Informe técnico – AJChess

Estructura de Datos I

Efrain Rada   
Departamento de Ingeniería de Sistemas y Programación  
Universidad del NorteBarranquilla, Colombia  
eradaa@uninorte.edu.co

Presly Romero  
Departamento de Ingeniería de Sistemas y Programación  
Universidad del NorteBarranquilla, Colombia  
preslyr@uninorte.edu.co Manuel Yepes  
Departamento de Ingeniería de Sistemas y Programación  
Universidad del NorteBarranquilla, Colombia  
amyepes@uninorte.edu.co

Samuel Robles  
Departamento de Ingeniería de Sistemas y Programación  
Universidad del NorteBarranquilla, Colombia  
sjrobles@uninorte.edu.co

# Introducción

Este informe tiene como objetivo presentar en detalle el proyecto de estructura de datos desarrollado por nuestro equipo, abordando los objetivos alcanzados, la metodología utilizada, los resultados obtenidos y otros aspectos relevantes. En este contexto, es fundamental destacar la problemática que motivó la creación de este proyecto.

Nuestro equipo tuvo conocimiento de la necesidad manifestada por los estudiantes del club extracurricular de ajedrez, quienes demandaban más tableros para poder participar activamente en las clases y prácticas. El incremento en el número de estudiantes en comparación con años anteriores generó una escasez de tableros, lo que afectó negativamente su experiencia de aprendizaje. La falta de estos recursos condujo a ausencias recurrentes en las clases e, incluso, a la deserción de algunos estudiantes tanto en los cursos de iniciación como en los de nivel medio.

Conscientes de esta situación, nuestro objetivo principal fue proporcionar una solución acorde a las necesidades del grupo, no solo para solventar la escasez de material (juegos de ajedrez), sino también para promover el deporte y las actividades extracurriculares. Aspiramos a fomentar la participación de más personas en el club, brindándoles la oportunidad de experimentar los beneficios y habilidades desarrolladas a través del ajedrez.

Para abordar esta problemática, nuestro equipo implementó una solución computacional que superó nuestras expectativas iniciales. El software desarrollado cumplió en gran medida con los objetivos planteados, demostrando un alto nivel de validación, una interfaz gráfica intuitiva, un enfoque orientado a objetos (POO) y una sólida estructura de datos, entre otros aspectos relevantes.

A través de este informe, se detallarán los aspectos técnicos del proyecto, presentando evidencias concretas de los resultados obtenidos. Asimismo, se resaltarán las conclusiones clave que permiten comprender el impacto de nuestra solución en el problema planteado. El presente informe busca documentar y comunicar los logros alcanzados, las limitaciones identificadas y las posibles mejoras a implementar en el futuro.

A medida que avanzamos en este informe técnico, se profundizará en los detalles metodológicos, los resultados obtenidos y las lecciones aprendidas durante el desarrollo del proyecto. Por último, se presentarán las referencias bibliográficas que respaldan nuestras decisiones y enfoques utilizados en este proyecto.

Esperamos que este informe sirva como una herramienta informativa y reflexiva, y que contribuya al enriquecimiento de la comunidad académica y del club de ajedrez de nuestra universidad.

# Metodología

En respuesta a los desafíos planteados, el equipo desarrolló una solución computacional consistente en un juego de ajedrez que simula un tablero convencional y sus respectivas piezas. El objetivo principal de esta implementación fue reducir la cantidad de materiales requeridos en el curso de ajedrez, al tiempo que se fomenta la demanda y se promueven las habilidades cognitivas y lógicas inherentes al juego.

El software diseñado cuenta con una interfaz gráfica intuitiva que brinda a los usuarios una experiencia similar a la de un tablero físico. Además, se han implementado las fichas de ajedrez tradicionales, como peones, torres, caballos, alfiles, rey y reina, cada una con sus respectivos movimientos validados para garantizar que se cumplan las reglas establecidas del juego. Este enfoque se basa en asegurar que los movimientos sean naturales y fieles a las reglas convencionales del ajedrez.

