

**Proyecto Final – IE-0117**  
**Programación Bajo Plataformas Abiertas**

Juego de Ajedrez en C utilizando la biblioteca **ncurses**

Emily Maryan Flores Rojas

II Ciclo 2025

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Objetivos</b>	<b>4</b>
2.1. Objetivo General . . . . .	4
2.2. Objetivos Específicos . . . . .	4
<b>3. Biblioteca Open Source Utilizada: ncurses</b>	<b>5</b>
<b>4. Diseño del Sistema</b>	<b>6</b>
4.1. 1. Motor del Juego (Lógica) . . . . .	6
4.2. 2. Interfaz mediante ncurses . . . . .	6
<b>5. Detalles de Implementación</b>	<b>7</b>
5.1. Compilación . . . . .	7
5.2. Ejecución . . . . .	7
5.3. Funciones Principales del Sistema . . . . .	7
<b>6. Resultados</b>	<b>8</b>
<b>7. Conclusiones</b>	<b>9</b>
<b>8. Trabajo Futuro</b>	<b>9</b>

# 1. Introducción

El presente documento corresponde al proyecto final del curso *IE-0117 Programación Bajo Plataformas Abiertas*. El objetivo principal del proyecto consiste en diseñar e implementar un programa funcional en lenguaje C que integre al menos una biblioteca *open source*. Para este trabajo se desarrolló un **juego de ajedrez completo** en consola, utilizando la biblioteca `ncurses` para mejorar la interfaz, habilitar colores y manejar la entrada de manera más eficiente.

El proyecto toma como base un motor de ajedrez previamente programado en C y se expande incorporando una interfaz más amigable, detección de jaque, jaque mate, ahogado, enroque, promoción de peones y validación general de movimientos.

## **2. Objetivos**

### **2.1. Objetivo General**

Desarrollar un juego de ajedrez completamente funcional en lenguaje C, integrando una biblioteca de software libre que mejore la experiencia en consola y promueva el uso de herramientas open source.

### **2.2. Objetivos Específicos**

- Implementar una interfaz visual utilizando la biblioteca `ncurses`.
- Adaptar el motor de ajedrez existente para interactuar con la nueva interfaz.
- Validar movimientos, jaque, jaque mate, ahogado, enroque y promoción.
- Documentar adecuadamente el proyecto siguiendo buenas prácticas.
- Crear un repositorio público que incluya el código completo y un README informativo.

### 3. Biblioteca Open Source Utilizada: **ncurses**

La biblioteca **ncurses** es una herramienta ampliamente utilizada en sistemas tipo UNIX para controlar interfaces de texto avanzadas. Permite:

- Crear interfaces basadas en terminal.
- Aplicar colores.
- Controlar posicionamiento del cursor.
- Gestionar entradas `getnstr()`, teclas especiales, ventanas, etc.

Fue elegida debido a su facilidad de integración, su amplia disponibilidad y porque cumple con el requisito del proyecto de emplear una biblioteca open source en el desarrollo del software.

La instalación se realiza con:

```
sudo apt install libncurses5-dev libncursesw5-dev
```

## 4. Diseño del Sistema

El programa se divide conceptualmente en dos grandes módulos:

### 4.1. 1. Motor del Juego (Lógica)

Incluye:

- Representación del tablero con una matriz 8x8.
- Estructura `Piece` para almacenar tipo y color.
- Validación de movimientos: peón, caballo, torre, alfil, dama y rey.
- Reglas especiales:
  - Enroque corto y largo.
  - Promoción de peón.
  - Detección de jaque.
  - Detección de jaque mate.
  - Detección de ahogado.

### 4.2. 2. Interfaz mediante ncurses

Incluye:

- Dibujo del tablero en la terminal.
- Colores diferenciados para piezas blancas y negras.
- Mensajes dinámicos en pantalla.
- Solicitud de jugadas en formato algebraico.
- Pausas para interacción del usuario.

## 5. Detalles de Implementación

### 5.1. Compilación

Para compilar el programa:

```
gcc ajedrez_ncurses.c -o ajedrez_ncurses -std=c99 -Wall -Wextra -lncurses
```

### 5.2. Ejecución

```
./ajedrez_ncurses
```

El usuario debe ingresar jugadas en formato algebraico simple como:

- e2e4
- g1f3
- e7e8Q

Para salir del juego:

```
salir
```

### 5.3. Funciones Principales del Sistema

Entre las funciones más relevantes:

- `initialize_board()`: crea el tablero inicial.
- `display_board_ncurses()`: dibuja el tablero con ncurses.
- `parse_algebraic_notation()`: interpreta movimientos.
- `mover()`: ejecuta jugadas, valida reglas y actualiza el estado.
- `is_in_check()`: determina si un rey está en jaque.
- `can_player_make_any_legal_move()`: determina mate/ahogado.

## 6. Resultados

El programa desarrollado cumple con:

- Juego completo de ajedrez en consola.
- Interfaz visual mejorada con colores gracias a `ncurses`.
- Detección precisa de jaque, jaque mate y ahogado.
- Enroque funcional.
- Promoción configurable (Q, R, N, B).



Figura 1: Captura de pantalla de la interfaz ncurses (tablero).

## 7. Conclusiones

El desarrollo de este proyecto permitió reforzar conocimientos de programación en C, manejo de estructuras de datos, validación de reglas formales y uso de bibliotecas open source. La integración con `ncurses` mejora significativamente la interacción, haciendo el juego más claro y visualmente atractivo sin recurrir a interfaces gráficas complejas.

Asimismo, el repositorio documentado y organizado cumple los lineamientos solicitados en el curso.

## 8. Trabajo Futuro

- Implementación de *en passant*.
- Regla de 50 movimientos.
- Detección de repetición triple.
- Exportar y cargar partidas en JSON usando `cJSON`.
- Crear interfaz gráfica con `SDL2` o `raylib`.