colores distintivos para resaltar situaciones importantes, como el jaque, lo que facilita que los jugadores

Comentado [AM41]: Texto texto, pero a ver como queda enseña pantallazos de si es responsive!! Eso es lo que hay que ver y si al ahcerse responsive quita algún menú y lo pone en hamburguesa esas cosas tienes que explicar tambien

identifiquen rápidamente las posiciones críticas durante el juego. También he mejorado la experiencia de seguimiento al agregar indicadores visuales que muestran los movimientos posibles de cada pieza, lo que permite a los usuarios planificar estrategias con mayor precisión y comprender mejor el estado actual del juego.

4 Plan de negocio

El plan de negocio en caso de que la aplicación saliese a la venta necesitaría entre otras cosas la identificación y captación de clientes potenciales, destacando ante todo las ventajas competitivas y estableciendo un sólido marco de gestión para garantizar el éxito de la aplicación. Con un enfoque claro en las necesidades del cliente, la diferenciación competitiva y una ejecución eficiente, se alcanzan los objetivos comerciales y se satisfacen las expectativas de los clientes.

4.1 Identificación de Clientes o Consumidores Potenciales

Esta plataforma de ajedrez está dirigida a una amplia gama de usuarios, desde entusiastas del ajedrez hasta jugadores casuales y competitivos. Entre los clientes potenciales se encuentran:

- Jugadores de ajedrez de todas las edades y niveles de habilidad.
- Escuelas, clubes y organizaciones que buscan herramientas para enseñar y promover el ajedrez.
- Entrenadores y profesionales del ajedrez que buscan una plataforma para gestionar partidas y torneos.
- Aficionados al ajedrez que desean mejorar sus habilidades y participar en una comunidad activa.

4.2 Ventajas y Desventajas Competitivas de la Plataforma

En esta sección, se evaluarán las ventajas y desventajas competitivas de la aplicación en comparación con los de la competencia. Se realizará un análisis

detallado del mercado, identificando a nuestros principales competidores y sus fortalezas y debilidades. Se destacarán las características únicas y los beneficios diferenciadores de la plataforma.

1. Ventajas:

- Interfaz intuitiva y atractiva que facilita la experiencia del usuario.
- Amplia gama de funciones, incluyendo partidas en línea, tutoriales, análisis de partidas y torneos.
- Sistema de emparejamiento inteligente que garantiza partidas equilibradas y desafiantes.
- Integración con redes sociales y capacidad de compartir partidas y logros.
- Servicio de soporte al cliente receptivo y eficiente.

2. Desventajas:

- Competencia en un mercado saturado con varias plataformas de ajedrez en línea.
- Necesidad de una estrategia de marketing sólida para destacar en un mercado competitivo.
- Requerimientos de desarrollo continuo para mantenerse al día con las demandas y expectativas de los usuarios.
- Posibles desafíos en la monetización del servicio, especialmente en un mercado donde muchas plataformas ofrecen funciones gratuitas.

4.3 Gestión del Proceso y Responsabilidades

La gestión del proceso de desarrollo, comercialización y operación de la plataforma estará a cargo del equipo de desarrollo, liderado por el fundador y desarrollador principal. Las responsabilidades incluirían:

- Desarrollo y mantenimiento continuo de la plataforma, incluyendo actualizaciones de características y correcciones de errores.
- Estrategias de marketing y promoción para aumentar la visibilidad y la base de usuarios.
- Gestión de la comunidad y soporte al cliente para garantizar la satisfacción del usuario y fomentar la retención.
- Monitoreo del rendimiento de la plataforma y análisis de datos para identificar áreas de mejora y oportunidades de crecimiento.
- Colaboraciones con otras plataformas y organizaciones de ajedrez para ampliar el alcance y ofrecer beneficios adicionales a los usuarios.

Comentado [AM42]: Pon alguna más

5 Autoevaluación y conclusiones

En este apartado de autoevaluación y conclusiones, reflexionaré sobre el proceso de desarrollo del proyecto, evaluando los desafíos encontrados, la valoración de la herramienta creada y las lecciones aprendidas durante el proceso. Además, ofreceré conclusiones sobre el diseño, la implementación y el tiempo empleado en el proyecto, destacando los conocimientos adquiridos y su impacto en mi desarrollo profesional como desarrollador de software.

