LABORATORIO 1

GONZALO DE VARONA

A00358687

ALGORTIMOS Y PROGRAMACIÓN 2

UNIVERSIDAD ICESI

AGOSTO 13 DE 2019

**Objetivos**

**Unidad 2: Pruebas Automáticas y Tipos de Excepción**

Al finalizar esta unidad, el estudiante estará en capacidad de:

OE2.1 Reconocer el mecanismo de manejo de excepciones señalando las implicaciones de la propagación versus el control.

OE2.2 Usar e implementar distintos tipos de excepción como parte de un programa, de manera que sea posible clasificar los tipos de error que se pueden presentar y asociarles en el programa distintas maneras de recuperarse ante el problema.

OE2.3 Diseñar pruebas unitarias automáticas que permitan validar el adecuado funcionamiento de las operaciones del sistema desarrolladas para soportar los requerimientos funcionales.

OE2.4 Desarrollar las clases y los métodos necesarios para implementar las pruebas unitarias automáticas, que ayudan a comprobar el correcto funcionamiento de un programa.

**Enunciado**

Buscaminas es un videojuego para un solo jugador inventado por Robert Donner en 1989. El objetivo del juego es despejar un campo de minas sin detonar ninguna. El juego adquirió mucha popularidad ya que viene incorporado en todas las versiones del sistema operativo Windows desde su versión 3.1.

El juego consiste en despejar todas las casillas de una pantalla que no oculten una mina. Las casillas que no ocultan una mina tienen un número, el cual indica la cantidad de minas que hay en las casillas circundantes. Así, si una casilla tiene el número 3, significa que de las ocho casillas que hay alrededor (si no es en una esquina o borde) hay 3 con minas y 5 sin minas. Si se descubre una casilla con número 0 indica que ninguna de las casillas vecinas tiene mina.

Si se descubre una casilla con una mina se pierde la partida.

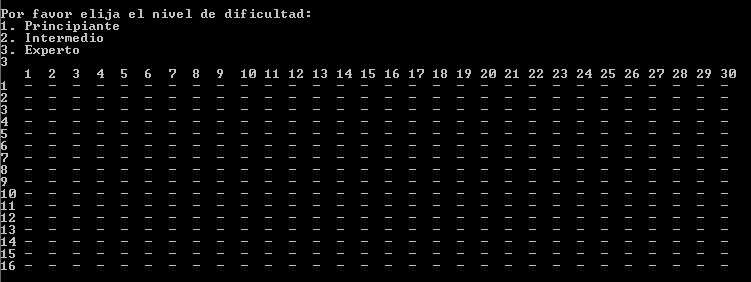
Se puede poner una marca en las casillas que el jugador piensa que hay minas para ayudar a descubrir las que están cerca.

El juego posee diferentes niveles de dificultad, en ellos se cambian la cantidad de minas en el mapa, y el tamaño del tablero donde se encuentran las minas. Los niveles son:

* Nivel principiante: 8 × 8 casillas y 10 minas.
* Nivel intermedio: 16 × 16 casillas y 40 minas.
* Nivel experto: 16 × 30 casillas y 99 minas.

Tu labor es terminar el desarrollar un programa que que ya fue iniciado para jugar buscaminas en la consola del computador. Para poder jugar el usuario podrá seleccionar el nivel. Una vez seleccionado el nivel se mostrará en la consola el tablero que representa el buscaminas y además se decidió que para ayudar al usuario a ubicarse en el tablero se mostrarán las posiciones asociadas a cada una de las filas y las columnas (Puede ver un ejemplo de cómo se vería un tablero experto en la imagen 1). Como esta versión del buscaminas es en la consola, no es necesaria implementar la opción en la cual se marca una casilla como mina.

Un usuario debe saber cuando perdió o cuando gano el juego, también debe tener una opción que muestre la solución del buscaminas en caso de que usuario está cansado de jugar y desee ver la solución, claramente, cuando se le muestra la solución no se considera como ganador. Un usuario podrá abrir una casilla del buscaminas utilizando las coordenadas que se muestran en la pantalla, una vez ingresadas las coordenadas correctamente se debe mostrar de nuevo el estado del tablero con la casilla deseada destapada. Como parte de una ayuda pedagógica para aprender el juego, se decidió agregar la opción llamada dar pista, dar pista consiste en que el programa abre la primera casilla que no sea una Mina y cuya cantidad de minas alrededor sea diferente de 0, una vez se muestre la pista se mostrará el tablero con la casilla abierta y se debe comunicar qué casilla abrió la pista. Un usuario se considera ganador del juego cuando destapa todas las casillas que no son mina.

