**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.

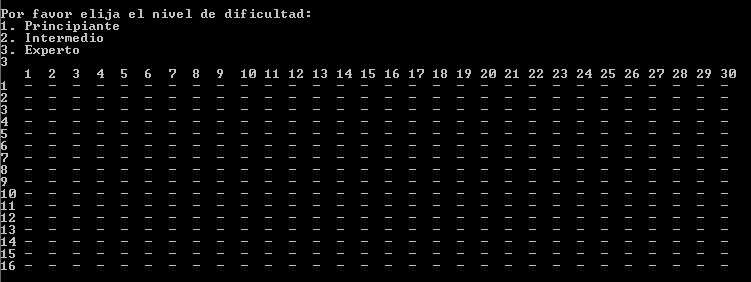
Se puede poner una marca en las casillas que el jugador piensa que hay minas para ayudar a descubrir las que están cerca.

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.

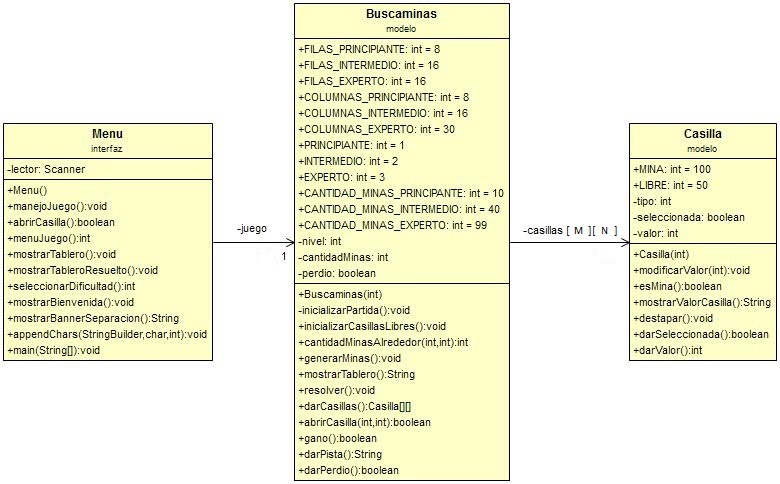


*Imagen 1 - Ejemplo de cómo se debe ver el tablero*

**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).

**Requerimientos funcionales**

|  |  |
| --- | --- |
| **NOMBRE** | Permitir al usuario elegir el nivel del juego |
| **RESUMEN** | Para poder jugar el usuario debe seleccionar cual nivel desea entre las tres opciones: principiante, intermedio y experto. |
| **ENTRADAS** | |
| * Nivel | |
| **RESULTADOS** | |
| El nivel del juego ha sido seleccionado | |

|  |  |
| --- | --- |
| **NOMBRE** | Mostrar el tablero del juego que representa el buscaminas |
| **RESUMEN** | El programa debe mostrar en consola el tablero del juego, dependiendo del nivel seleccionado por el usuario. |
| **ENTRADAS** | |
| < Ninguna > | |
| **RESULTADOS** | |
| El tablero del juego ha sido mostrado correctamente al usuario. | |

|  |  |
| --- | --- |
| **NOMBRE** | Mostrar las posiciones asociadas a cada una de las filas y columnas del tablero de juego |
| **RESUMEN** | El programa debe mostrar las posiciones asociadas a las filas y columnas del tablero con el fin de ayudar al usuario a ubicarse en el tablero. |
| **ENTRADAS** | |
| < Ninguna > | |
| **RESULTADOS** | |
| Las posiciones del tablero se muestran correctamente | |
| **NOMBRE** | Especificar cuando un usuario perdió la partida del juego |
| **RESUMEN** | El programa debe mostrarle al usuario que perdió, cuando este seleccione una casilla con una mina. |
| **ENTRADAS** | |
| < Ninguna > | |
| **RESULTADOS** | |
| Se muestra cuando el usuario pierde correctamente | |

