

Elaborado por:

* Hernández Muñoz, Diego Iván
* Ramírez Navarro, Angel Alexander
* De la Cruz Ascencio, Luis Angel

**Sietemesinos**

Seminario de Ing. De Software

Entrega de primer milestone

Mtro. Rubén Adrián Gil Rivera

**Proyecto colaborativo**

**“Chess.mx”**

***Índice***

[**Descripción del proyecto de software** 3](#_Toc180170438)

[**Justificación** 4](#_Toc180170439)

[**Viabilidad** 5](#_Toc180170440)

[**Objetivo general** 7](#_Toc180170441)

[**Objetivos específicos** 7](#_Toc180170442)

[**Riesgos** 8](#_Toc180170443)

[**Tabla de requerimientos funcionales** 9](#_Toc180170444)

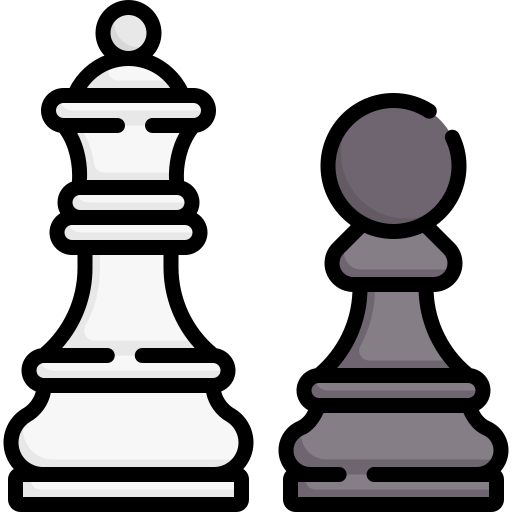
[**Tabla de requerimientos no funcionales** 11](#_Toc180170445)



# **Descripción del proyecto de software**

Este proyecto será desarrollado utilizando el lenguaje de programación Java. Aunque actualmente no contamos con una gran experiencia en este lenguaje, lo hemos elegido por su adaptabilidad y relevancia en la industria del software.

El objetivo principal será crear un juego de ajedrez con funciones adicionales. Inicialmente, nos enfocaremos en implementar partidas entre dos jugadores en una misma pantalla, y posteriormente se añadirá la posibilidad de jugar contra la computadora. Además, los usuarios deberán crear un perfil para poder ingresar al juego, y los puntajes de las partidas serán almacenados en dicho perfil. Con base en estos puntajes, se generará una tabla de clasificación (leaderboard) que mostrará a los usuarios con mejores resultados.



# **Justificación**

Si bien, en un principio, podría parecer que la creación de un juego de ajedrez no aborda problemáticas significativas más allá del entretenimiento del usuario, es importante destacar que la realización y el éxito de este proyecto contribuirán de manera sustancial al desarrollo de las habilidades técnicas de los desarrolladores, a la aplicación práctica de sus conocimientos y, por supuesto, al disfrute de los usuarios que hagan uso de él.

Dado que aún no se posee un dominio extenso del lenguaje Java y que este representa un entorno relativamente nuevo para el equipo, el desarrollo de este proyecto les permitirá adquirir competencias clave en programación orientada a objetos, así como aplicar diversos algoritmos y estructuras de datos para gestionar el comportamiento y los movimientos de cada una de las piezas del ajedrez. Además, pondrán en práctica sus conocimientos sobre diseño de interfaces gráficas para el desarrollo del tablero y la presentación visual del juego

Asimismo, se aprovecharán las habilidades adquiridas en la implementación de las reglas del ajedrez, incluyendo movimientos complejos como el enroque o el jaque mate, así como el manejo adecuado de excepciones, con el fin de crear una aplicación sólida que sirva como un valioso ejercicio de aplicación de teorías en un entorno práctico.

Por último, el entretenimiento será una consecuencia directa de la creación de **Chess.mx**, dado que el ajedrez es uno de los juegos más populares a nivel mundial. Al trasladarlo a una versión digital desarrollada en Java, no solo se proporcionará una herramienta de diversión, sino también una plataforma para el aprendizaje y la competición entre los usuarios.

