

## PSP04.- Aplicaciones cliente-servidor.

### Orientaciones para la tutoría

El objetivo de la unidad es que el alumnado profundice en la programación de aplicaciones Cliente/Servidor. Para ello, primero ampliará los conceptos teóricos relacionados con el modelo Cliente/Servidor para más tarde poder aplicar dichos conceptos a la programación en java de aplicaciones que utilicen Sockets.

#### Datos generales de la Unidad de Trabajo

Nombre completo del <u>MP</u>	Programación de servicios y procesos.	Siglas <u>MP</u>	<u>PSP</u>
Nº y título de la <u>UT</u>	04.- Aplicaciones cliente-servidor.	Horas	7/70
Índice o tabla de contenidos	<ul style="list-style-type: none"><li>1.- Paradigma Cliente/Servidor.<ul style="list-style-type: none"><li>1.1.- Características básica.</li><li>1.2.- Ventajas y desventajas.</li><li>1.3.- Modelos.</li><li>1.4.- Programación.</li><li>1.5.- Ejemplo <u>I</u>.<ul style="list-style-type: none"><li>1.5.1.- Ejemplo <u>I</u> (<u>II</u>).</li></ul></li></ul></li><li>2.- Optimización de sockets.<ul style="list-style-type: none"><li>2.1.- Atender múltiples peticiones simultáneas.</li><li>2.2.- Threads.</li><li>2.3.- Ejemplo <u>II</u>.<ul style="list-style-type: none"><li>2.3.1.- Ejemplo <u>II</u> (<u>III</u>).</li></ul></li><li>2.4.- Pérdida de información.</li><li>2.5.- Transacciones.</li><li>2.6.- Ejemplo <u>III</u>.</li><li>2.7.- Monitorizar tiempos de respuesta.</li><li>2.8.- Ejemplo <u>IV</u>.</li></ul></li></ul>		
Objetivos	<ul style="list-style-type: none"><li>✓ Conocer los conceptos más importantes sobre el modelo Cliente/Servidor.</li><li>✓ Utilización de hilos en la programación de aplicaciones en red.</li><li>✓ Programación de aplicaciones cliente/servidor.</li><li>✓ Depuración y gestión de transacciones en la utilización de sockets.</li><li>✓ Monitorización de los tipos de respuesta.</li></ul>		
Consejos y recomendaciones	Se recomienda que el alumnado compile y ejecute todo el código que se muestra a lo largo de la unidad.		
Tarea	Criterios de puntuación		
	<ul style="list-style-type: none"><li>✓ <b>Actividad 4.1</b> (2 puntos).</li><li>✓ <b>Actividad 4.2</b> (2 puntos).</li><li>✓ <b>Actividad 4.3</b> (6 puntos).<ul style="list-style-type: none"><li>1.- Diagrama de estados - 1 punto.</li><li>2.- Cliente – 1 punto.</li><li>3.- Servidor – 4 puntos.</li></ul></li></ul>		
	Criterios de corrección		

	<ul style="list-style-type: none"> <li>✓ Diagrama de estados (1 punto) – Penalizar 0,1 punto por cada estado incorrecto.</li> <li>✓ Cliente (1 punto).</li> <li>✓ Servidor . <ul style="list-style-type: none"> <li>➡ Transición entre estados – 1punto.</li> <li>➡ Mostrar el fichero – 1 punto.</li> <li>➡ Mostrar el directorio – 1 punto.</li> <li>➡ Claridad del código – 1 punto.</li> </ul> </li> </ul>
	<p style="text-align: center;"><b>Solución propuesta</b></p>
<p style="text-align: center;"><b>Temas de debate</b></p>	<ul style="list-style-type: none"> <li>✓ Tema de debate 4.1. Hebras de ejecución. En este debate se aborda la utilización de hebras de ejecución (threads) en las aplicaciones cliente/Servidor. Hay que hacer hincapié que además del modelo tradicional en el que un hilo atiende a un cliente, es posible necesitar múltiples hilos para atender al cliente.</li> <li>✓ Tema de debate 4.2. Autómatas finitos para la gestión de sockets. El objetivo de este debate es promover al alumnado la utilización de autómatas para la gestión de transacciones en los sockets.</li> </ul>

