

# Análisis PL2: Scala Jewels Legend

## Hero

A continuación, se realiza un análisis de las funciones necesarias para obtener el primer apartado y posteriormente se especifica la asignación de cómo se va a repartir el trabajo:

### Funciones necesarias

**main:** función principal que tiene de entrada los argumentos de la línea de comandos, hace llamadas a otras funciones para iniciar el juego y crea el bucle del juego (gameloop) que el usuario puede parar cuando quiera salir del juego. Devuelve Unit.

**generarTablero:** se le pasa una lista de 7x9 y usa Random para generar números aleatorios entre 1 y 6. Devuelve la lista con los números generados.

**comprobarTablero:** Se le pasa el tablero y comprueba si existe algún movimiento legal. Devuelve un Boolean (True si existe ese movimiento legal y False en caso contrario)

**comprobarMovimiento:** Tiene como entrada el tablero y se comprueba si el movimiento que se quiera realizar es legal (que pueda formar tres diamantes en horizontal). Devuelve un Boolean (True si dicho movimiento es legal y False en caso contrario)

**mover:** De entrada se le pasa el tablero, posición inicial (x, y) y posición final (x,y). Se puede mover un diamante cualquiera (sea legal o no por si no existe la posibilidad de alineamiento de tres diamantes que se determina por comprobarTablero). Devuelve la lista (tablero) con el movimiento ya realizado.

**recolocar:** Se le pasa el tablero, elimina donde haya tres diamantes en horizontal, baja los diamantes y genera nuevos en la parte de arriba del tablero. Esto devuelve la lista (tablero) ya recolocado.

**estrategia:** Se le pasa el tablero y realiza la estrategia del computador que en principio será aleatoria, pero después se hará un trabajo de optimización para que elija una jugada más óptima. Devuelve el tablero con el movimiento de la máquina.

**imprimirTablero:** entra una lista (el tablero) y se imprime por pantalla. Devuelve un Unit.

**Nota:** Las funciones antes expuestas son las principales, pero en el proceso de programación hará falta crear pequeñas funciones que faciliten la programación funcional.

## Repartición del trabajo

Integrante	Tareas
<u>Ana Cortés Cercadillo</u>	Se hará cargo de las funciones de comprobarTablero y comprobarMovimiento. También creará la función mover en la que se usa la función anterior de comprobarMovimiento.
<u>Carlos Javier Hellín Asensio</u>	Realizará las funciones básicas necesarias para no usar funciones de orden superior, programar el main con el generarTablero y el bucle del juego. Por último, imprimir tablero como llamada que se realizará dentro del bucle del juego.
<u>Daniel Ferreiro Rodríguez</u>	Las funciones que realizará son la de recolocar y estrategia. Esta última es la que se implementa en la optimización posterior.