

Requerimientos funcionales:

R1	Seleccionar nivel
Resumen	El programa permite que el usuario pueda seleccionar la dificultad al iniciar el juego. De esto dependerá el tamaño del tablero y la cantidad de minas que se generen en él.
Entradas	Nivel seleccionado.
Salida	Se selecciona el nivel.

R2	Mostrar tablero
Resumen	El programa muestra el tablero de juego una vez que el usuario haya escogido la dificultad. Dicho tablero tiene un formato específico para indicar el número de filas y columnas. Además, este se actualiza cada que se abre una casilla.
Entradas	Ninguna.
Salida	Se muestra el tablero.

R3	Verificar ganó
Resumen	El programa permite determinar cuándo un usuario gana el juego. Para ganar, el usuario deberá seleccionar todas las casillas que no sean minas.
Entradas	Ninguna.
Salida	El programa indica si el usuario ganó o no en el momento de su ejecución.

R4	Verificar perdió
Resumen	El programa permite determinar cuándo un usuario perdió el juego y terminar la partida en tal caso. La única manera de perder en el juego es seleccionando una mina.
Entradas	Ninguna.
Salida	El programa indica que el usuario perdió cuando este selecciona una mina.

R5	Resolver buscaminas
Resumen	El programa está en la capacidad de resolver el buscaminas y el usuario así lo desea. Resolverlo significaría abrir todas las casillas que no sean minas.
Entradas	Ninguna.
Salida	Se resuelve el tablero.

R6	Abrir casilla
Resumen	El programa permite al usuario abrir una casilla cualquiera que se encuentre en el tablero y no haya sido seleccionada antes. En caso de haber sido seleccionada antes, no podrá abrirla.
Entradas	Número de fila y columna.
Salida	Se abre o no la casilla.

R7	Dar pista
Resumen	El programa permite al usuario obtener una pista si este lo desea. La pista consiste en abrir la primera casilla libre con al menos una mina alrededor.
Entradas	Ninguna.
Salida	Se abre la primera casilla libre del tablero con un valor mayor a 0.

R8	Poner bandera
Resumen	El programa permite al usuario marcar una casilla con una bandera, lo cual bloqueará el botón para que la casilla no pueda ser destapada.
Entradas	Ninguna.
Salida	Se pone la bandera en la casilla seleccionada.

R9	Reiniciar nivel
Resumen	El programa permite al usuario reiniciar el nivel creando un nuevo tablero de juego con la misma dificultad.
Entradas	Ninguna.
Salida	Se reinicia el nivel.

R10	Mostrar menú
Resumen	El programa muestra un menú interactivo que permite al usuario interactuar con todas las funcionalidades del programa. Este es lo suficientemente robusto como para que el usuario no pueda causar errores internos.
Entradas	Ninguna.
Salida	Se muestra el menú.

Diseño de las pruebas Casilla

Objetivo: Probar que el método es capaz de definir si una casilla es mina o no.

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	esMina();	Escenario 1	---	Verdadero, la casilla en efecto es una mina.
Casilla	esMina();	Escenario 2	---	Falso, la casilla no es una mina.

Objetivo: Probar que el método es capaz de mostrar el valor de una casilla según el estado en el que se encuentre la misma (libre seleccionada, libre no seleccionada, mina seleccionada y mina no seleccionada).

Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	<code>mostrarValorCasilla();</code>	Escenario 1	---	El valor que se muestra de la casilla debe ser “-” mientras no haya sido destapada.
Casilla	<code>mostrarValorCasilla();</code>	Escenario 3	---	El valor que se muestra de la casilla debe ser “*” ya que es una mina.
Casilla	<code>mostrarValorCasilla();</code>	Escenario 2	---	El valor de la casilla debe ser “-” mientras no haya sido destapada.
Casilla	<code>mostrarValorCasilla();</code>	Escenario 4	---	El valor que se muestra de la casilla debe ser “-1” ya que es una casilla libre.

Objetivo: Probar que el método permite modificar el estado del atributo “seleccionada” de una casilla a true.

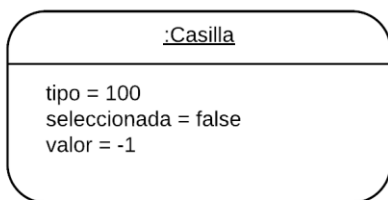
Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	<code>destapar();</code>	Escenario 1	---	El estado del atributo seleccionada cambia a true.

Objetivo: Probar que el método permite modificar el valor de una casilla dando dicho valor por parámetro.

