

**Universidad Francisco José de Caldas**

**Manual técnico**

**Juego Ajedrez**

Desarrollado en java



David Santiago Buitrago C. 20221020085

Henry Ricaurte 20221020084

Julian Andres Vargas M. 20221020069

Bogotá 12 de septiembre 2023

Contenido

[1.](#_gjdgxs) Introducción 3

[2.1 Objetivo 3](#_30j0zll)

[2.2 Audiencia 3](#_1fob9te)

[2.](#_3znysh7) Marco teórico 4

[3.](#_2et92p0) Descripción General de la Aplicación 5

[4.1Funcionalidades clave. 5](#_tyjcwt)

[5. Instalación y Configuración 22](#_3dy6vkm)

[6.Uso de la aplicación 23](#_1t3h5sf)

[7.Conclusiones 25](#_4d34og8)

# Introducción

## 2.1 Objetivo

En este laboratorio de programación, el objetivo principal es desarrollar un ajedrez básico utilizando el lenguaje de programación de JAVA. El juego de ajedrez será capaz de :

**Pasos y Componentes Clave:**

Representación del Tablero: Creamos una representación del tablero de ajedrez utilizando estructuras de datos adecuadas. Esto incluirá la ubicación de las piezas, el estado de las casillas, el turno del jugador y la información sobre enroques, capturas al paso, etc.

Reglas del Ajedrez: Implementar las reglas básicas del ajedrez, como la forma en que se mueven las piezas, las reglas para el jaque y el jaque mate, el enroque y la captura al paso.

Generación de Movimientos Válidos: Desarrollar algoritmos para generar todos los movimientos válidos para el jugador en turno. Esto incluirá movimientos de todas las piezas y verificar si un movimiento resulta en un estado ilegal del tablero.

Juego de Ajedrez Completo: Integrar todas las piezas y reglas para que el motor pueda jugar una partida completa de ajedrez contra otro jugador humano o contra sí mismo.

Interfaz de Usuario: Si se dispone de tiempo, los estudiantes pueden desarrollar una interfaz de usuario básica para que los jugadores humanos interactúen con el motor de ajedrez.

Pruebas y Depuración: Los estudiantes deben probar exhaustivamente su motor de ajedrez y corregir cualquier error o comportamiento inesperado.

## 2.2 Audiencia

El manual técnico del juego ajedrez está escrito pensando en un público técnico, centrándose en particular en los ingenieros de sistemas y los profesores. Está destinado a personas con conocimientos técnicos y conocimientos fundamentales de principios de informática y programación. El objetivo principal de esta guía es brindar a los jugadores instrucciones completas sobre cómo jugar AJEDREZ, incluidas reglas del juego, consejos para ganar y técnicas efectivas para resolver acertijos.Si los profesores de ingeniería de sistemas quieren utilizar el juego como recurso didáctico en sus clases, puede haber información adicional en el manual que deben conocer. Esto podría incluir ideas para incorporar juegos en lecciones sobre razonamiento, toma de decisiones y resolución de problemas en el contexto de la informática.

# Marco teórico

El ajedrez tiene su origen en la India, más concretamente en el Valle del Indo, y data del siglo VI d.C. Originalmente conocido como Chaturanga, o juego del ejército, se difundió rápidamente por las rutas comerciales, llegó a Persia, y desde allí al Imperio bizantino, extendiéndose posteriormente por toda Asia.

**Conceptos clave:**

Tablero de Ajedrez: El tablero de ajedrez es un cuadrado de 8x8 con 64 casillas alternas de colores (generalmente blanco y negro). Cada jugador tiene una fila de piezas al frente del tablero.

Piezas de Ajedrez: El ajedrez consta de seis tipos diferentes de piezas: el rey, la dama, la torre, el alfil, el caballo y el peón. Cada pieza tiene reglas específicas para su movimiento y captura.

Movimiento y Captura: Cada pieza tiene un conjunto específico de movimientos permitidos. Los movimientos se hacen en un patrón de cuadrícula y las piezas se pueden mover horizontal, vertical o diagonalmente. Para capturar una pieza enemiga, una pieza se mueve a la casilla ocupada por la pieza enemiga.

Rey: El rey es la pieza más importante. El objetivo del juego es poner al rey en jaque mate, lo que significa que el rey está amenazado y no puede moverse a ninguna casilla segura.

Dama: La dama es la pieza más poderosa. Puede moverse en cualquier dirección (horizontal, vertical o diagonal) cualquier cantidad de casillas.