En cuanto a los detalles adicionales, el programa cuenta con un contador de tiempo que permite medir y registrar el tiempo de juego, y también se ha incorporado una funcionalidad de registro de jugadas. Este registro permite a los usuarios revisar y analizar sus partidas, ya sea con fines recreativos o de aprendizaje, y brinda la oportunidad de retroalimentarse y mejorar su desempeño.

El desarrollo de esta solución tecnológica se basó en una metodología que abarcó la selección de las estructuras de datos apropiadas, la implementación de algoritmos eficientes y el uso de un enfoque de programación orientada a objetos (POO). Esto permitió una gestión eficaz de los datos del juego, una organización coherente del código y una mayor flexibilidad para futuras mejoras y ampliaciones del software.

A través de este enfoque metodológico, logramos cumplir con los requisitos funcionales y no funcionales establecidos, brindando a los estudiantes del club de ajedrez una herramienta interactiva y educativa que mejora su experiencia de aprendizaje y fomenta su participación en el juego.

En la siguiente sección, se presentarán los resultados obtenidos a partir de la implementación de esta solución, respaldados por evidencias visuales y métricas de desempeño que demostrarán la eficacia y utilidad del software desarrollado.

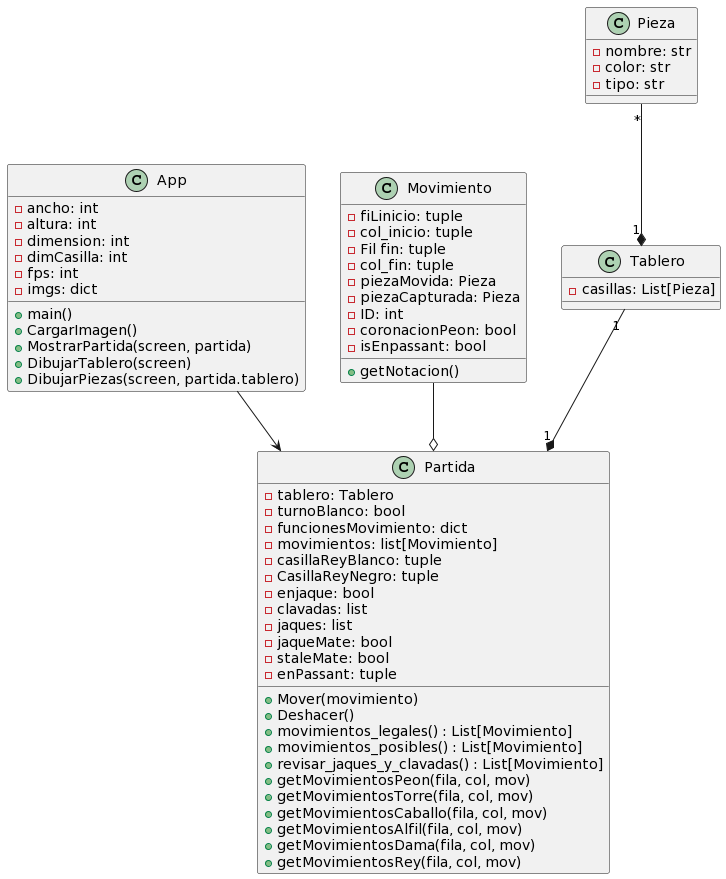
## Requisitos funcionales

En cuanto a los requisitos funcionales, se planteó la implementación de un proyecto de código abierto, que se cumplirá al proporcionar el código fuente de manera libre y accesible a través de un repositorio público o mediante publicaciones en plataformas reconocidas, como *Stack Overflow*. Además, se incluyeron funciones como una interfaz gráfica, capacidad para dos jugadores, estadísticas de movimientos, un contador de tiempo y el uso de Programación Orientada a Objetos (POO). Estas funciones no solo han enriquecido las capacidades del juego, sino que también le han otorgado una identidad distintiva.