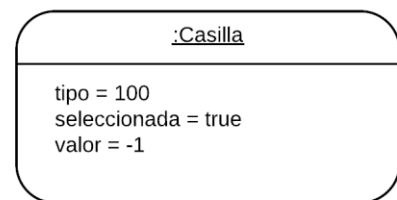
Clase	Método	Escenario	Valores de entrada	Resultado
Casilla	modificarValor ();	Escenario 2	Nuevo valor = -469846232	El valor que tenga la casilla en el momento se modifica por el valor de entrada.

Escenarios para el test de Casilla:

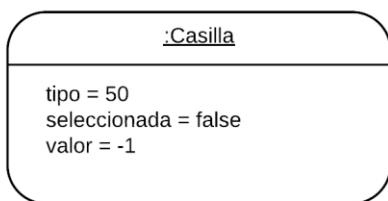
Escenario 1



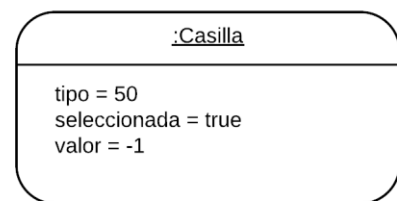
Escenario 3



Escenario 2



Escenario 4



Diseño de pruebas Buscaminas

Objetivo: Probar que el método cuenta correctamente las minas que se encuentran alrededor de la casilla seleccionada, sin importar la posición de esta en el tablero (esquinas y en donde sus posibles casillas alrededor se encuentren dentro del tablero).

Clase	Método	Escenario	Valores de entrada	Resultado
-------	--------	-----------	--------------------	-----------

Buscaminas	<u>cantidadMinasAlrededor(int i, int j);</u>	Escenario 1	i = 0 j = 0	El método debe retornar un entero igual a 1 ya que solamente hay una mina alrededor.
Buscaminas	cantidadMinasAlrededor(int i, int j);	Escenario 2	i = 7 j = 7	El método debe retornar un entero igual a 2 ya que hay dos minas alrededor.
Buscaminas	cantidadMinasAlrededor(int i, int j);	Escenario 3	i = 0 j = 7	El método debe retornar un entero igual a 1 ya que solamente hay una mina alrededor.
Buscaminas	cantidadMinasAlrededor(int i, int j);	Escenario 4	i = 7 j = 0	El método debe retornar un entero igual a 3 ya que hay tres minas alrededor.
Buscaminas	cantidadMinasAlrededor(int i, int j);	Escenario 5	i = 1 j = 3	El método debe retornar un entero igual a 4 ya que hay cuatro minas alrededor.

Objetivo: Probar que el método genera la cantidad de minas adecuada según el nivel escogido y que, además, las genera aleatoriamente.

Clase	Método	Escenario	Valores de entrada	Resultado
-------	--------	-----------	--------------------	-----------

Buscaminas	<code>generarMinas();</code>	Escenario 6	---	El método generó correctamente 99 minas a lo largo de todo el tablero.
Buscaminas	<code>generarMinas();</code>	Escenario 7	---	El método generó correctamente 40 minas a lo largo de todo el tablero.
Buscaminas	<code>generarMinas();</code>	Escenario 8	---	El método generó correctamente 10 minas a lo largo de todo el tablero.

Objetivo: Probar que el método inicializa las casillas libres con el número de minas que estas tienen alrededor. Se prueba para los tres tipos de tablero.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>inicializarCasillasLibres();</code>	Escenario 6	---	El método modificó únicamente las casillas que no eran minas, agregándoles un valor dado por el método “contarMinasAlrededor();”
Buscaminas	<code>inicializarCasillasLibres();</code>	Escenario 7	---	El método modificó únicamente las casillas que no eran minas, agregándoles un valor dado por el método “contarMinasAlrededor();”

Buscaminas	<code>inicializarCasillasLibres();</code>	Escenario 8	---	El método modificó únicamente las casillas que no eran minas, agregándoles un valor dado por el método “ <code>contarMinasAlrededor();</code> ”
------------	---	-------------	-----	---

Objetivo: Se prueba que el método genera y muestra el tablero adecuadamente, es decir, con su respectivo formato y mostrando el valor de las casillas según su estado.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>mostrarTablero();</code>	Escenario 9	---	Al mostrar el tablero, este tendrá el formato adecuado (incluyendo los números guías para las filas y columnas), las casillas sin seleccionar se mostrarán como “-” y las seleccionadas con su respectivo valor.
Buscaminas	<code>mostrarTablero();</code>	Escenario 10	---	Al mostrar el tablero, este tendrá el formato adecuado (incluyendo los números guías para las filas y columnas) y las casillas mostrarán su valor (excepto por la casilla de tipo mina, la cual mostrará un “*”))

