

ACCBM

Chess Simulator

DOCUMENTACIÓN

TABLA DE CONTENIDO

1	Objetivos.....	3
1.1	Objetivo general	4
1.2	Objetivos específicos.....	4
2	Definiciones y Especificaciones de Requisitos.....	4
2.1	Definición general del software.....	4
2.2	Público objetivo.....	4
2.3	Especificación de requerimientos del proyecto	4
2.3.1	Requerimientos del programa.....	6
2.3.2	Requerimientos funcionales	6
3	Procedimientos de Desarrollo	6
3.1	Arquitectura del Sistema	6
3.1.1	Descripción jerárquica:.....	7
3.1.2	Descripción individual de los paquetes:	8
3.2	Documentación técnica.....	7
4	Procedimiento de Pruebas.....	8

1 OBJETIVOS

1.1 OBJETIVO GENERAL

- Aplicar un diseño de software válido y funcional que permita jugar al ajedrez mediante el desarrollo de un aplicativo realizado en JAVA utilizando el Paradigma Orientado a Objetos.

1.2 OBJETIVOS ESPECÍFICOS

- Abstraer los conceptos y reglas reales que constituyen y abarcan al juego de ajedrez en el programa.
- Desarrollar un sistema extensible, entendible y modulable para el usuario, con el fin de que sea intuitivo en su uso.
- Definir un manual técnico que ayude a entender la lógica del código implementado en el programa.

2.1 DEFINICIÓN GENERAL DEL SOFTWARE

Este proyecto tiene como propósito general el diseño, emulación e implementación de un programa que permita jugar partidas de ajedrez mediante una interfaz gráfica de usuario con salida en pantalla en 2D, donde, por teclado se ingresan los nombres del jugador y los movimientos de piezas y acciones se ejecutan por ratón mediante la representación graficas de piezas en la interfaz.

2.2 PÚBLICO OBJETIVO

El sistema está enfocado a jugadores entusiastas del ajedrez quienes deberán tener un conocimiento básico de las reglas del juego. Este programa fue diseñado con el fin de que la utilización de este sea fácil de manejar por parte de los jugadores. En este sentido, únicamente se le solicita al usuario saber la manipulación básica de un computador.

2.3 ESPECIFICACIÓN DE REQUERIMIENTOS DEL PROYECTO

Conforme al desarrollo de una partida de ajedrez se establecieron los requerimientos que construye al proyecto realizado. Estos se muestran a continuación:

2.3.1 Requerimientos del programa

El sistema para emular partidas de ajedrez aplica los principios del POO con el fin de entregar al cliente un correcto diseño de software en la realización de este. De ahí que se enlista información necesaria del funcionamiento del ajedrez para aplicarlo y diseñarlo.

En el programa se emulan partidas de ajedrez, las cuales son correspondientes a las reglas reales de la misma. Se toma en cuenta la existencia de 16 piezas cada una según la cantidad indicada por las reglas de ajedrez en el tablero. Es así como también se pondrán a hacer movimientos, capturas de piezas y ganar partidas conforme a las reglas establecidas según el turno del usuario habilitado y los requerimientos de cada pieza.

De esta manera, se pusieron en estima los movimientos y consideraciones propios de cada pieza presente en el programa. Por ende, después de un arduo esfuerzo investigativo, se implementaron las funcionalidades en el programa en función a cada pieza. No está de más recordar que la partida se dará como finalizada cuando un rey haya muerto.

2.3.2 Requerimientos funcionales

Al momento de la ejecución el sistema permite al usuario la elección de jugar una nueva partida o si gusta cargar una previamente guardada. Si se inicia una nueva partida en la pantalla se desplegará un submenú para ingresar los nombres de los jugadores. Por defecto, el jugador 1 será el que llevará las piezas claras y el jugador 2 las piezas negras. Posteriormente, se mostrará un tablero de ajedrez con todas las piezas organizadas conforme el funcionamiento de la vida real.

Por otro lado, si el usuario desea cargar una partida guardada, esta contará con las posiciones de cada una de las piezas tal y como se ubicaron en el momento de guardar dicha partida. De esta manera, si los jugadores deciden terminar el encuentro en otro momento, se mantendrán todos los movimientos previos.

En curso de partida, el jugador podrá mover una pieza en su respectivo turno arrastrándola por el tablero. Si el usuario desea una ayuda para el movimiento de las piezas, bastará con darle clic a la ficha requerida para saber las posibles posiciones en las que se puede mover. Del mismo modo, si desea una ayuda general de jugabilidad o del cómo se mueven las piezas podrá mantener el clic izquierdo en el botón de mostrar ayuda. Obviamente, cada jugador podrá comer piezas del contrincante y no de sí mismo.

Las piezas, como se mencionó en el apartado anterior, tienen movimientos y consideraciones propias, por lo que se las especifica a continuación (véase la Tabla 1.):

Tabla 1.
Requerimientos generales de cada pieza de ajedrez.