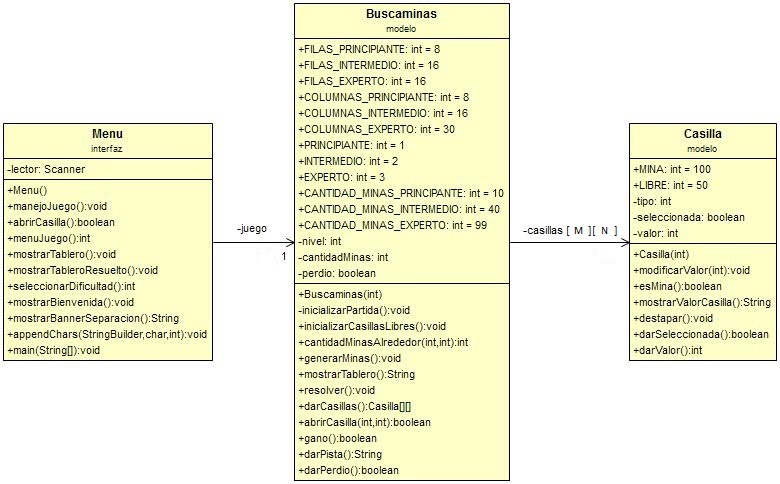


*Imagen 1 - Ejemplo de cómo se debe ver el tablero*

**Entregables**

1. Especificación de Requerimientos Funcionales y no funcionales.
2. Diagrama de Clases Completo (incluye el Modelo y el Main en la interfaz). El modelo debe ser elaborado digitalmente, pero NO generado automáticamente (por ejemplo, no es válido entregar modelos generados por ObjectAid o ninguna otra herramienta).
3. Trazabilidad del Análisis al Diseño. Una tabla a tres columnas en la que se relaciona cada requerimiento con el método o métodos que permiten satisfacer dicho requerimiento.
4. Implementación en Java.
5. Pruebas de todas las funcionalidades del aplicativo.
6. Excepciones que garanticen el buen funcionamiento del programa.

Se hace entrega de una versión sin terminar y usted deberá completar, realizar todas las excepciones y pruebas necesarias que requiera para que la aplicación funcione correctamente. A Continuación un diagrama inicial sugerido.



**Nota:** Usted debe entregar un archivo comprimido en formato zip de un directorio con únicamente 2 archivos: 1 archivo en formato pdf con toda la documentación (análisis, diseño, tabla de trazabilidad, Api) y otro archivo comprimido de un directorio con los archivos de codificación en sus respectivos paquetes.

El nombre del archivo comprimido debe tener el formato: PRIMERAPELLIDO\_PRIMERNOMBRE.zip (tenga en cuenta que el separador entre el primer apellido y el primer nombre es un guión al piso).

**Desarrollo**

**1.**

|  |  |
| --- | --- |
| **Nombre** | **R.F. 1. Generar el juego** |
| **Resumen** | El usuario debe ingresar la dificultad del juego para que el sistema cree un tablero con unas dimensiones específicas, además de ubicar aleatoriamente las minas en el tablero. |
| **Entradas** | |
| -Dificultad del juego | |
| **Resultados** | |
| Tablero de juego. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 2. Abrir una casilla** |
| **Resumen** | El usuario debe ingresar la información de una zona al sistema para poder ser utilizada dentro del programa |
| **Entradas** | |
| -Coordenada en x de la casilla  -Coordenada en y de la casilla | |
| **Resultados** | |
| Registrar la información de una zona en el sistema. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 3. Dar una pista** |
| **Resumen** | El usuario debe ingresar la opción de pedir una pista, para que el sistema destape la casilla primera casilla más cercana a (0,0) que tenga alredor 1 mina por lo menos |
| **Entradas** | |
| -Opción de dar una pista | |
| **Resultados** | |
| Casilla de pista destapada. | |

|  |  |
| --- | --- |
| **Nombre** | **R.F. 4. Resolver el juego** |
| **Resumen** | El usuario debe ingresar la opción resolver el juego para que se muestre el tablero destapado por completo y se finalice la partida |
| **Entradas** | |
| -Opción de resolver el juego. | |
| **Resultados** | |
| Tablero resuelto. | |