Buscaminas	<code>mostrarTablero();</code>	Escenario 11	---	Al mostrar el tablero, este tendrá el formato adecuado (incluyendo los números guías para las filas y columnas), las casillas sin seleccionar se mostrarán como “-” y las seleccionadas con su respectivo valor.
Buscaminas	<code>mostrarTablero();</code>	Escenario 12	---	Al mostrar el tablero, este tendrá el formato adecuado (incluyendo los números guías para las filas y columnas) y las casillas mostrarán su valor (excepto por la casilla de tipo mina, la cual mostrará un “*”))
Buscaminas	<code>mostrarTablero();</code>	Escenario 13	---	Al mostrar el tablero, este tendrá el formato adecuado (incluyendo los números guías para las filas y columnas), las casillas sin seleccionar se mostrarán como “-” y las seleccionadas con su respectivo valor.
Buscaminas	<code>mostrarTablero();</code>	Escenario 14		Al mostrar el tablero, este tendrá el formato adecuado (incluyendo los

			---	números guías para las filas y columnas) y las casillas mostrarán su valor (excepto por la casilla de tipo mina, la cual mostrará un “*”)
--	--	--	-----	---

Objetivo: Probar que el método abre todas las casillas del tablero que no sean minas en todos los tipos de tablero posibles.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>resolver();</code>	Escenario 6	---	El método modificó el estado del atributo seleccionada de todas las casillas a true.
Buscaminas	<code>resolver();</code>	Escenario 7	---	El método modificó el estado del atributo seleccionada de todas las casillas a true.
Buscaminas	<code>resolver();</code>	Escenario 8	---	El método modificó el estado del atributo seleccionada de todas las casillas a true.

Objetivo: Probar que el método abre las casillas únicamente cuando esta no ha sido seleccionada. Y que en caso de ser una mina, cambia el atributo perdió de false a true.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>abrirCasilla(int i, int j);</code>	Escenario 15	i = 0 j = 0	Verdadero. La casilla de abre con éxtio ya que no ha sido seleccionada.
Buscaminas	<code>abrirCasilla(int i, int j);</code>	Escenario 15	i = 1 j = 1	Falso. No es posible abrir la casilla debido a que ya ha sido seleccionada.
Buscaminas	<code>abrirCasilla(int i, int j);</code>	Escenario 15	i = 2 j = 2	Verdadero. La casilla de abre con éxtio ya que no ha sido seleccionada pero también cambia el estado del atributo perdió de buscaminas a true.

Objetivo: Probar que el método define, en todos los casos, si el usuario ganó según el estado del tablero.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>gano();</code>	Escenario 16	---	Verdadero. Todas las casillas libres han sido descubiertas.
Buscaminas	<code>gano();</code>	Escenario 17	---	Falso. Aún falta una casilla libre por descubrir.

Buscaminas	<code>gano();</code>	Escenario 18	---	Verdadero. Todas las casillas libres han sido descubiertas.
Buscaminas	<code>gano();</code>	Escenario 19	---	Falso. Aún falta una casilla libre por descubrir.
Buscaminas	<code>gano();</code>	Escenario 20	---	Verdadero. Todas las casillas libres han sido descubiertas.
Buscaminas	<code>gano();</code>	Escenario 21	---	Falso. Aún falta una casilla libre por descubrir.

Objetivo: Probar que el método da abre la primera casilla libre con valor mayor o igual a 1 que encuentre en el tablero generado.

Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>darPista();</code>	Escenario 22	---	El método abre la primera casilla libre que no ha sido seleccionada que encuentre y devuelve un String con su posición.

Objetivo: Probar que el método verifica si una fila escogida por el usuario está fuera del límite del tablero o no.