Piezas	
Movimientos	Consideraciones
Peón	
<ul style="list-style-type: none"> Al inicio pueden empezar con un movimiento de 1 o 2 casillas. El segundo movimiento de cada peón sólo va de una casilla en diagonal (cuando come) o un movimiento vertical de 1 casilla. No puede retroceder. Si existe una pieza enemiga delante de ellos no se pueden mover. 	<ul style="list-style-type: none"> Ataca solo en diagonal en caso de haber una pieza en esa casilla de llegada. Si llega al principio del lado contrario puede tomar el estado de cualquier figura que el jugador desee.
Rey	
<ul style="list-style-type: none"> Se puede mover una casilla alrededor en cualquier dirección. 	<ul style="list-style-type: none"> Según su estado se evalúa el estado de la partida
Reina	
<ul style="list-style-type: none"> Se puede mover diagonal, vertical y horizontalmente cuantas casillas sea posible. 	<ul style="list-style-type: none"> No pueden atravesar piezas durante su trayectoria.
Alfil	
<ul style="list-style-type: none"> Se puede mover diagonalmente cuantas casillas sea posible. 	<ul style="list-style-type: none"> No puede realizar un movimiento atravesando las piezas.
Torre	
<ul style="list-style-type: none"> Se puede mover horizontal y verticalmente cuantas casillas sea posible. 	<ul style="list-style-type: none"> No pueden atravesar piezas durante su trayectoria.
CABALLO	
<ul style="list-style-type: none"> Se puede mover libremente en forma de “L” por el tablero. 	<ul style="list-style-type: none"> Es la única pieza que puede pasar por encima de otras piezas. Se considera un aumento largo y otro corto.

3.1 ARQUITECTURA DEL SISTEMA

Principalmente, el programa se subdivide en **logica**¹ la cual maneja la programación de objetos y sus comportamientos según cada requerimiento de cada pieza conforme al POO. Por otro lado, la **Graphic User Interface**, como ya se lo mencionó, es la que construye la parte visual e interactiva del sistema realizado por el equipo de trabajo.

3.1.1 Descripción módulos:

El proyecto se subdivide en: [UML_package.png](#)

3.1.1.1 Lógica (logica)

Define el comportamiento de la partida y las acciones implicadas en una partida de ajedrez, controlando las consideraciones y posibles excepciones que pueda llegar a tener un movimiento. Dentro de este módulo se tiene varios paquetes, explicados en 3.1.2.

Enlace del UML del software: [UML_Chess.png](#)

3.1.1.2 Graphical User Interface (GUI_Chess)

El funcionamiento de este módulo depende del módulo **logica**¹, por lo tanto, toda modificación en dicho modulo se reflejará en el programa visual debido a la dependencia mencionada. Sin embargo, se debe tomar en cuenta que la programación de la GUI también tiene su propio apartado, por lo que, cambios más significativos fuera de las reglas básicas del ajedrez en **logica**¹ pueden que tengan que ser diseñados individualmente en **GUI_Chess**¹.

Enlace del UML del GUI: [UML_GUI_Chess.png](#)

3.1.2 Descripción individual de los paquetes:

3.1.2.1 Exceptions (excepción²)

Paquete en el que se hallan todas las excepciones creadas para identificar los posibles errores dentro del código. La clase Excepción principal o de más alto nivel en el diseño es ExcepcionAjedrez que, como todas las excepciones propias creadas dentro de un código, extiende de Exception. Mediante un UML se explican las relaciones entre estas excepciones.

Enlace al UML de las excepciones: [exceptions.png](#)

3.1.2.2 Interfaces de movimiento (interfaz²)

En este paquete se hallan las interfaces que permiten controlar los movimientos de todas las piezas del ajedrez. Las fichas creadas implementan de una interfaz acorde su movimiento para poder ser correctamente diseñada en su funcionamiento. Como es de esperar, cada interfaz de movimiento también determinará las excepciones que se esperan que sucedan.

¹ Es el nombre propio del módulo utilizado en el programa, por ende, no se colocó con tilde en ninguna parte del documento que haga referencia al módulo.

² Es el nombre propio del paquete utilizado en el programa, por ende, se omiten reglas de puntuación en todos los títulos.

3.1.2.3 Piezas (*pieza*²)

Se encuentran alojadas las clases de todas las piezas creadas. Todas las fichas extienden o se heredan de la clase padre Pieza presente en este paquete. Algunas clases hacen uso de clases externas al paquete para enlazar su funcionalidad dentro del programa realizado.

Enlace al javadoc que describe las clases de este paquete: [pieza](#)

3.1.2.4 Color de las piezas (*color*²)

Se halla la enumeración ColorAjedrez que contiene claro y oscuro.

