

**Desarrollado por:** Gilberto Hernández

## **Documentación**

(Ajedrez IA)

### **Índice**

1. Introducción
2. Objetivos del Proyecto
3. Alcance del Sistema
4. Tecnologías Utilizadas
5. Arquitectura General del Sistema
6. Diseño y Arquitectura del Sistema
7. Backend – Servidor e Inteligencia Artificial
  - 7.1 Gestión de partidas
  - 7.2 Motor de IA
  - 7.3 Algoritmo Minimax con poda alfa-beta
  - 7.4 Evaluación avanzada del tablero
  - 7.5 Dificultades de juego
8. Frontend – Interfaz de Usuario
  - 8.1 Flujo de juego
  - 8.2 Manejo del tablero
  - 8.3 Comunicación con el backend
  - 8.4 Mensajes de estado del juego
9. Reglas de Ajedrez Implementadas
10. Seguridad y control de errores
11. Pruebas realizadas
12. Limitaciones del sistema
13. Posibles mejoras futuras
14. Conclusión
15. Imágenes

# 1. Introducción

El presente proyecto consiste en el desarrollo de un sistema de ajedrez profesional con Inteligencia Artificial, accesible desde un navegador web.

El usuario juega con las piezas blancas contra una IA que controla las negras, con distintos niveles de dificultad, incluyendo un **modo profesional** con comportamiento estratégico avanzado.

El sistema está diseñado bajo una arquitectura **cliente-servidor**, utilizando Python con **Flask** como backend y **JavaScript** para la interacción en tiempo real del frontend.

# 2. Objetivos del Proyecto

## Objetivo General

Desarrollar un juego de ajedrez web con una IA capaz de tomar decisiones estratégicas realistas, simulando el comportamiento de un jugador humano avanzado.

## Objetivos Específicos

- Implementar las reglas oficiales del ajedrez.
- Desarrollar una IA basada en el algoritmo Minimax.
- Permitir múltiples niveles de dificultad.
- Proveer una interfaz clara, intuitiva y profesional.
- Garantizar estabilidad y control de errores durante la partida.

# 3. Alcance del Sistema

- Juego completo de ajedrez
- IA con múltiples dificultades
- Promoción de peones
- Detección de jaque, jaque mate y tablas
- Interfaz gráfica web
- Comunicación en tiempo real

# 4. Tecnologías Utilizadas

Tecnologías	Uso
Python	Backend e IA
Flask	Servidor Web
python-chess	Motor de reglas del ajedrez
JavaScript	Lógica del frontend
HTML y CSS	Interfaz grafica
Fetch Api	Comunicación Cliente-Servidor

