

## Práctica final: comparativa de tecnologías MOM

Ingeniería de Sistemas Telemáticos.

Grado en Ing. en Tecnologías de la Telecomunicación, 2016-2017.

Escuela Técnica Superior de Ingeniería de Telecomunicación.

Universidad Rey Juan Carlos.

Fecha de entrega: **Defensa presencial, lunes 8 de mayo de 2017, 16:00 - 18:00.**

Modalidad: Por parejas o individual.

---

### 1. Objetivo de la práctica

El objetivo de la práctica es comparar dos tecnologías de MOM para Java:

- Java Message Service (JMS).
- ZeroMQ.

La práctica se centra en la implementación de dos sistemas de sincronización mediante paso de mensajes, con esquemas de comunicación similares pero diferentes propiedades. El alumno debe comprender la implementación de cada uno de los dos sistemas propuestos (código línea a línea), así como las funciones que proporcionan y limitaciones de cada uno de los dos sistemas.

### 2. Enunciado de la práctica

Calificación máxima: **10 puntos.**

#### 2.1. Implementación de MOM con JMS

##### Proyecto `urjc.ist.jms.pubsubexample`

La primera parte de la práctica consiste en importar y hacer funcionar correctamente el proyecto de ejemplo `urjc.ist.jms.pubsubexample` disponible en el Tema 5 del Aula Virtual.

En particular, en la práctica final nos centraremos en la combinación de la clase `PubSubSender` y `PubSubAsyncReceiver`. Se deben probar varias combinaciones de ejecución, incluyendo:

- Primero el servidor y luego el cliente.
- Primero el cliente y luego el servidor.
- Varios servidores y varios clientes, simultáneamente (alternando el orden de ejecución de los mismos).

Para la última opción, se recomienda añadir retardos más largos entre los mensajes enviados al *broker*, enviar un mayor número de mensajes (para dar tiempo a que más clientes se conecten y los reciban) o una combinación de ambas estrategias.

##### Mejora opcional

Como mejora opcional para esta parte se propone mejorar la implementación de la clase `PubSubAsyncReceiver`, para que se realice por medio de hilos ejecutores de tareas (`Executor`, `ExecutorService`, etc.), tal y como se vió en el Tema 3.

## 2.2. Implementación de MOM con ZeroMQ

### Patrón Pub-Sub en ZeroMQ (básico)

En la sección [Getting the Message Out](#) de la ZeroMQ Guide podemos encontrar la explicación y ejemplos de código para un patrón de comunicación similar al del apartado anterior con JMS.

Se pide:

- Importar en un paquete nuevo las clases `wuserver` (<http://zguide.zeromq.org/java:wuserver>) y `wuclient` (<http://zguide.zeromq.org/java:wuclient>) de la guía de ZeroMQ.
- Instalar la biblioteca de ZMQ para Java en nuestro sistema, tal y como se ha indicado en clase, y ejecutar el ejemplo para comprobar la funcionalidad.
- Como en el caso anterior, se deben ejecutar las clases en distinto orden (primero el publisher y luego el subscriber, y viceversa; varios publishers y subscribers, etc.).

### Mejora opcional

Extender el esquema de comunicación anterior con el patrón de diseño presentado en la sección [The Dynamic Discovery Problem](#) de la ZeroMQ Guide, para el escenario antes planteado (según se describe en la Figura 13 de la guía, en esa misma sección).

Se puede encontrar, además, un ejemplo resuelto de device intermediario para este caso: <http://zguide.zeromq.org/java:wuproxy>.

## 3. Evaluación

La evaluación de esta práctica final se realizará el día del examen final (8 de mayo de 2017), tras la parte teórica de la prueba.

La práctica se podrá realizar de forma individual o por parejas. **No se admitirá, en ningún caso, grupos de 3 o más personas.**

La evaluación consistirá en una prueba de ejecución *in situ* de ambos proyectos, tras la cual el profesor realizará varias preguntas a cada alumno para confirmar que se han asimilado correctamente todos los conceptos clave y que se conoce la implementación de la solución propuesta.

La implementación de las partes opcionales propuestas ofrecerá de partida más puntos a los alumnos, es decir, se obtendrá mejor nota si se cometen fallos similares en la presentación que otra persona o pareja que presente la práctica básica.