3.2 DOCUMENTACIÓN TÉCNICA

Si desea conocer más acerca del funcionamiento específico del programa deberá descargar la carpeta con el javadoc generado, por ende, la documentación técnica realizada está disponible en:

Enlace: [javadocGenerado](#)

4 PROCEDIMIENTO DE PRUEBAS

Figura 1.

Excepción de auto ataque en el programa.



Figura 2.

Excepción del atravesamiento de piezas para un movimiento.

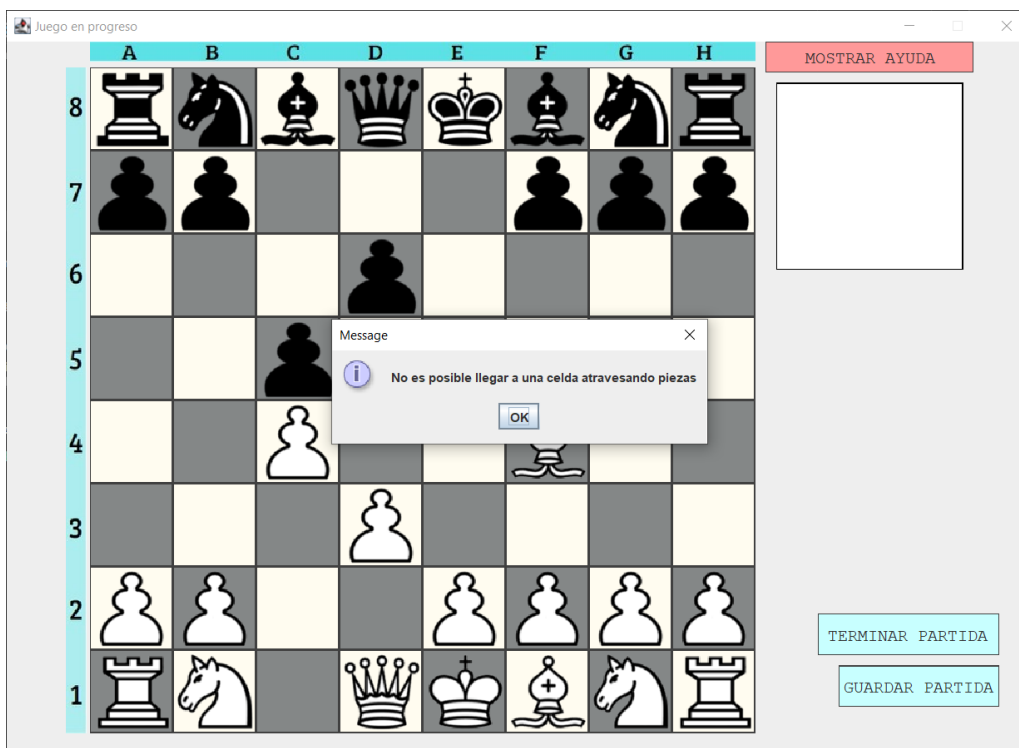


Figura 3.

Excepción de turno.

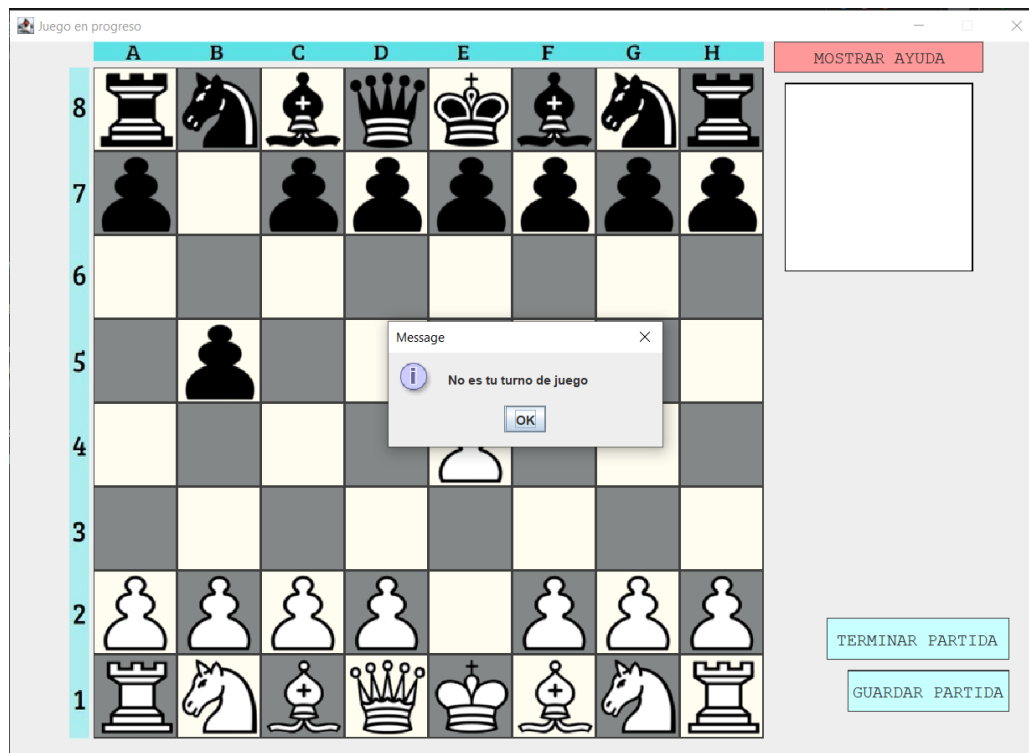


Figura 4.

