

ACTIVIDAD 6.18 (hasta 10 puntos) ***TRES EN RAYA***

Actividad 6.18 – Tres en Raya (Ampliación hasta 10 puntos)

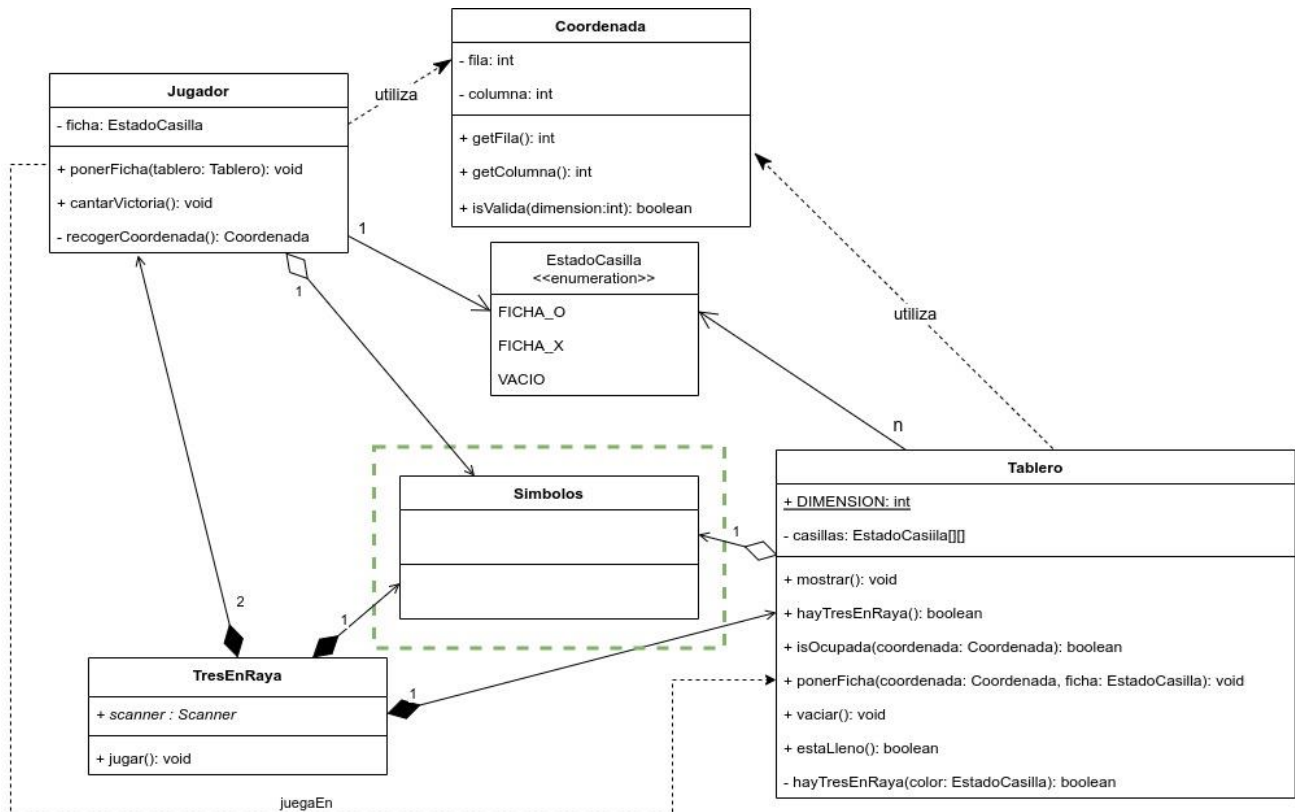
Enunciado

❗ Para desarrollar esta nueva funcionalidad, deberás crear **una nueva rama de desarrollo en tu sistema de control versiones (git)** y NO FUSIONAR (merge) con tu rama principal una vez acabada la funcionalidad con la que actualmente tienes para la versión inicial del juego. Por supuesto, deberá estar esta rama accesible desde tu repositorio remoto.

Vamos a añadir una nueva funcionalidad al juego del **tres en raya**. En este caso queremos que el usuario pueda seleccionar la pareja de símbolos que representarán a la ficha X y O del juego original (dando a elegir entre diferentes posibilidades de parejas de símbolos) con las que jugar.