**3.**

|  |  |  |
| --- | --- | --- |
| **REQUERIMIENTO FUNCIONAL** | **CLASES** | **METODOS** |
| **R.F. 1. Generar el juego** | Menu  Buscaminas  Casilla | +Menu()  +mostrarBienvenida():void  +mostrarBannerSeparacion():void  +seleccionarDificultad():int  +Buscaminas(int)  +manejoJuego():void  +mostrarTablero():void  +mostrarTablero():void  +mostrarValorCasilla():int  +mostrarTableroResuelto():void |
| **R.F. 2. Abrir una casilla** | Menu  Buscaminas  Casilla | +abrirCasilla():boolean  +darSeleccionada():boolean  +esMina():boolean  +destapar():void  +darPerdio():boolean  +mostrarTablero():void  +gano():boolean |
| **R.F. 3. Dar una pista** | Menu  Buscaminas  Casilla | +darPista():boolean  +darValor():int  +darSeleccionada():boolean  +esMina():boolean  +destapar():void |
| **R.F. 4. Resolver el juego** | Menu  Buscaminas  Casilla | +resolver():void  +destapar():void  +mostrarTablero():void |

**Diseño de casos de pruebas unitarias**

**Buscaminas:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valor Entrada** | **Resultado** |
| Buscaminas | inicializarCasillasLibres() | El escenario inicializa un buscaminas de nivel principiante que es una matriz 8x8 con diez minas ubicadas aleatoriamente en ella | - | Verdadero, porque al contar todas las casillas libres del la matriz son 54=8x8-10 |
| Buscaminas | inicializarCasillasLibres() | El escenario inicializa una buscaminas de nivel intermedio que es una matriz 16x16 con 40 minas ubicadas aleatoriamente en ella r | - | Verdadero, porque al contar todas las casillas libres del la matriz son 216=16x16-40 |
| Buscaminas | inicializarCasillasLibres() | El escenario inicializa un buscaminas de nivel experto que es una matriz de 16x30 con 99 minas ubicadas aleatoriamente en ella |  | Verdadero, porque al contar todas las casillas libres del la matriz son 381=16x30-99 |

**Objetivo:** Probar si el método de inicializar las casillas libres genera la cantidad de casillas libres para cada nivel de manera correcta.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valor Entrada** | **Resultado** |
| Buscaminas | inicializarPartida() | El escenario inicializa un buscaminas de nivel principiante con sus valores por defecto | - | Verdadero, puesto que el método solo puede trabajar con 3 niveles, principiante(1), intermedio(2), y experto(3) |
| Buscaminas | inicializarPartida() | El escenario inicializa un buscaminas de nivel intermedio con sus valores por defecto | - | Verdadero , puesto que el método solo puede trabajar con 3 niveles, principiante(1), intermedio(2), y experto(3) |
| Buscaminas | inicializarPartida() | El escenario inicializa un buscaminas de nivel experto con sus valores por defecto | - | Verdadero, puesto que el método solo puede trabajar con 3 niveles, principiante(1), intermedio(2), y experto(3) |

**Objetivo:** Probar si la partida se inicializa según alguna escala de dificultad dada.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valor Entrada** | **Resultado** |
| Buscaminas | generarMinas() | El escenario inicializa un buscaminas de nivel principiante con sus valores por defecto, ubicando sus minas correspondientes de manera aleatoria | - | Verdadero, puesto que el contador de minas es el mismo con el numero de minas que debe haber en el nivel principiante |
| Buscaminas | generarMinas() | El escenario inicializa un buscaminas de nivel intermedio con sus valores por defecto, ubicando sus minas correspondientes de manera aleatoria | - | Verdadero, puesto que el contador de minas es el mismo con el numero de minas que debe haber en el nivel intermedio |
| Buscaminas | generarMinas() | El escenario inicializa un buscaminas de nivel experto con sus valores por defecto, ubicando sus minas correspondientes de manera aleatoria | - | Verdadero, puesto que el contador de minas es el mismo con el numero de minas que debe haber en el nivel experto |