Excepción de movimiento diagonal.

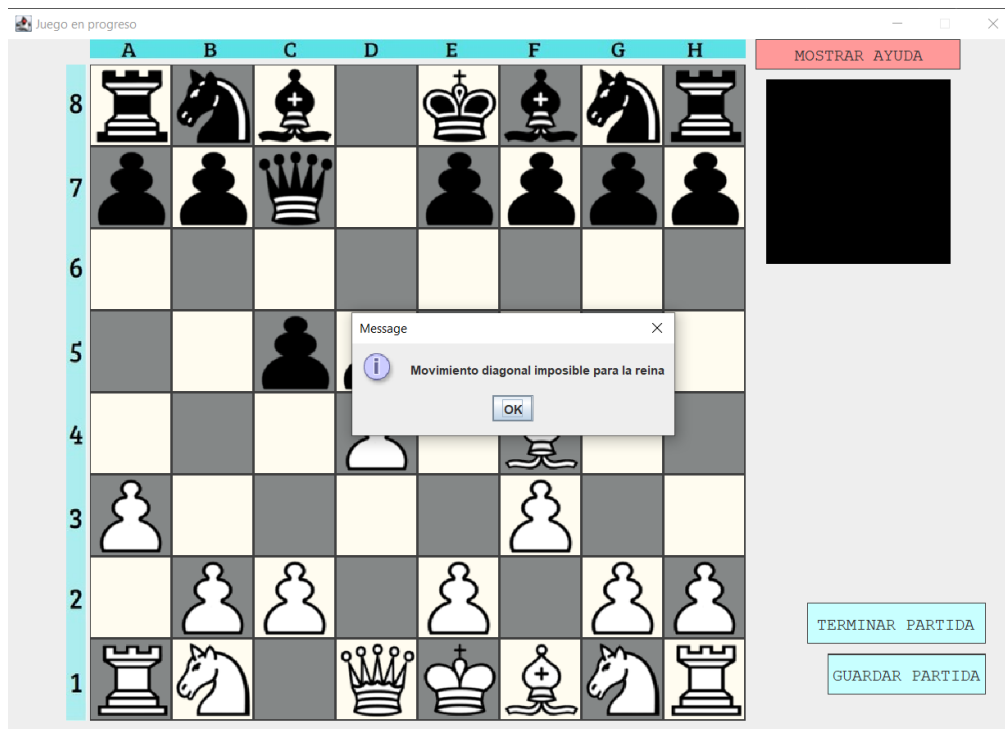


Figura 5.

Excepción de movimiento en L para el caballo.

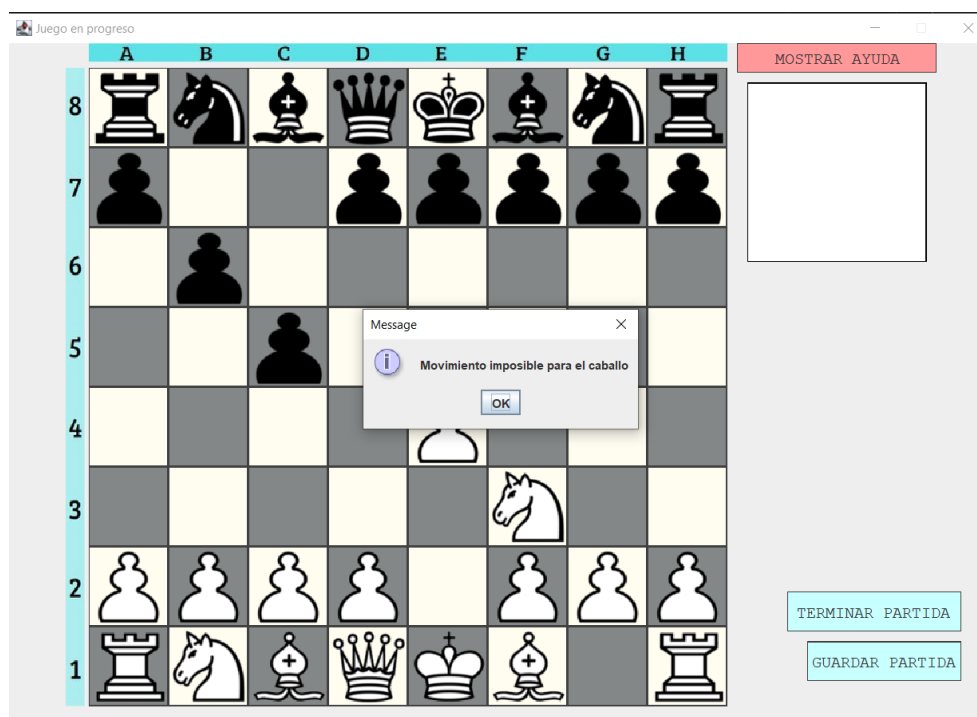


Figura 6.

