

# Reto 1: “Tres en raya”

---

DENTRO DEL PROYECTO INTEGRADOR DEL MÓDULO DE  
PROGRAMACIÓN DE JUEGOS REUNIDOS



**Unidades Didácticas: UD 1,2,3,4**

**PRUEBA DE EVALUACIÓN: TIPO B**

**AGRUPAMIENTO: INDIVIDUAL**

**OBJETIVOS DIDÁCTICOS:**

1. ***Realizar un programa de complejidad media que use estructuras de control: condicionales y bucles*** (asociado a RA3 – CE3(a,b,c,e,f,g,i)).
2. ***Definir y llamar a métodos estáticos*** (RA2 – CE2(e,f)).
3. ***Hacer uso de estructuras de datos: arrays o vectores*** (RA6 – CE6(a)).
4. ***Usar adecuadamente variables, expresiones, operadores del lenguaje, así como conversión de tipos y comentarios*** (RA1 – CE1(todos)).

# Proyecto integrador del módulo de Programación: "Juegos Reunidos"

A lo largo del curso vas a desarrollar una serie de juegos de escritorio multiplataforma empleando el lenguaje de programación Java.

Cada uno de los juegos o modificaciones de juegos ya realizados se plantearán en forma de retos.



## Reto 1: "Tres en raya" (Individual)

### Desarrollo de una partida de ejemplo

1. Jugar una partida
2. Mostrar estadísticas
3. Salir

Seleccione una opción:

1

```
-----  
| 1 | 2 | 3 |  
-----  
| 4 | 5 | 6 |  
-----  
| 7 | 8 | 9 |  
-----
```

Introduzca movimiento (Jugador 1):

1

```
-----  
| X |  |  |  
-----  
|  |  |  |  
-----  
|  |  |  |  
-----
```

Introduzca movimiento (Jugador 2):

2

```
-----  
| X | 0 |  |  
-----  
|  |  |  |  
-----  
|  |  |  |  
-----
```

-----  
Introduzca movimiento (Jugador 1):

4

-----  
| X | O | |  
-----  
| X | | |  
-----  
| | | |  
-----

Introduzca movimiento (Jugador 2):

3

-----  
| X | O | O |  
-----  
| X | | |  
-----  
| | | |  
-----

Introduzca movimiento (Jugador 1):

7

-----  
| X | O | O |  
-----  
| X | | |  
-----  
| X | | |  
-----

Ha Ganado el Jugador 1!

1. Jugar una partida
2. Mostrar estadísticas
3. Salir

### Instrucciones para la programación del juego

**Variables globales**, solo podrás usar 2:

```
static Scanner in = new Scanner(System.in)
```

```
static char[] tablero = new char[9] : solo  
puede tener 3 valores: ' '(vacía, valor  
inicial), 'X' (ficha jugador 1), 'O' (ficha del  
jugador 2).
```

**Funciones o métodos estáticos obligatorios mínimos:**

***public static void mueveJugador1(int pos):*** el jugador 1 pone una ficha en una posición válida (entre 0 y 8, además está vacía)

***public static void mueveJugador2(int pos)***

***public static boolean movimientoValido(int pos):*** se comprueba que el movimiento es válido antes de poner la ficha (entre 0 y 8, además está vacía).

***public static void iniciar():*** pone el tablero vacío (0).

***public static boolean quedanCasillas():*** comprueba si ya están ocupadas todas las posiciones del tablero.

***public static void dibujaTablero():*** dibuja el tablero tal y como aparece en el ejemplo de partida.

***public static boolean ganaJugador1():*** comprueba si el jugador 1 tiene tres fichas en raya, ganando la partida.

***public static boolean ganaJugador2()***

## Criterios de calificación (Rúbrica)

### 1. Funcionalidad (5 puntos)

INDICADORES	Se cumple y ajusta al enunciado	Se cumple con algún fallo leve	No se cumple o no se ajusta al enunciado	OBJETIVOS DIDÁCTICOS
<b>Juego funcional (1,5 puntos):</b> El juego funciona correctamente, permitiendo a los jugadores realizar movimientos alternativos y detectando el ganador alineando tres fichas en raya.	1,5	0,75	0	OD1,2,3,4
<b>Movimientos válidos y manejo de errores (1,5 puntos):</b> Los movimientos de los jugadores son válidos y se comprueba antes de poner una ficha si la posición está vacía y dentro del rango de 1-9. Se controlan los errores y aparecen mensajes de error correspondientes.	1,5	0,75	0	OD1,2,3,4
<b>Finalización de la partida (0,25 puntos):</b> El juego permite finalizar una partida de manera adecuada y muestra el ganador o empate cuando sea necesario.	0,25	-	-2*	OD1,4
<b>Reinicio del juego (0,25 puntos):</b> Después de una partida, el juego permite a los jugadores jugar nuevamente sin problemas.	0,25	-	-2	OD1,4
<b>Visualización del tablero (1 punto):</b> El tablero se muestra de manera adecuada, tal como se ejemplificó en la partida de muestra.	1	0,5	0	OD1,2,3,4
<b>Estadísticas (0,5 puntos):</b> El juego permite a los jugadores ver estadísticas, como la cantidad de partidas ganadas por cada jugador.	0,5	-	0	OD1,4

(\*) la calificación final del reto, no será en ningún caso negativa. La calificación mínima es de 0 puntos.

## 2.Código, estructura y comentarios (4 puntos)

INDICADORES	Se cumple y ajusta al enunciado	Se cumple con algún fallo leve	No se cumple o no se ajusta al enunciado	OBJETIVOS DIDÁCTICOS
<b>Variables globales (0,5 puntos):</b> Se han utilizado las variables globales in y tablero correctamente, y no se han utilizado variables adicionales innecesarias.	0,5	-	-2	OD4
<b>Funciones estáticas obligatorias (1,5 puntos):</b> Se han implementado las funciones estáticas requeridas con la cabecera especificada,	1,5	0,75	-2	OD2
<b>Estructura del código (1 punto):</b> El código está organizado y comentado de manera clara y legible. Sigue buenas prácticas de programación y es fácil de entender.	1	0,5	0	OD1
<b>Array unidimensional de enteros (1,5 puntos):</b> Se usa un array unidimensional de enteros para almacenar todo el desarrollo de la partida.	1	-	-2	OD3

## 3.Mejoras (1 punto):

- Se incorpora la funcionalidad de recrear última partida, para ello se ha de usar un array bidimensional de enteros (9x9) en el que se va guardando el estado del tablero para cada jugada.