# **Viabilidad**

El desarrollo de **Chess.mx** es viable y factible, considerando los recursos con los que cuenta el equipo de desarrolladores.

**Viabilidad técnica**

El proyecto es técnicamente factible debido a que el lenguaje de programación que será utilizado para su desarrollo es versátil y ampliamente utilizado en el desarrollo de aplicaciones de escritorio y juegos. Las librerías que ofrece facilitan el diseño de interfaces gráficas llamativas e intuitivas, y al ser un lenguaje orientado a objetos es idóneo para el modelaje de las piezas de ajedrez y sus comportamientos.

Además, el equipo desarrollador cuenta con Entornos de Desarrollo Integrados, también conocidos como IDEs, que proporcionan soporte gratuito para llevar a cabo la programación en el lenguaje Java, así como herramientas de control de versiones que permitirán llevar una gestión correcta del proyecto, tal como lo es GitHub.

Y en cuanto a los conocimientos adquiridos, a pesar de que el equipo de desarrollo se encuentra en una etapa de aprendizaje a profundidad del lenguaje Java, se posee una base sólida en cuanto a los fundamentos de la programación, los algoritmos y las estructuras de datos.

**Viabilidad Económica**

El desarrollo de **Chess.mx** no conllevará de barreras económicas de gran amplitud debido a que los recursos que se estarán utilizando por parte de los desarrolladores serán de libre acceso. Por otro lado, el tiempo a invertir en el desarrollo del proyecto no será mayor a tres meses, por lo que costos adicionales ligados al tiempo invertido no se encuentran previstos.

**Viabilidad Operativa**

Todos los integrantes del equipo programador tienen la capacidad de operación para desarrollar e implementar el proyecto en los plazos de tiempo previstos, cumpliendo con lo establecido previamente. Se utilizarán los conocimientos adquiridos y se usarán los aprendidos durante la creación del juego. Además, se ha establecido un cronograma de actividades para el desarrollo del proyecto, dividiéndolo en fases manejables como el diseño de la interfaz gráfica, la implementación de las reglas de juego, el desarrollo de la lógica de los movimientos de las piezas y la fase de pruebas.

**Viabilidad legal**

Al pertenecer el juego de ajedrez al dominio público, no conllevará algún problema o restricción legal en cuanto a la creación de una versión digital. Además, toda librería o herramienta a utilizar cumplen con las licencias de uso.

**Viabilidad ambiental**

El software digital no requiere del uso de recursos materiales o energéticos más allá del necesario para su desarrollo en los equipos de cómputo, por lo que no conllevará de impactos ambientales significativos.



# **Objetivo general**

Desarrollar un juego de ajedrez funcional para computadoras, que permita a dos usuarios jugar por turnos en la misma pantalla, con un sistema de puntuación integrado.

# **Objetivos específicos**

* Desarrollar un sistema de inicio de sesión que permita a los usuarios registrarse, jugar, y acceder a su información personal y al historial de puntajes.
* Implementar una tabla de clasificación (leaderboard) que muestre a los usuarios con los mejores puntajes de las partidas.
* Crear un sistema de almacenamiento que asocie los puntajes de las partidas con el perfil del usuario.
* Diseñar una interfaz intuitiva y atractiva, con botones grandes y colores y fuente adecuados, procurando que sea fácil de usar. El equipo de desarrollo intentará superar la curva de aprendizaje del lenguaje Java para lograr este objetivo.
* Desarrollar una base de datos optimizada y normalizada para garantizar tiempos de respuesta rápidos y eficientes.
* Cumplir con la entrega del software en la fecha establecida (25 de noviembre de 2024), asegurando que cumpla con los requisitos acordados.
* Garantizar la seguridad de los datos del usuario (nombre de usuario, correo y contraseña) para prevenir posibles ataques al sistema.
* Validar que la contraseña del usuario cumpla con los requisitos de seguridad adecuados para proteger su información.
* Verificar que no existan nombres de usuario duplicados en la base de datos, evitando la redundancia.