Excepción de movimiento horizontal.

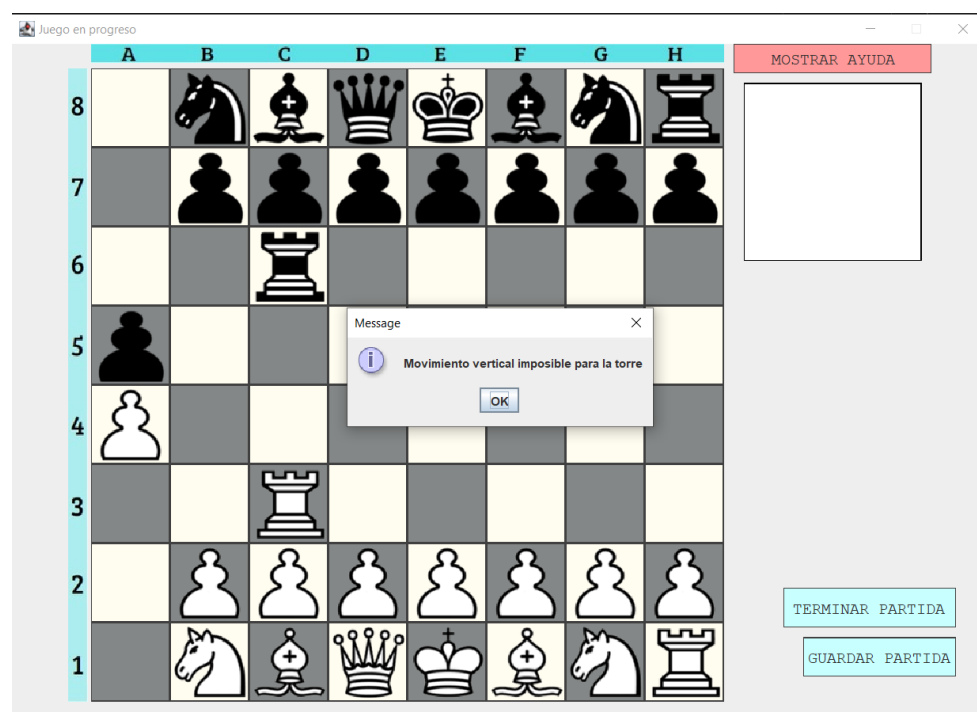


Figura 7.

Excepción de movimiento vertical.

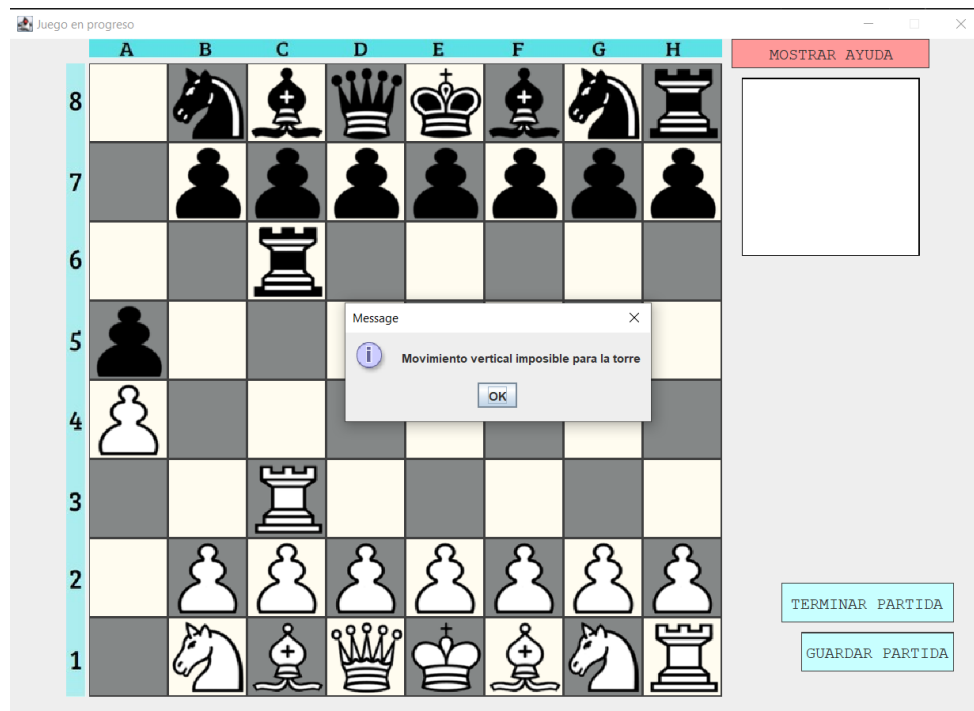


Figura 8.

