MANUAL

TÉCNICO

**ÍND****ICE**

Introducción …………………………………………………………………………………………………3

Tecnologías Utilizadas……………………………………………………………………………………3

Descripción……………………………………………………………………………………………………4

Captura 1……………...…………………………………..………………………..…………………..4

Captura 2a……………..…………………………………………………………..……………………5

Captura 2b……………..…………………………………………………………..……………………5

Captura 3………………………………………………………………………………………………….6

Código Fuente y Documentación…………………………………………………………………7

Notas…………………………………………………………………………………………………………….7

**INTRO****DUCCIÓN**

El buscaminas es un videojuego inventado por Robert Donner en 1989, su objetivo es despejar todas las casillas sin detonar ninguna mina. El buscaminas se compone de una matriz que representa un campo de minas. Este campo de minas tiene tanto minas como números, los cuales representa el número de minas que hay alrededor de ese número.

Por ejemplo, si en una casilla se encuentra el número 2, esto significa que alrededor de está casilla se hallan 2 minas, y el resto son números, sean cero o no.

Por su naturaleza y al usar una matriz como principal objeto de juego, el buscaminas tiende a almacenar casillas con el número 0, y por tanto se desperdicia memoria en tiempo de ejecución.

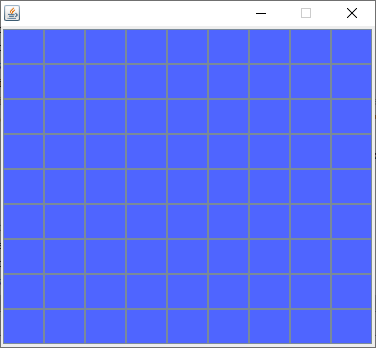
Este programa busca implementar la gran ventaja que tiene el uso de matrices dispersas frente a las matrices convencionales, el cual es ahorrar espacio en memoria almacenando todo valor que no se considere un valor vacío, en este caso, todo valor diferente de 0.

**TECNOLOGÍ****AS UTILIZADAS**

* Se utilizó **Java** como lenguaje de programación, esto debido a su gran popularidad respecto la programación orientada a objetos (POO).
* Para la parte gráfica, el uso de **Java Swing** fue la opción más agradable y sencilla.
* En cuando a los IDE’s se utilizó tanto **Eclipse** como **Netbeans**, el segundo para agilizar la creación y edición de la interfaz gráfica, y el primero para una mejor depuración y corrección de errores mayormente en la parte lógica.

**DESCRIPCIÓN**

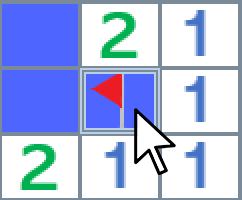
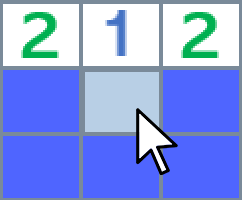
Para empezar a implementar el uso de matrices dispersas en el buscaminas, entre las 3 formas de matrices dispersas vistas en el curso, se decidió utilizar la **Forma 1**, debido a su fácil recorrido y su desempeño frente a las demás. Así, el componente principal de este programa se centra en la clase **CampoDeMinas**, que hereda de la clase **MatrizForma1**.

Para la interacción del usuario con el buscaminas, en la parte de la interfaz se tiene un botón por cada casilla del campo de minas, de aquí surge la clase **CasillaB** la cual es derivada de la clase **JButton** de Java **Swing**, con la particularidad de que esta contiene la posición de la casilla en el campo de minas.

Captura 1

Los botones a su vez interactúan con el campo de minas utilizando los métodos de interacción principales de la clase **CampoDeMinas**, los cuales son:

* **tocarCasilla(***int***,** *int***)**
* **posibleMinas(***int***,** *int***)**



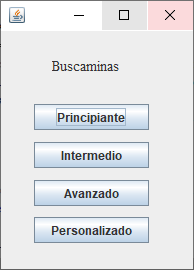
Captura 2a

Captura 2b

Estos dos métodos corresponden a los clicks efectuados en determinado botón, click izquierdo para tocar una casilla (*Captura 2a*), y click derecho para marcar o desmarcar una posible mina (*Captura 2b*). Las interacciones se controlas dentro de la clase **CampoDeMinas**, según sea el estado de la casilla con la cual se interactúe (*oculta*, *posibleMina*, *descubierta*), y según sea su valor (*mina*, *0*, *1-8*). Para más información véase la [*documentación java*.](#Javadoc)

Los niveles o modos de juego son 4:

* **Principiante:** Campo de 8x8 con 10 minas.
* **Intermedio:** Campo de 16x16 con 40 minas.
* **Experto:** Campo de 16x30 con 99 minas
* **Personalizado:** El usuario elige el tamaño del campo y el número de minas, siempre que sea menor a 30x30 y el número de minas sea mayor a 0 y menor al tamaño del campo.

Estos niveles controlan el tamaño y las minas del campo de minas. En la clase **Menu** se tienen botones para cada nivel, todos estos botones inician un nuevo juego según el nivel que se elige.

Captura 3

De *Principiante* a *Avanzado* se crea una nueva ventana (*Captura 1*) con la clase **GUI** que extiende de **JFrame** (Java Swing), con sus respectivos valores predefinidos. Para la opción *Personalizado* se le pide al usuario que ingrese los valores para crear la ventana, esto haciendo uso del método **showInputDialog(***String***)**  perteneciente a la clase **JOptionPane** de Java Swing. Si el usuario ingresa datos incorrectos se regresa al menu principal (*Captura 3)*.

La nueva ventana que se crea al elegir un nivel contiene el campo de minas con tamaño definido según el nivel elegido, esta ventana es la intermediaria entre el usuario y el campo de minas, invocando a los métodos de interacción principales, y dibujando el campo de minas. Además, se pueden crear tantas ventanas (partidas) como lo permita el entorno de Java.

Por último, al terminar una partida (sea ganada o perdida) se pregunta por medio de **showConfirmDialog(***Component***,** *String***,** *String***,** *int***)**perteneciente a **JOptionPane**, si se quiere seguir jugando nuevas partidas, si se elige *“no”* el programa termina su ejecución, en el otro caso se cerrará la ventana que contiene el juego terminado, y se esperará a la creación de una nueva ventana con un nuevo juego.

**CÓDIGO** **FUENTE Y DOCUMENTACIÓN**

* **Código fuente:** <https://github.com/Sneyderbg/Buscaminas>
* **Java****doc:** <https://github.com/Sneyderbg/Buscaminas/tree/main/doc>

**N****OTAS**

* Se utilizó el **JDK 8u281** para la realización de este programa.
* Se recomienda el uso de Eclipse para abrir el proyecto de forma correcta.
* Se recomienda usar la codificación **UTF-8** para la correcta visualización de los comentarios.