

Universidad Peruana de Ciencias Aplicadas

Ingeniería de Software



# COMPLEJIDAD ALGORÍTMICA

## INFORME DEL TRABAJO FINAL

---

### TF

**Project:** Ajedrez

**Teacher:**

Canaval Sanchez, Luis Martin

**Team Members:**

- Lostaunau Pereira, Estefano Sebastian
- Garcia Moscoso, Andrea Joselyn
- Coraje Bayona, Jair André

**Section:** WS6B

Ciclo 2023 - 02  
Septiembre

## CONTENT

Registro de Versiones del Informe.....	3
Descripción del problema.....	4
Descripción del conjunto de datos (dataset).....	7
Propuesta.....	8
Diseño del aplicativo.....	9
Validación de resultados y pruebas.....	11
Conclusiones.....	11
Referencias Bibliográficas.....	12

## Registro de Versiones del Informe

El objetivo de esta sección es resumir las modificaciones relevantes que se realizan al informe durante el ciclo de vida del proyecto. Esta sección inicia en una página nueva y se incluye un cuadro con la siguiente estructura.

Versión	Fecha	Autor	Descripción de modificación	Desarrollo de las Competencias Específicas: ABET 4
TP	24/09/23	Grupo 4	Descripción del problema. Descripción del conjunto de datos(dataset). Propuesta.	Todos participamos.
HITO 2	11/11/2023	Grupo 4	Descripción del problema. Descripción del conjunto de datos(dataset). Propuesta. Diseño del aplicativo.	Todos participamos.

## Descripción del problema

El ajedrez es un juego de estrategia atemporal que ha desafiado a la mente humana durante siglos. Es una disciplina que va mucho más allá de las simples reglas y movimientos de las piezas en un tablero de 8x8 casillas. Implica la capacidad de anticipar y planificar movimientos futuros, evaluar las consecuencias de cada jugada y adaptarse rápidamente a las decisiones del oponente. El ajedrez es un campo de batalla mental donde la creatividad y la lógica se entrelazan de manera única.

Resolver problemas de ajedrez no solo es una excelente manera de mejorar las habilidades tácticas y estratégicas, sino que también fomenta un pensamiento crítico agudo. Cada problema plantea un desafío único que requiere un enfoque meticuloso y una visión profunda del tablero. Los jugadores deben calcular con precisión, visualizar múltiples movimientos por delante y evaluar las implicaciones de cada opción.

Además de desarrollar habilidades cognitivas, resolver problemas de ajedrez mejora la memoria a corto y largo plazo. Los jugadores deben recordar patrones de jugadas exitosas, tácticas efectivas y posiciones clave. Esta mejora en la memoria no solo se traduce en un mejor rendimiento en el ajedrez, sino que también puede beneficiar otras áreas de la vida, como el estudio, el trabajo y la toma de decisiones en general.

La propuesta de crear un resolutor de ajedrez es una respuesta a la necesidad de los jugadores de ajedrez de contar con una herramienta versátil y accesible para mejorar sus habilidades. La aplicación que se busca desarrollar no solo permitirá a los jugadores practicar la resolución de problemas de ajedrez, sino que también ofrecerá una retroalimentación valiosa sobre sus soluciones. Esta retroalimentación puede incluir análisis de jugadas, estadísticas de desempeño y recomendaciones para mejorar.

Este proyecto es viable y factible en la era digital actual, donde la tecnología puede facilitar la creación de una aplicación interactiva y educativa. Además, el ajedrez es un juego con una comunidad global apasionada y en crecimiento, lo que garantiza una audiencia potencialmente amplia para la herramienta.

En última instancia, el proyecto de resolver problemas de ajedrez tiene un impacto real en la vida de los jugadores. Les brinda la oportunidad de desarrollar habilidades que no solo se aplican al ajedrez, sino que también se traducen en un pensamiento más estratégico en otros aspectos de su vida, desde la toma de decisiones hasta la resolución de problemas cotidianos. Es una inversión en el crecimiento personal y cognitivo de los jugadores, y puede ser una herramienta valiosa en su viaje hacia la maestría en el ajedrez y más allá.



