

# ALGORITMOS

Autor: Abel Gallart Bonome

# BUSCAMINAS

En JavaFX Swing



**Buscaminas** (nombre original en inglés: *Minesweeper*) es un videojuego para un jugador creado por Curt Jhonson y Robert Donner en 1989. El objetivo del juego es despejar un campo de minas sin detonar ninguna. El juego consiste en descubrir **todas las ubicaciones de un tablero cuadrado en las que hay minas**. El juego empieza con todo el tablero cubierto y para empezar a jugar tendremos que pulsar en cualquier casilla, lo cual nos mostrará su contenido, el cual puede ser: una mina, en cuyo caso perderemos la partida; un número, que indicará la cantidad de minas en las 8 casillas adyacentes; o puede aparecer como una casilla en blanco, lo cual indica que no tiene ninguna mina en las casillas que están a su alrededor. Cuando pulsamos en una casilla sin minas en ocasiones se nos abrirán las casillas de alrededor que no tengan minas en sus respectivas casillas adyacentes hasta que el espacio abierto esté rodeado o bien de los límites del mapa o de

casillas con minas en sus alrededores, aunque esto no quiere decir que haya casillas más allá de las que se han mostrado sin minas ya que solo se mostrarán las casillas con minas adyacentes que tengan una casilla sin número en sus casillas cercanas. **Conociendo la cantidad de minas en las casillas adyacentes de varias posiciones al mismo tiempo se puede ir deduciendo el emplazamiento de las minas**, las cuales deberemos ir marcando hasta haber señalado el número de minas escondidas inicialmente, una vez hecho esto si hemos acertado con las casillas marcadas **habremos ganado la partida**.

El juego se puede configurar para jugar en **diferentes dificultades** ya prediseñadas o se puede personalizar manualmente el tamaño del tablero y la cantidad de minas que queremos que se escondan en él, ajustando así la dificultad al gusto de cada jugador. Independientemente de los ajustes, **cada partida será única** ya que las minas se ponen en el tablero de forma aleatoria en cada ocasión.

### **¿Para qué sirve el Buscaminas?**

El buscaminas fue **uno de los primeros juegos que se implementaron en ordenadores personales modernos**, junto con otros juegos como el Solitario o los Corazones, todos estos **no se crearon buscando entretener a los usuarios**, por el contrario, perseguían **que la gente se habituara a usar el ratón**, haciendo que en estos juegos necesitaras **pulsar en cuadrados pequeños con precisión** como en el caso del buscaminas o **pulsar y arrastrar** las cartas como en el caso del solitario. Estos juegos con los años han seguido formando parte de los sistemas operativos de Microsoft de forma nativa, sufriendo actualizaciones y cambios en la interfaz hasta **Windows 8, que es la última versión de Windows que los incluía**.

El buscaminas **clásico usa una cuadrícula** de distintas dimensiones, pero con los años han ido apareciendo otras versiones que llevan el concepto más allá introduciendo **tableros triangulares o con otras formas geométricas** además de nuevas mecánicas como la posibilidad de jugar **en tres dimensiones o que puedan aparecer varias minas en la misma casilla**.

Neurólogos británicos realizaron un estudio entre los que juegan y los que no juegan. Los resultados sorprendieron al público. Resultó que las personas que juegan al buscaminas aprenden habilidades motoras mucho más rápido y su **cerebro absorbe** nueva información más rápido. Bonificación: mejora la percepción de los jugadores de los pequeños componentes del todo.

