

MOM

MESSAGE ORIENTED MIDDLEWARE

SERGIO JULIO GARCÍA URDIALES

18/04/18

Funcionamiento

La práctica tiene 3 partes diferenciadas.

- Cliente: La consola del sistema, por la que pasa toda la información de los sensores y manda señales a los actuadores. Localizada en el archivo *consola.js*.
- Servidores: En este caso tenemos 2, cada una de las oficinas. Tienen su lógica en los archivos *oficina1.js* y *oficina2.js*
- Activemq: El encargado de gestionar nuestra mensajería, con unos *topics* creados en función de la información que contengan.

Lo primero que hacen los tres archivos es conectarse al servidor de *Activemq*, de forma que puedan enviar y recibir mensajes de los *topics* correspondientes. Para poder tener información sobre los sensores, esta información se ha simulado y va cambiando en el tiempo.

Aleatoriedad de los datos.

Cada oficina cuenta con unos simuladores de datos de luz y temperatura. En la parte superior de cada archivo hay algunas constantes que permiten decidir cada cuanto se enviaran mensajes a la consola, o cual es el mínimo o máximo que pueden alcanzar sus sensores para indicar peligro (ponerse en rojo).

Además, hay unos valores que indican cómo se van a simular el comportamiento de la temperatura y de la luz. La secuencia que sigue por cada elemento a medir de cada oficina (luz y temperatura de cada uno) es la siguiente.

1. Se genera un valor aleatorio que indicará si el sensor empieza a subir o bajar. Este valor se vuelve a generar en función de una de las constantes, por defecto a 20 segundos, por lo que pasado este tiempo la temperatura puede volver a subir o bajar durante los siguientes 20 segundos si los activadores no actúan.
2. Una vez fijado el comportamiento (subir o bajar), se generará aleatoriamente un valor, entre 20 y 80 para la luz, y entre 2 y 5 para la temperatura. En función de una de las constantes, los valores subirán o bajarán de forma aleatoria dentro de ese rango. Por defecto, cada segundo los valores suben o bajan.
3. Si se activa un activador, cambiará el valor actual de subir o bajar a su contrario.

Mandar datos a la consola.

El envío de mensajes está regulado por una constante que hará que se repita el envío de mensajes. Estos mensajes son enviados a un topic de *Activemq* con un nombre que les identifica. Cada sensor envía un mensaje con dos apartados.

- Origin: Identifica qué sensor es.
- Value: Valor del sensor en ese momento.

La consola.

La consola se conecta a ActiveMQ y se suscribe a los *topics*, y controla la información que recibe. Dispone de cuatro constantes en las que se puede indicar cuando quieren que se activen los activadores.

En función de los datos recibidos y comprobando si sobrepasan o no las constantes, envía mensajes a los activadores con la acción que deben de tomar, si es subir o bajar cualquiera de sus valores. Pero esto no lo hace directamente, lo que hace es enviar mensajes de nuevo a ActiveMQ, al *topic* correspondiente en el que le indica la acción que debe tomar.

Los activadores.

Los activadores están suscritos a los *topic* de ActiveMQ correspondientes, que son los que envía la consola. Estos *topics* tiene un valor dentro del mensaje que indica la acción a tomar para modificar los valores de temperatura o luz de la oficina correspondiente.

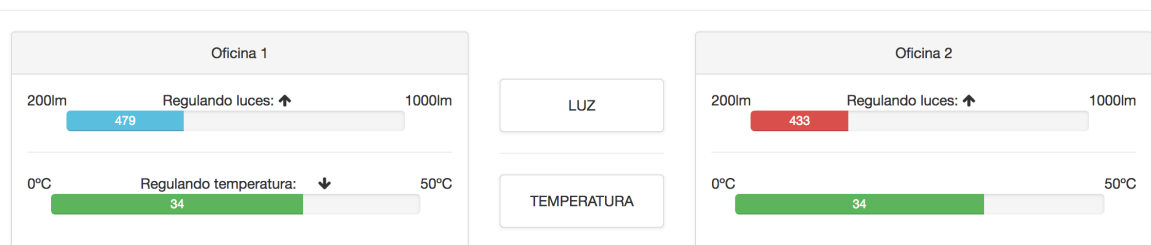
Puesta en marcha

Lo principal es lanzar el servidor Activemq, para esto hay que ir a la carpeta correspondiente y ejecutarlo.

```
./apache-activemq-5.15.3/bin/activemq console
```

Para poder ver de forma visual lo que está pasando, la practica cuenta con un HTML en el que se cargan las oficinas. Nada más abrir el HTML se cargarán las oficinas y consola, y se realizarán las conexiones pertinentes. El resultado es algo como esto:

Oficinas Disponibles



En la imagen se ve como la luz de la oficina dos indica que el activador correspondiente está subiendo las luces. Lo mismo ocurre en la oficina uno, alcanzando ya niveles estables, por lo que el siguiente mensaje que se mande hará que se desactiven los activadores correspondientes.

La primera ejecución tiene un tiempo de espera mayor para reaccionar, puesto que se están realizando todas las conexiones. Pasados unos 5 segundos, la aleatoriedad de los números y acciones, junto con el proceso de evaluar y actuar sobre los activadores harán que la simulación sea completa y muy visual.

Fuentes

Para la realización de la práctica se necesita usar *Stomp.js*, incluido en la carpeta JS, que nos permitirá realizar la conexión con Activemq en este lenguaje de programación.

Además, un tutorial realizado por Miguel Arlandy Rodríguez ([Fuente](#)) explica el proceso de creación de un chat con este proceso, muy útil para tomar contacto con ActiveMq