- Estado final

El estado final en una partida de ajedrez se representa mediante la disposición final de las piezas en el tablero al concluir la partida. La representación de un estado final puede variar según la posición específica. Los estados finales mas comunes en el ajedrez son:

- Jaque Mate: En un jaque mate, uno de los reyes ha sido capturado, lo que indica la finalización de la partida.

En esta representación, ambas partes han realizado sus movimientos finales, y el rey que ha sido jaqueado se muestra como capturado (generalmente con una "X" o una cruz) para indicar el jaque mate.

- Empate o Acuerdo de Tablas: En un empate o cuando se acuerda un empate, la posición final puede variar según la causa del empate. Puede representarse de manera similar a la posición inicial o con la posición de las piezas al momento del acuerdo de tablas.
- Resultado Pendiente: Si la partida aún no ha concluido, se representaría la posición actual del tablero con las piezas en sus ubicaciones actuales y se indicaría que la partida está "en curso" o "pendiente".

Finalmente, este también puede ser representado por "1-0" o "0-1" y en caso de empate como "1/2-1/2".

- Transiciones

Para nuestro trabajo, vamos a usar la librería pygame para implementar las transiciones de piezas a través del mouse. Para ello, vamos a crear una clase para cada pieza del juego, con un atributo para la posición actual de la pieza y un método para mover la pieza. El método para mover la pieza usará el evento pygame.MOUSEBUTTONDOWN para detectar cuando el usuario hace clic en la pieza. Para que la transición sea fluida, voy a usar el efecto de animación `pygame.transform.smoothscale()`.

- Representación(estructuras de datos a utilizar), atributos y tipo de dato

Se van a utilizar unas matrices para poder poner cada ficha dentro del tablero, los tipos de dato dependen de los atributos de las fichas, ya que cada fichar será de una clase distinta, los atributos que tendran en comun cada uno de ellos, son los de sus movimientos, pero cada uno tendrá su tipo de movimiento respectivo como otro atributo. Los tipos de datos para cada movimiento serán int, para la validación de los movimientos unos booleanos, tanto como para validar si la ficha enemiga fue comida por la nuestra, se deberán usar ints para la posición de los ejes de cada ficha.

Para la ficha de la Reina, se deberá crear un método a parte, que indicara su zona de peligro, si es que puede ser atacada, sea la única que pueda moverse, así como un método para saber si se realiza un jaque mate a la Reina.

- Cálculo del Tamaño(espacio de búsqueda)

En el ajedrez, el espacio de búsqueda es el conjunto de todos los estados posibles del juego. Cada estado se define por la posición de las piezas en el tablero. El tamaño del espacio de búsqueda de ajedrez es enorme. Se estima que es de aproximadamente  $10^{120}$ . Esto significa que es imposible explorar el espacio de búsqueda completo usando un algoritmo de búsqueda exhaustiva. Por lo tanto, los motores de ajedrez modernos usan algoritmos de búsqueda heurística para reducir el tamaño del espacio de búsqueda. Estos algoritmos se basan en una función de evaluación que estima la calidad de un estado.

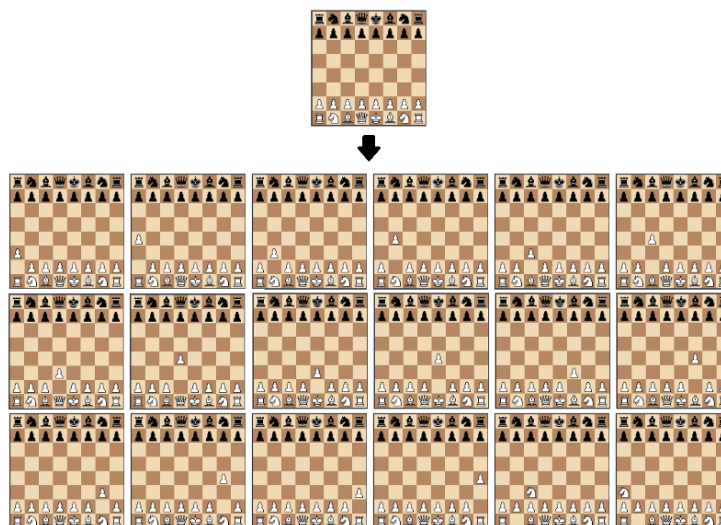
La función de evaluación se usa para seleccionar los estados más prometedores para explorar. Esto permite a los motores de ajedrez encontrar soluciones de alta calidad en un tiempo razonable.