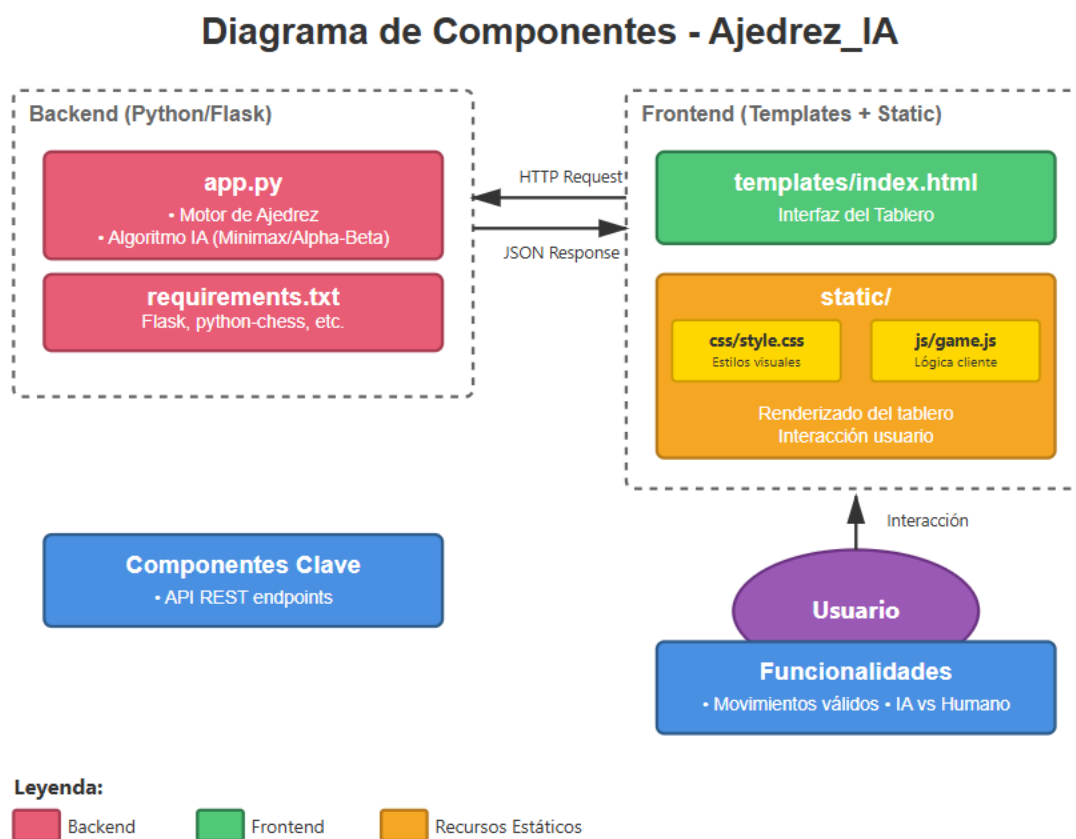
## 5. Arquitectura General del Sistema

El sistema sigue una arquitectura **cliente-servidor**:

- **Frontend**
  - Renderiza el tablero
  - Captura interacciones del usuario
  - Envía movimientos al backend
- **Backend**
  - Valida movimientos
  - Ejecuta la IA
  - Controla el estado del juego

La comunicación se realiza mediante **peticiones HTTP POST** con datos en formato JSON.

## 6. Diseño y Arquitectura del Sistema



## 7. Backend – Servidor e Inteligencia Artificial

### 7.1 Gestión de Partidas

Cada partida se almacena en memoria usando un identificador único (game\_id).

Esto permite:

- Mantener múltiples partidas simultáneas.
- Aislar el estado de cada juego.

### 7.2 Motor de Inteligencia Artificial

La IA está encapsulada en la clase ChessAI, responsable de:

- Evaluar posiciones
- Analizar jugadas futuras
- Seleccionar el mejor movimiento posible

### 7.3 Algoritmo Minimax con Poda Alfa-Beta

La IA utiliza Minimax, un algoritmo de búsqueda adversarial, optimizado con poda alfa-beta, lo que reduce drásticamente el número de nodos evaluados.

Características:

- Negras → maximizan puntuación
- Blancas → minimizan puntuación
- Profundidad variable según dificultad

### 7.4 Evaluación Avanzada del Tablero

La función evaluate() considera:

- **Material** (valor de piezas)
- **Jaque y jaque mate**
- **Capturas**
- **Rey expuesto**
- **Ventaja posicional**

Esto permite decisiones más humanas y agresivas en dificultad profesional.

### 7.5 Dificultades de Juego

Dificultad	Profundidad	Aleatoriedad
Principiante	1	Alta
Intermedio	2	Media
Avanzado	3	Baja
Profesional	4-5	Ninguna

## **8. Frontend – Interfaz de Usuario**

### **8.1 Flujo de Juego**

1. Selección de dificultad
2. Inicio de partida
3. Movimiento del jugador
4. Respuesta de la IA
5. Detección de final de juego

### **8.2 Manejo del Tablero**

- El tablero se genera dinámicamente a partir de FEN.
- Las casillas válidas se resaltan visualmente.
- Se evita mover piezas cuando no es el turno correspondiente.

### **8.3 Comunicación con el Backend**

Todas las acciones se envían mediante fetch():

- /new\_game
- /move
- /get\_legal\_moves

### **8.4 Mensajes de Estado**

El sistema muestra mensajes claros:

- **GANASTE**
- **PERDISTE**
- **TABLAS**

Cuando el juego finaliza, se bloquean las interacciones.

