

Instruccions per implementar el Tres en Ratlla de manera modular en Java

Aquest projecte modularitza el joc en diferents classes per fer-lo més **organitzat i mantenible**. Segueix aquests passos per implementar-lo.



Estructura del projecte

El projecte tindrà **tres classes** principals:

- 1 **Tauler** – Gestiona el tauler de joc.
- 2 **Joc** – Controla el flux del joc.
- 3 **Main** – Punt d'entrada del programa.

Crea **tres fitxers Java** amb aquests noms en el mateix directori o projecte.



Classe `Tauler` (Gestió del tauler)

Aquesta classe s'encarrega de:

- ☒ **Inicialitzar el tauler** amb una graella buida.
- ☒ **Mostrar el tauler** per pantalla.
- ☐ **Permetre posar una jugada** en una casella lliure.
- ☐ **Comprovar si hi ha un guanyador**.
- ☐ **Comprovar si el tauler està ple** (per detectar un empat).



Passos a seguir:

1. Declara una **matriu de 3x3** per representar el tauler.
 2. Crea un **mètode per inicialitzar** el tauler amb valors buits (-).
 3. Implementa un **mètode per mostrar** el tauler per pantalla.
 4. Afegeix un **mètode per marcar una casella** amb la jugada d'un jugador (x o o), només si la posició està lliure.
 5. Crea una funció que **comprovi si un jugador ha guanyat** (revisant files, columnes i diagonals).
 6. Implementa una funció que **comprovi si el tauler està ple** per determinar si hi ha un empat.
-

◆ Classe `Joc` (Control del flux del joc)

Aquesta classe s'encarrega de:

- ✓ **Controlar els torns** dels jugadors (x i o).
- **Gestionar la interacció amb l'usuari** per demanar les jugades.
- **Alternar els torns** entre els dos jugadors.
- **Finalitzar el joc quan hi ha un guanyador o empat.**

◆ Passos a seguir:

1. Declara una instància de la classe `Tauler`.
 2. Declara una variable per saber **quin jugador està jugant** (x o o).
 3. Implementa un **bucle principal** que faci el següent:
 - Mostrar el tauler.
 - Demanar a l'usuari una fila i una columna.
 - Intentar posar la jugada al tauler.
 - Comprovar si el jugador actual ha guanyat.
 - Si ningú ha guanyat, canviar el torn al següent jugador.
 4. Si hi ha un guanyador, mostrar un missatge i acabar el joc.
 5. Si el tauler està ple, mostrar un missatge d'empat i acabar el joc.
-

◆ Classe `Main` (Punt d'entrada del programa)

Aquesta classe només:

- ✓ **Crea una instància de `Joc`.**
- **Inicia el joc** cridant el mètode corresponent.

◆ Passos a seguir:

1. Dins del mètode `main`, crea una instància de `Joc`.
 2. Crida el mètode per iniciar la partida.
-

🔗 Extres i millores opcionals

- ◆ Afegir una **IA** perquè l'ordinador jugui contra l'usuari.
 - ◆ Implementar un **menú** per jugar diverses partides seguides.
 - ◆ Permetre que el jugador **esculli si vol ser x o o**.
-

🚩 Amb aquestes instruccions, podràs implementar un Tres en Ratlla modular en Java!

```
Benvingut al tres en ratlla!!  
-----  
- - -  
- - -  
- - -  
-----  
Jugador X escull una fila
```

```
Jugador X escull una fila  
1  
Jugador X escull una columna  
1  
-----  
- - -  
- X -  
- - -
```

```
Jugador 0 escull una fila  
2  
Jugador 0 escull una columna  
5  
Introdueix una posició dins del tauler si us plau  
Jugador 0 escull una columna  
|
```

[Link Github Tauler Activitat](#)

[Link Joc Activitat](#)

Instrucciones para implementar el Tres en Raya de forma modular en Java

Este proyecto modulariza el juego en diferentes clases para hacerlo más **organizado y mantenible**. Sigue estos pasos para implementarlo.



Estructura del proyecto






El proyecto tendrá **tres clases** principales:

- 1 **Tablero** – Gestiona el tablero del juego.
- 2 **Juego** – Controla el flujo del juego.
- 3 **Main** – Punto de entrada del programa.

Crea **tres archivos Java** con estos nombres en el mismo directorio o proyecto.

◆ Clase `Tablero` (Gestión del tablero)

Esta clase se encarga de:

-  **Inicializar el tablero** con una cuadrícula vacía.
-  **Mostrar el tablero** en pantalla.
-  **Permitir colocar una jugada** en una casilla libre.
-  **Comprobar si hay un ganador**.
-  **Comprobar si el tablero está lleno** (para detectar un empate).

◆ Pasos a seguir:

1. Declara una **matriz de 3x3** para representar el tablero.
 2. Crea un **método para inicializar** el tablero con valores vacíos (-).
 3. Implementa un **método para mostrar** el tablero en pantalla.
 4. Añade un **método para marcar una casilla** con la jugada de un jugador (x o o), solo si la posición está libre.
 5. Crea una función que **compruebe si un jugador ha ganado** (revisando filas, columnas y diagonales).
 6. Implementa una función que **compruebe si el tablero está lleno** para determinar si hay un empate.
-

◆ Clase `Juego` (Control del flujo del juego)

Esta clase se encarga de:

- **Controlar los turnos** de los jugadores (x y o).
- **Gestionar la interacción con el usuario** para pedir las jugadas.
- **Alternar los turnos** entre los dos jugadores.
- **Finalizar el juego cuando hay un ganador o empate.**

◆ **Pasos a seguir:**

1. Declara una instancia de la clase `Tablero`.
 2. Declara una variable para saber **qué jugador está jugando** (x o o).
 3. Implementa un **bucle principal** que haga lo siguiente:
 - Mostrar el tablero.
 - Pedir al usuario una fila y una columna.
 - Intentar colocar la jugada en el tablero.
 - Comprobar si el jugador actual ha ganado.
 - Si nadie ha ganado, cambiar el turno al siguiente jugador.
 4. Si hay un ganador, mostrar un mensaje y terminar el juego.
 5. Si el tablero está lleno, mostrar un mensaje de empate y terminar el juego.
-

◆ **Clase `Main` (Punto de entrada del programa)**

Esta clase solo:

- ✓ **Crea una instancia de `Juego`.**
- **Inicia el juego** llamando al método correspondiente.

◆ **Pasos a seguir:**

1. Dentro del método `main`, crea una instancia de `Juego`.
 2. Llama al método para iniciar la partida.
-

🔗 Extras y mejoras opcionales

- ◆ Agregar una **IA** para que el ordenador juegue contra el usuario.
 - ◆ Implementar un **menú** para jugar varias partidas seguidas.
 - ◆ Permitir que el jugador **elija si quiere ser x o o**.
-

🚀 Con estas instrucciones, podrás implementar un **Tres en Raya modular en Java**.