# **Riesgos**

1. **Falta de conocimiento en Java**: El equipo no tiene experiencia previa en el lenguaje Java, lo que podría generar dificultades técnicas y desafíos de aprendizaje que ralenticen el avance del proyecto.

* **Mitigación:** Dedicar tiempo a estudiar los conceptos básicos y practicar con pequeños proyectos antes de abordar objetivos más grandes.

1. **Sobrecarga de funcionalidades:** Incluir demasiadas funcionalidades complejas (como implementar un algoritmo de IA para poder lograr el juego contra la computadora) podría hacer que el proyecto se vuelva difícil de manejar, lo que pondría en riesgo el cumplimiento de los plazos.

* **Mitigación:** Priorizar las funcionalidades esenciales, como el juego entre dos jugadores. Dejar funcionalidades más complejas como opcionales según el tiempo disponible.

1. **Problemas de integración de componentes:** Integrar diferentes módulos (sistema de log in, base de datos, y lógica del juego) podría causar errores imprevistos.

* **Mitigación:** Probar cada módulo antes de combinarlo con otros. Utilizar control de versiones (Git) para manejar cambios.

1. **Problemas con la base de datos:** Un mal diseño o implementación de la base de datos podría generar tiempos de respuesta lentos o errores en la asociación de datos.

* **Mitigación:** Diseñar la base de datos con anticipación, además de su respectiva normalización.

1. **Bugs no detectados o mal manejo de errores:** El juego podría presentar errores que afecten la jugabilidad o la experiencia del usuario.

* **Mitigación:** Implementar pruebas unitarias y de integración de forma continua.

1. **Falta de documentación:** La falta de documentación del código podría dificultar su comprensión en un futuro, así como el mantenimiento y la adición de futuras mejoras del sistema.

* **Mitigación:** Mantener una buena documentación desde el inicio, incluyendo comentarios en el código y manuales de usuario.

1. **No adoptar un estilo de codificación:** La ausencia de un estilo de codificación uniforme puede llevar a un código desordenado y difícil de leer.

* **Mitigación:** Establecer y documentar un estilo de codificación claro y uniforme para todo el equipo. Esto puede incluir convenciones de nombres de variables, clases, estructura de archivos, formato de código, y comentarios.

1. **Gestión inadecuada del tiempo:** La falta de una buena planificación podría llevar a que el proyecto no se termine dentro del plazo establecido (25 de noviembre de 2024).

* **Mitigación:** Seguir, revisar y modificar el cronograma de actividades regularmente, según convenga.