Torre: La torre se mueve horizontal o verticalmente cualquier cantidad de casillas.

Alfil: El alfil se mueve en diagonal cualquier cantidad de casillas y siempre se mantiene en el mismo color de casilla en el que comenzó el juego.

Caballo: El caballo tiene un movimiento especial en forma de "L". Puede avanzar dos casillas en una dirección (horizontal o vertical) y luego una casilla en una dirección perpendicular.

Peón: Los peones se mueven hacia adelante una casilla a la vez, pero capturan diagonalmente. En su primer movimiento, un peón puede avanzar dos casillas.

Enroque: El enroque es una maniobra defensiva en la que el rey y una de las torres se mueven juntas. Puede ser del lado corto o del lado largo. El enroque tiene reglas específicas, como que el rey y la torre no deben haberse movido antes y que no debe haber piezas entre ellas.

Promoción del Peón: Cuando un peón llega a la última fila del tablero, se promociona a cualquier otra pieza (generalmente una dama) que elija el jugador.

Captura al Paso: Si un peón avanza dos casillas desde su posición inicial y queda al lado de un peón enemigo, este último puede capturar al peón que se movió dos casillas como si hubiera avanzado solo una casilla.

Jaque: Cuando el rey está amenazado por una pieza enemiga, se dice que está en jaque. El jugador en jaque debe tomar medidas para proteger al rey.

Jaque Mate: El objetivo del juego es dar jaque mate al rey enemigo, lo que significa que el rey está en jaque y no puede escapar a un movimiento legal. El juego termina cuando se produce el jaque mate.

**Algoritmos y estructuras de datos:**

**Estructura del Tablero:** El tablero de ajedrez se puede representar mediante una matriz bidimensional de 8x8 o una estructura de datos similar. Cada celda de la matriz puede contener información sobre la pieza presente en esa posición, como el tipo de pieza (rey, dama, torre, etc.) y el color (blanco o negro).

**Generación de Movimientos Válidos:** Para determinar los movimientos legales en una posición dada, necesitas un algoritmo que pueda generar todos los movimientos posibles para cada pieza en función de las reglas del ajedrez. Esto incluye movimientos de piezas como el enroque, la captura al paso y la promoción de peones.

**Programación y diseño de software:**

El marco teórico proporcionado aquí es un punto de partida y puede expandirse o personalizarse según las necesidades específicas del laboratorio de programación y el nivel de los estudiantes. El objetivo principal es proporcionar a los estudiantes una comprensión sólida de los conceptos fundamentales y las habilidades de programación necesarias para crear un juego de Buscaminas funcional.

# Descripción General de la Aplicación

## 4.1Funcionalidades clave.

* **Tablero de juego:** La aplicación debe proporcionar un tablero de juego que consta de un campo cuadriculado, donde se encuentran las casillas del juego.
* **Minas:** Se deben colocar minas de forma aleatoria en el tablero antes de que comience el juego. El número de minas y su ubicación también puede variar según la dificultad seleccionada.
* **Casillas:** Cada casilla en el tablero puede estar en uno de los siguientes estados:
  + Cubierta: El jugador aún no ha interactuado con ella.
  + Descubierto: El jugador ha hecho clic en la casilla y se revela su contenido.
  + Marcada: El jugador ha marcado una casilla sospechosa de contener una mina con una bandera o un símbolo equivalente.
* **Contenido de las casillas:** Las casillas pueden contener una mina (en cuyo caso el juego termina si se descubre), un número que indica cuántas minas están adyacentes a esa casilla o están vacías.
* **Interacción del jugador:** Los jugadores deben poder hacer clic en las casillas para descubrir su contenido o marcarlas como sospechosas de contener una mina. También deben poder deshacer acciones si cometen un error.
* **Reglas del juego:** La aplicación debe seguir las reglas estándar del Buscaminas, como el hecho de que si se descubre una casilla vacía, se revelen automáticamente todas las casillas adyacentes vacías hasta que se alcance una casilla con números.
* **Dificultad ajustable:** Los jugadores deben poder elegir entre diferentes niveles de dificultad que afecten al tamaño del tablero y la cantidad de minas presentes.
* **Personalización**: La aplicación puede permitir a los jugadores personalizar el aspecto del juego, como el diseño del tablero y los iconos utilizados para las minas y las casillas marcadas.

Estas son las funcionalidades clave que se esperan en una aplicación de Buscaminas típica, aunque algunas versiones pueden agregar características adicionales para aumentar la diversión y el desafío.

