

Programación de servicios y procesos - 2324

[Área personal](#) ▶ [Mis cursos](#) ▶ [FORMACIÓN PROFESIONAL 23/24](#) ▶ [CIFP CARLOS III](#) ▶
[CGS Desarrollo de Aplicaciones Multiplataforma 23/24](#) ▶ [Segundo\(DAM\)](#) ▶
[Programación de servicios y procesos - 2324](#) ▶ [1ª CONVOCATORIA](#) ▶
[Tarea para PSP04. \(hasta el 07/12/23 23:55\)](#)

Tarea para PSP04. (hasta el 07/12/23 23:55)

Tarea para PSP04.

Detalles de la tarea de esta unidad.

Enunciado.

No utilices un jdk superior al 17. Recuerda que debes entregar el proyecto que pueda abrirse en NetBeans correctamente. No es necesario la generación de ningún .jar.

Actividad 1 (3 puntos)

Modifica el servidor TCP que entregaste en la unidad 3 para que el servidor permita trabajar de forma concurrente con varios clientes.

En los mensajes que imprimes en el servidor debe poder distinguirse con qué cliente está jugando.

No hay que implementar ningún mecanismo de sincronización y/o comunicación entre hilos.

Para esta entrega debes tener ambos proyectos hechos, aunque solamente se va evaluar y probar que el servidor sea concurrente.

Debes incluir en la entrega también el cliente aunque no modifiques nada de él.

Ejemplo de ejecución de un servidor con varios clientes a la vez

```

Soy el servidor TCP esperando a que alguien se conecte
Cliente1: ¡Se ha conectado un nuevo jugador!
Soy el servidor TCP esperando a que alguien se conecte
Cliente1: Fila recibida 1 Columna recibida 1
Cliente1: Después de la jugada del cliente
- - -
- x -
- - -
Cliente1: Después de la jugada del servidor
- - -
- x -
o - -
Cliente2: ¡Se ha conectado un nuevo jugador!
Soy el servidor TCP esperando a que alguien se conecte
Cliente2: Fila recibida 0 Columna recibida 0
Cliente2: Después de la jugada del cliente
x - -
- - -
- - -
Cliente2: Después de la jugada del servidor
x - -
- - -
o - -
Cliente1: Fila recibida 0 Columna recibida 2
Cliente1: Después de la jugada del cliente
- - x
- x -
o - -
Cliente1: Después de la jugada del servidor
- - x
o x -
o - -
Cliente1: Fila recibida 0 Columna recibida 0
Cliente1: Después de la jugada del cliente
x - x
o x -
o - -
Cliente1: Después de la jugada del servidor
x - x
o x -
o o -
Cliente2: Fila recibida 2 Columna recibida 2
Cliente2: Después de la jugada del cliente
x - -
- - -
o - x
Cliente2: Después de la jugada del servidor
x - -
- o -
o - x
Cliente1: Fila recibida 0 Columna recibida 1
Cliente1: Después de la jugada del cliente
x x x
o x -
o o -
Cliente1: El cliente ha ganado
Cliente2: Fila recibida 0 Columna recibida 2
Cliente2: Después de la jugada del cliente
x - x
- o -
o - x
Cliente2: Después de la jugada del servidor
x - x
- o -
o o x
Cliente3: ¡Se ha conectado un nuevo jugador!
Soy el servidor TCP esperando a que alguien se conecte
Cliente3: Fila recibida 1 Columna recibida 1
Cliente3: Después de la jugada del cliente
- - -
- x -
- - -
Cliente3: Después de la jugada del servidor

