Universidad Técnica Nacional

Sede Regional San Carlos

Ingeniería en Tecnologías de Información

Logotipo

Descripción generada automáticamente

Programación I, II-25, SC-01

Proyecto programado II | Buscaminas

Profesor:

[Andrés Joseph Jiménez Leandro](https://campusvirtual.utn.ac.cr/user/view.php?id=18905&course=84030)

Propuesto por:

Emesis Mayerli Mairena Sevilla

Jairo Steven Herrera Romero

AÑO: 2025

Tabla de contenidos

[Resumen ejecutivo 3](#_Toc206215981)

[Descripción 5](#_Toc206215982)

[Abstracto 6](#_Toc206215983)

[Objetivos 8](#_Toc206215984)

[Objetivo general 8](#_Toc206215985)

[Objetivos específicos 8](#_Toc206215986)

[Introducción 9](#_Toc206215987)

[Marco teórico 10](#_Toc206215988)

[Desarrollo 12](#_Toc206215989)

[Descripción del problema 12](#_Toc206215990)

[Diagrama de Clases 14](#_Toc206215991)

[Análisis de la solución 15](#_Toc206215992)

[Discusión de resultados obtenidos 16](#_Toc206215993)

[Conclusión 17](#_Toc206215994)

[Recomendaciones 18](#_Toc206215995)

[Cronograma de trabajo 19](#_Toc206215996)

[Bitácora 20](#_Toc206215997)

[Referencias bibliográficas 21](#_Toc206215998)

Resumen ejecutivo

Este proyecto estará implementado en el lenguaje de java, con el propósito de crear un video juego que fue muy conocido en los años noventa, este consta de un ambiente grafico bastante perceptible a la vista, para que el usuario pueda experimentar como era el juego en aquellos tiempos, este se llama Buscaminas el consta de una particularidad de tener suerte y perspectiva de poder adivinar en que sitios se encuentran las bombas y no tocarlas para no perder la partida.

Este juego es conocido por destapar casillas en las cuales dependiendo de la técnica que se esté utilizando puedes conseguir un campo vacío y que todos los que se encuentren cerca también se destapen y tener esa ventaja de saber de esos lugares, en otros casos se puede llegar a encontrar con un numero los cuales indicaran cuantas bombas se encuentran cercas de esa casilla en particular, todo el juego llega a ser muy interactivo ya que hay que tener mucho cuidado para no pisar una mina, uno de los principales objetivos es que al momento de jugar el usuario pueda llegar a experimentar un juego fluido y fácil de poder jugar.

Este proyecto se está desarrollando con el objetivo de poder poner a prueba todos aquellos conocimientos sobre la creación de clases, interfaz, y que los códigos de lógica se encuentren bien administrados para un buen desarrollo del ambiente gráfico.

La aplicación será en un ambiente de pura interfaz donde al momento de iniciar el juego, el usuario va a poder seleccionar que tan grande quiere las dimensiones de la tabla, las minas se distribuirán de forma aleatoria, esto se lograra en la parte de la lógica del programa, los niveles de dificultad dependerán de que tan grande sea la tabla seleccionada por el participante.

El desarrollo de este proyecto nos va a contribuir a un mejor aprendizaje, llegar a adquirir mejores conocimientos teóricos en el área de programación, como también nos ayudara a fomentar nuestros conocimientos y competencias para un futuro trabajo con mayor planeación, análisis y problemas vamos a poder tener el conocimiento necesario para poder afrontarlos y resolverlos con éxito.

Descripción

Este proyecto del buscaminas tiene como objetivo implementar temas sobre la orientación a objetos y uso de matrices, esto con el fin de poder adquirir técnicas de buen uso y distribución de clases, objetos y mediciones de cuadriculas para la creación del juego más conocido de los años noventa, este video juego nos incita a el uso de condiciones de lógica, flujo de control, estadísticas, diseños de interfaz interactivos, también nos proporciona un estímulo y habilidad para las planificaciones, análisis y resolución de problemas que nos atribuyen a los ámbitos de desarrollo de programas.