**4.3 Diagrama UML**

# 5. Instalación y Configuración

**5.1 Requisitos de hardware y software:**

Estos son los requisitos que deberá tene tu dispositivo para poder ejecutar el programa con buena eficacia

**Requisitos de Hardware:**

* **CPU:** Cualquier procesador moderno debería ser suficiente. No se requiere una CPU potente para jugar Buscaminas.
* **Memoria RAM:** Una pequeña cantidad de RAM es suficiente para ejecutar el juego de Buscaminas. Incluso 1 GB de RAM es más que suficiente.
* **Tarjeta gráfica:** No se requiere una tarjeta gráfica dedicada. La mayoría de las tarjetas gráficas integradas en las computadoras modernas son adecuadas para el juego.
* **Espacio en disco duro:** El juego de Buscaminas es muy liviano y ocupa una cantidad insignificante de espacio en disco, generalmente menos de 100 MB.
* **Sistema operativo:** El juego de Buscaminas está disponible en una variedad de sistemas operativos, incluyendo Windows, macOS y Linux. Asegúrate de que tu sistema operativo sea compatible con el juego.

**Requisitos de Software:**

* **Sistema operativo:** El juego de Buscaminas suele estar preinstalado en sistemas Windows, por lo que en la mayoría de las computadoras con Windows, no se necesita software adicional. Si deseas jugar en macOS o Linux, es posible que necesites una versión del juego específica para tu sistema operativo o usar una implementación de código abierto.
* **Java (opcional):** Algunas versiones del juego de Buscaminas están escritas en Java. Si estás utilizando una versión basada en Java, necesitarás tener Java Runtime Environment (JRE) instalado en tu computadora.
* **Resolución de pantalla:** La mayoría de las computadoras modernas cumplen con los requisitos de resolución de pantalla para jugar Buscaminas. No se requiere una resolución alta.
* **Sonido (opcional):** Si deseas disfrutar de efectos de sonido en el juego, necesitarás una tarjeta de sonido y altavoces o auriculares conectados.

**5.2 Instrucciones para la instalación**

* Descarga el código fuente: Asegúrate de que tienes la carpeta que contiene el código fuente del juego de Buscaminas en tu computadora. Puede ser un archivo ZIP que descomprime previamente.
* Instala las dependencias (si es necesario): Verifica si el juego requiere la instalación de algún software adicional, como un entorno de desarrollo específico o una versión específica de un lenguaje de programación.
* Compila el código (si es necesario): Si el código fuente no está precompilado, abre una terminal o línea de comandos, navega hasta la carpeta que contiene el código fuente y compila el código utilizando el comando o la herramienta adecuada. Esto puede variar según el lenguaje de programación del juego.
* Ejecuta el juego: Una vez que el código se haya compilado (si es necesario), busca el archivo de inicio o el archivo principal del juego..
* Disfruta del juego: El juego de Buscaminas debería iniciarse y mostrarse en tu pantalla. Juega el juego siguiendo las reglas y los controles proporcionados por el juego.

# 6.Uso de la aplicación

**6.1 Inicio rápido**

* Abrir el juego: Abre el juego Buscaminas en tu sistema operativo. En Windows, puedes encontrarlo en el menú "Inicio" o buscarlo en el cuadro de búsqueda.
* Seleccionar una casilla: Haz clic izquierdo en una casilla para revelarla. El juego comenzará a revelar casillas, y si no hay una mina en la casilla seleccionada, verás un número que indica cuántas minas están adyacentes a esa casilla. Esto te proporciona pistas.
* Identificar minas: Utiliza las pistas numéricas para identificar la ubicación de las minas. Por ejemplo, si una casilla muestra el número "1", significa que una mina está adyacente a esa casilla.
* Marcar minas con banderas: Si crees que una casilla contiene una mina, haz clic derecho en ella para colocar una bandera. Esto marca la casilla como sospechosa de contener una mina.
* Continuar explorando: Continúa seleccionando casillas y marcando minas con banderas hasta que hayas revelado todas las casillas vacías. El juego termina cuando todas las casillas vacías están descubiertas o cuando haces estallar una mina.
* Ganar o perder: Si revelas todas las casillas vacías sin hacer estallar una mina, ganas el juego. Si haces estallar una mina, pierdes y el juego termina