## **9. Reglas de Ajedrez Implementadas**

- Movimientos legales
- Capturas
- Promoción de peones
- Jaque y jaque mate
- Tablas por ahogado
- Material insuficiente

## **10. Seguridad y Control de Errores**

- Validación de movimientos ilegales
- Manejo de excepciones en backend
- Bloqueo de acciones inválidas en frontend

## 11. Pruebas Realizadas

- Pruebas manuales de cada dificultad
- Validación de promociones
- Verificación de estados finales
- Pruebas de jugadas ilegales

## 12. Limitaciones del Sistema

- No guarda historial de partidas
- IA no utiliza apertura teórica ni tablas finales
- No usa multithreading (una IA por petición)

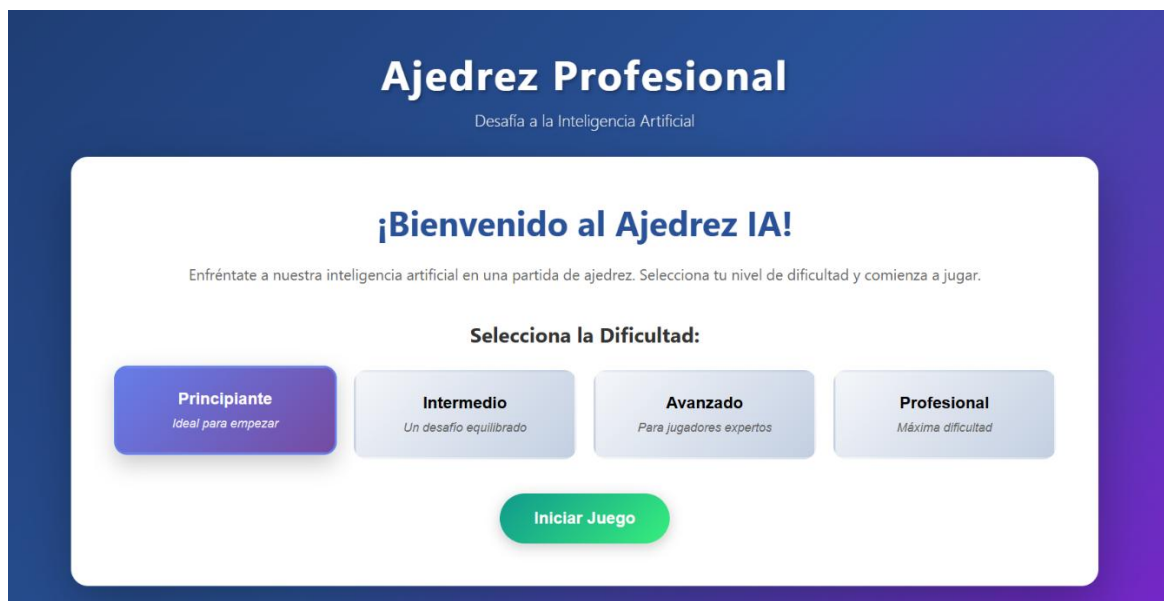
## 13. Posibles Mejoras Futuras

- Integración de base de datos
- Multijugador online
- Aperturas predefinidas
- Evaluación con tablas de finales
- Implementación de Stockfish como motor externo

## 14. Conclusión

Este proyecto demuestra la aplicación práctica de **algoritmos de inteligencia artificial**, **arquitectura web** y **desarrollo full stack**, integrando lógica compleja con una interfaz amigable y profesional.

## 15. Imágenes



# Ajedrez Profesional

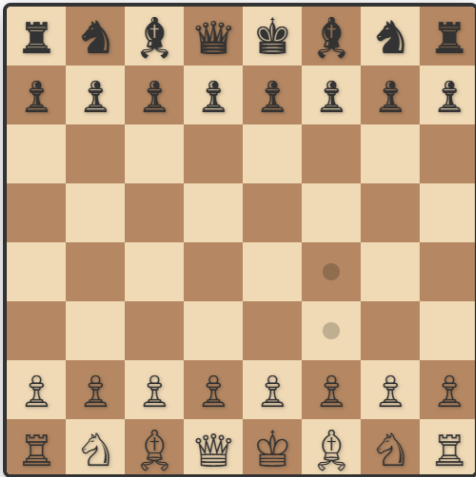
Desafía a la Inteligencia Artificial

Tú (Blancas)

Tu turno

IA (Negras)

Dificultad: Beginner



Reiniciar Partida

Nueva Configuración

# Ajedrez Profesional

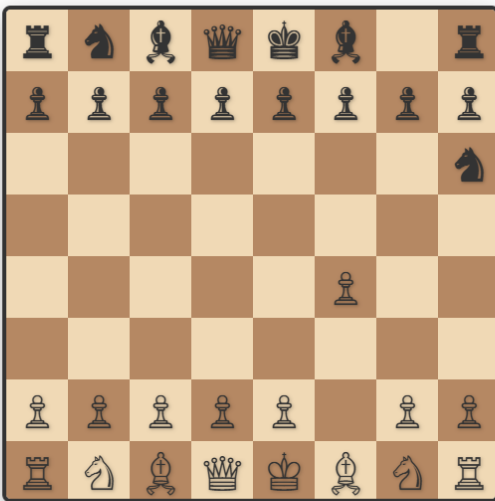
Desafía a la Inteligencia Artificial

Tú (Blancas)