|  |  |
| --- | --- |
| **NOMBRE** | Mostrar la solución del buscaminas. |
| **RESUMEN** | El programa debe mostrar la solución del buscaminas al usuario cuando este seleccione la opción. |
| **ENTRADAS** | |
| < Ninguna > | |
| **RESULTADOS** | |
| La solución del buscaminas se muestra correctamente. | |

|  |  |
| --- | --- |
| **NOMBRE** | Abrir una casilla del juego |
| **RESUMEN** | El programa abre una casilla seleccionada por el usuario dependiendo de las coordenadas mostradas en el tablero del juego |
| **ENTRADAS** | |
| * Posición fila * Posición columna | |
| **RESULTADOS** | |
| La casilla del juego se abre correctamente. | |

|  |  |
| --- | --- |
| **NOMBRE** | Dar una pista al usuario del juego |
| **RESUMEN** | El programa debe darle una pista al usuario cuando este seleccione la opción. Una pista consiste en mostrar una casilla que no sea una mina y cuyo valor sea mayor a 0 |
| **ENTRADAS** | |
| <Ninguna> | |
| **RESULTADOS** | |
| Se muestra correctamente la pista al usuario del juego | |

|  |  |
| --- | --- |
| **NOMBRE** | Especificar cuando un usuario es ganador en el juego |
| **RESUMEN** | El programa debe mostrar cuando el usuario gana la partida. Para que un usuario gane, se deben haber mostrado todas las casillas sin seleccionar una mina. |
| **ENTRADAS** | |
| <Ninguna> | |
| **RESULTADOS** | |
| Se muestra correctamente si el usuario gano la partida del juego | |

|  |  |
| --- | --- |
| **NOMBRE** | Crear interfaz gráfica |
| **RESUMEN** | El programa debe tener una interfaz gráfica. |
| **ENTRADAS** | |
| <Ninguna> | |
| **RESULTADOS** | |
|  | |

**Requerimientos no funcionales**

* No es necesario implementar la opción en la cual se marca una casilla como mina.
* Cuando se muestra la solución no se considera como ganador al usuario.
* El nivel principiante debe tener únicamente 8 filas, 8 columnas y 10 minas
* El nivel intermedio debe tener únicamente 16 filas, 16 columnas y 40 minas.
* El nivel experto debe tener únicamente 16 filas, 30 columnas y 99 minas.
* Al dar una pista, la casilla mostrada no puede ser una mina
* Al dar una pista, la casilla mostrada debe ser un valor mayor a 0

**Trazabilidad**

|  |  |  |
| --- | --- | --- |
| **Requerimientos funcionales** | **Clase** | **Método** |
|
| Permitir al usuario elegir el nivel del juego | Menu | +SeleccionarDificultad() |
| Buscaminas | +inicializarPartida() |
| Mostrar el tablero de juego | Buscaminas | +MostrarTablero() |
| Menu | +MostrarTablero() |
| Mostrar las posiciones asociadas a filas y columnas | Buscaminas | +MostrarTablero() |
| Especificar cuando un usuario perdió la partida | Buscaminas | +darPerdio() |
| Mostrar la solución del buscaminas | Menu | +MostrarTableroResuelto() |
| Abrir una casilla del juego | Buscaminas | +AbrirCasilla() |
| Menu | +abrirCasilla() |
| Dar una pista al usuario del juego | Buscaminas | +darPista() |
| Menu | +manejoJuego() |
| Especificar cuando el usuario gano | Buscaminas | +gano() |
| Menu | +manejoJuego() |

**Documentación pruebas**

**Objetivo:** Probar que el método crea la cantidad correcta de minas y el tamaño de las filas al igual que las columnas es el correcto según el nivel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | inicializarPartida(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | nivelEsperado = PRINCIPIANTE;  ColumnasEsperadas = 8;  FilasEsperadas = 8;  CantidadMinasEsperadas = 10; | Verdadero, el método asigna la cantidad correcta de minas al igual que el tamaño de la matriz. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | inicializarPartida(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = INTERMEDIO;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | nivelEsperado = INTERMEDIO;  ColumnasEsperadas = 16;  FilasEsperadas = 16;  CantidadMinasEsperadas = 40; | Verdadero, el método asigna la cantidad correcta de minas al igual que el tamaño de la matriz. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | inicializarPartida(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | nivelEsperado = PRINCIPIANTE;  ColumnasEsperadas = 30;  FilasEsperadas = 16;  CantidadMinasEsperadas = 99; | Verdadero, el método asigna la cantidad correcta de minas al igual que el tamaño de la matriz. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método asigna la cantidad correcta de casillas libres