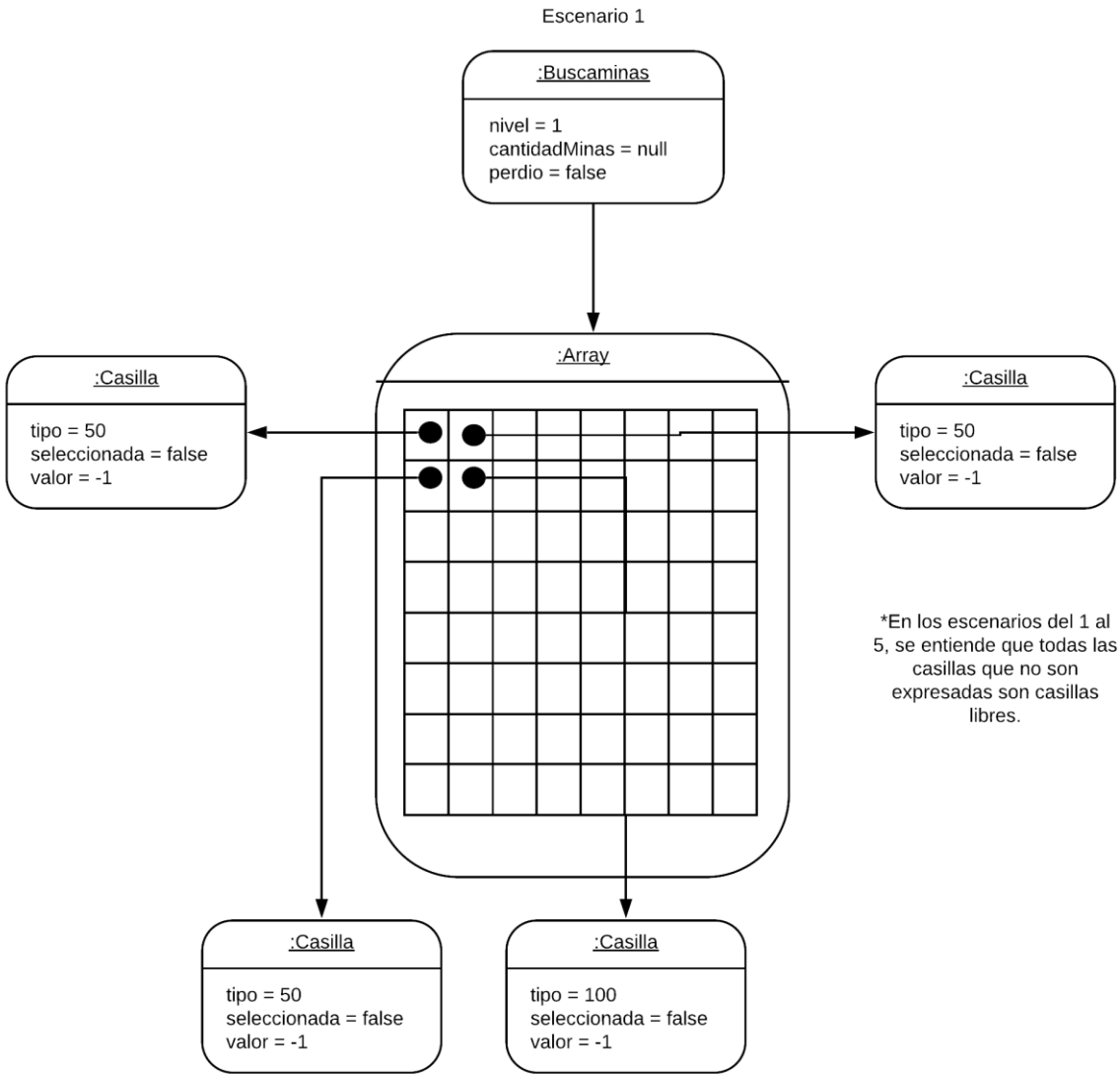
Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>rowVerifier(int i);</code>	Escenario 6	i = -1	Falso. El número introducido no hace parte de las filas de la matriz.
Buscaminas	<code>rowVerifier(int i);</code>	Escenario 6	i = 0	Verdadero. El número introducido hace parte de las filas de la matriz.
Buscaminas	<code>rowVerifier(int i);</code>	Escenario 6	i = 98	Falso. El número introducido no hace parte de las filas de la matriz.

Objetivo: Probar que el método verifica si una columna escogida por el usuario está fuera del límite del tablero o no.

