

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 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.

Por favor elija el nivel de dificultad:

1. Principiante

2. Intermedio

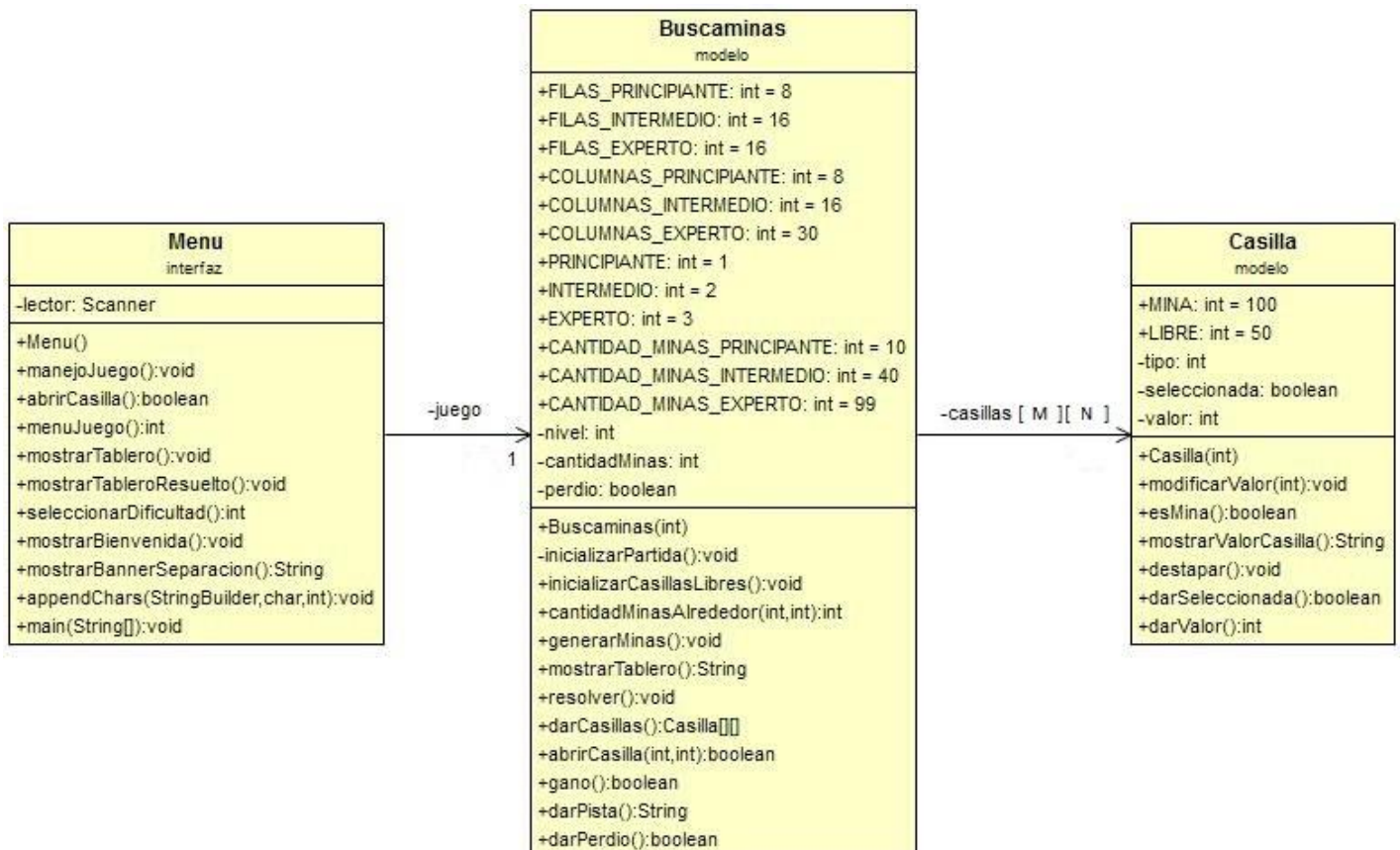
3. Experto

3	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

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.



Requerimientos funcionales

Nombre	R#1 Seleccionar nivel de dificultad
Descripción	El usuario puede escoger el nivel de dificultad del tablero para su partida, los niveles son: principiante, intermedio y experto.
Entrada	Un entero que representa el nivel de dificultad para del juego.
Salida	Se imprimirá en consola el tablero de juego con su respectiva dificultad.