5.1 Valoración del trabajo y dificultades encontradas

Durante el proceso de desarrollo de este proyecto, me enfrenté a una serie de desafíos que contribuyeron significativamente a mi aprendizaje y crecimiento como desarrollador. Uno de los principales obstáculos que encontré fue la gestión eficiente del tiempo. Con un plazo de apenas 12 semanas para completar el proyecto, fue crucial aprender a administrar mis recursos y priorizar tareas de manera efectiva. Esta limitación temporal me obligó a desarrollar habilidades sólidas en planificación y organización, aspectos fundamentales en cualquier entorno laboral.

También, una de las dificultades que surgieron durante el desarrollo fue la gestión del estado de las partidas y su recuperación desde la base de datos. Esta tarea resultó desafiante debido a la necesidad de mantener una sincronización precisa entre el estado actual del juego y su representación en la base de datos. Superar este obstáculo implicó un análisis exhaustivo de la lógica de la aplicación y la implementación de estrategias eficientes para garantizar la integridad y

coherencia de los datos. Aunque fue un desafío significativo, el proceso me otorgó una valiosa experiencia en el diseño de sistemas robustos y la resolución de problemas complejos en el desarrollo de aplicaciones web.

Para la construcción de la plataforma, he elaborado un diagrama de Gantt que detalla las tareas realizadas semana a semana. Este diagrama proporciona una visión clara y estructurada del proyecto.

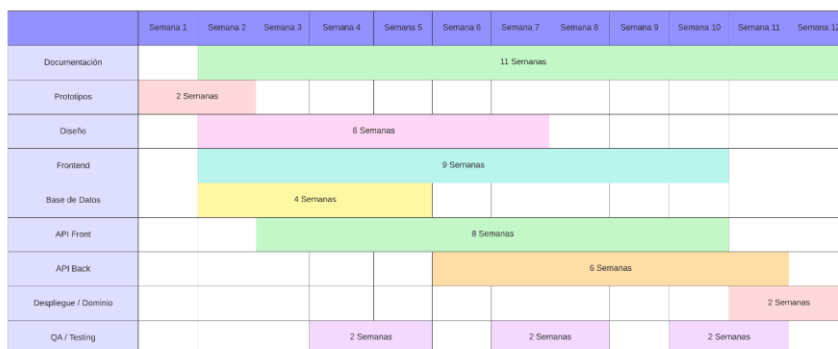


Ilustración 28 - Diagrama de Gantt

Comentado [AM43]: Esto que hace aquí?

5.2 Valoración de la herramienta o aplicación desarrollada

En cuanto a la valoración de la herramienta desarrollada, considero que he logrado crear una aplicación funcional y satisfactoria que cumple con los objetivos establecidos inicialmente. Sin embargo, siempre hay espacio para mejorar y expandir las funcionalidades ofrecidas. Creo que, con más tiempo, tendría la oportunidad de implementar características adicionales que enriquecerían aún más la experiencia del usuario. Por ejemplo, podría explorar la inclusión de salas de chat entre jugadores para fomentar la interacción social, así como un sistema de notificaciones para recordatorios de turnos, lo que mejoraría la experiencia de juego y la retención de usuarios entre otras cosas.

Comentado [AM44]: Perfecto

5.3 Conclusiones finales

En términos de conocimientos adquiridos, este proyecto ha sido una experiencia significativa para mi crecimiento profesional. He aprendido sobre el desarrollo de aplicaciones web utilizando tecnologías modernas como React, así como sobre

conceptos fundamentales como la componentización y la gestión de datos en el frontend y el backend. Además, he mejorado mis habilidades en diseño de interfaces, desarrollo de API y despliegue de aplicaciones en entornos de producción. En resumen, este proyecto me ha otorgado una base sólida de conocimientos y habilidades que sin duda me prepararán para futuros desafíos en mi carrera profesional como desarrollador web.

6 Bibliografía

ooowebhost. (s.f.). Obtenido de <https://www.ooowebhost.com>

Base64. (s.f.). Obtenido de <https://base64.guru>

Bootstrap. (s.f.). Obtenido de <https://getbootstrap.com>

ChatGPT. (s.f.). Obtenido de <https://chatgpt.com>

Developer.mozilla.org. (s.f.). Obtenido de <https://developer.mozilla.org>

LucidChart. (s.f.). Obtenido de <https://www.lucidchart.com>

PHP. (s.f.). Obtenido de <https://www.php.net>

Sourcetreeapp. (s.f.). Obtenido de <https://www.sourcetreeapp.com>

StackOverFlow. (s.f.). Obtenido de <https://es.stackoverflow.com>

Taniascascia. (s.f.). Obtenido de <https://www.taniascascia.com>