Práctica 4

Descripción:

- La práctica consiste en el desarrollo de una especie de *Central* que coordina la información de supuestos *Arduinos* y controla la iluminación y la temperatura de cada *Oficina* mediante el paso de mensajes de *ActiveMQ*.

Clases:

- **Central:** Es la clase encargada de gestionar los datos a distancia de todas las oficinas, en este desarrollo usaremos 2 oficinas pero se podrían usar tantas como quisiéramos.
- **OficinaN:** Es la clase encargada de leer la luz y la temperatura capturada por un supuesto *Arduino* y le envía los datos constantemente a la *Central* y en caso de ser necesario se actualizarán los datos. Cada N es de una oficina, teniendo en el ejemplo Oficina1 y Oficina2.
- **Oficina:** Es la clase que lleva la información básica de la *Oficina* y sus datos. Es usada en las *Oficinas* (*Oficina1 y Oficina2*) y en la *Central* a través de *Actuador*.
- **Actuador:** Es una clase básica para gestionar el envío de datos a través del topic *Actuador*, pero podría llegar a ser una clase prescindible modificando las demás (se ha dejado por falta de tiempo).

Topics de ActiveMQ:

- **Lecturas:** Estos topics son enviados por las *Oficinas* y recibidos por la *Central*. Representan la información actual que tiene una *Oficina*, desde los datos actuales (luz actual) como los ideales (temperatura ideal) y las variaciones (variación de luz aceptada).
- **Actuador:** Estos topics son enviados por la Central y recibidos por las *Oficinas*. Representan la información que debe cambiar (si es necesario) en los controladores de temperatura y luz de las oficinas.

Herramientas:

- **Eclipse:** como IDE de desarrollo de los servicios web y contratos *WSDL* además del servicio de orquestación con *BPEL*.
- **ActiveMQ:** como servicio encargado del paso de mensajes

Puesta en marcha:

1. Iniciar *ActiveMQ*.
2. Iniciar clases (no importa el orden, la *Central* recogerá los datos en cuanto se conecte y las Oficinas trabajarán con datos preestablecidos que serán modificados cuando se conecte la *Central*).

Demostración:

- Se adjuntó un video demostrativo con los pasos por los que se hace uso de TODAS las utilidades.

Aspectos a destacar:

- Se ha hecho uso de programación con hilos (*Threads*) para tener un hilo que escucha y modifica los datos estáticos tanto en *Central* como en *Oficina*.
- Se ha hecho uso de programación asíncrona en ambos archivos.
- Aunque una pequeña refactorización se podría usar tantas oficinas como quisiéramos añadiendo el nombre al comienzo de la ejecución de la *Oficina* hemos hecho dos archivos *.java*.

Dificultades encontradas:

No se han encontrado grandes dificultades, la única es tener que hacer la práctica con la mayor brevedad posible por falta de tiempo y por ello varias cosas no han sido modificadas mediante refactorización para ahorrar tiempo (como fusionar *Oficina1 y Oficina2,* prescindir de *Actuador.java*, etc).

A nivel personal me ha parecido sencillo y enriquecedor aunque he recurrido a tener que implementar muchos métodos como extraer un campo *XML* y generador *XML*.