La elección de este proyecto del buscaminas nos atribuye una gran motivación de seguimiento de los pasos que se requieren para la creación y finalización de este, ya que este nos da un vistazo sobre los procesos que conllevan los programas, la creatividad y simplicidad de reglas que hay que seguir para un resultado.

Abstracto

This project will be implemented in Java, with the purpose of creating a video game that was very popular in the nineties. It consists of a fairly visually perceptible graphic environment, so that the user can experience what the game was like in those days. This game is called Minesweeper. It consists of a particularity of having luck and perspective to be able to guess where the bombs are located and not touch them to avoid losing the game.

This game is known for uncovering squares in which, depending on the technique being used, you can get an empty field, and everyone nearby will also be uncovered, giving you the advantage of knowing those locations. In other cases, you can even find a number which will indicate how many bombs there are near that particular square. The entire game becomes very interactive since you must be very careful not to step on a mine. One of the main objectives is to ensure that when playing, the user can experience a fluid and easy-to-play game. This project is being developed with the goal of putting to the test all the knowledge regarding class creation and interfaces and ensuring that the logic codes are well-managed for a successful graphical environment.

The application will be in a purely interface environment where, upon starting the game, the user will be able to select the desired table size. The mines will be randomly distributed. This will be achieved in the logic section of the program. The difficulty levels will depend on the size of the table selected by the participant.

The development of this project will contribute to improved learning, help us acquire greater theoretical knowledge in the area of programming, and will also help us develop our knowledge and skills for future work with greater planning, analysis, and problem-solving skills. We will have the necessary knowledge to successfully address and solve them.

The motivation for having selected this topic is that it will help us to have an idea about how applications are created, what aspects must be taken into account in order to develop them correctly, since it must be taken into account that desktop applications are often simpler but in the field of games it becomes a little more complex, because all the scenarios that may occur at the time of its execution must be taken into account.

Objetivos

Objetivo general

El objetivo de este proyecto es crear un juego funcional sobre un buscaminas en un lenguaje de programación de java, en este llegaremos a aplicar uno de los temas principales de programación I, el cual es orientación a objetos e interfaces graficas, esto con el propósito de reforzar nuestros conocimientos, mejorar nuestras técnicas y tener un mejor manejo de resolución de problemas.

Objetivos específicos

1. Desarrollar la lógica que va a tener el juego, en esta se va a utilizar matrices para formar las cuadriculas, gestionarlas y poder ubicarlas las minas de forma aleatoria.
2. Se diseñará la interfaz que necesita para que visualmente sea atractiva para el usuario, donde esta sea fácil de entender y ejecutar.
3. Se implementará métodos para el manejo de las minas, su ubicación, mensajes de que gano o perdió, tiempo de duración.
4. Se realizarán pruebas para poder determinar que el juego funciona correctamente.

Introducción

En este proyecto del buscaminas estamos buscando poder implantarlo en el lenguaje java, debido a que muchos otros lo han logrado pero en otros lenguajes, nuestro reto principal es intentarlo en este lenguaje y obtener un resultado exitoso, este videojuego es super clásico muchas personas lo han jugado en algún momento de sus vidas, es super conocido porque se necesita una estrategia para poder descubrir que casillas de nuestro tablero contiene alguna mina, el punto clave para poder descubrirlas es destapando algunas y seguir las pistas numéricas que nos brindan esas casillas.

La implementación del proyecto en este entorno de prácticas ha sido de mucha ayuda, porque primeramente nos da retos para poder lograr un resultado exitoso, luego debemos de realizar búsquedas para poder adquirir conocimiento e implementarlo dentro de nuestro Código, las gráficas que lleguemos a realizar deben de ser entendibles y fáciles de usar, de este modo le damos vida a nuestro juego.

También se tiene que agregar que tener esta posibilidad de recrear este juego que ha brindado tanto entretenimiento a lo largo de los años es super valioso, ya que tenemos la oportunidad de fomentar nuestro análisis, nuestra creatividad y las formas de resolución de problemas, el buscaminas se complementa tanto de lógica y de grafica esto atribuye muchas técnicas para otros futuros proyectos.

