



UNIVERSIDAD MARIANO GÁLVEZ

Nombre Jose Daniel Hernández Camey

9959 24 1434

programación 1

Ingeniería en sistemas

ANALICIS DE CODIGO BUSCA MINAS

CELDA_H

```
#ifndef JUEGO_H
```

```
#define JUEGO_H
```

```
#include "Tablero.h" // Se incluye la clase Tablero, que probablemente maneja el tablero del juego
```

```
class Juego
```

```
{
```

```
private:
```

```
    Tablero tablero; // Objeto de la clase Tablero que representa el tablero del juego
```

```
    int cantidadMinas; // Número total de minas en el juego
```

```
    // Métodos privados que generan números aleatorios dentro de un rango
```

```
    int aleatorio_en_rango(int minimo, int maximo);
```

```
    int filaAleatoria(); // Devuelve una fila aleatoria dentro del tablero
```

```
    int columnaAleatoria(); // Devuelve una columna aleatoria dentro del tablero
```

```
public:
```

```
    Juego(Tablero tablero, int cantidadMinas); // Constructor que inicializa el juego con un tablero y una cantidad de minas
```

```
    void colocarMinasAleatoriamente(); // Método para colocar las minas en posiciones aleatorias dentro del tablero
```

```
    int solicitarFilaUsuario(); // Solicita al usuario que ingrese una fila
```

```
    int solicitarColumnaUsuario(); // Solicita al usuario que ingrese una columna
```

```
    bool jugadorGana(); // Verifica si el jugador ha ganado el juego
```

```
    void iniciar(); // Inicia el juego y maneja la lógica principal
```

```
    void dibujarPortada(string nombreArchivo); // Dibuja una portada basada en un archivo de entrada
```

```
};
```

```
#endif // JUEGO_H // Fin de la directiva de preprocesador para evitar inclusiones múltiples
```

CONFING_H

```
#ifndef CONFIG_H // Inicia una directiva de preprocesador para evitar múltiples inclusiones del archivo
```

```
#define CONFIG_H // Define la constante CONFIG_H para evitar que el archivo se incluya más de una vez
```

```
class Config // Declaración de la clase Config, que manejará la configuración del juego
```

```
{
```

```
    public:
```

```
        Config(int filasTablero, int columnasTablero, int minasTablero, bool modoDesarrolladorTablero, int vidasTablero); // Constructor de la clase, inicializa los atributos con los parámetros proporcionados
```

```
        int getfilasTablero(); // Método getter que retorna el número de filas del tablero
```

```
        int setfilasTablero(int filasTablero); // Método setter que establece el número de filas del tablero
```

```
        int getcolumnasTablero(); // Método getter que retorna el número de columnas del tablero
```

```
        int setcolumnasTablero(int columnasTablero); // Método setter que establece el número de columnas del tablero
```

```
        int getminasTablero(); // Método getter que retorna la cantidad de minas en el tablero
```

```
        int setminasTablero(int minasTablero); // Método setter que establece la cantidad de minas en el tablero
```

```
        bool getmodoDesarrolladorTablero(); // Método getter que retorna si el "modo desarrollador" está activo o no
```

```
        bool setmodoDesarrolladorTablero(bool modoDesarrolladorTablero); // Método setter que establece el valor de "modo desarrollador"
```

```
        int getvidasTablero(); // Método getter que retorna el número de vidas del jugador
```

```
        int setvidasTablero(int vidasTablero); // Método setter que establece el número de vidas del jugador
```

```
        void menuConfiguracion(); // Método que probablemente muestra un menú para cambiar la configuración del juego
```

```
    protected:
```

```
private:

    int filasTablero; // Atributo privado que almacena el número de filas del tablero

    int columnasTablero; // Atributo privado que almacena el número de columnas del tablero

    int minasTablero; // Atributo privado que almacena la cantidad de minas en el tablero

    bool modoDesarrolladorTablero; // Atributo privado que indica si el "modo desarrollador" está activo

    int vidasTablero; // Atributo privado que almacena el número de vidas del jugador

};

#endif // CONFIG_H // Fin de la directiva de preprocesador, marca el final del archivo de cabecera
```