Excepción mínima para el movimiento de una pieza.

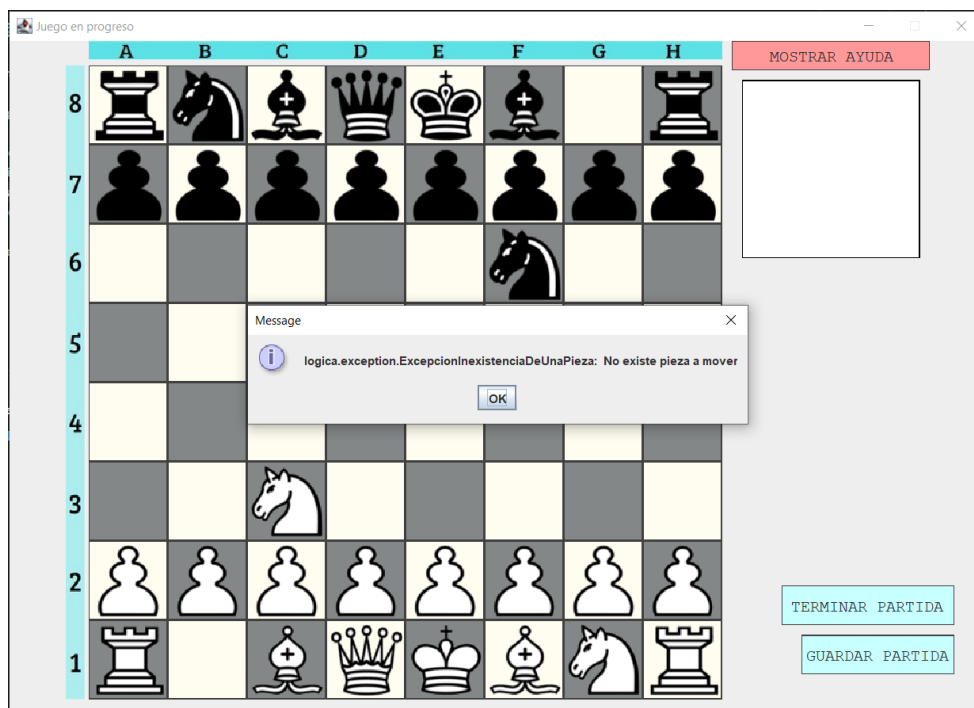


Figura 9.

Excepción para posición fuera de rango en el tablero.

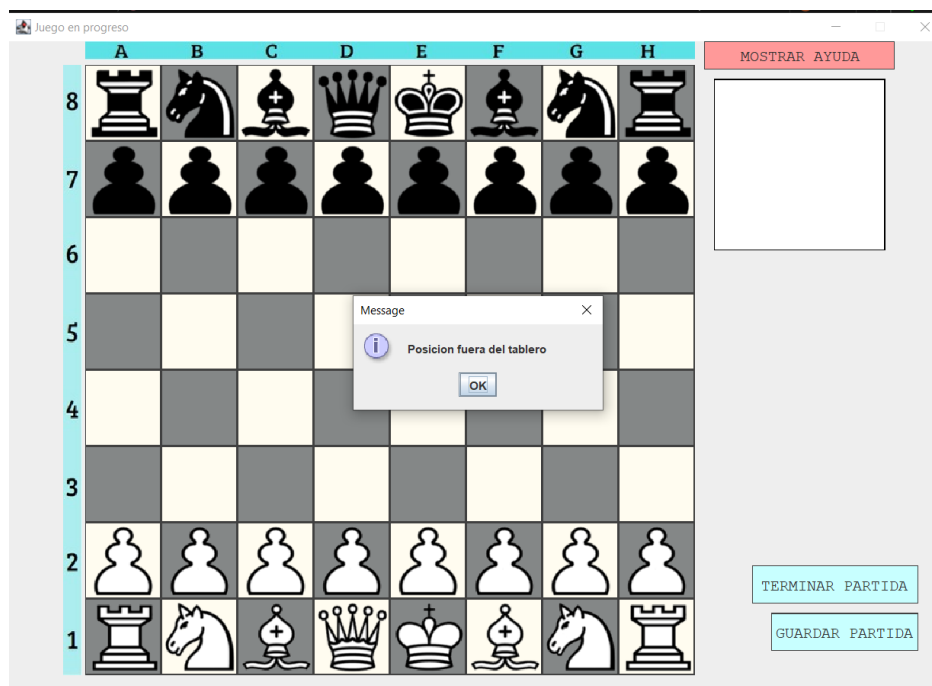


Figura 10.