Marco teórico

**1. Buscaminas**

El Buscaminas es un videojuego de lógica cuyo objetivo es descubrir todas las casillas libres de minas en un tablero dividido en celdas cuadradas. El jugador utiliza pistas numéricas para deducir la ubicación de las minas sin detonarlas. Este juego se popularizó en los sistemas operativos Microsoft Windows desde la década de 1990 y ha servido como ejercicio de razonamiento lógico y memoria espacial.

En su funcionamiento, el juego presenta las siguientes características esenciales:

* **Tablero**: matriz cuadrada o rectangular de celdas ocultas.
* **Minas**: distribuidas aleatoriamente, sin repetición en la misma celda.
* **Numeración**: cada celda destapada indica la cantidad de minas en sus casillas adyacentes.
* **Acciones del jugador**: marcar, desmarcar o destapar celdas.
* **Condiciones de victoria y derrota**: el jugador gana al marcar todas las minas correctamente y pierde al destapar una mina.

**2. Programación en Java**

Java es un lenguaje de programación orientado a objetos ampliamente utilizado por su portabilidad, robustez y amplio ecosistema de bibliotecas. Sus características relevantes para este proyecto son:

* **Portabilidad**: ejecución en cualquier sistema con una máquina virtual Java (JVM).
* **Bibliotecas gráficas**: como Swing y AWT, que permiten la creación de interfaces gráficas interactivas.
* **Gestión de eventos**: mecanismos para capturar interacciones del usuario (clics, entradas de texto).
* **Estructuras de datos**: manejo eficiente de matrices y colecciones para representar el estado del tablero.

**3. Interfaz Gráfica Swing**

Swing es una biblioteca de Java para interfaces gráficas que incluye componentes como botones (JButton), tablas (JTable) y paneles (JPanel). En este proyecto, Swing se utiliza para:

* Representar el tablero como una grilla interactiva.
* Mostrar menús y cuadros de diálogo (JOptionPane).
* Actualizar el estado del juego en tiempo real según las acciones del jugador.

**4. Uso de Matrices y Estructuras de Datos**

El tablero se implementa como una **matriz bidimensional** de enteros o celdas personalizadas, donde:

* **Valor -1** representa una mina.
* **Valores ≥ 0** representan el número de minas adyacentes.
* Se utilizan estructuras auxiliares para controlar las casillas destapadas y marcadas.  
  La generación aleatoria de minas se realiza con funciones como Random, asegurando no repetir coordenadas.

Desarrollo

Descripción del problema

El problema consiste en desarrollar una versión funcional del juego Buscaminas que cumpla con especificaciones técnicas y de usabilidad definidas. El usuario debe poder ingresar el tamaño del tablero (L), que debe ser mayor a 2, y el sistema debe generar automáticamente 2×L minas en posiciones aleatorias y únicas. El jugador interactúa con el tablero mediante acciones de marcar, destapar o desmarcar casillas.

El juego debe terminar si el jugador destapa una casilla con mina (pierde) o si logra marcar correctamente todas las minas (gana). Al perder, se deben revelar todas las minas. Si el jugador destapa una casilla con cero minas vecinas, se deben destapar automáticamente todas las casillas adyacentes no minadas hasta alcanzar casillas con números. Además, se requiere un sistema de estadísticas y un menú funcional.

Los principales desafíos incluyen la gestión de la lógica del juego, la sincronización entre la interfaz y el modelo, y la implementación de algoritmos recursivos para la expansión de zonas vacías.

Para resolver el problema, se adoptó una metodología basada en el ciclo de desarrollo de software: análisis, diseño, implementación, prueba y documentación.

Se utilizó el patrón Modelo-Vista-Controlador (MVC) para organizar el código:

* Modelo: Clases Casilla y Coordenada.
* Vista: Interfaces guiPrincipal y TableroJuego.
* Controlador: Clase Controlador, que gestiona la lógica del juego.

Se implementaron algoritmos para:

* Generar minas evitando duplicados.
* Calcular minas vecinas.
* Expandir zonas vacías recursivamente.
* Validar entradas y estados del juego.

La interfaz se desarrolló con Swing, y se utilizó GitHub para el control de versiones.