**6.2 Funciones detalladas.**

* Tablero de Juego: El juego se juega en un tablero cuadriculado donde se ocultan casillas que pueden contener minas o números que indican cuántas minas están adyacentes.
* Minas: El objetivo principal es evitar las minas. Si seleccionas una casilla que contiene una mina, el juego termina y pierdes.
* Pistas Numéricas: Las casillas que no contienen minas muestran un número que indica cuántas minas están adyacentes. Estas pistas ayudan a los jugadores a identificar la ubicación de las minas.
* Selección de Casillas: Puedes hacer clic izquierdo en una casilla para revelar su contenido. Si no hay mina en esa casilla, se mostrará un número o estará vacía, revelando casillas adyacentes automáticamente si son seguras.
* Marcación de Minas con Banderas: Puedes hacer clic derecho en una casilla para marcarla con una bandera, indicando que crees que esa casilla contiene una mina. Esto es útil para recordar qué casillas consideras peligrosas.
* Descubrimiento de Casillas Vacías: Cuando seleccionas una casilla vacía (sin número ni mina adyacente), se revelarán automáticamente todas las casillas vacías adyacentes, lo que facilita la progresión en el juego.
* Niveles de Dificultad: El juego generalmente ofrece diferentes niveles de dificultad, que varían en el tamaño del tablero y la cantidad de minas ocultas. Los niveles más difíciles tienen tableros más grandes y más minas.

# 7.Conclusiones

El desarrollo de un juego de Buscaminas en el lenguaje de programación Java ha sido un proceso educativo y práctico que ha permitido comprender y aplicar diversos conceptos y técnicas de programación. A continuación, se presentan las conclusiones clave derivadas de este laboratorio:

1. Aprendizaje de Java y Programación Orientada a Objetos (POO)

Desarrollar un juego de Buscaminas en Java ha proporcionado una oportunidad valiosa para aprender y aplicar los principios de la Programación Orientada a Objetos (POO). Se ha creado una estructura de clases que representan objetos del mundo real, como el tablero, las celdas, los jugadores y las minas. La modularidad de Java ha facilitado la organización y la reutilización de código.

2. Gestión de Eventos y GUI

La creación de la interfaz gráfica de usuario (GUI) para el juego ha sido un desafío importante. Se ha aprendido a gestionar eventos de usuario, como clics de ratón, y a diseñar una interfaz de usuario intuitiva y atractiva. El uso de bibliotecas como Swing ha simplificado la creación de componentes de interfaz de usuario.

3. Lógica del Juego

Implementar la lógica del juego Buscaminas ha requerido un análisis cuidadoso y una planificación detallada. Se han desarrollado algoritmos para generar el tablero, colocar las minas, calcular las pistas y manejar las acciones del jugador. Esto ha demostrado la importancia de la lógica detrás del juego para su funcionamiento correcto y desafiante.

4. Tratamiento de Excepciones

Durante el desarrollo, se ha tenido en cuenta la gestión de excepciones para manejar situaciones inesperadas o errores del usuario. Esto ha mejorado la robustez del juego y ha evitado posibles bloqueos o comportamientos incorrectos.

5. Pruebas y Depuración

El proceso de desarrollo ha incluido pruebas exhaustivas y depuración para identificar y corregir errores y problemas de funcionamiento. La depuración ha sido esencial para garantizar que el juego sea confiable y esté libre de errores críticos.

6. Mejoras y Futuras Extensiones

El juego de Buscaminas desarrollado en este laboratorio sirve como punto de partida para futuras mejoras y extensiones. Se pueden agregar características adicionales, como estadísticas, niveles de dificultad o modos multijugador, para hacer el juego aún más atractivo y desafiante.

7. Colaboración y Trabajo en Equipo

Si se realizó el laboratorio en un entorno de trabajo en equipo, se ha experimentado la colaboración y la comunicación efectiva entre los miembros del grupo. El trabajo en equipo es esencial en proyectos de desarrollo de software y se han adquirido habilidades valiosas en este aspecto.

8. Continuación del Aprendizaje

El desarrollo de un juego de Buscaminas en Java es solo el comienzo del aprendizaje en programación y desarrollo de software. Se ha desarrollado una base sólida de conocimientos y habilidades que se pueden aplicar en proyectos más complejos y desafiantes en el futuro.

El desarrollo de un Buscaminas ha sido una experiencia educativa enriquecedora que ha permitido aplicar conceptos de programación, trabajar en la creación de una GUI, comprender la lógica del juego y mejorar las habilidades de depuración. Este laboratorio sienta las bases para continuar aprendiendo y desarrollando habilidades en el campo de la programación y el desarrollo de software.

.