Cambio del peón a otra pieza al llegar al final del lado opuesto a su color.

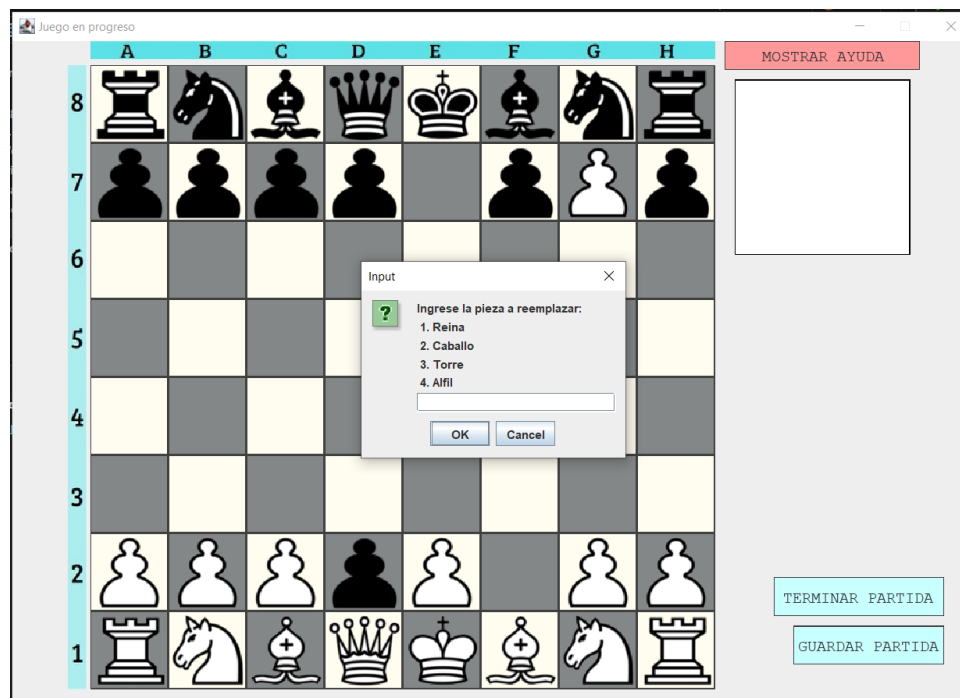


Figura 11.

Verificación del ganador de la partida acorde al turno correspondiente.

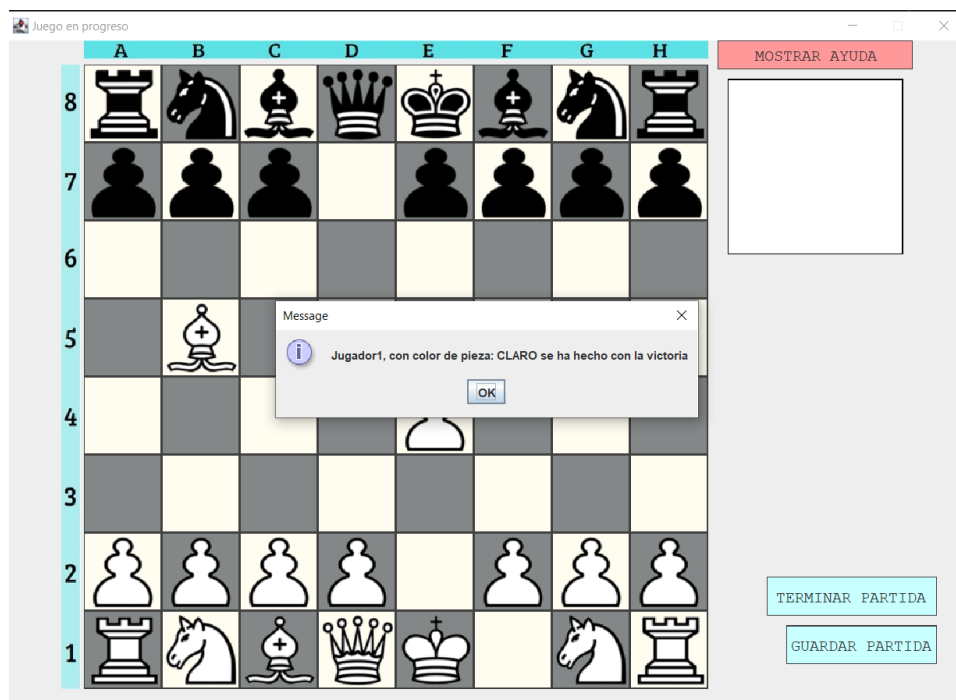


Figura 12.

Excepción de cómo captura el peón.