Tu turno

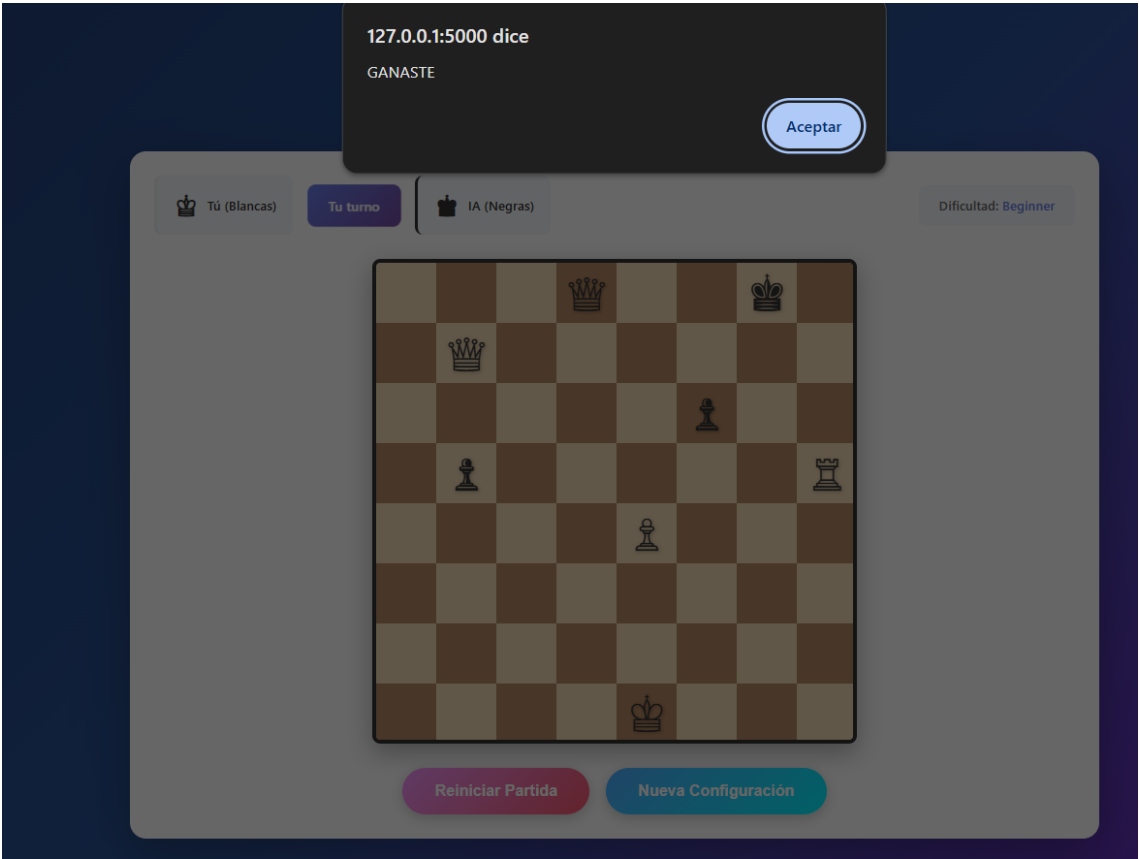
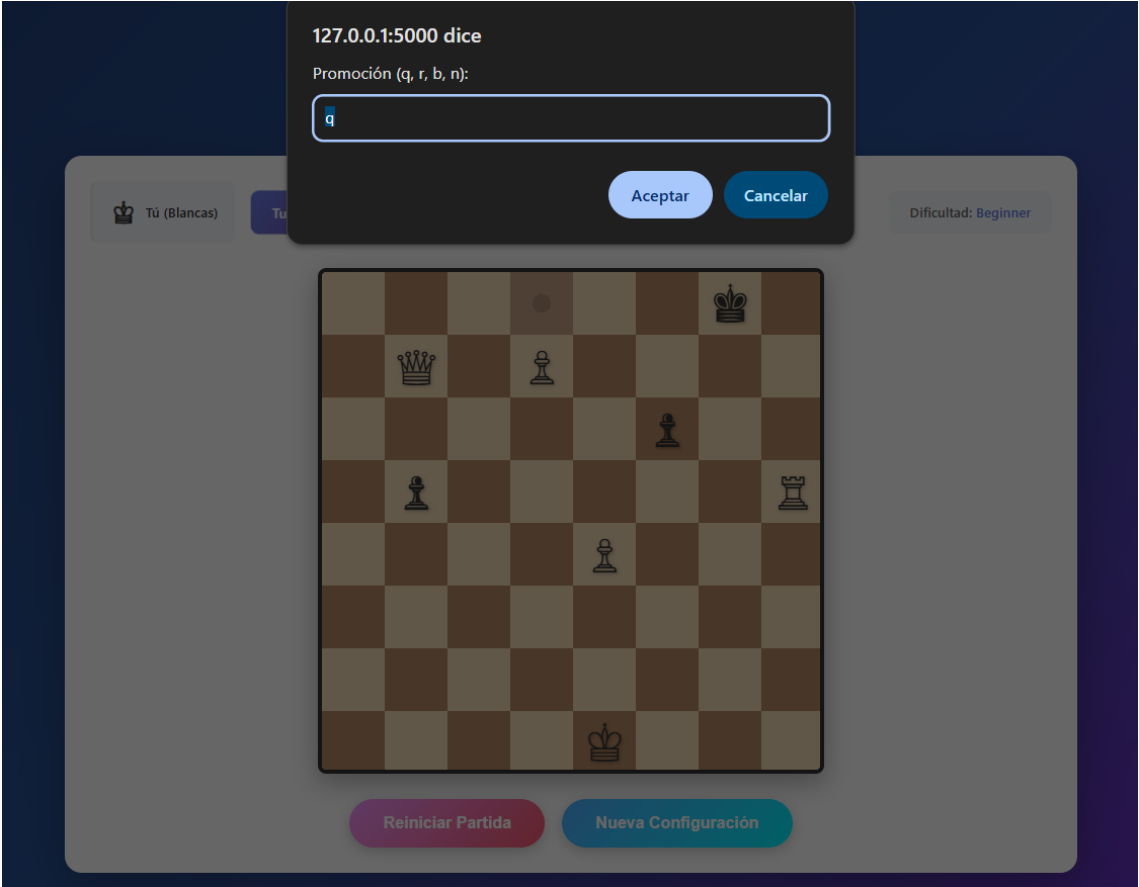
IA (Negras)

