

Actividad Principal

“3 en raya multijugador”







Módulo: Programación de Servicios y Procesos

2.º DAM

 Desarrolla un **servidor concurrente multijugador** que permita gestionar partidas de 3 en raya de forma simultánea usando procesos e hilos.

¿Qué debe hacer tu aplicación?

Tu programa debe:

-  Actuar como **servidor TCP** que acepta múltiples conexiones
-  Gestionar **varias partidas en paralelo**
-  Ejecutar **cada partida** como un **proceso o hilo independiente**
-  Asignar jugadores automáticamente según disponibilidad
-  Sincronizar los turnos y comunicar los movimientos entre jugadores
-  Mantener el estado del tablero correctamente durante toda la sesión

Detalles técnicos

- Los jugadores actúan como **clientes TCP** que se conectan al servidor
- El servidor:
 - Detecta nuevas conexiones
 - Empareja jugadores
 - Crea procesos/hilos para cada partida
 - Controla la comunicación y los turnos
- Las partidas se gestionan de forma **aislada y concurrente**

Conceptos que se trabajan

- Procesos e hilos (threads)
- Sincronización de tareas concurrentes
- Comunicación entre procesos (IPC) mediante **sockets TCP**
- Planificación y gestión de recursos compartidos
- Arquitectura cliente-servidor


Opcionales y ampliaciones

Si completas la funcionalidad principal puedes añadir:

- Interfaz gráfica (Swing, JavaFX, Tkinter...)
- Almacenamiento de partidas en fichero o base de datos
- Emparejamiento de jugadores por nombre/sala
- Registro de logs y estadísticas
- Sistema de turnos por tiempo o desconexiones

Entregables

- Código fuente completo y funcional
- Capturas o vídeo de ejecución
- Diagrama breve de arquitectura (cliente-servidor)
- Documentación del diseño (opcional)
- Explicación de cualquier mejora opcional añadida

 ¡Demuestra que puedes construir un servidor concurrente real con procesos coordinados y comunicación remota!