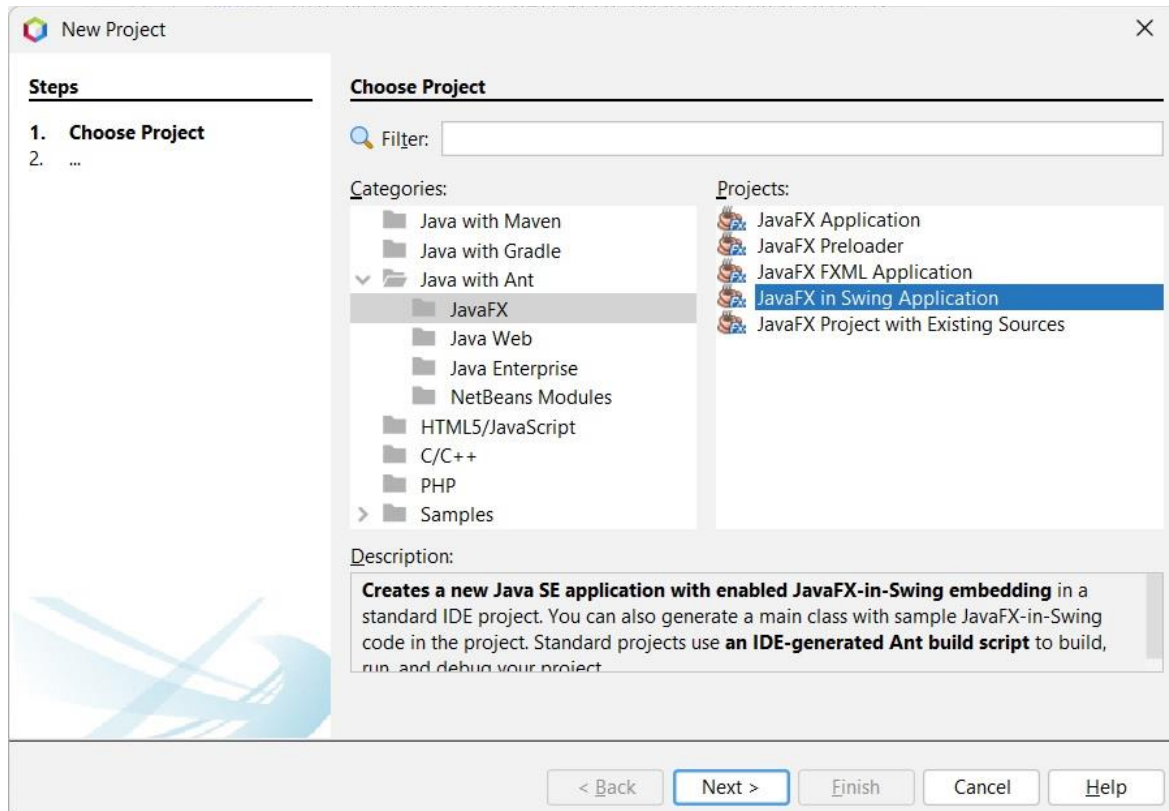
La serie de juegos Buscaminas en línea merece atención. A pesar del **nombre militante**, el juego en línea Buscaminas es **bastante pacífico**. El juego está incluido en el conjunto de juguetes gratuitos para todas las ventanas. ¡Una trama simple no requiere un estudio profundo de las reglas! Pero el juego requiere atención e ingenio por parte de los jugadores. Un movimiento falso resultará en una explosión. Un paso en falso hacia la explosión y la pérdida. Tendremos que empezar de nuevo. El juego Buscaminas es tan atractivo que puede cautivarte fácilmente durante varias horas.

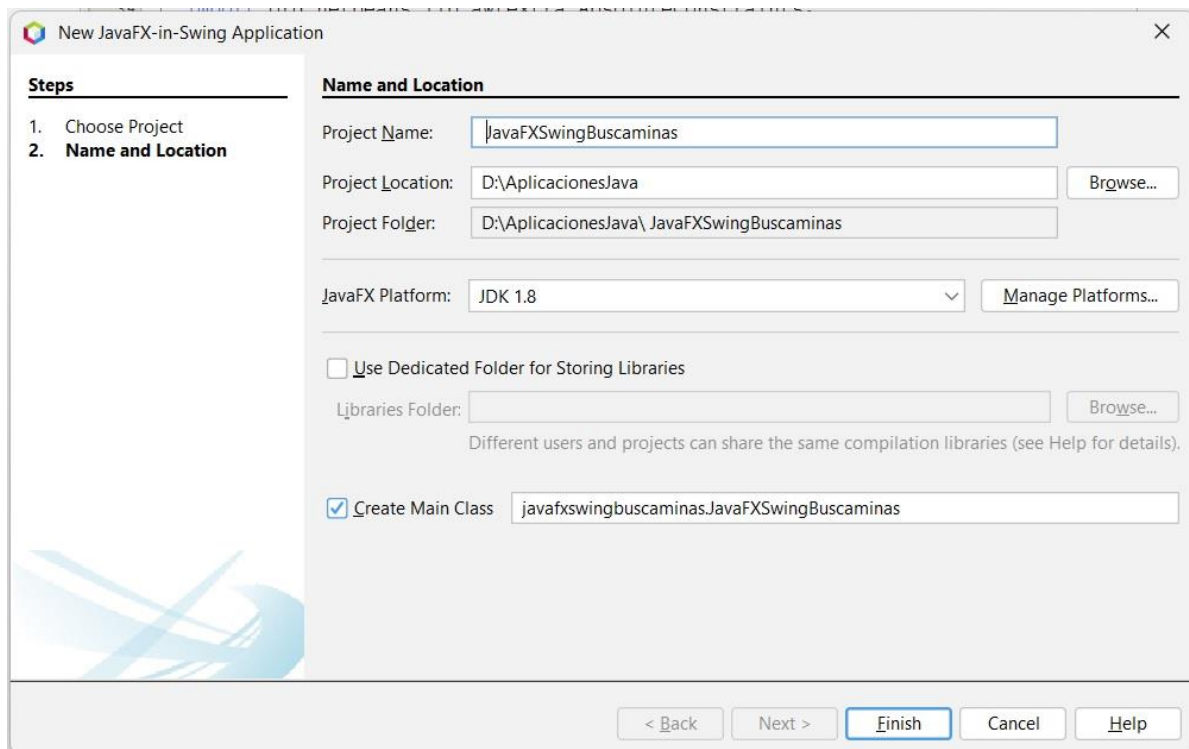
Hay que tener en cuenta que puedes perder desde el primer movimiento. Si aterrizas accidentalmente en una mina detrás de cuadrados ocultos, tendrás que empezar de nuevo. No se moleste, pero la mano misma dirige el mouse de la computadora para reiniciar el proceso.