```

```
- - 0
- x -
- - -
```

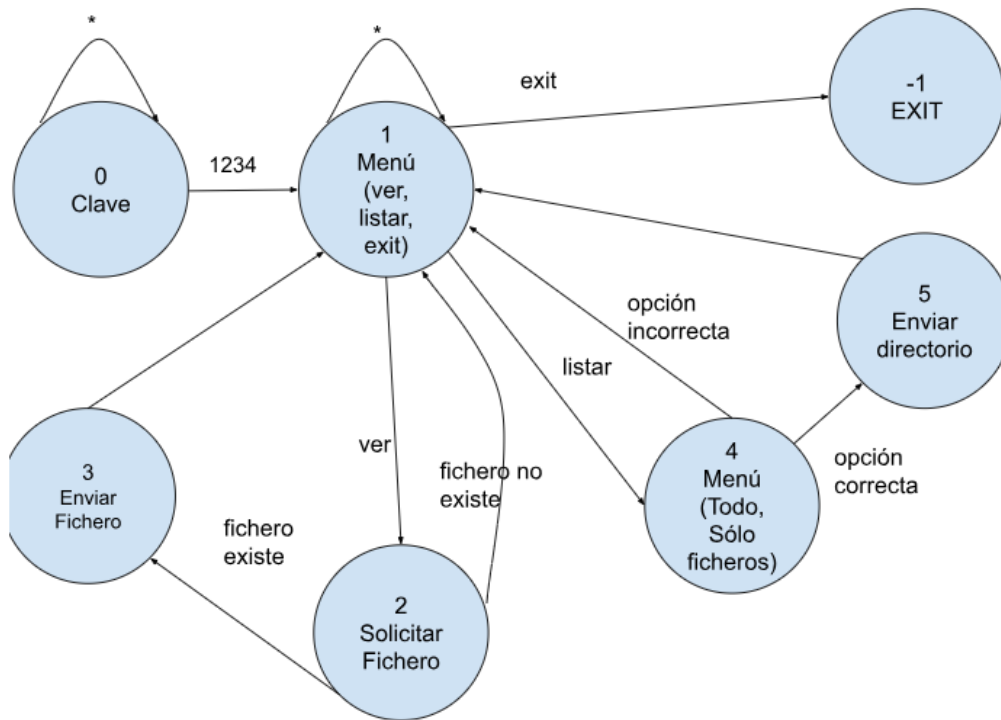
Cliente2: Fila recibida 3 Columna recibida 3

Cliente2: El cliente ha introducido valores erróneos. Terminará el juego.

Actividad 2 (7 puntos)

Implementa un **servidor al que los clientes se conectan mediante TCP** para consultar y visualizar el contenido de un directorio o ver un fichero no binario (txt, xml, etc). Debes implementar tanto el servidor y el cliente teniendo en cuenta lo siguiente:

- - El servidor va a estar continuamente esperando peticiones de los clientes en un puerto adecuado.
 - El servidor debe ser capaz de atender varias peticiones de clientes de forma concurrente.
 - Cuando los clientes se conectan tendrán que introducir una clave de acceso que les requerirá el servidor. Esta clave de acceso será 1234.
 - El servidor estará continuamente solicitando la clave hasta que el cliente le envíe 1234.
 - Una vez el cliente ha introducido una clave válida, el servidor le muestra las opciones disponibles (1. VER, 2.LISTAR y 3.EXIT).
 - Si elige VER el servidor pregunta al cliente por el nombre del fichero a visualizar. Sólo visualizamos ficheros no binarios como txt, xml, etc.
 - Si el fichero existe enviamos al cliente el contenido del fichero para que lo imprima por pantalla y volvemos al menú (1. VER, 2.LISTAR y 3.EXIT).
 - Si el fichero dado por el usuario no existe informamos al usuario y volvemos al menú (1. VER, 2.LISTAR y 3.EXIT).
 - Si elige LISTAR mostraremos al usuario un menú con las opciones (1. Visualizar sólo ficheros. 2. Visualizar directorios y ficheros). En ambos casos, mostraremos el contenido del directorio actual con la visualización elegida por el usuario. Una vez hecho esto volveremos al menú (1. VER, 2.LISTAR y 3.EXIT). Si introduce una opción incorrecta también volvemos al menú (1. VER, 2.LISTAR y 3.EXIT) pero sin listar nada evidentemente.
 - Si elige EXIT saldremos de la aplicación. El cliente podrá salir de la aplicación cuando el servidor le da la opción de poder salir. Fíjate en el diagrama de estados que sólo desde un estado le haremos caso al exit que nos pueda llegar del usuario.
 - La solución debe implementar el siguiente diagrama de estados. Además debe verse claramente en el código que estás usando estados (una variable por ejemplo) y la forma en que pasas de un estado a otro (modificando el valor de dicha variable y utilizando un switch para comprobar en el estado en el que se encuentra).
 - La lógica de la máquina de estados (transiciones) está en el servidor, el cliente únicamente se encarga de pedir datos al usuario por teclado, enviarlos al servidor e imprimir las respuestas.
 - Puedes utilizar el directorio raíz del proyecto para mostrar los ficheros y contenido del directorio o crear un directorio de ejemplo en el proyecto con algunos ficheros y directorios de prueba. El usuario no escribe ninguna ruta.
 - Escribe mensajes en el servidor para saber qué está pasando en cada momento.
 - Utiliza algún mecanismo en el servidor para identificar cada cliente como hiciste en la anterior actividad.