Nombre	R#2 Mostrar solución
Descripción	Cuando el usuario no desee seguir jugando porque es muy difícil debe tener la opción de ver la solución para la partida actual. Si decide ver la solución esto hará automáticamente que pierda la partida.
Entrada	Un entero que activa la opción de “mostrar solución”
Salida	Se imprimirá en consola el tablero solucionado.

Nombre	R#3 Dar pista
Descripción	El usuario debe tener la opción de ver una pista que le ayude a ganar el juego. Se destapará la primera casilla que no sea una mina y cuyo numero de minas alrededor sea mayor a 0.
Entrada	Un entero que activa la opción de “dar pista”
Salida	<Ninguna>

Nombre	R#4 Abrir Casilla
Descripción	El usuario debe ser capaz de abrir una casilla del tablero para así poder jugar al busca minas jugar al busca minas en la consola.
Entrada	Int i: representa la fila para la casilla que se desea abrir. Int j: representa la columna para la casilla que se desea abrir.
Salida	

Requerimientos no funcionales

- Mostrar las posiciones asociadas a cada una de las filas y las columnas.
- Un usuario debe saber cuando perdió o cuando gano el juego.

- Una vez ingresadas las coordenadas correctamente se debe mostrar de nuevo el estado del tablero con la casilla deseada.

Trazabilidad del análisis al diseño

Requerimiento	Método(s)	Clase
R#1 Seleccionar nivel de dificultad	Menu(); seleccionarDificultad(); InicializarPartida(); Buscaminas();	Menu Menu Buscaminas Buscaminas
R#2 Mostrar solución	menuJuego(); manejoJuego(); mostrarTableroResuelto(); resolver(); destapar();	Menu Menu Menu Buscaminas Casilla
R#3 Dar pista	menuJuego(); manejoJuego(); darPista(); darValor(); esMina(); destapar();	Menu Menu Buscaminas Casilla Casilla Casilla
R#4 Abrir casilla	menuJuego(); manejoJuego(); abrirCasilla(); abrirCasilla(); destapar();	Menu Menu Menu Buscaminas Casilla

Pruebas unitarias

Configuración de los escenarios:

Clase	Escenario	Descripción
CasillaTest	setupEscenario1();	Se crea una instancia de Casilla. Se inicializa con su atributo tipo == 50, es decir que es una casilla libre.
CasillaTest	setupEscenario2();	Se crea una instancia de Casilla. Se inicializa con su atributo tipo == 100, es decir que es una casilla de tipo MINA.
BuscaminasTest	setupEscenario1();	

		Se crea una instancia de Buscaminas. Cuyo atributo de nivel será == 1. Por ende su nivel de dificultad será la de principiante, su matriz de casillas (tablero de juego) será de 8x8 y tendrá 10 Minas distribuidas aleatoriamente en la matriz.
BuscaminasTest	setupEscenario2();	Se crea una instancia de Buscaminas. Cuyo atributo de nivel será == 2. Por ende su nivel de dificultad será la de principiante, su matriz de casillas (tablero de juego) será de 16x16 y tendrá 40 Minas distribuidas aleatoriamente en la matriz.
BuscaminasTest	setupEscenario3();	Se crea una instancia de Buscaminas. Cuyo atributo de nivel será == 3. Por ende su nivel de dificultad será la de principiante, su matriz de casillas (tablero de juego) será de 16x30 y tendrá 99 Minas distribuidas aleatoriamente en la matriz.
BuscaminasTest	setupEscenario4();	Se crea una instancia de Buscaminas. Cuyo atributo de nivel será == 1. Pero esta vez en el tablero se posicionan manualmente 7 minas. Para verificar métodos como: destapar();, esMina(); y cantidadMinasAlrededor();

Casos de prueba:

Objetivo de la prueba: Verificar que una casilla es mina

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	esMina();	setupEscenario1();	<ninguno>	True. Se hace la aserción de que la casilla no es mina

Objetivo de la prueba: Verificar que una casilla es mina

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	esMina();	setupEscenario2();	<ninguno>	True. Se hace la aserción de que la casilla es mina