Algunos de los factores que influyen en el tamaño del espacio de búsqueda de ajedrez son:

**El número de piezas:** El ajedrez tiene 32 piezas, cada una de las cuales puede moverse de diferentes maneras. Esto aumenta el número de estados posibles.

**El número de movimientos posibles:** Cada pieza tiene varios movimientos posibles. Esto aumenta el número de estados posibles.

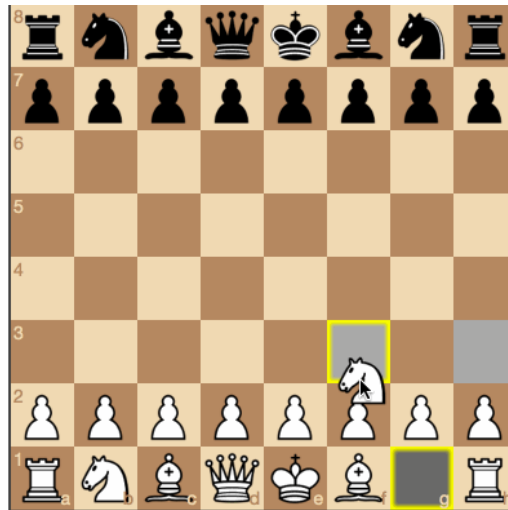
**El número de estados terminales:** El ajedrez tiene tres estados terminales: jaque mate, tablas y empate. Esto reduce el número de estados posibles.



- Visualizar:

El espacio de búsqueda de ajedrez es un conjunto enorme y complejo de estados posibles, cada uno de los cuales se define por la posición de las piezas en el tablero. Para comprender mejor el espacio de búsqueda de ajedrez, podemos visualizarlo mediante un gráfico de una porción representativa del espacio. Una forma de visualizar el espacio de búsqueda de ajedrez es mediante un gráfico de dispersión. En este gráfico, cada punto representa un estado del juego. La posición del punto en el gráfico representa la posición de las piezas en el tablero. Otra forma de visualizar el espacio de búsqueda de ajedrez es mediante un gráfico de árbol. En este gráfico,

cada nodo del árbol representa un estado del juego. Los nodos hijos de un nodo representan los estados que se pueden alcanzar desde el estado representado por el nodo padre.



## Descripción del conjunto de datos (dataset)

Para nuestro proyecto utilizaremos un dataset que contenga partidas jugadas en torneos, de los cuales utilizaremos las jugadas del TOP 10 jugadores de ajedrez en el mundo por elo, con el fin de obtener los mejores resultados para nuestro bot.

Utilizaremos los datos obtenidos para que el bot pueda analizar la partida que esté jugando contra el jugador apoyándose del dataset. Los datos que contendrá el dataset será, la partida jugada, el ganador de la partida y de ahí los movimientos realizados durante la partida, usaremos la partida jugada y el ganador, para poder analizar el tablero, y el color del bot que estará jugando, los movimientos realizados también nos servirá para ir viendo el estado de los tableros al momento de jugar, ya que cada movimiento actualiza el tablero, por lo tanto el bot analizaría el estado actual del tablero y compararía con las jugadas del dataset, pudiendo validar que movimiento es el más certero al momento de jugar, gracias al uso del dataset mejoramos la eficacia del algoritmo utilizado para los movimientos del bot, porque ya no evaluará todas las posibilidades que tiene el juego de ajedrez, sino las que le estamos entregando a partir del dataset.

