

# Ejercicios WebSockets

1. Cree un servidor simple de WebSockets que simplemente devuelva el mensaje que se le envíe.
2. Cree una página en JavaScript con un cuadro de texto y un botón que envíe el mensaje almacenado en el cuadro de texto al servidor websocket, que sirva para comprobar el funcionamiento del servidor anterior. Debe tener un div en el que se muestren los mensajes recibidos desde el servidor. Debe ser similar a la siguiente:

[onopen] Conexión abierta. Mandando datos al servidor

[onmessage] Mensaje recibido del servidor: Hola 4

[onmessage] Mensaje recibido del servidor: Hola 3

[onmessage] Mensaje recibido del servidor: Hola 2

[onmessage] Mensaje recibido del servidor: Hola 1

3. Crear un cliente de ejemplo en Java que mande un mensaje de prueba al servidor y lo muestre en pantalla.

**Nota:** Usar la biblioteca tyrus. Se pueden descargar los archivos jar desde:

<https://mvnrepository.com/artifact/org.glassfish.tyrus.bundles/tyrus-standalone-client/2.1.2>

Se compila con:

```
javac -cp tyrus-standalone-client-2.1.2.jar Ejer3.java
```

Se ejecuta con:

```
java -cp .:tyrus-standalone-client-2.1.2.jar Ejer3
```

## Sesiones

4. Modifique el servidor del ejercicio 1 para que sólo permita mandar mensajes por websocket si la sesión está iniciada. Debe tener dos métodos GET que respondan las urls “inicio” y “cierre”. “inicio” debe abrir un sesión y cierre debe cerrarla.  
**Nota:** Si verifica que el usuario no ha abierto la sesión, puede cerrar el socket que se ha abierto con “ws.close()”.  
**Nota:** Verifique que si se cierra la sesión el socket sigue abierto. Es decir, el mecanismo de sesiones que se ha planteado no puede cerrar el socket una vez abierto.
5. Busque soluciones al problema del cierre de la sesión del ejercicio anterior. ¿A qué puede deberse?

## Lista de sockets y json

6. Analice el ejemplo del chat y localice la lista en la que se almacenan todos los sockets abiertos y las líneas de código en las que se almacenan los sockets, en las que se borran y en las que se usan para enviar mensajes a los clientes.
7. Modifique el ejemplo del chat para añadir un campo más al JSON que se usa para intercambiar mensajes en que se almacene uno o varios campos de su elección. Puede ser el nombre de usuario, la fecha y hora del mensaje,...