



# **Docuementación de Proyecto: Buscaminas**

Nombre: Alejandro José Castellanos Zavala

Número de cuenta: 12441410

Asignatura: 892 Programación I

Catedrático: Ing. Darío Alexander Cardona Aguilar

Fecha: domingo 15 de septiembre de 2024

# **Introducción**

Este proyecto consistió en desarrollar el clásico juego Buscaminas en Java. el juego cuenta con una interfaz gráfica que permite al usuario hacer click sobre los botones para

El juego consiste en despejar todas las celdas que no contengan una mina de entre una matriz de celdas.

Algunas celdas tienen un número, el cual indica la cantidad de minas que hay en las celdas circundantes. Por ejemplo, si una celda tiene el número 3, significa que de las ocho celdas que hay alrededor, tres tienen minas y cinco no. Si se descubre una celda sin número, indica que ninguna de las celdas vecinas contiene minas, y éstas se descubren automáticamente.

Si se descubre una celda con una mina, se pierde la partida.

Es posible colocar una marca en las celdas que el jugador cree que contiene minas para ayudar a descubrir las que están cerca. Generalmente esta marca es una bandera.

# Elementos del Buscaminas

La implementación del buscaminas creada para el proyecto, consistió principalmente en dos elementos: la matriz secreta y la matriz mostrada.



## Matriz Secreta

La matriz secreta contiene la posición de todas las minas y el número de minas circundantes para el resto de las celdas. La matriz secreta es utilizada para comparar y revelar los números del tablero cuando el usuario presiona los botones de la interfaz gráfica.

## Colocar Minas

Para inicializar la matriz secreta, primero se colocan las minas aleatoriamente.

```
public void inicializar() {  
    limpiarTablero(); // Borrar cualquier mina presente  
    for (int i = 0; i < minas; i++) { // Generar minas  
        colocarMina();  
    }  
}
```

Donde el método colocarMinas() es un método recursivo que se llama a sí mismo si la celda elegida aleatoriamente ya contiene una mina.

```
private void colocarMina() {  
    int x = random.nextInt(filas);  
    int y = random.nextInt(columnas);  
    if (matriz[x][y] != MINA) {  
        matriz[x][y] = MINA;  
    }  
}
```

```

        return;
    }
    colocarMina();
}

```

Esto asegura de que se coloque exáctamente la cantidad de minas ingresada. Nótese que se podría haber realizado con ciclo while, pero para efectos de aprendizaje, se decidió hacerlo como se indicó anteriormente.

## Generar Números

Una vez las minas han sido colocadas, se procede a generar los números para el resto de las celdas. Para esto, recorren todos los elementos de la matriz y para cada uno se hace lo siguiente:

1. Si la celda contiene una mina, saltarla.
2. Se crea una variable cuenta y se inicializa en 0.
3. Se revisan las ocho celdas circundantes con la ayuda de dos ciclos for que van desde [-1, 1].

```

cuenta = 0;
for (int di = -1; di <= 1; di++) {
    for (int dj = -1; dj <= 1; dj++) {
        // Saltar el centro de la celda
        if (di == 0 && dj == 0) {
            continue;
        }
        int newFila = i + di;
        int newCol = j + dj;
        if (esValido(newFila, newCol)) {
            if (matriz[newFila][newCol] == MINA) {
                cuenta++;
            }
        }
    }
}

```

```
}  
matriz[i][j] = (char) (cuenta + CERO);
```

En el código anterior, newFila y newColumna son las coordenadas de las ocho esquinas de una determinada celda.

Para evitar el error `OutOfBoundsException`, se revisa si las coordenadas de las celdas circundantes están dentro de los límites de la matriz, antes de revisar lo que contiene.

```
private boolean esValido(int fila, int columna) {  
    return fila >= 0 && fila < filas && columna >= 0 && columna < columnas;  
}
```

A continuación, en la [Figura 1](#) se muestra un diagrama que muestra el proceso mencionado anteriormente.

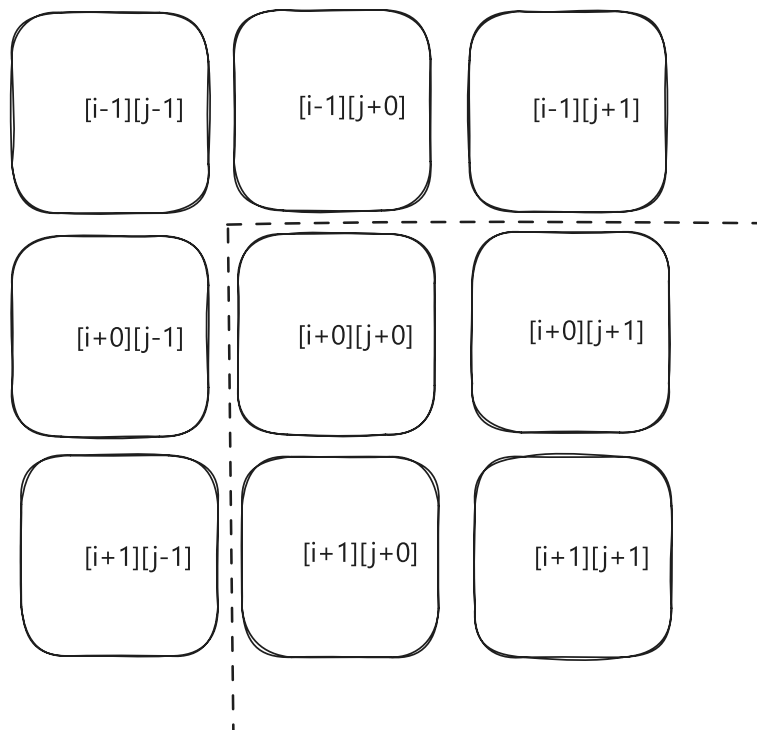


Figura 1: Diagrama de las celdas circundantes para una determinada celda.

• • • • • • • • • • • • • • • • • •

# Matriz de Botones

Cuando el usuario presiona uno de los botones de la matriz de botones (que es la interfaz gráfica), se descubre el número que contiene la matriz secreta en esa posición.

• • • • • • • • • • • • • • • • • •

# Otros Elementos

# **Conclusiones**

# **Recomendaciones**