

Práctica 6

Desarrollo dirigido por pruebas (TDD)

Objetivos:

- Aplicar las prácticas propias del desarrollo software dirigido por pruebas, traduciendo los requisitos en casos de prueba que originan el desarrollo de nuevo código y la refactorización del código global resultante.

Descripción del proyecto

El proyecto plantea la implementación de ciertas funcionalidades de un juego de mesa que se desarrolla en un tablero de ajedrez 4x4 sobre el que dos jugadores, B y N, deben ir colocando, de forma alternada, 2 torres del color asignado a cada uno de ellos. Entre las funcionalidades del juego se encuentra la de comprobar, una vez colocadas todas las piezas del juego, si la configuración resultante presenta amenazas entre las torres de los dos jugadores.

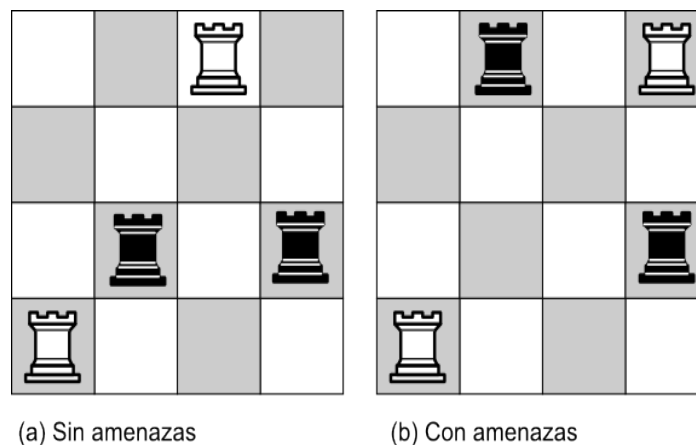


Figura 1 Dos disposiciones del tablero.

Para dar soporte al juego se plantean los siguientes requisitos:

1. **Posicionamiento de piezas:** el sistema debe identificar cuando una posición de una torre resulta válida o inválida (posición dentro del tablero y no ocupada) y colocar la pieza en su caso.
2. **Dos jugadores:** el sistema debe dar soporte a que participen dos jugadores de forma alterna, independientemente de quien comience la partida.
3. **Determinar amenazas:** el sistema debe comprobar si la configuración final del tablero comporta amenazas entre torres de los dos equipos.

En el caso de que el sistema reciba una posición inválida se lanzará una excepción de tipo `RuntimeException`.

Ejercicios

1. Descarga el proyecto que se proporciona en los recursos del Aula Virtual e impórtalo a Eclipse. Cambia el nombre del proyecto a la secuencia: P6-TDD-ApellidosNombre.
2. Considera el requisito del sistema 1 para implementar un método (`jugar(int x, int y)`) de la clase Juego que implemente la funcionalidad de dicho requisito. Realiza los siguientes pasos:
 - a. Identifica las comprobaciones a realizar en un tablero ideal 4x4 sobre la posición de una torre.
 - b. Para cada comprobación se debe generar de forma secuencial (uno a uno) los test que comprueban el correcto funcionamiento del método. Se ejecutará el test, que producirá un fallo, se incorporará la funcionalidad mínima para que el test no falle, se refactorizará el código si es necesario y se repetirá el paso b hasta tener toda la funcionalidad del requisito 1 implementada.
 - c. Implementa el resto de funcionalidad del método aplicando el ciclo TDD.

Importante: Los test deben quedar escritos de forma secuencial tanto en términos de requisito como en términos de incorporación de test y código. Cada vez que se realice la secuencia de procesos codificar-refactorizar se comentará la clase, se realizará una copia del estado anterior de la clase y se incorporará el código asociado al test que falla y la respectiva refactorización (indicar, mediante un comentario, en qué ha consistido la refactorización). Es imprescindible que se pueda observar el desarrollo seguido para implementar la funcionalidad.
3. Realizar los pasos del ejercicio 2, con su correspondiente adaptación, para el requisito 2. La finalidad es implementar un método que indique el jugador al que le corresponde el turno siguiente (método `siguienteJugador()`). Tened presente las indicaciones del apartado “Importante” del ejercicio 2.
4. Realizar los pasos del ejercicio 2, con su correspondiente adaptación, para el requisito 3. Tened presente las indicaciones del apartado “Importante” del ejercicio 2.

MUY IMPORTANTE:

Tened presente que durante los ciclos TDD el proceso de refactorización incorporará modificaciones en el diseño del código (no en la funcionalidad) y que es posible que para dar respuesta a determinados test sea necesario incorporar estructuras de datos que inicialmente no aparecen ya que su necesidad depende de la codificación que soluciona el fallo de un test propuesto.

Por último, recordad que cuando se incorpora un test que falla, el proceso de codificación sólo debe incorporar la menor cantidad de código necesaria para que el test deje de fallar con independencia de que un test posterior pueda provocar un fallo. La idea es solucionar el caso de prueba actual y no pensar en los casos de prueba futuros.

Entregables

La entrega se realizará en la tarea del Aula Virtual asignada a la práctica con fecha de entrega 16/12/2022 23:55 horas y se debe entregar:

Cuando termines se debe subir a la tarea del Aula Virtual el proyecto Eclipse completo. Para ello se debe comprimir la carpeta del proyecto, no el espacio de trabajo Eclipse.