Dificultad: Beginner



Reiniciar Partida

Nueva Configuración





# Ajedrez Profesional

Desafía a la Inteligencia Artificial

## ¡Bienvenido al Ajedrez IA!

Enfrentate a nuestra inteligencia artificial en una partida de ajedrez. Selecciona tu nivel de dificultad y comienza a jugar.

Selecciona la Dificultad:

**Principiante**

*Ideal para empezar*

**Intermedio**

*Un desafío equilibrado*

**Avanzado**

*Para jugadores expertos*

**Profesional**

*Máxima dificultad*

Iniciar Juego

# Ajedrez Profesional

Desafía a la Inteligencia Artificial



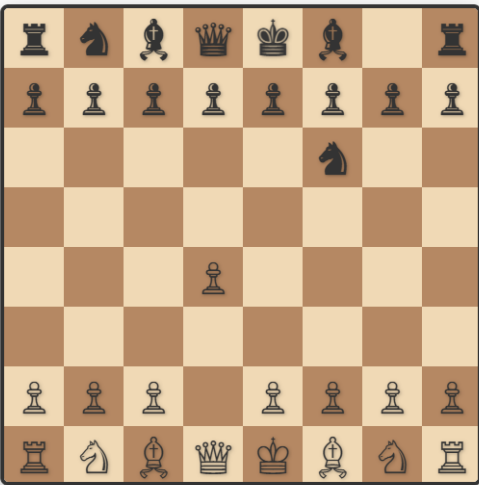
Tú (Blancas)

Tu turno



IA (Negras)

Dificultad: **Intermedia**



Reiniciar Partida

Nueva Configuración

127.0.0.1:5000 dice

PERDISTE

Aceptar



Tú (Blancas)

Tu turno



IA (Negras)

Dificultad: [Intermediate](#)



Reiniciar Partida

Nueva Configuración