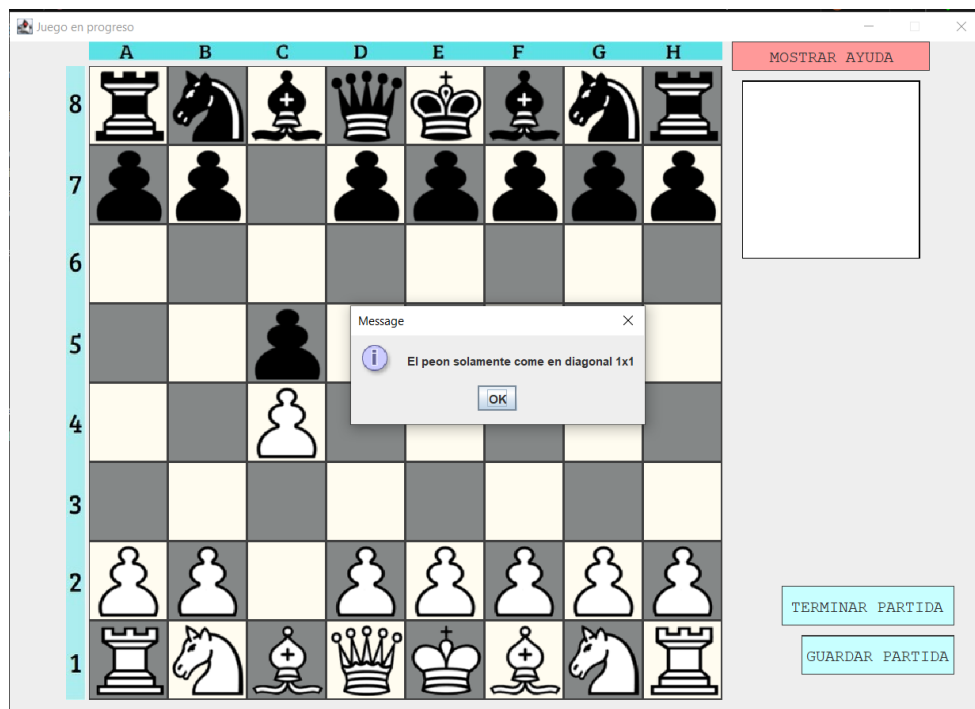
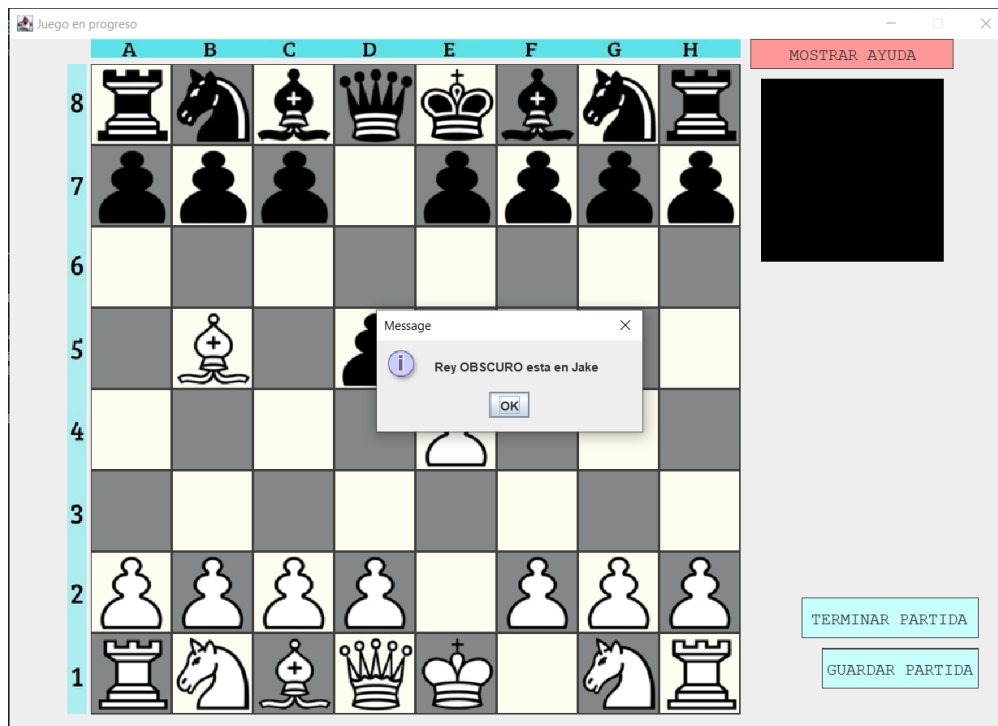


Figura 13

Mensaje de advertencia de Jaque



5 REFERENCIAS BIBLIOGRÁFICAS

Braga, F., Criado, P., Minzer, C., & Montoto J. (2006). *Manual de Ajedrez*. Biblioteca virtual de Madrid-España.

W3Schools. (2021). Java Iterator. Obtenido de [Java Iterator \(w3schools.com\)](https://www.w3schools.com/java/java_iterator.asp).