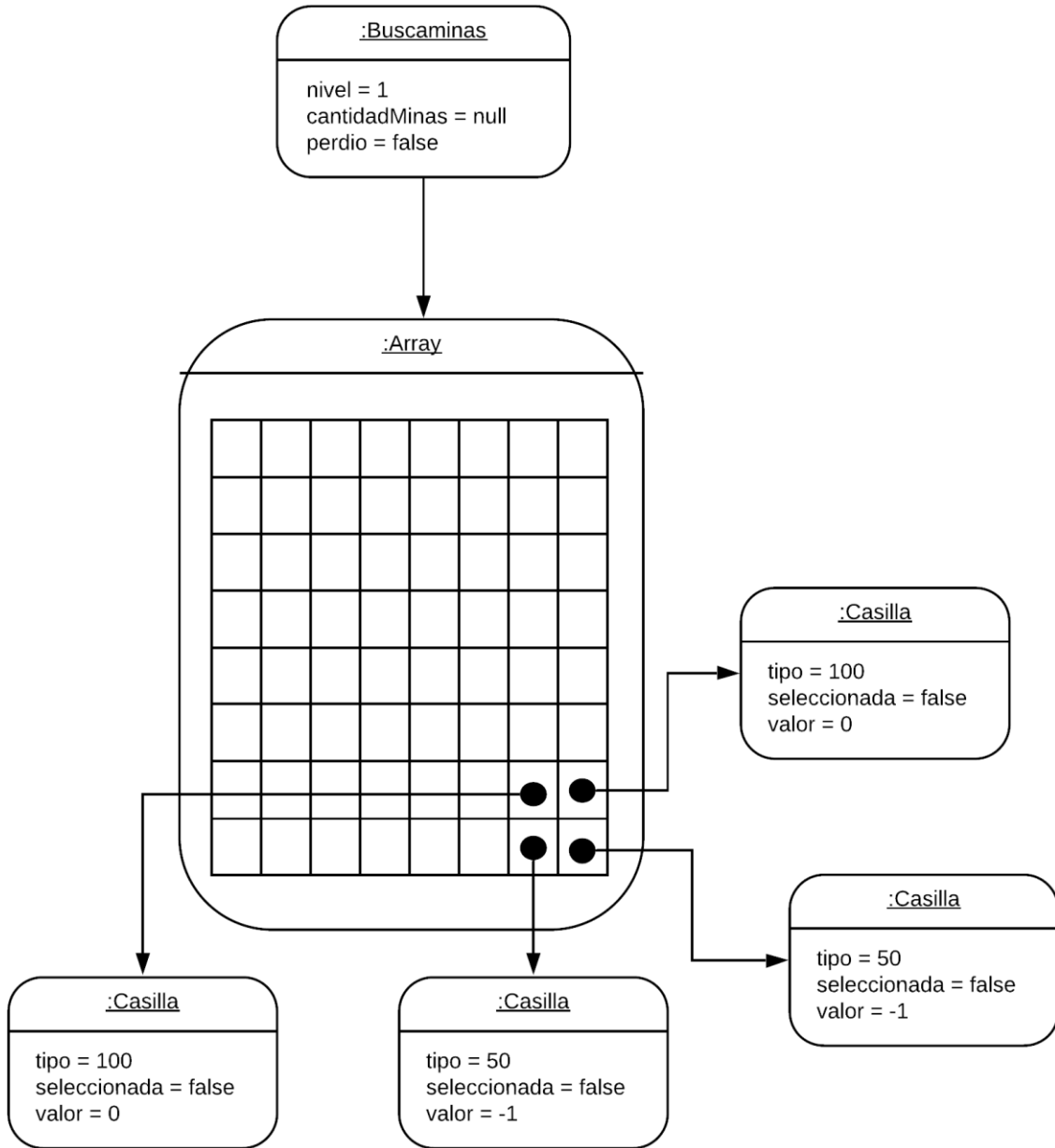
Clase	Método	Escenario	Valores de entrada	Resultado
Buscaminas	<code>columnVerifier (int i);</code>	Escenario 6	i = -1	Falso. El número introducido no hace parte de las filas de la matriz.
Buscaminas	<code>columnVerifier (int i);</code>	Escenario 6	i = 0	Verdadero. El número introducido hace parte de las filas de la matriz.

Buscaminas	<code>columnVerifier(int i);</code>	Escenario 6	i = 98	Falso. El número introducido no hace parte de las filas de la matriz.
------------	-------------------------------------	-------------	--------	---

