


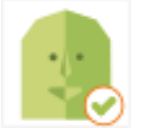
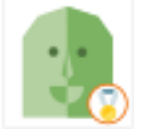


COMPETENCIAS TÉCNICAS PSP	 0	 2.5	 5	 7.5	 10
<b>Modelo de Comunicaciones (sockets)</b> RA3 100%	No se logra <b>conexión</b> estable <b>entre cliente y servidor</b> . No se usan correctamente los streams o los sockets.	La <b>conexión</b> se establece de forma básica, pero hay <b>fallos</b> en el <b>envío/recepción</b> de <b>mensajes</b> o en la <b>liberación de recursos</b> .	Se establece la <b>conexión</b> y se intercambian <b>mensajes privados y/o los recursos se liberan parcialmente al finalizar</b> .	Comunicación bidireccional estable con <b>mensajes públicos y privados</b> , usando adecuadamente <b>ObjectInputStream/ObjectOutputStream</b> y <b>liberación correcta de recursos</b> .	<b>Comunicación completamente funcional</b> , con <b>manejo robusto de errores</b> , <b>mensajes públicos/privados eficientes</b> y <b>cierre ordenado de todos los recursos</b> .
<b>Gestión de Hilos (Cliente)</b> RA2 40%	La <b>interfaz</b> se <b>bloquea</b> durante el envío o recepción de mensajes y <b>no hay separación</b> entre el <b>hilo principal</b> y el de <b>recepción de mensajes</b> .		Se usan <b>hilos independientes para recepción/envío</b> y la interfaz se actualiza en tiempo real con la información siguiente: <b>mensajes recibidos, diferenciando públicos y privados</b> usando técnicas adecuadas ( <b>SwingWorker, invokeLater, etc.</b> ).	<b>Interfaz fluida y actualizada en tiempo real</b> usando hilos con la información siguiente: <b>usuario cliente conectado-desconectado y mensajes recibidos, diferenciando públicos y privados</b> usando técnicas adecuadas ( <b>SwingWorker, invokeLater, etc.</b> ).	<b>Todo lo anterior</b> , y además se <b>informa de errores al cliente</b> (p.e cliente desconectado)
<b>Gestión de Hilos (Servidor)</b> RA2 55%	El servidor <b>solo</b> acepta <b>una conexión</b> . No hay manejo de <b>conurrencia</b> ni <b>sincronización</b> .		El servidor <b>acepta múltiples conexiones a la vez</b> , los <b>hilos (conexión y recepción/envío de mensajes)</b> son <b>independientes</b> y hay registro de acciones en el <b>log</b> .	<b>Todo lo anterior</b> y además, <b>sincronización básica</b> en secciones críticas.	<b>Manejo concurrente de clientes con sincronización de métodos efectiva</b> , <b>logs detallados</b> , <b>control de número máximo de conexiones</b> y <b>rechazo controlado de excedentes</b> .
<b>Monitorización</b> RA2 5%	No existe un mecanismo de <b>monitorización</b> o <b>no funciona</b> .		Se implementa un <b>hilo monitor</b> que <b>informa periódicamente del estado</b> , presentando estadísticas: <b>clientes conectados</b> .		Se implementa un <b>hilo monitor</b> que <b>informa periódicamente del estado</b> , presentando estadísticas: <b>clientes conectados, tiempo activo, último mensaje</b> .

**COMPETENCIAS  
TÉCNICAS PSP**

**Nota:**

**Durante** la calificación del reto se requiere no solo que el programa funcione, sino también que el alumnado explique **qué alternativas se evaluaron, por qué eligieron una, así como saber responder a las preguntas sobre el código que se realicen.** La falta de justificación o explicación reducirá la nota o supondrá suspender incluso si el código funciona sin errores.