

# Proyecto UT5 Explorador (para hacer en casa y entregar en GitHub)

## Antes de empezar

Este ejercicio es para realizar individualmente.

Crea el proyecto en IntelliJ.

Cada vez que se realice un desarrollo del proyecto, genera un Commit y un push a GitHub comentando dicho desarrollo.

**No olvides poner público el repositorio de GitHub.**

Nombra el repositorio de GitHub como:

*DAM1B\_ExploradorNombre1Apellido1Nombre2Apellido2.*

Se valorará en la corrección que el programa esté probado (compila y ejecuta bien) y que esté claramente escrito y organizado (se respetan las reglas de estilo del lenguaje Java, nombres descriptivos, código no duplicado, ...)

**La fecha tope de entrega es el Miércoles 8 enero hasta las 15:00h.**

**Se deberá entregar en el moodle el enlace al repositorio de Github.**

Se anulará automáticamente la corrección del ejercicio y se **evaluará con un 0** si se detecta que ha sido copiado o dejado copiar a algún compañero/a

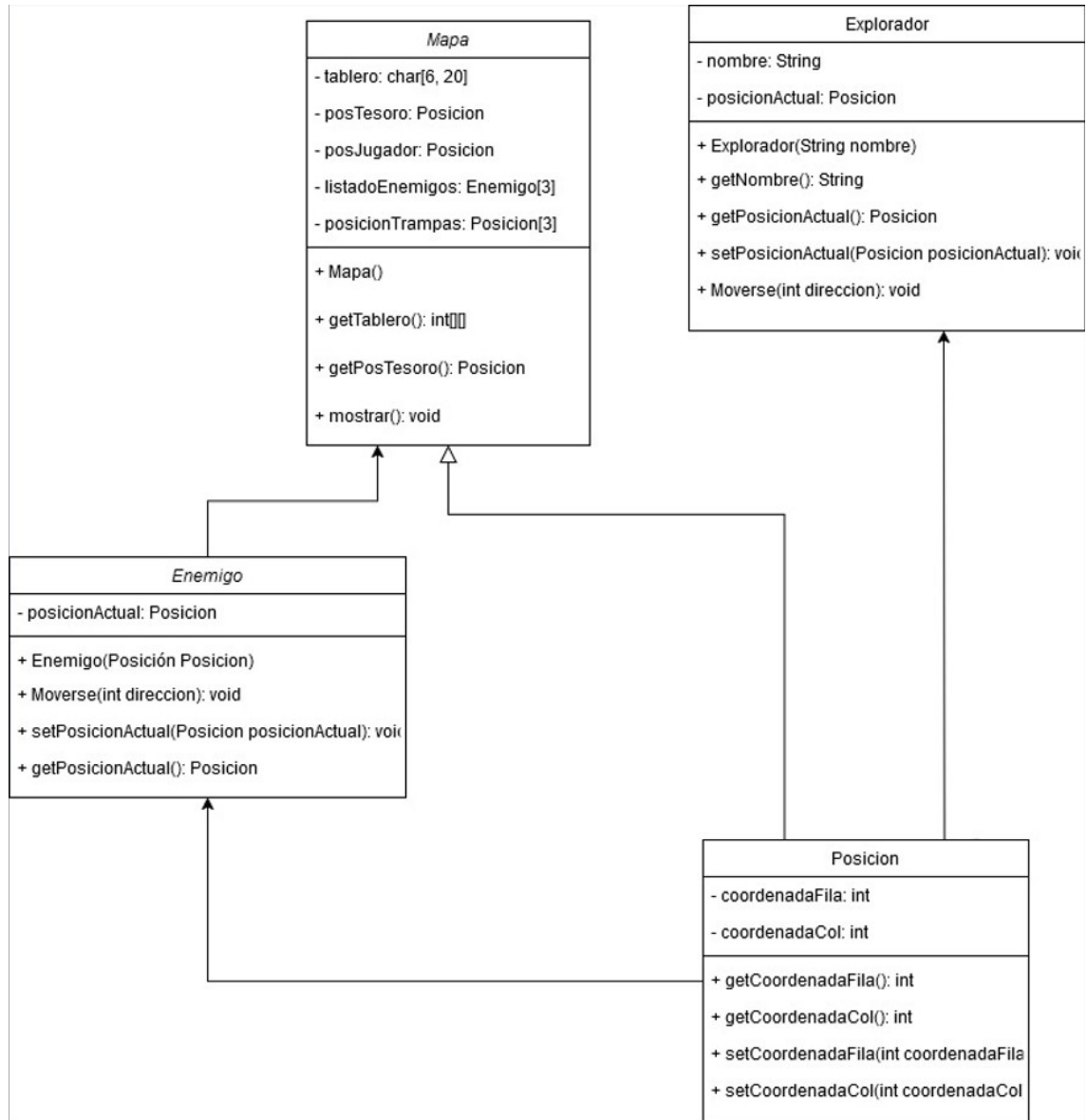
Se penalizará si no se siguen las normas de entrega del ejercicio :