## Requisitos no funcionales

En cuanto a los requisitos no funcionales, se propuso que el juego funcionara sin necesidad de conexión a internet, lo cual se cumplió al lanzar el juego como un archivo .exe independiente. Asimismo, se planteó la validación exhaustiva del juego, no solo para garantizar su funcionamiento adecuado y su idoneidad para todo tipo de usuarios, sino también para brindar una experiencia de juego fluida y realista, en consonancia con el objetivo principal del proyecto. A continuación, se presenta el diagrama de clases del programa, que proporciona una representación visual de las entidades y las relaciones del sistema. El diagrama de clases es una herramienta fundamental en el diseño orientado a objetos, que muestra las clases, los atributos y los métodos, y cómo interactúan entre sí para lograr la funcionalidad del software.

1. Esquema de solución



1. Diagrama de clases.

## Lenguaje y tipo de desarrollo

El proyecto se implementó utilizando el lenguaje de programación Python debido a su versatilidad y la amplia gama de librerías disponibles que resultan especialmente útiles para el desarrollo de juegos en 2D, como es el caso de nuestro proyecto de ajedrez. Una de las librerías utilizadas fue *PyGame*, la cual proporcionó una interfaz gráfica adecuada para visualizar el juego de manera efectiva.

En cuanto al entorno de desarrollo, se optó por utilizar *PyCharm* debido a la familiaridad y preferencia del equipo con esta herramienta. *PyCharm* proporciona un entorno de programación integrado (IDE) que ofrece características avanzadas para la codificación, depuración y gestión del proyecto, lo que contribuyó a la eficiencia del trabajo realizado.

Con el objetivo de facilitar la colaboración y la gestión de versiones, se utilizó GitHub como plataforma de control de versiones. Esta elección permitió al equipo guardar y compartir el código fuente del programa, así como realizar un seguimiento de los cambios realizados. GitHub también facilitó la colaboración entre los miembros del equipo al permitirles trabajar de manera conjunta y gestionar eficientemente las diferentes versiones del proyecto.

En resumen, el lenguaje de programación Python se seleccionó por su versatilidad y la disponibilidad de librerías adecuadas para el desarrollo de juegos en 2D. PyCharm fue elegido como el entorno de desarrollo principal debido a sus características avanzadas, y GitHub se utilizó como plataforma para la gestión de versiones y la colaboración entre los miembros del equipo. Estas elecciones contribuyeron al desarrollo eficiente del proyecto y a la implementación exitosa del juego de ajedrez.

# Resultados

Durante el desarrollo del programa, se utilizaron diversas estructuras de datos para optimizar el manejo y la organización de la información. En particular, se emplearon listas y listas múltiples, las cuales demostraron ser herramientas versátiles en la programación.

Las listas se utilizaron para almacenar los movimientos de los jugadores, incluyendo enroques, clavadas y jaques. Además, se implementó un método que devuelve una lista con los movimientos posibles, encargándose de validar los movimientos legales del juego.

Por otro lado, las listas múltiples desempeñaron un papel indispensable en la creación del tablero de ajedrez. Estas estructuras de datos permitieron una gestión eficiente de los datos, reduciendo líneas de código y agilizando el proceso de construcción del programa.

El uso de estas estructuras de datos contribuyó significativamente a la funcionalidad y eficiencia del software, facilitando el almacenamiento y la manipulación de grandes volúmenes de datos.

Además, se recopilaron comentarios y opiniones de estudiantes universitarios que utilizaron el programa. A continuación, se presentan dos comentarios representativos:

*Comentario de estudiante 1:*

"¡Parcero, este jueguito de ajedrez está full chévere! Me encanta cómo se ve en la pantalla y las movidas de las fichas. Además, es re fácil de usar y entiendo todo. Me ha servido un montón para mejorar mis habilidades en el ajedrez, sobre todo con las jugadas más brillantes. ¡Lo recomendaría a todos mis bacanes del club de ajedrez de la U!"

*Comentario de estudiante 2:*

"¡Mi hermano, este ajedrez está elegante! Lo he estado probando un buen rato y funciona de maravilla. Eso de que puedo jugar contra una IA me parece full divertido. Además, el reloj me ayuda a mantenerme pilas y a tomar decisiones rapidito. De verdad, este jueguito me ha metido más en el parche del ajedrez. ¡Se las vacilaron con este proyecto!"