Aquí se encuentra el dataset:

[https://github.com/JairCoraje/TrabajoComplejidad/blob/main/data\\_set\\_-\\_Grupo4.txt](https://github.com/JairCoraje/TrabajoComplejidad/blob/main/data_set_-_Grupo4.txt)



## Propuesta

Las técnicas de búsqueda que se pueden utilizar para resolver problemas de ajedrez incluyen:

**Búsqueda en profundidad:** La búsqueda en profundidad comienza en el nodo actual y explora todos los nodos hijos hasta que encuentra una solución o llega a un nodo sin hijos.

**Búsqueda A:** La búsqueda A es una técnica de búsqueda informada que utiliza una función de evaluación para guiar la búsqueda hacia la solución más probable.

**Búsqueda en anchura:** La búsqueda en anchura comienza en el nodo actual y explora todos los nodos hijos en un orden de anchura.

**Búsqueda en amplitud:** Esta técnica consiste en explorar todos los posibles movimientos de un jugador, de forma que cada movimiento se explora antes de explorar el siguiente. La búsqueda en amplitud es más eficiente que la búsqueda en profundidad para problemas complejos, pero puede no encontrar la mejor solución.

**Búsqueda con retroceso:** Esta técnica consiste en explorar un árbol de juego, comenzando por la raíz y avanzando hacia las hojas. La búsqueda con retroceso se puede utilizar para combinar las ventajas de la búsqueda en profundidad y la búsqueda en amplitud.

**Búsqueda iterativa:** Esta técnica consiste en realizar varias búsquedas, cada una con una profundidad diferente.

**Búsqueda en paralelo:** Esta técnica consiste en realizar varias búsquedas simultáneamente.

**Búsqueda con aprendizaje automático:** Esta técnica utiliza algoritmos de aprendizaje automático para mejorar la eficacia de la búsqueda.

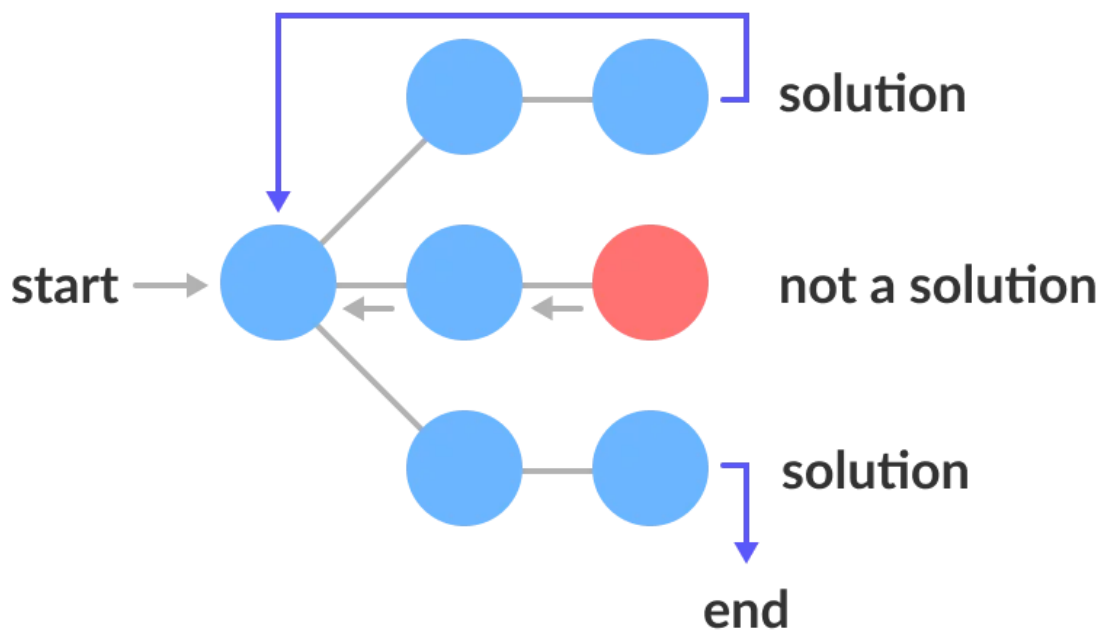
**Backtracking:** Esta técnica se puede utilizar en ajedrez para resolver problemas como encontrar el mejor movimiento para un jugador, o encontrar una secuencia de movimientos que lleve a un jaque mate.