- *Los commit no tienen comentarios.*
- *Se han realizado pocos commit, no siendo posible observar una evolución del trabajo.*
- *hay algún commit posterior a esta fecha de entrega*

El profesorado podrá convocar al alumno/a para defender oralmente el proyecto

## Especificaciones

Crea un proyecto en IntelliJ llamado *DAM1B\_ExploradorNombreApellido* que contenga la siguiente estructura:



## Clase Explorador

**Constructor:** Tendrá como parámetro de entrada un string, que será el nombre del explorador. Y en el mismo se asignará la posición actual del explorador de tal forma que estará en la primera columna, y la fila se obtendrá de manera aleatoria.


**Moverse:** Se modificará la posición actual del explorador. El parámetro de entrada será un valor entero entre 1 y 4 (utilizar constantes *ARRIBA*, *ABAJO*, *DERECHA*, *IZQUIERDA*). Se deberá tener en cuenta que no se puede mover fuera de los límites del mapa.

Explorador	
- nombre: String	
- posicionActual: Posicion	
+ Explorador(String nombre)	
+ getNombre(): String	
+ getPosicionActual(): Posicion	
+ setPosicionActual(Posicion posicionActual): void	
+ Moverse(int direccion): void	

## Clase Enemigo

**Constructor:** Tendrá como parámetro de entrada un string, que será el nombre del explorador. Y en el mismo se asignará la posición actual del explorador de tal forma que estará en la primera columna, y la fila se obtendrá de manera aleatoria.

**Moverse:** Se modificará la posición actual del enemigo El parámetro de entrada será un valor aleatorio entero entre 1 y 4 (utilizar constantes *ARRIBA*, *ABAJO*, *DERECHA*, *IZQUIERDA*). Se deberá tener en cuenta que no se puede mover fuera de los límites del mapa y a posiciones donde se encuentre otro enemigo o una trampa.

	<i>Enemigo</i>
- posicionActual: Posicion	
+ Enemigo(Posición Posicion)	
+ Moverse(int direccion): void	
+ setPosicionActual(Posicion posicionActual): void	
+ getPosicionActual(): Posicion	

## Clase Mapa

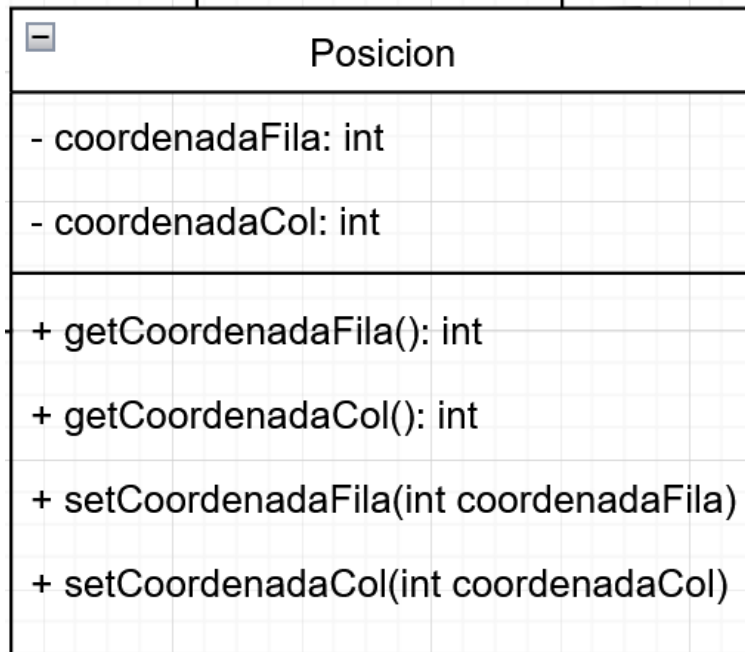
Constructor: En el constructor se generará el tablero de la clase de tamaño 6 x 20 siendo chars. Cada posición del tablero del mapa podrá tener uno de los siguientes valores:

- **J** – Representa al jugador.
- **E** – Representa a cada uno de los 3 enemigos.
- **T** – Representa a cada una de las 3 trampas.
- **' '** - Representa un espacio donde no hay nada. Puede estar oculto el tesoro.

Mapa
<ul style="list-style-type: none"><li>- tablero: char[6, 20]</li><li>- posTesoro: Posicion</li><li>- posJugador: Posicion</li><li>- listadoEnemigos: Enemigo[3]</li><li>- posicionTrampas: Posicion[3]</li></ul>
<ul style="list-style-type: none"><li>+ Mapa()</li><li>+ getTablero(): int[][]</li><li>+ getPosTesoro(): Posicion</li><li>+ mostrar(): void</li></ul>

## **Clase Posicion**

Esta clase se proporcionará completa al alumno, por lo que no se deberá modificarla ni completar nada. Sólo se deberá revisarla para que se comprenda.



## Clase Main

\*\*\*\*\*

*Bienvenido al juego del explorador*

\*\*\*\*\*

```
-----
| E | | | | | | | | | | | | | | | | | P | | | |
-----
| | | | | J | | E | | | | | | | | | | | | |
-----
| | | | | | | | | | | | | | | | | | | |
-----
| | | | | | | P | | | | | | | | | P | | | |
-----
| | | | | | | | | | | | | | | | | | | |
-----
| | E | | | | | | | | | | | | | | | | |
-----
```

W=Arriba    A=Izquierda    S=Abajo    D=Derecha

¿Que accion quieres realizar? A

Se vuelve a mostrar el menú inicial con el mapa actualizado.

Durante el turno de movimiento el jugador moverá primero, no podrá moverse donde ya hay un enemigo ni fuera del mapa. Si cae en una trampa morirá y se acaba el juego. Una vez finalizado en movimiento del jugador le tocan a los 3 enemigos. Lo harán por orden de posición en el array que ocupan. La dirección de movimiento será aleatoria y moverán una posición. Un enemigo no podrá moverse a una posición ocupada por un enemigo ni por una trampa. Las 3 trampas son estáticas, no se pueden mover. Se posicionarán al inicio del juego en posiciones aleatorias con una única regla, la distancia entre coordenadas X y coordenadas Y de las trampas será de 3.

El tesoro estará oculto en una casilla vacía.

El juego finalizará cuando:

- 1- El explorador encuentra el tesoro posicionándose en una casilla ocupada por el tesoro, gana el juego.
- 2- El explorador se topa con un enemigo o cae en una trampa, entonces muere y pierde el juego.

Se deberá crear un cuarto enemigo que sea inteligente, es decir, cuando este enemigo se mueva lo hará donde está el explorador persiguiéndolo. Para moverse deberá cotejar primero una coordenada ( $x$  o  $y$ ) y luego la otra coordenada, si ya se ha movido en la primera coordenada cotejada no lo hará en la siguiente. El orden de prioridad de las coordenadas se dejará a elección del alumno. Este enemigo tendrá las mismas limitaciones que el resto, no podrá moverse a la posición ocupada por una trampa u otro enemigo. Este enemigo se representará mediante un asterisco (\*) de color rojo. Para la creación del enemigo se creará una nueva clase llamada *EnemigoInteligente*.