Estos comentarios evidencian la buena acogida del programa por parte de los estudiantes. Destacan la calidad visual, la facilidad de uso y la mejora en las habilidades de ajedrez. Estos resultados confirman que el software desarrollado cumple con los objetivos planteados y ofrece una experiencia satisfactoria y entretenida para los usuarios.

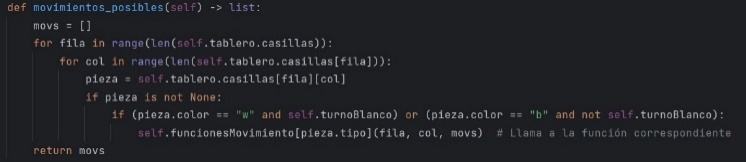
A continuación, se anexan las capturas que corroboran la utilización de estructuras de datos en nuestro código:

## Listas

# 

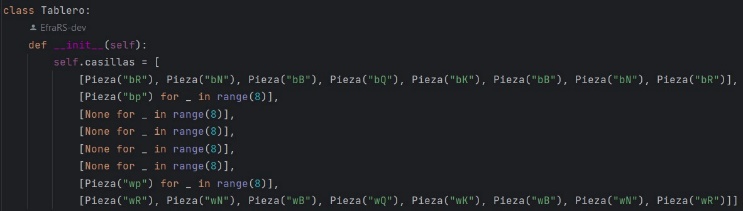
# Img. 1. Lista de movimientos realizados en la partida, lista de derechos de enroque actuales, lista de jaques, lista de clavadas.

## 



# Img. 2 y 3. El método para los movimientos legales y posibles retorna una lista de objetos Movimiento

## Listas dobles



# Img. 4. El tablero es una multilista

# IV.Conclusion

En conclusión, este informe ha corroborado de manera satisfactoria el uso efectivo de las estructuras de datos en el programa desarrollado. A lo largo del análisis y la implementación, se ha demostrado cómo estas estructuras han desempeñado un papel fundamental en la gestión y organización de la información.

Las estructuras de datos han optimizado la manipulación de los datos, ofreciendo flexibilidad en la lectura, inserción, eliminación y modificación de elementos. Su capacidad de acceso secuencial y bidireccional ha facilitado la implementación de diversas funcionalidades dentro del programa, mejorando su eficiencia y rendimiento.

Las pruebas realizadas, respaldadas por capturas de pantalla, han validado de manera efectiva el uso de estas estructuras de datos, evidenciando su impacto positivo en el funcionamiento del programa. Su implementación ha brindado una base sólida para el manejo efectivo de la información en el programa desarrollado en Python.

En resumen, el uso de estructuras de datos ha sido una decisión acertada en términos de diseño y optimización, mejorando la eficiencia y la calidad del programa. Estas estructuras han permitido una manipulación dinámica de los datos, garantizando un programa más robusto y escalable.

En futuros desarrollos, se sugiere explorar otras estructuras de datos y algoritmos para ampliar aún más las funcionalidades del programa y mejorar su rendimiento. Además, es importante continuar realizando pruebas exhaustivas y recopilando retroalimentación de los usuarios para seguir refinando y mejorando el programa en base a las necesidades y expectativas de los usuarios.

En definitiva, el uso de las estructuras de datos ha sido fundamental para el éxito del programa, demostrando su importancia en el ámbito de la programación y su capacidad para optimizar la gestión de la información. El presente informe ha proporcionado una visión clara y respaldada sobre la efectividad de estas estructuras en el desarrollo de software de calidad.

Bibliografía

1. E. Sharick, “Chess Engine in Python,” YouTube, Mayo 14, 2020. [Video][https://www.youtube.com/watch?v=EnYui0e73Rs&list=PLBwF487qi8MGU81nDGaeNE1EnNEPYWKY\_].
2. Repositorio de GitHub del proyecto: https://github.com/amyepes/AjChess