**Búsqueda con poda:** Esta técnica consiste en evitar que el algoritmo explore movimientos que no tienen posibilidades de conducir a una solución. Hay varias técnicas de poda que se pueden utilizar en ajedrez, como la poda alfa-beta y la poda de profundidad.

**Búsqueda con heurística:** Esta técnica utiliza una función de evaluación para guiar la búsqueda hacia la solución más probable. La función de evaluación puede tener en cuenta factores como la posición de las piezas, el material disponible y el estado del juego.

**Búsqueda con aprendizaje automático:** Esta técnica utiliza algoritmos de aprendizaje automático para mejorar la eficacia de la búsqueda. Los algoritmos de aprendizaje automático pueden aprender de los datos de entrenamiento para mejorar su capacidad de predecir la mejor jugada.





### Backtracking

Se concluyó que los algoritmos de backtracking y búsqueda con poda son los mejores para resolver problemas de ajedrez porque son capaces de encontrar la mejor jugada para un jugador y encontrar una secuencia de movimientos que lleve a un jaque mate. Además, son eficientes y pueden explorar el espacio de búsqueda de forma rápida.

En nuestro caso estos algoritmos son utilizados para las siguientes tareas:

- Encontrar la mejor jugada para el jugador activo.
- Encontrar una secuencia de movimientos que lleve a un jaque mate.
- Evaluar la posición del tablero.

Estos algoritmos pueden utilizarse para mejorar el rendimiento de un revolvedor de ajedrez, ya que pueden ayudar a encontrar la mejor jugada para el jugador activo y a encontrar una manera de ganar la partida.

## Diseño del aplicativo

Este diseño del aplicativo proporciona un marco sólido de su desarrollo utilizando las etapas de la ingeniería de software. Cada etapa se encarga de garantizar una interfaz intuitiva y un motor de ajedrez eficiente.

### 1. Requisitos:

- **Análisis del Problema:**

Identificación de Funcionalidades Clave: Interactividad del tablero, retroalimentación del jugador, motor de ajedrez.

- **Requisitos Funcionales y No Funcionales:**

Funcionales: Mover piezas, retroalimentación en tiempo real, estadísticas de desempeño.

No Funcionales: Eficiencia en la búsqueda de movimientos, interfaz intuitiva.

## 2. Diseño:

- **Arquitectura del Aplicativo:**

Componentes: Interfaz de Usuario (IU), Motor de Ajedrez, Integración con Pygame.

Comunicación: La IU interactúa con el Motor de Ajedrez para mostrar movimientos y recibir retroalimentación.

- **Diseño de Algoritmos:**

Motor de Ajedrez: Algoritmos de búsqueda en profundidad, evaluación heurística, validación de movimientos y detección de jaque mate.

- **Diseño de la Interfaz de Usuario:**

Prototipo: Tablero interactivo, menús de opciones, mensajes informativos.

## 3. Implementación:

- **Codificación:**

Módulos: Implementación de módulos para la IU, el Motor de Ajedrez y la integración con Pygame.

Algoritmos: Implementación de algoritmos de búsqueda y evaluación heurística en el Motor de Ajedrez.

- **Integración de Componentes:**

Conexión: Asegurar la correcta comunicación entre la IU y el Motor de Ajedrez.

## 4. Pruebas:

- **Pruebas Unitarias:**

Validación de Movimientos: Verificar que el Motor de Ajedrez genera movimientos válidos.

Retroalimentación: Confirmar que la IU proporciona retroalimentación precisa.

- **Pruebas de Integración:**

Transiciones: Asegurar que las transiciones entre movimientos sean suaves y precisas.

## 5. Despliegue:

- **Optimización:**

Rendimiento: Ajustes para mejorar la eficiencia del Motor de Ajedrez.

- **Despliegue en Plataformas Digitales:**

Publicación: Desplegar en plataformas digitales para acceder al aplicativo.

## 6. Mantenimiento:

- Actualizaciones y Mejoras:

Retroalimentación de Usuarios: Recopilar feedback para realizar mejoras.

Actualizaciones Periódicas: Corregir errores, agregar nuevas funciones según sea necesario.