Para ello vamos a **añadir una nueva clase que se encargará de gestionar los packs (parejas) de símbolos** seleccionados para jugar, es decir, en el juego se podrá seleccionar previamente si se quiere jugar con los símbolos (X, O), o bien con (\$, €) o bien con (!, ?), por poner varios ejemplos.

El sistema se verá ampliado con una nueva clase `Simbolos` tal y como se muestra en el siguiente diagrama:



La nueva **clase Simbolos** contendrá los **packs** (lotes de parejas de símbolos de fichas que representarán a la **ficha X** y a la **ficha O** en el juego) y que el usuario podrá seleccionar. Para ello, utilizará una matriz de **N X 2**, siendo **N** el número de **parejas que existen** y **2** cada uno de los símbolos que representan a los dos tipos de fichas (es decir, la X y la O). La siguiente tabla representaría un ejemplo de la lista de packs:

	FICHA_X	FICHA_O
pack1 (fila 1)	X	O
pack2 (fila 2)	\$	€
pack3 (fila 3)

En el **momento en el que se instancie la clase**, se seleccionará una de las parejas como pareja por defecto (utilizaremos un atributo **packSeleccionado** de tipo **entero**). Esta pareja podrá ser cambiada a posteriori por el usuario a través de un menú de opciones. Si se selecciona una opción no existente se mostrará un mensaje de error. Además, se ofrecerá

la opción de obtener aleatoriamente la pareja de iconos con la que jugar. (ver ejemplo de ejecución).

Cuando se termina cada partida, si el usuario decide jugar una nueva, **se deberá de seleccionar de nuevo la pareja con la que se desea jugar la nueva partida**, A continuación, se describe una serie de métodos que debería tener la clase **Simbolos** :

`public void seleccionar()`: Mostrará los packs de que dispone el usuario para elegir y pedirá que el usuario seleccione una opción. Una vez seleccionada, deberá guardar en un atributo de instancia la opción que ha seleccionado el usuario, dando la opción de seleccionar de manera aleatoria cualquier pareja de símbolos preestablecidos (cualquiera de las opciones anteriores).

`public String obtenerSimbolo(EstadoCasilla estadoCasilla)`: A partir del tipo de casilla que recibe, devolverá el símbolo correspondiente según el pack de iconos elegido por el usuario.

❗ Si observas con detenimiento el diagrama de clases, verás que, para poder **utilizar el pack de iconos seleccionado por el usuario**, las clases **Jugador** y **Tablero** recibirán un objeto de tipo **Simbolos** en el constructor para posteriormente utilizarlo ambos como atributo.

❗ **No se evaluará NADA acerca de esta ampliación** si no se ha realizado anteriormente la versión anterior del programa en su totalidad.

❗ **Sólo** debes cambiar la forma en que **se visualizan** las **casillas** del tablero y las **fichas** con las que juega un jugador (para ello, deberás llamar al método `obtenerSimbolo` creado en la clase **Simbolos**, en lugar de llamar al `toString` definido para cada opción del enum). La forma de gestionarlas en el juego seguirá siendo a través del **enumerado** `EstadoCasilla`

A continuación, se presenta un ejemplo de la utilización conjunta de la clase **Simbolos**.

```
// Creamos el objeto de tipo Icono
Simbolos packDeSimbolos = new Simbolos();

// Preguntamos al usuario para que seleccione el pack de Iconos
packDeSimbolos.seleccionar();

// Pasamos al usuario en el constructor el gestor de iconos

Jugador jugador1 = new Jugador(EstadoCasilla.FICHA_X, packDeSimbolos);

// Mostramos el icono correspondiente a la ficha que juega el usuario
System.out.println("Tira el jugador con " +
packDeSimbolos.obtenerSimbolo(EstadoCasilla.FICHA_X));
```

Ejemplo parcial de ejecución

Vamos a jugar al "Tres en Raya"

Vamos a seleccionar los iconos con los que jugar

1) X 0

2) € \$

3) ? !

4) Aleatorio

Selecciona una opción [1-4]:6

Debe introducir un valor dentro del rango establecido

Selecciona una opción [1-4]:hola

Debe introducir un entero

Selecciona una opción [1-4]:2

```
| |1|2|3|
|1| | | |
|2| | | |
|3| | | |
```

Jugador con \$

Introduce fila [1-3]: 1

Introduce columna [1-3]: 1

```
| |1|2|3|
|1|$| | |
|2| | | |
|3| | | |
```

Jugador con €

Introduce fila [1-3]:

.....