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | inicializarCasillasLibres(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | casillasLibresEsperadas = 54; | Verdadero, el método asigna 54 casillas libres. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | inicializarCasillasLibres(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | casillasLibresEsperadas = 216; | Verdadero, el método asigna 216 casillas libres |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | inicializarCasillasLibres(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | casillasLibresEsperadas = 381; | Verdadero, el método asigna 38| casillas libres |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método no asigna un número mayor a 8 y tampoco menor a 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | cantidadMinasAlrededor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | correctoEsperado = true; | Verdadero, el método no asigna un número mayor a 8 y tampoco menor a 0 . |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | cantidadMinasAlrededor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | correctoEsperado = true; | Verdadero, el método no asigna un número mayor a 8 y tampoco menor a 0 . |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | cantidadMinasAlrededor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | correctoEsperado = true; | Verdadero, el método no asigna un número mayor a 8 y tampoco menor a 0 . |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método genera la cantidad de minas correcta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | generarMinas(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | cantidadMinasEsperadas = 10; | Verdadero, método no genera más de 10 minas, ni menos de 10. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | generarMinas(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | cantidadMinasEsperadas = 40; | Verdadero, método no genera más de 10 minas, ni menos de 10. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | generarMinas(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | cantidadMinasEsperadas = 99; | Verdadero, método no genera más de 10 minas, ni menos de 10. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método no considera ganador al usuario cuando muestra la respuesta del tablero de juego.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | resolver(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | perdioEsperado = true; | Verdadero, el método no considera ganador a un usuario que eligió la opción de mostrar respuesta. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | resolver(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | perdioEsperado = true; | Verdadero, el método no considera ganador a un usuario que eligió la opción de mostrar respuesta. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | resolver(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | perdioEsperado = true; | Verdadero, el método no considera ganador a un usuario que eligió la opción de mostrar respuesta. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que método dar casillas muestra la casilla en la posición correcta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | abrirCasilla(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | valorEsperado = 5; | Verdadero, el método retorna la casilla correcta. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | abrirCasilla(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | valorEsperado = 5; | Verdadero, el método retorna la casilla correcta |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | abrirCasilla(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | valorEsperado = 5; | Verdadero, el método retorna la casilla correcta |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método no declara al usuario como ganador si este abre una mina

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | gano(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | esperado = false; | Verdadero, el método no asigna como ganador al usuario si la casilla que abrió es una mina |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | gano(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | esperado = false; | Verdadero, el método no asigna como ganador al usuario si la casilla que abrió es una mina |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | gano(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | esperado = false; | Verdadero, el método no asigna como ganador al usuario si la casilla que abrió es una mina |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método retorna si el atributo perdió correctamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | darPerdio(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | Esperado = true; | Verdadero, el método retorna el atributo perdió correctamente. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darPerdio(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40 | Esperado = true; | Verdadero, el método retorna el atributo perdió correctamente. |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darPerdio(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | Esperado = true; | Verdadero, el método retorna el atributo perdió correctamente. |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Demostrar que el método retorna el atributo correcto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | darNivel(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | nivelEsperado = PRINCIPIANTE; | Verdadero, el método retorna el nivel principiante (1) |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darNivel(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | nivelEsperado = INTERMEDIO; | Verdadero, el método retorna el nivel intermedio (2) |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darNivel(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | nivelEsperado = EXPERTO; | Verdadero, el método retorna el nivel experto (3) |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** Demostrar que el método retorna el atributo correcto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Buscaminas | darCantidadMinas(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | cantidadMinasEsperado = 10 | Verdadero, el método retorna la cantidad de minas correcta |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darNivel(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | cantidadMinasEsperado = 40 | Verdadero, el método retorna la cantidad de minas correcta |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darNivel(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | cantidadMinasEsperado = 99 | Verdadero, el método retorna la cantidad de minas correcta |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método modifica el valor correctamente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Casilla | modificarValor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | valorEsperado = 4 | Verdadero, el método cambia correctamente el valor de la casilla |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | modificarValor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | valorEsperado = 4 | Verdadero, el método cambia correctamente el valor de la casilla |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | modificarValor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | valorEsperado = 4 | Verdadero, el método cambia correctamente el valor de la casilla |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que el método retorna un boolean correctamente dependiendo si la casilla es de tipo mina o no