Buscaminas es adecuado para **niños y adultos**. No cabe duda que este minijuego tiene beneficios, los cuales son los siguientes:

- Mejora el cerebro;
- Desarrolla el pensamiento;
- Desarrolla la capacidad de responder con eficacia;
- Desarrolla habilidades motoras en un niño de 3 a 9 años.
- Ayuda al cerebro a mantenerse en forma.

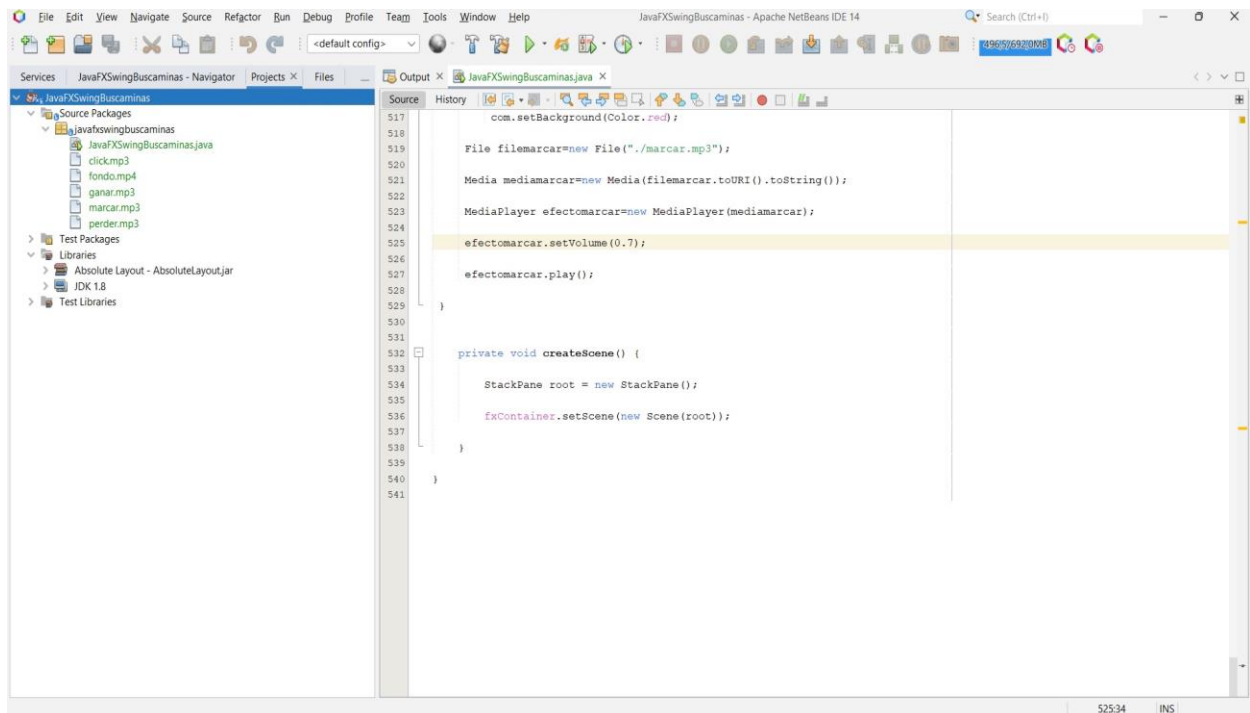
Para comenzar abrimos NetBeans y creamos nuestro proyecto JavaFX Swing.