El problema presenta varios retos técnicos:

* Validación de entrada: Asegurar que L > 2.
* Generación de minas: Evitar duplicados y garantizar exactamente 2×L minas.
* Interfaz gráfica: Sincronizar eventos de clic con la lógica del juego.
* Expansión recursiva: Implementar un algoritmo eficiente para destapar zonas vacías.
* Gestión de estado: Determinar cuándo el jugador gana o pierde.

La solución requiere una correcta abstracción de objetos y una clara separación de responsabilidades.

Diagrama de clases

Diagrama

El contenido generado por IA puede ser incorrecto.

Análisis de la solución

**Se logró implementar completamente todos los requisitos:**

* **Tamaño de tablero variable con validación.**
* **Generación de minas sin duplicados.**
* **Interacción con clic izquierdo/derecho.**
* **Expansión de las zonas vacías.**
* **Estadísticas y menú funcional.**

**Qué faltó: No se implementó un contador de minas restantes ni un temporizador.**

**Mejoras posibles:**

* **Añadir niveles de dificultad.**
* **Incluir un sistema de guardado de partidas.**
* **Mejorar la interfaz con iconos más bonitos.**

**Detalles técnicos:**

* **Se usó recursión para la expansión.**
* **Se retrasó la generación de minas para evitar perder en el primer clic.**
* **Se utilizó GridLayout para organizar el tablero.**

Discusión de resultados obtenidos

El proyecto alcanzó un alto grado de completitud. La aplicación es funcional, estable y cumple con los requisitos académicos. Los errores iniciales (como el doble clic) fueron corregidos mediante la actualización forzada de la interfaz. La arquitectura MVC permitió una fácil depuración y mantenimiento.

El uso de GitHub con múltiples commits refleja un trabajo progresivo y bien organizado. La documentación es coherente y detallada.

Conclusión

En conclusión, este juego del buscaminas me permitió poder aplicar varios de mis conocimientos de programación que he logrado adquirir con el pasar del tiempo, el tema de orientación a objetos, gestión de eventos, estructuras de control, diseño de las interfaces en este proyecto fue muy reconfortante ya que pude adquirir más conocimiento de la organización de las clases, métodos, el uso de las matrices, con todo esto se logró poder representar gráficamente las cuadriculas del juego que este sea interactivo y entendible.

Después de todo el tiempo invertido se logró un gran objetivo ya que este proyecto proporciono herramientas para otros proyectos a futuro, me brindo conocimiento clave para un mejor pensamiento y distribución del desarrollo de un juego, la experiencia fue super buena ya que de un video juego clásico logre obtener más creatividad y facilidad de diseño.

Recomendaciones

Se recomienda a futuros estudiantes comenzar el proyecto con anticipación, dividiendo el trabajo en etapas pequeñas y manejables, como el diseño de clases, la lógica del juego y la interfaz gráfica. Es fundamental utilizar desde el inicio un sistema de control de versiones como GitHub, haciendo commits frecuentes y descriptivos para reflejar el avance real del trabajo. Evitar depender exclusivamente del diseñador visual de NetBeans es clave, ya que puede generar código difícil de modificar; es preferible combinarlo con programación manual para mayor control.

Además, se sugiere probar el juego con tamaños de tablero pequeños al principio, lo que facilita la detección de errores lógicos o de interfaz. La documentación debe ir avanzando en paralelo con el desarrollo, ya que facilita la redacción final del informe. Por último, se recomienda mantener una comunicación constante entre los integrantes del grupo, si el proyecto es colaborativo, para asegurar una integración fluida de las diferentes partes del sistema.