## Validación de resultados y pruebas

Para el trabajo final.

## Conclusiones

- Se adoptó una definición de "espacio temporal" para evaluar y medir el rendimiento de nuestro resolutor de ajedrez. Este espacio temporal se compone de tres elementos esenciales: el número de piezas en el tablero, el número de movimientos realizados por cada jugador y el número de estados terminales alcanzados en una partida. Este enfoque nos permite evaluar la eficiencia y la efectividad de nuestro algoritmo en la toma de decisiones estratégicas y tácticas en el juego de ajedrez, lo que a su vez nos ayudará a mejorar y perfeccionar nuestra herramienta en futuras iteraciones.

- El backtracking es un algoritmo de búsqueda recursiva que puede ser utilizado para resolver problemas de ajedrez. El algoritmo comienza en el estado inicial y explora todos los posibles movimientos desde ese estado. Si alguno de los movimientos lleva a un estado terminal, el algoritmo devuelve ese estado. Si ninguno de los movimientos lleva a un estado terminal, el algoritmo retrocede y explora otro movimiento.
- La valoración de las piezas en el tablero es otro componente esencial en la toma de decisiones ajedrecísticas, ya que nos ayuda a evaluar la posición relativa de las piezas y a determinar qué sacrificios o intercambios son beneficiosos para nuestra estrategia. Este conocimiento nos brinda una ventaja significativa al jugar ajedrez y puede marcar la diferencia entre la victoria y la derrota.
- En conclusión, la implementación de técnicas de búsqueda como el backtracking y la búsqueda con poda ofrece un gran potencial para mejorar el desempeño en el ajedrez. Estos algoritmos pueden ser fundamentales para encontrar la mejor jugada, secuencias de movimientos que conduzcan al jaque mate y evaluar posiciones de manera eficiente. Su aplicación en motores de ajedrez puede elevar significativamente la calidad de las decisiones tomadas por los jugadores y permitir un juego más sofisticado y estratégico. La combinación de estas técnicas con el aprendizaje automático representa un futuro prometedor para la optimización continua del ajedrez.

## Referencias Bibliográficas

2022/, J. S. (2023, enero 24). *Guía paso a paso para crear una IA de ajedrez sencilla*. freecodecamp.org.

<https://www.freecodecamp.org/espanol/news/guia-paso-a-paso-para-crear-una-ia-de-ajedrez-sencilla/>

Barrera, R. (2016, diciembre 5). *La Gestión de Activos Vista como un Juego de Ajedrez*. Reliabilityweb.

<https://reliabilityweb.com/sp/articles/entry/strategy-of-asset-management-seen-as-a-chess-game>

de Carrera, T. F. (s/f). *UNIVERSIDAD POLITÉCNICA DE MADRID*. Upm.es.

[https://oa.upm.es/51640/1/PFC\\_JESUS\\_M\\_FERNANDEZ\\_ORCHANDO.pdf](https://oa.upm.es/51640/1/PFC_JESUS_M_FERNANDEZ_ORCHANDO.pdf)

Gavilán, I. G. R. (2022, enero 24). *El ajedrez y el factor escala en el aprendizaje algorítmico*. Ignacio G.R. Gavilán.

<https://ignaciogavilan.com/el-ajedrez-y-el-factor-escala-en-el-aprendizaje-algoritmico/>

Ines, P. (2021, octubre 22). *¿Cuál es el Algoritmo detrás de un juego de Ajedrez por ordenador?* Techlib.net.

<https://techlib.net/blog/cual-es-el-algoritmo-detras-de-un-juego-de-ajedrez-por-ordenador/>

*Live Chess On Chessbase.com*. (s/f). Chessbase.com. <https://live.chessbase.com/en/>

Sánchez, J. S., & TTorrents, A. S. (s/f). *Aprendizaje basado en juegos: El ajedrez como método de aprendizaje de la estrategia empresarial*. Upc.edu.

<https://upcommons.upc.edu/bitstream/handle/2117/26447/DIT+02-2015+Aprendizaje+basado+juegos+ajedrez.pdf?sequence=1>