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Casilla | esMina(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | esperado = true; | Verdadero, el método retorna correctamente en caso de que la casilla sea o no una mina |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | esMina(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | esperado = true; | Verdadero, el método retorna correctamente en caso de que la casilla sea o no una mina |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | esMina(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | esperado = true; | Verdadero, el método retorna correctamente en caso de que la casilla sea o no una mina |
|
|
|
|
|
|
|
|
|
|

**Objetivo:** demostrar que dependiendo el tipo de casilla, el método muestra correctamente su valor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Casilla | mostrarValorCasilla(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | valorEsperadoCasillaSinSeleccionar = “-“;  valorEsperadoCasillaSeleccionadaMina = “\*”;  ValorEsperadoCasillaSeleccionadaValor = “1”; | Verdadero, el método retorna correctamente el valor de la casilla |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | mostrarValorCasilla(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | valorEsperadoCasillaSinSeleccionar = “-“;  valorEsperadoCasillaSeleccionadaMina = “\*”;  ValorEsperadoCasillaSeleccionadaValor = “1”; | Verdadero, el método retorna correctamente el valor de la casilla |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | mostrarValorCasilla(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | valorEsperadoCasillaSinSeleccionar = “-“;  valorEsperadoCasillaSeleccionadaMina = “\*”;  ValorEsperadoCasillaSeleccionadaValor = “1”; | Verdadero, el método retorna correctamente el valor de la casilla |

**Objetivo:** Demostrar que el método cambia el atributo seleccionada cuando se abre una casilla

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Casilla | destapar(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | seleccionadaEsperado = true; | Verdadero, el método cambia el atributo seleccionada de la casilla abierta |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | destapar(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | seleccionadaEsperado = true; | Verdadero, el método cambia el atributo seleccionada de la casilla abierta |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | destapar(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | seleccionadaEsperado = true; | Verdadero, el método cambia el atributo seleccionada de la casilla abierta |

**Objetivo:** Demostrar que el método retorna correctamente el atributo seleccionada

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Casilla | darSeleccionada(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | seleccionadaEsperado = true; | Verdadero, el método retorna correctamente el atributo seleccionada |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darSeleccionada(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | seleccionadaEsperado = true; | Verdadero, el método retorna correctamente el atributo seleccionada |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darSeleccionada(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | seleccionadaEsperado = true; | Verdadero, el método retorna correctamente el atributo seleccionada |

**Objetivo:** Demostrar que el método retorna correctamente el atributo valor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de entrada** | **Resultado** |
|
| Casilla | darValor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = principiante;  Columnas = 8;  Filas = 8;  CantidadMinas = 10; | valorEsperado = 1; | Verdadero, el método retorna correctamente el atributo darValor |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darValor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = intermedio;  Columnas = 16;  Filas = 16;  CantidadMinas = 40; | valorEsperado = 1; | Verdadero, el método retorna correctamente el atributo darValor |
|
|
|
|
|
|
|
|
|
|
| Buscaminas | darValor(); | El escenario inicializa una matriz adquiriendo los siguientes valores:  Nivel = experto;  Columnas = 30;  Filas = 16;  CantidadMinas = 99; | valorEsperado = 1; | Verdadero, el método retorna correctamente el atributo darValor |