**Objetivo:** Probar si el método de generar minas, genera la cantidad de minas de manera aleatoria correspondiente a cada nivel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clase | Método | Escenario | Valor Entrada | Resultado |
| Buscaminas | resolver(); | El escenario inicializa un buscaminas con nivel principiante con los valores por defecto | - | Verdadero, puesto que marca todas las casillas del buscaminas como seleccionadas y las muestra por pantalla |
| Buscaminas | resolver(); | El escenario inicializa un buscaminas con nivel intermedio con los valores por defecto | - | Verdadero, puesto que marca todas las casillas del buscaminas como seleccionadas y las muestra por pantalla |
| Buscaminas | Resolver(); | El escenario inicializa un buscaminas con nivel experto con los valores por defecto | - | Verdadero, puesto que marca todas las casillas del buscaminas como seleccionadas y las muestra por pantalla |

**Objetivo:** Probar si el método resuelve la matriz

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valor Entrada** | **Resultado** |
| Buscaminas | abrirCasilla(); | El escenario inicializa un buscaminas de nivel principiante/intermedio/experto con sus valores por defecto | La casilla -1,-1 | Falso, puesto que esas coordenadas para una casilla no existen |
| Buscaminas | abrirCasilla(); | El escenario inicializa un buscaminas de nivel principiante/intermedio/experto con sus valores por defecto | La casilla 300,200 | Falso, puesto que esas coordenadas para una casilla exceden los límites de la matriz, por tanto no existen |
| Buscaminas | abrirCasilla(); | El escenario inicializa un buscaminas de nivel principiante/intermedio/experto con sus valores por defecto | La casilla  -1,200 | Falso, puesto que esas coordenadas para una casilla no existen |
| Buscaminas | abrirCasilla(); | El escenario inicializa un buscaminas de nivel principiante/intermedio/experto con sus valores por defecto | La casilla 1,1 | Verdadero, puesto que la casilla siempre existe y es posible destaparla |

**Objetivo:** Probar si el método abre las casillas dentro de una matriz especifica a una dificultad.

**Casilla:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Casilla | Casilla(int) | El escenario inicializa una casilla como mina y los valores por defecto del constructor | 100 | Verdadero, se crea la casilla de tipo mina junto con sus valores por defecto |
| Casilla | Casilla(int) | El escenario inicializa una casilla como libre y los valores por defecto del constructor | 50 | Verdadero, se crea la casilla de tipo mina junto con sus valores por defecto |

**Objetivo:** Probar si el método está creando correctamente las minas y las casillas libres.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Casilla | esMina() | El escenario inicializa una casilla como mina y los valores por defecto del constructor | - | Verdadero puesto que la casilla es de tipo mina |
| Casilla | esMina() | El escenario inicializa una casilla como libre y los valores por defecto del constructor | - | Falso, puesto que la casilla es de tipo libre. |

**Objetivo:** Probar si el método reconoce cuando una casilla es mina está libre.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Casilla | modificarValor(int) | El método modifica el valor de minas que una casilla libre tiene a su alrededor | Valor de minas que una casilla libre tiene a su alrededor | Verdadero, ya que se hace una comparación se hace con el mismo numero que se modifico |
| Casilla | modificarValor(int) | El método modifica el atributo valor de una casilla | Cualquier n>0 | Falso, ya que las minas siempre deben tener como valor -1 |

**Objetivo:** Probar si el método modifica el atributo valor para cualquier casilla

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores Entrada** | **Resultado** |
| Casilla | mostrarCasillaValor() | El método inicializa una casilla como Libre y sus valores por defecto | - | Verdadero, ya que la casilla es de tipo mina y muestra la cadena “ - ” |
| Casilla | mostarCasillaValor() | El método inicializa una casilla como Mina y sus valores por defecto | - | Verdadero, ya que la casilla es de tipo mina y muestra la cadena “ \* ” |

**Objetivo:** Probar si el método genera la cadena de String esperada según el atributo tipo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores entrada** | **Resultado** |
| Casilla | destapar() | El método destapa una casilla mina | - | Verdadero, puesto que la casilla puede destaparse |
| Casilla | destapar(); | El método destapa una casilla libre | - | Verdadero, puesto que la casilla puede destaparse |
| Casilla | destapar(); | El método destapa una casilla donde seleccionado = true | - | Falso, puesto que la casilla ya estaba destapada |

**Objetivo:** Probar si el método resuelve la partida