Cronograma de trabajo

|  |  |  |  |
| --- | --- | --- | --- |
| 24/07/2025 | Configuración inicial del proyecto y estructura básica | Integración con JavaFX no necesaria | TheJPlay2006 |
| 03/08/2025 | Renombrar proyecto a "Buscaminas" y agregar archivos de GUI | Ajuste de paquetes y estructura de carpetas | TheJPlay2006 |
| 07/08/2025 | Implementación inicial de clases:CasillayJuego | Diseño incompleto, sin validación | Eme2004 |
| 11/08/2025 | Renombrar proyecto y agregar formularios GUI | Conflictos menores en estructura de paquetes | TheJPlay2006 |
| 13/08/2025 | Implementación de clases clave:Coordenada,Controlador,Estadistica | Inconsistencias en métodos y constructores | Eme2004 |
| 13/08/2025 | Mejora de la interfaz principal y adición del tablero de juego | Sincronización entre lógica y vista | TheJPlay2006 |
| 14/08/2025 | Actualización y limpieza de clases | Errores de compilación por nomenclatura | TheJPlay2006 |
| 14/08/2025 | Fusión de ramas (main→Jairo,main→Emesis) | Conflictos de merge resueltos manualmente | TheJPlay2006, Eme2004 |
| 14/08/2025 | Creación y actualización delREADME.md | Falta de documentación inicial | TheJPlay2006 |
| 24/07/2025 | Configuración inicial del proyecto y estructura básica | Integración con JavaFX no necesaria | TheJPlay2006 |

Bitácora

|  |  |  |  |
| --- | --- | --- | --- |
| 24/07/2025 | Inicio del proyecto con estructura inicial y app JavaFX | Se usó JavaFX, no necesario para el proyecto | Se eliminó JavaFX y se mantuvo solo Swing |
| 03/08/2025 | Renombrado a "Buscaminas" y agregado formularios GUI | Estructura de paquetes desorganizada | Se reorganizó enModelo,Logica,gui |
| 07/08/2025 | Implementación deCasillayJuego | Métodos no coincidían con GUI | Se ajustaron nombres y parámetros |
| 11/08/2025 | Adición de GUI principal y tablero | Botones no respondían | Se vincularon eventos conControlador |
| 13/08/2025 | Implementación deCoordenada,Controlador,Estadistica | Errores por falta degetLadoTablero() | Se agregaron métodos faltantes |
| 13/08/2025 | Mejora de UI: menú, estadísticas, tamaño | Interfaz no se actualizaba | Se añadiórevalidate()yrepaint() |
| 14/08/2025 | Limpieza de código y corrección de errores | Doble clic requerido para destapar | Se forzó actualización tras primer clic |
| 14/08/2025 | Fusión de ramas (Jairo,Emesis) | Conflictos enEstadistica.java | Se resolvió con merge manual |
| 14/08/2025 | Creación delREADME.md | Documentación mínima | Se completó con descripción del proyecto |
| 24/07/2025 | Inicio del proyecto con estructura inicial y app JavaFX | Se usó JavaFX, no necesario para el proyecto | Se eliminó JavaFX y se mantuvo solo Swing |

Referencias bibliográficas

Deitel, P., & Deitel, H. (2019). *Java: Cómo programar* (10.ª ed.). Pearson Educación.

Oracle. (2025). *Java Platform, Standard Edition Documentation*. <https://docs.oracle.com/en/java/javase/index.html>

NetBeans. (2025). *Guía del usuario de NetBeans IDE*. <https://netbeans.apache.org/kb/index.html>

Sweigart, A. (2020). *Inventando juegos con Python* (4.ª ed.). <https://inventwithpython.com/invent4thed/>  
*(Nota: Aunque está en Python, su análisis del juego Buscaminas es aplicable a cualquier implementación algorítmica)*

Microsoft. (2025). *Historia de los juegos en Windows: Buscaminas*. <https://support.microsoft.com/es-es/help/11894/windows-7-games-minesweeper>  
*(Fuente oficial que describe las reglas, mecánicas y evolución del juego original)*

Horstmann, C. S. (2019). *Big Java: Early Objects* (7.ª ed.). Wiley.  
*(Libro ampliamente utilizado en cursos universitarios para enseñar POO y GUIs en Java)*

Liang, Y. D. (2020). *Introduction to Java Programming and Data Structures* (12.ª ed.). Pearson.  
*(Cubre desde conceptos básicos hasta interfaces gráficas y recursión, temas clave en este proyecto)*

Gamasutra. (2018). *The History and Design of Minesweeper*. <https://www.gamasutra.com/view/feature/134425/the_history_and_design_of_.php>

Anexos

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Imagen que contiene teclado

El contenido generado por IA puede ser incorrecto.

Link al repositorio en GitHub: <https://github.com/TheJPlay2006/Buscaminas/tree/main>