A continuación, la simulación de conexión de un cliente que se conecta al servidor con diferentes situaciones que pueden darse.

```

Introduce clave de acceso
3434
Introduce clave de acceso
1234
1.Ver
2.Listar
3.Exit
5
1.Ver
2.Listar
3.Exit
1
Introduzca el nombre del fichero a visualizar
adsad
Fichero no existe
1.Ver
2.Listar
3.Exit
1
Introduzca el nombre del fichero a visualizar
hola.txt
Contenido del fichero
hola
te doy mis dieses
1.Ver
2.Listar
3.Exit
2
1. Visualizar sólo ficheros
2. Visualizar directorios y ficheros
1
hola.txt
pom.xml
1.Ver
2.Listar
3.Exit
2
1. Visualizar sólo ficheros
2. Visualizar directorios y ficheros
2
hola.txt
pom.xml
src
target
1.Ver
2.Listar
3.Exit
2
1. Visualizar sólo ficheros
2. Visualizar directorios y ficheros
3
1.Ver
2.Listar
3.Exit
3

```

Criterios de puntuación. Total 10 puntos.

- **Actividad 1** (3 puntos).
- **Actividad 2** (7 puntos).

Recursos necesarios para realizar la Tarea.

Para realización de la actividad tan sólo es necesario tener instalado el entorno de desarrollo de Java.

Consejos y recomendaciones.

Leer detenidamente el contenido de la unidad.

Indicaciones de entrega.

Lo que debes entregar:

- Los **proyectos NetBeans** de las dos actividades **bien comentados**. **No es necesario Javadoc ni generar el .jar**. Cada ejercicio debe estar en una carpeta distinta, actividad-1 y actividad-2.
- **Manual con las pruebas realizadas**. Realiza un sólo manual con las dos actividades. Únicamente es necesario los pantallazos de la salida de tu programa y una pequeña explicación de lo que ocurre o si has utilizado algún parámetro de entrada a tu programa que deba poner también.
 - Para la primera actividad realiza los pantallazos de manera que se vea una ejecución con dos clientes que juegan a la vez. No es necesario probar todos los casos que pueden darse (perder, ganar, fallos, etc.), simplemente que se compruebe que dos o más clientes pueden conectarse y jugar a la vez.
 - Para la segunda actividad, probar distintos casos. Algo parecido a la prueba que te pongo en el enunciado del ejercicio.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo comprimido se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la cuarta unidad del MP de PSP, debería nombrar esta tarea como...

sanchez_manas_begona_PSP04_Tarea

Fecha de entrega: 7 de Diciembre de 2022 a las 23:55

Fecha con penalización: 10 de Diciembre de 2022 a las 23:55

Estado de la entrega

Número del intento	Este es el intento 1.
Estado de la entrega	No entregado
Estado de la calificación	Sin calificar
Fecha de entrega	jueves, 7 de diciembre de 2023, 23:55
Tiempo restante	4 días 21 horas
Última modificación	-
Comentarios de la entrega	+ Comentarios (0)

Agregar entrega

Todavía no has realizado una entrega.


Actividad previa

◀ [Tarea para PSP03 \(hasta el 19/11/23 23:55\)](#)

Ir a...

Mantente en contacto

Dirección General de Formación Profesional e Innovación

 Resumen de retención de datos

 Descargar la app para dispositivos móviles

TODOS LOS DERECHOS RESERVADOS 
El tema fue desarrollado por  por [conecti.me](#)