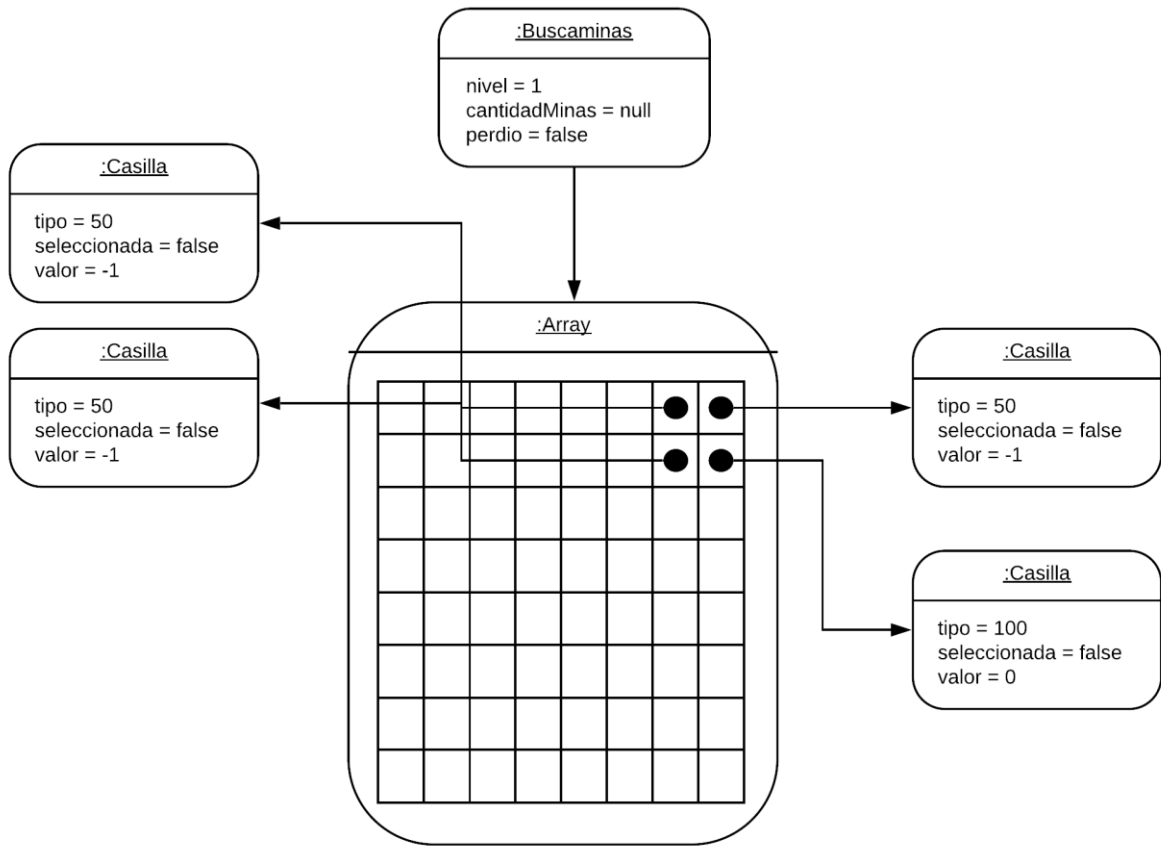
Escenarios para el test de Buscaminas:



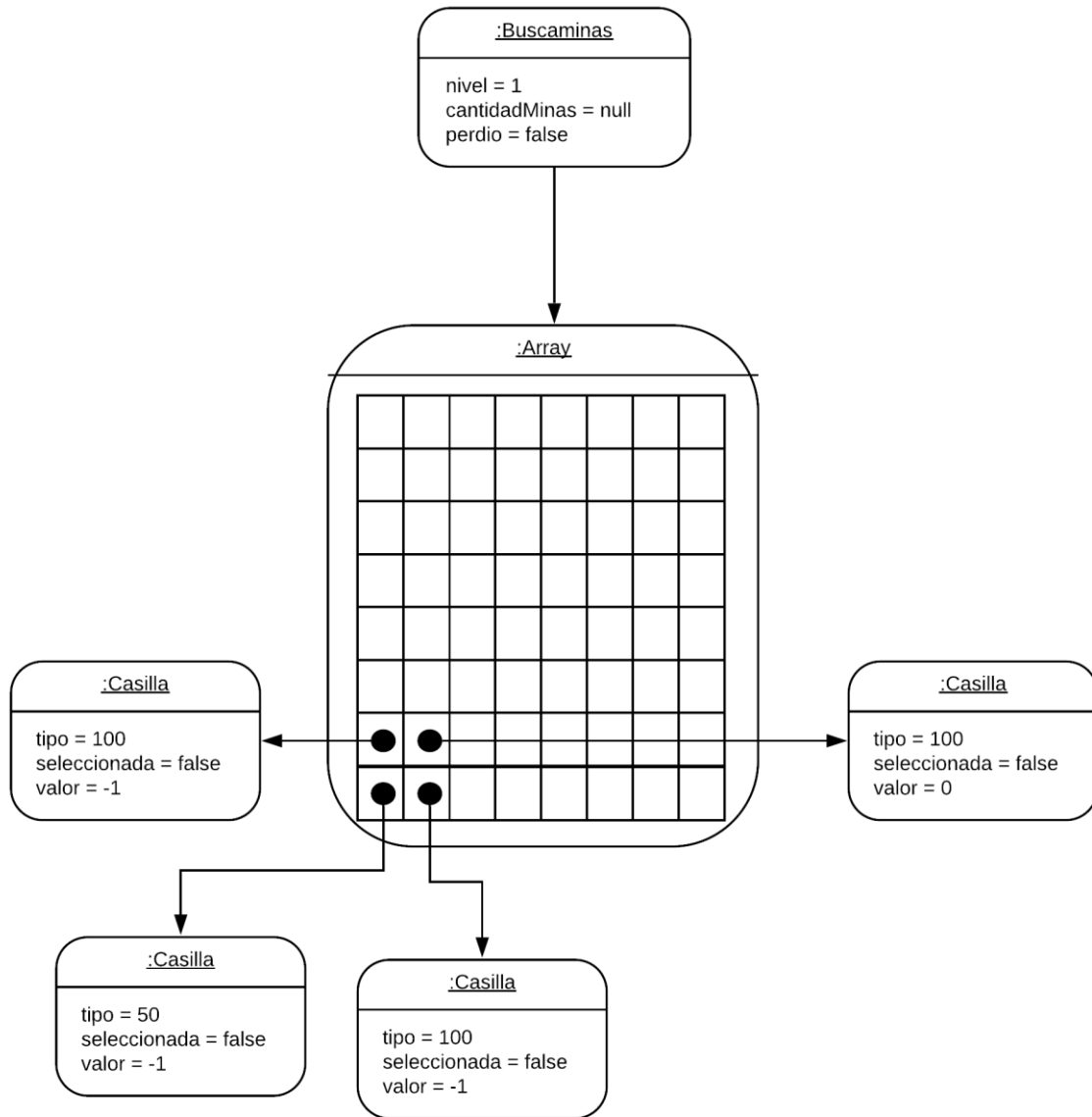
Escenario 2



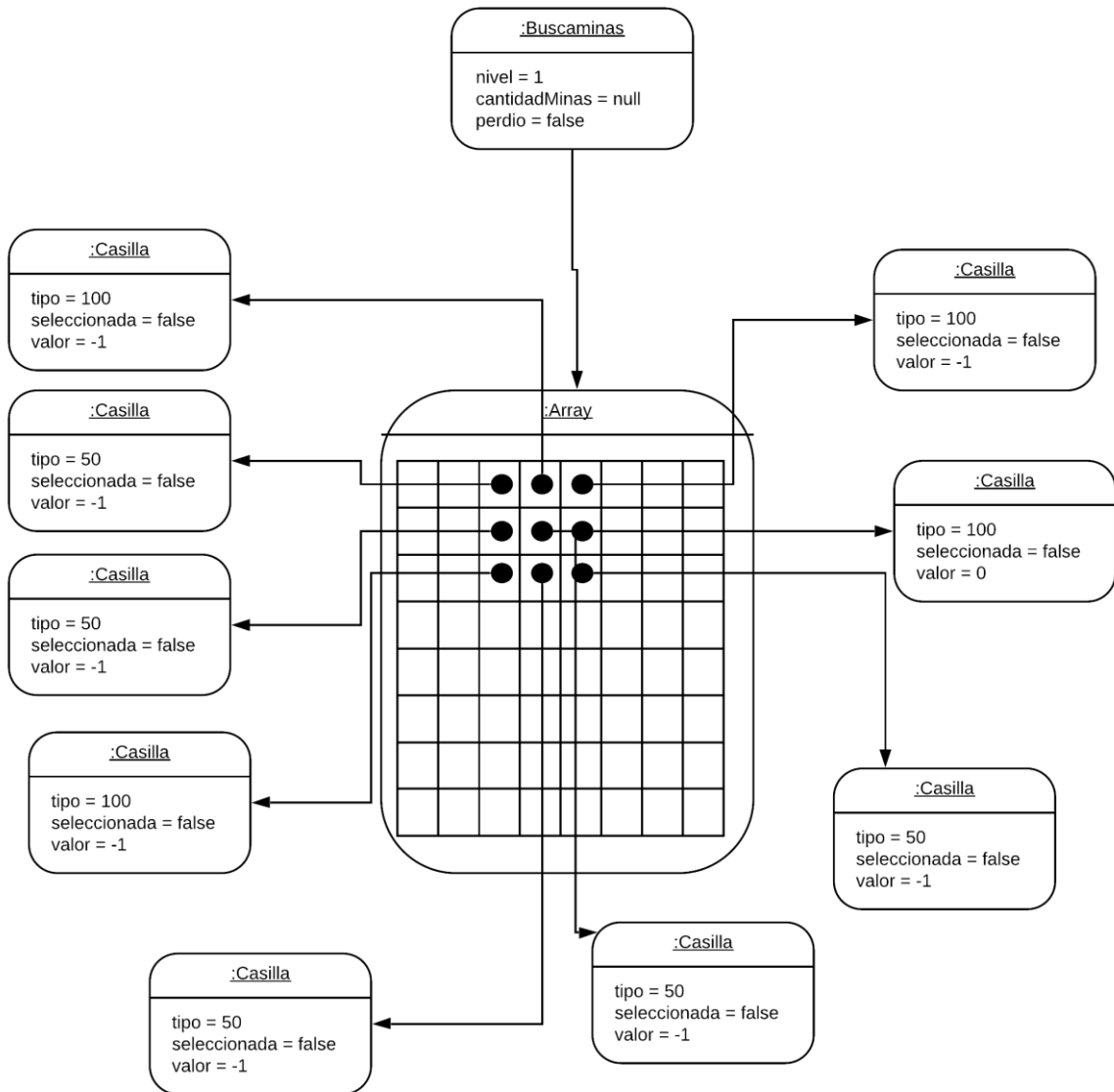
Escenario 3



Escenario 4



Escenario 5



Escenario 6:

Se crea un juego de buscaminas en nivel experto, perdio = false, con su respectivo tablero generado aleatoriamente.

Escenario 7:

Se crea un juego de buscaminas en nivel intermedio, perdio = false, con su respectivo tablero generado aleatoriamente.

Escenario 8:

Se crea un juego de buscaminas en nivel principiante, perdio = false, con su respectivo tablero generado aleatoriamente.

Escenario 9:

Se crea un juego de buscaminas con dificultad principiante. Todo el tablero se llena con casillas libres (con valor = 0 y seleccionada = false) excepto por la casilla en la posición [3][3] que es una casilla de tipo mina, y la casilla [1][1] que tiene su atributo seleccionada = true.

Escenario 10:

Se crea un juego de buscaminas con dificultad principiante. Todo el tablero se llena con casillas libres excepto por la casilla en la posición [1][2] que es una casilla de tipo mina. Todas las casillas tienen el atributo de seleccionada = true.

Escenario 11:

Se crea un juego de buscaminas con dificultad intermedio. Todo el tablero se llena con casillas libres (con valor = 0 y seleccionada = false) excepto por la casilla en la posición [2][3] que es una casilla de tipo mina, y la casilla [0][1] que tiene su atributo seleccionada = true.

Escenario 12:

Se crea un juego de buscaminas con dificultad intermedio. Todo el tablero se llena con casillas libres excepto por la casilla en la posición [8][6] que es una casilla de tipo mina. Todas las casillas tienen el atributo de seleccionada = true.

Escenario 13:

Se crea un juego de buscaminas con dificultad experto. Todo el tablero se llena con casillas libres (con valor = 0 y seleccionada = false) excepto por la casilla en la posición [12][14] que es una casilla de tipo mina, y la casilla [14][28] que tiene su atributo seleccionada = true.

Escenario 14:

Se crea un juego de buscaminas con dificultad experto. Todo el tablero se llena con casillas libres excepto por la casilla en la posición [10][17] que es una casilla de tipo mina. Todas las casillas tienen el atributo de seleccionada = true.

Escenario 15:

Se crea un juego de buscaminas en nivel principiante con un tablero donde la casilla [1][1] es libre y ya ha sido seleccionada; la casilla [2][2] es una mina y no ha sido seleccionada. Las demás casillas son libres y no han sido seleccionadas.

Escenario 16:

Se crea un juego de buscaminas en nivel principiante y se destapan todas las casillas que no son minas.

Escenario 17:

Se crea un juego de buscaminas en nivel principiante y se destapan todas las casillas que no son minas, excepto por una, la casilla [2][3].

Escenario 18:

Se crea un juego de buscaminas en nivel intermedio y se destapan todas las casillas que no son minas.

Escenario 19:

Se crea un juego de buscaminas en nivel intermedio y se destapan todas las casillas que no son minas, excepto por una, la casilla [8][8].

Escenario 20:

Se crea un juego de buscaminas en nivel experto y se destapan todas las casillas que no son minas.

Escenario 21:

Se crea un juego de buscaminas en nivel experto y se destapan todas las casillas que no son minas, excepto por una, la casilla [14][20].

Escenario 22:

Se reutiliza el escenario 1 pero se destapa la casilla [0][1].