Objetivo de la prueba: Verificar el valor de una casilla que no es mina

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	mostrarValorCasilla();	setupEscenario1();	<ninguno>	Cuando se invoca el método para la casilla que no es mina deberá retornar un String igual a "-1"

Objetivo de la prueba: Verificar el valor de una casilla que no es mina

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	mostrarValorCasilla();	setupEscenario1();	<ninguno>	Cuando se invoca el método para la casilla que no es mina deberá retornar un String igual a "-1"

Objetivo de la prueba: Verificar el valor de una casilla que es mina

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	mostrarValorCasilla();	setupEscenario2();	<ninguno>	Cuando se invoca el método para la casilla que es mina deberá retornar un String igual a "**"

Objetivo de la prueba: Verificar apertura de casilla

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	destapar();	setupEscenario2();	<ninguno>	Se destapa la casilla y se verifica que esta allá sido destapada. Debe dar true si se hace la aserción de que la casilla esta seleccionada.

Objetivo de la prueba: Verificar la creación de la cantidad correcta de minas para el nivel principiante.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	generarMinas();	setupEscenario1();	<ninguno>	Las minas que se generan para este tablero deberán ser 10 . Se crea un contador inicializado en 0 el cual debe recorrer todo el tablero de juego el cual incrementa cada vez que encuentre una mina. Al final del recorrido el contador deberá ser igual a la cantidad de minas.

Objetivo de la prueba: Verificar la creación de la cantidad correcta de minas para el nivel intermedio.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	generarMinas();	setupEscenario2();	<ninguno>	Las minas que se generan para este tablero deberán ser 40 . Se crea un contador inicializado en 0 el cual debe recorrer todo el tablero de juego el cual incrementa cada vez que encuentre una mina. Al final del recorrido el contador deberá ser igual a la cantidad de minas.

Objetivo de la prueba: Verificar la creación de la cantidad correcta de minas para el nivel experto.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	generarMinas();	setupEscenario2();	<ninguno>	Las minas que se generan para este tablero deberán ser 99. Se crea un contador inicializado en 0 el cual debe recorrer todo el tablero de juego el cual incrementa cada vez que encuentre una mina. Al final del recorrido el contador deberá ser igual a la cantidad de minas.

Objetivo de la prueba: Verificar la creación del tablero de juego con las dimensiones especificadas para el nivel principiante.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	inicializarPartida();	setupEscenario1();	<ninguno>	Se multiplican las filas por las columnas y este resultado se compara con el resultado esperado. Deberá retornar true si ambos son iguales.

Objetivo de la prueba: Verificar la creación del tablero de juego con las dimensiones especificadas para el nivel intermedio.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	inicializarPartida();	setupEscenario2();	<ninguno>	Se multiplican las filas por las columnas y este resultado se compara con el resultado esperado. Deberá retornar true si ambos son iguales.

Objetivo de la prueba: Verificar la creación del tablero de juego con las dimensiones especificadas para el nivel experto.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	inicializarPartida();	setupEscenario3();	<ninguno>	Se multiplican las filas por las columnas y este resultado se compara con el resultado esperado. Deberá retornar true si ambos son iguales.

Objetivo de la prueba: Verificar la apertura de una casilla.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	abrirCasilla();	setupEscenario1();	Int i = 0; Int j = 0;	Se destapa la casilla en la posición i,j y se hace la aserción de que esta casilla está seleccionada. Debe retornar true.

Objetivo de la prueba: Verificar la cantidad de minas alrededor

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	cantidadDeMinasAlrededor();	setupEscenario4();	Int i = 1; Int j = 1;	Se destapa la casilla en la posición i,j y se hace la aserción de que alrededor de dicha casilla hay 7 minas ya que se pusieron manualmente y se sabe donde están. La cantidad que retorne el método para la cantidad i,j deberá ser entonces igual a 7.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	cantidadDeMinasAlrededor();	setupEscenario4();	Int i = 7; Int j = 7;	Se destapa la casilla en la posición i,j y se hace la aserción de que alrededor de dicha casilla hay 0 minas ya que se pusieron manualmente y se sabe donde están. La cantidad que retorne el método para la cantidad i,j deberá ser entonces igual a 0.