# **Tabla de requerimientos funcionales**

|  |  |
| --- | --- |
| ID de requerimiento | *RF01* |
| Tipo | Funcional |
| Funcionalidad | Registro de usuario |
| Requerimiento | El sistema debe permitir que los jugadores se registren en *Chess.mx* creando un nombre de usuario y contraseña. |
| Comentarios | Validar que el nombre de usuario no exista previamente en la base de datos para evitar redundancia.  Verificar que la contraseña cumpla con los siguientes requisitos:   * Longitud mínima de ocho caracteres. * Contener al menos una letra mayúscula. * Incluir al menos un número. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF02* |
| Tipo | Funcional |
| Funcionalidad | Inicio de sesión |
| Requerimiento | El sistema debe permitir a los usuarios registrados iniciar sesión en *Chess.mx* para poder jugar, acceder a su perfil y ver el historial de partidas. |
| Comentarios | Si el nombre de usuario o contraseña es incorrecto, se debe mostrar un mensaje de error adecuado. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF03* |
| Tipo | Funcional |
| Funcionalidad | Partida entre dos jugadores |
| Requerimiento | El sistema debe permitir que dos usuarios jueguen una partida de ajedrez por turnos de forma local (en la misma pantalla). |
| Comentarios | Mostrar el nombre del jugador en turno |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF04* |
| Tipo | Funcional |
| Funcionalidad | Sistema de puntaje |
| Requerimiento | El sistema debe registrar el resultado de cada partida (victoria, derrota o empate) y asociarlo al perfil de los usuarios que participaron en la partida. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF05* |
| Tipo | Funcional |
| Funcionalidad | Tabla de clasificación |
| Requerimiento | Implementar una tabla de clasificación (leaderboard) que muestre a los usuarios con los mejores puntajes de las partidas. |
| Comentarios | La tabla debe permitir la visualización de los 10 mejores jugadores. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF06* |
| Tipo | Funcional |
| Funcionalidad | Iniciar una partida |
| Requerimiento | El sistema debe permitir al usuario iniciar una nueva partida de ajedrez desde la pantalla principal. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF07* |
| Tipo | Funcional |
| Funcionalidad | Tablero |
| Requerimiento | El sistema debe mostrar un tablero de ajedrez de 8x8 con las piezas en sus respectivas posiciones iniciales. |
| Comentarios | El tablero deberá tener sus respectivas guías de números y letras. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF08* |
| Tipo | Funcional |
| Funcionalidad | Piezas |
| Requerimiento | El sistema debe permitir al usuario mover las piezas del tablero según las reglas oficiales del ajedrez. |
| Comentarios | El sistema debe verificar que el movimiento realizado sea válido de acuerdo con las reglas del ajedrez.  Debe actualizarse el estado del tablero después de cada movimiento. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF09* |
| Tipo | Funcional |
| Funcionalidad | Reglas de juego |
| Requerimiento | El sistema debe implementar movimientos especiales como enroque, coronación y captura al paso. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF10* |
| Tipo | Funcional |
| Funcionalidad | Mostrar resultado de la partida |
| Requerimiento | El sistema debe mostrar un sistema de victoria un mensaje de victoria, derrota o empate al final de cada partida. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RF11* |
| Tipo | Funcional |
| Funcionalidad | Historial de partidas del usuario registrado |
| Requerimiento | El juego debe permitir consultar el historial de partidas de cada jugador registrado. |
| Comentarios | Se debe mostrar un mensaje de alerta en caso que el jugador no tenga partidas registradas.  Mostrar una tabla que incluya un historial de las partidas jugadas por el usuario, con columnas para la fecha y hora, oponente y el resultado de cada partida. |
| Id\_Test\_Case\_Satisfied | N/A |

# **Tabla de requerimientos no funcionales**

|  |  |
| --- | --- |
| ID de requerimiento | *RNF01* |
| Tipo | No funcional |
| Funcionalidad | Rendimiento |
| Requerimiento | El juego debe garantizar tiempos de respuesta rápidos para las acciones de los usuarios, como el inicio de sesión, el registro y el movimiento de las piezas durante una partida. |
| Comentarios | El tiempo de respuesta para las acciones no debe exceder de 3 segundos. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RNF02* |
| Tipo | No funcional |
| Funcionalidad | Usabilidad |
| Requerimiento | La interfaz del usuario debe ser funcional, intuitiva y fácil de usar para garantizar una experiencia de usuario agradable. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RNF03* |
| Tipo | No funcional |
| Funcionalidad | Mantenimiento |
| Requerimiento | El sistema debe ser fácilmente mantenible para permitir actualizaciones y mejoras futuras sin requerir grandes modificaciones. |
| Comentarios | Documentar el código y seguir un estilo de codificación para que el código sea legible. |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RNF04* |
| Tipo | No funcional |
| Funcionalidad | Seguridad |
| Requerimiento | El sistema debe de contar con cifrado para las contraseñas de los usuarios. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RNF05* |
| Tipo | No funcional |
| Funcionalidad | Estabilidad |
| Requerimiento | El sistema debe de ser estable y no debe presentar cierres inesperados sobre su uso. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |

|  |  |
| --- | --- |
| ID de requerimiento | *RNF06* |
| Tipo | No funcional |
| Funcionalidad | GUI |
| Requerimiento | El tablero y la interfaz deben ser visualmente atractivos y con una gama de colores adecuado. |
| Comentarios | N/A |
| Id\_Test\_Case\_Satisfied | N/A |