Y agregamos a nuestro paquete los archivos multimedia que usaremos como efectos de sonido, en concreto los archivos de sonido ganar.mp3, perder.mp3, click.mp3, marcar.mp3, y fondo.mp4 un video que reproduce al fondo de nuestro juego, para hacer más visual el juego.

## Código fuente:



Y ahora si pasamos a codificar nuestro algoritmo, tenemos que expresar en lenguaje Java la dinámica del juego.

```
/*
```

```
* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
```

```
* Click nbfs://nbhost/SystemFileSystem/Templates/javafx/FXSwingMain.java to edit this template
```

```
*/
```

```
package javafxswingbuscaminas;
```

**Importamos las clases que vamos a necesitar**

```
import java.awt.Color;
```

```
import java.awt.Component;
```

```
import java.awt.Dimension;
```

```
import java.awt.Font;
```

```
import java.awt.event.MouseAdapter;
```

```
import java.awt.event.MouseEvent;
```

```
import java.io.File;
```

```
import java.net.URISyntaxException;
```

```
import java.util.Random;
```

```
import java.util.logging.Level;
```

```
import java.util.logging.Logger;
```

```
import javafx.application.Platform;
```

```
import javafx.embed.swing.JFXPanel;
```

```
import javafx.scene.Group;
```

```
import javafx.scene.Scene;
```

```
import javafx.scene.layout.StackPane;
```

```
import javafx.scene.media.Media;
```

```
import javafx.scene.media.MediaPlayer;
```

```
import javafx.scene.media.MediaView;
```

```
import javax.swing.BorderFactory;
import javax.swing.JApplet;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import org.netbeans.lib.awtextra.AbsoluteConstraints;
import org.netbeans.lib.awtextra.AbsoluteLayout;

/**
 *
 * @author Abel Gallart Bonome
 */
public class JavaFXSwingBuscaminas extends JApplet {
```

**Declaramos las variables de la clase**

```
int JFXPANEL_WIDTH_INT = 1000;
```

```
int JFXPANEL_HEIGHT_INT = 1000;
```

```
JFXPanel fxContainer;
```

```
static int ancho=30;
```

```
static int largo=30;
```

```
static int bombas=30;
```

```
static int puntos=0;
```

```
static int nivel=0;
```

```
Random ran;
```

```
String [][]matris;
```

```
JButton b[][];
```

```
JLabel l[][];
```

```
static JFrame frame;
```

```
static JApplet applet;
```

```
private JLabel jLabel1;
```

```
private JLabel jLabel2;
```

```
private JLabel jLabel3;
```

```
private JPanel jPanel1;
```

```
private JPanel jPanel2;
```

```
private JPanel jPanel3;
```

En nuestro método main creamos la ventana y mostramos una instancia de nuestra clase.

```
public static void main(String[] args) {

    SwingUtilities.invokeLater() -> {

        try {

            UIManager.setLookAndFeel("com.sun.java.swing.plaf.nimbus.NimbusLookAndFeel");

        }

        catch (Exception e){ }

        frame = new JFrame("Buscaminas");

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setUndecorated(true);

        frame.setOpacity((float) 1.0);

        applet = new JavaFXSwingBuscaminas();

        applet.init();

        frame.setContentPane(applet.getContentPane());

        frame.pack();

        frame.setLocationRelativeTo(null);
```

```
        frame.setVisible(true);

        applet.start();
    });
}
```

En el metodo init() se inicializan las variables y objetos necesarios tanto para gráfica como para la lógica.

@Override

```
public void init() {
```

```
    jPanel2 = new JPanel();
```

```
    jLabel1 = new JLabel();
```

```
    jLabel2 = new JLabel();
```

```
    jLabel3 = new JLabel();
```

```
    jPanel3 = new JPanel();
```

```
    jPanel2.setSize(new Dimension(720, 720));
```

```
    jPanel2.setOpaque(false);
```

```
    jPanel2.setLayout(new AbsoluteLayout());
```

```
    jLabel1.setFont(new Font("Segoe UI", 3, 36)); // NOI18N
```

```
    jLabel1.setForeground(new Color(255, 255, 255));
```