

Guión de prácticas – Sockets en Java

Índice de contenido

Guión de prácticas – Sockets en Java	1
Objetivo del documento	1
Ejemplo 1. Eco (Echo)	1
Ejemplo 2. Consulta de cita (Quote)	1
Ejemplo 3. Multidifusión de citas (Multicast)	1
Ejemplo 4. Eco (Echo) con Java NIO (Non-blocking IO)	2
Licencia	3

Objetivo del documento

El presente documento presenta **cuatro casos de estudio de programación cliente-servidor con sockets**, a partir de los códigos de ejemplo del tutorial de Java y otra documentación, disponible en la web¹.

Como variación a lo propuesto en dicho tutorial, el desarrollo se realiza con **Maven**, desglosándose cada ejemplo, en dos proyectos con Maven: uno para el servidor y otro para el cliente.

Los proyectos están preparados para su ejecución con Maven a través del plugin *exec-maven-plugin* (<http://www.mojohaus.org/exec-maven-plugin/>).

Ej: `$> mvn exec:java`

Se deben revisar los códigos correspondientes, compilar y ejecutar los proyectos, comprobando su correcto funcionamiento ejecutando cliente y servidor en distintas máquinas del laboratorio.

Ejemplo 1. Eco (Echo)

Ejemplo de cliente-servidor con TCP, donde el servidor responde con el mismo texto enviado por el cliente, realizando el típico eco (echo).

Se debe observar con cuidado que el servidor sólo admite un cliente (se bloquea en espera de la primera petición), quedando pendiente como ejercicio modificar el código para permitir múltiples clientes (lanzando un hilo por petición aceptada).

Ejemplo 2. Consulta de cita (Quote)

Ejemplo de cliente servidor con UDP, donde el servidor responde a peticiones del cliente enviando una cita (quote) al cliente. Las citas se leen de un fichero externo. Sigue el modelo *pull*, donde los clientes realizan la petición.

Ejemplo 3. Multidifusión de citas (Multicast)

Ejemplo de multidifusión con UDP, donde el servidor reenvía citas (leídas de un fichero de texto) a **todos** los miembros del grupo de multidifusión, siguiendo un modelo *push*, donde el servidor envía los mensajes.

El proyecto depende de clases del proyecto previo, `QuoteDatagramServer`. En Eclipse es necesario indicar esta dependencia en el *build path* del proyecto, indicando la dependencia del proyecto (además el proyecto debe estar abierto).

¹Consultar <http://docs.oracle.com/javase/tutorial/networking/sockets/index.html>.

Para su compilación y ejecución con Maven, se ha incluido su dependencia en el `pom.xml`. Para que la biblioteca esté disponible es necesario que previamente se haya instalado el proyecto `QuoteDatagramServer` en el repositorio local:

```
$> mvn install
```

Si el proceso ha sido correcto, en el directorio `.m2\repository` se habrá instalado el `.jar` correspondiente. Comprobando este punto, se puede realizar la compilación y ejecución del proyecto en Maven.

Ejemplo 4. Eco (Echo) con Java NIO (Non-blocking IO)

Ejemplo de cliente-servidor con TCP utilizando la variante con la API Java NIO, donde el servidor responde con el mismo texto enviado por el cliente, realizando el típico eco (echo).

Se debe observar con cuidado que el servidor puede gestionar múltiples conexiones sin necesidad de hilos independientes puesto que el objeto `Selector` gestiona múltiples canales (`SocketChannel`). Ver Ilustración 1. Esquema de gestión de canales en un selector.

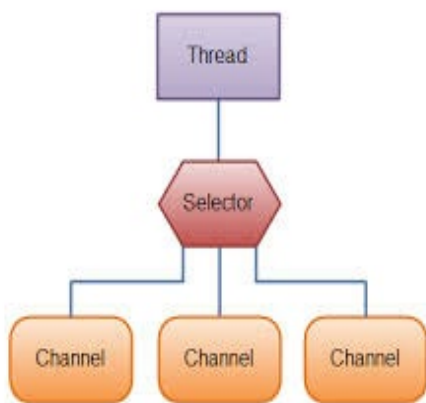


Ilustración 1: Esquema de gestión de canales en un selector

Se trabaja en modo **no bloqueante**, asíncrono, gestionándose los eventos en el bucle principal, y trabajando con el concepto de buffer (`ByteBuffer` en el ejemplo).

Se recomienda, además de revisar el ejemplo concreto, consultar la documentación adicional en:

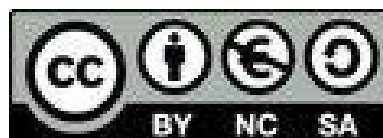
Java NIO Tutorial.

The Rox Java NIO Tutorial

Scalable IO in Java (PDF)

Licencia

Autor: Joaquín Seco & Raúl Marticorena
Área de Lenguajes y Sistemas Informáticos
Departamento de Ingeniería Informática
Escuela Politécnica Superior
UNIVERSIDAD DE BURGOS



Esta obra está bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 3.0 Unported. No se permite un uso comercial de esta obra ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula esta obra original

Licencia disponible en <http://creativecommons.org/licenses/by-nc-sa/3.0/>