



# Proyecto: Buscaminas. Parte 2

## 1 Descripción del juego

Se incorporan ahora dos nuevas características a nuestro buscaminas: descubrimiento de islas de casillas vacías y acciones asociadas a las casillas, según el contenido de estas.

En primer lugar, es necesario crear un nuevo proyecto minesweeper\_sprint2 importando el anterior y realizar los cambios pertinentes sobre este nuevo proyecto.

**Recordad** que, al finalizar el juego, se gane o se pierda, se debe mostrar el tablero completo.

## 2 Introducir el concepto de Action

Llamaremos **acción** al conjunto de sucesos que se desencadena al hacer stepOn() sobre un objeto Square.

Hay **tres tipos** de acciones : **descubrimiento de casillas** adyacentes al hacer stepOn() en una casilla vacía, señalar el **tablero como explotado** al hacer stepOn() en una casilla con mina o **hacer nada**.

En nuestra nueva versión del buscaminas, se asocia una acción a cada casilla. Esa acción se ejecutará cuando se invoque el método stepOn() sobre la casilla, siempre y cuando la casilla no haya sido descubierta con anterioridad.

### 2.1 Nuevos tipos java

A todos y cada uno de los objetos Square se les asociará una única acción que es posible elegir entre tres tipos distintos de acciones. Esta acción se almacenará como un atributo del objeto Square que **debe ser de tipo Action**.

Por ello, será necesario implementar **tres nuevas clases para definir las posibles acciones que se pueden asociar con una casilla**. Estas tres nuevas clases se detallan en el siguiente apartado.

Todas ellas ofrecen un único método público **activate()** pero recuerda que puedes **añadir** tantos **métodos y atributos privados** como consideres necesarios.

**void activate()**

Activa la acción que corresponda.

Para proporcionar a un objeto Square su acción asociada, **añade el siguiente método público a Square**:

**void setAction(Action action)**

Asigna la acción recibida como parámetro al objeto sobre el que se invoca este método.



Además, cada vez que se llame al método `stepOn()` de un objeto `Square`, se actuará como sigue:

- En caso de que el estado de la casilla sea **CLOSED**:
  1. Cambiar **estado a OPEN**.
  2. **Activar la acción** asociada a la casilla, provocando la ejecución de su método `activate()`.

## 2.2 Clases Action

### BlowUpAction

Está pensada para ser asociada a una casilla de mina.

Métodos públicos:

**void** `activate()`

Provoca que el tablero se marque como explotado.

`MineAction(Board board)`

**Constructor.** Recibe un parámetro de tipo **Board** que apunta a la instancia del tablero que contiene la casilla a la que se asocia esta acción.

### ClearAction

Está pensada para ser asociada a una casilla vacía.

Métodos públicos:

**void** `activate();`

Provoca que se ejecute el método `stepOn` de las casillas adyacentes (8-vecinos) a esta casilla.

`ClearerAction(List<Square> neighbouringSquares)`

**Constructor.** Recibe como parámetro una lista de objetos `Square` que se corresponden con aquellas casillas adyacentes a la casilla a la que la acción está asociada. Consideramos casillas adyacentes a una casilla dada a todas aquellas que están a una posición de distancia en cualquier dirección (horizontal, vertical, diagonal).

### NullAction

Está pensada para ser asociada a casillas con pista numérica.

Métodos públicos:

**void** `activate();`

No tiene ningún efecto.

## 2.3 Adaptación de código

Además, las nuevas clases añadidas probablemente provoquen que sea necesario **reescribir o mover parte de la lógica de tu código**. Algunas comprobaciones o acciones, como el momento en que el tablero se marca como explotado, que hasta ahora se disparaban desde el tablero a través de evaluar estados de casillas, ahora se llevarán a cabo dentro de las acciones asociadas a Square.

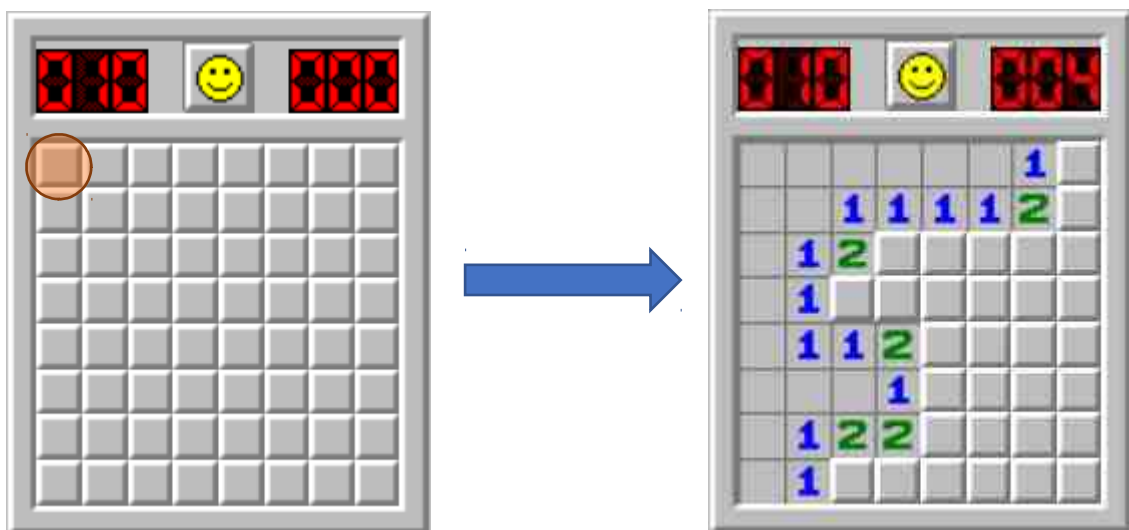
## 3 Descubrimiento de islas de casillas vacías

En el juego del buscaminas original, cuando se descubre una casilla vacía, se descubre el contenido tanto de esa casilla como de todas las casillas adyacentes vacías o con pista numérica.

A su vez, si alguna de estas nuevas casillas descubiertas es vacía, este efecto también se contagia a sus casillas adyacentes.

De esta forma, con un solo click se descubre un grupo de casillas vacías contiguas las unas a las otras y todas las pistas numéricas que las rodean.

Las siguientes imágenes se corresponden con capturas de la implementación online del buscaminas accesible en <http://buscaminas.eu/>. En la primera de muestra una partida en la que aún no se ha descubierto ninguna casilla. Al **hacer** click sobre la casilla (0,0), el tablero descubre posiciones tal y como se muestra en la segunda imagen: una isla de casillas vacías rodeadas de pistas numéricas.



En esta versión de nuestro buscaminas deberemos implementar este comportamiento. Cada vez que el usuario acierte en una casilla vacía se descubrirán casillas hasta que todo el grupo de casillas vacías contiguas y de pistas que les rodeen queden descubiertas.



## 4 Estructura de paquetes

Se debe mantener una estructura de paquetes como la que sigue, además de ceñirse escrupulosamente a los nombres que se indican en los documentos.



## 5 Tests

En caso de modificar la lógica de algunos métodos que hayas implementando en el sprint1, **es muy posible que tengas que replantear sus tests también.**

Además, deberás desarrollar nuevos tests para las siguiente clases y métodos:

### Clase ClearAction

activate()

### Clase BlowUpAction

activate()