



Universidad de Murcia

Grado en Ingeniería Informática
Curso 2020/2021

Programación para las comunicaciones

Profesor: Humberto Martínez Barberá

Práctica VI

Jaime Ortiz Aragón - 49196689B - jaime.ortiza@um.es

- ESCENARIO

El escenario de esta práctica parte del implementado en la práctica anterior, pero añadiendo el servicio REST.

El funcionamiento final será más completo que en la práctica anterior, ya que en esta no se contemplaba el envío de mensajes *unicast* a los servidores para cambiar propiedades de sus envíos con los datos correspondientes a sus sensores.

En este caso, a la posibilidad de mostrar los últimos datos de los sensores, se añaden las funcionalidades de aumentar la velocidad de envío, disminuirla y parar el envío de datos. Para realizar esto es necesario el tratamiento del recurso al que se está accediendo, como se explicará en el despliegue.

Sumado a esto, se sigue contando con la funcionalidad de pedir los datos haciendo uso del mismo correo de Gmail que se ofrecía en la práctica 5, mediante los servicios POP/SMTP.

- DESPLIEGUE

Para la realización de esta práctica, se han creados dos nuevas clases : *HttpRest.java* y *HttpsRest.java*. La lógica de estas clases es similar a la de las prácticas 1 y 4, ya que el servidor escuchará en un puerto una petición y formará una respuesta que envíe al navegador.

Para el acceso a los datos, sin hacer uso de la API REST se utilizan los puertos 9999 y 4430 para las conexiones HTTP y HTTPS respectivamente, por lo que para las conexiones de la API, los servidores estarán escuchando esta vez en los puertos 10000 y 4431 respectivamente.

Como se ha comentado, la principal diferencia con las peticiones de las prácticas anteriores es que esta vez es necesario analizar a qué url se está accediendo. Si se accede a la raíz del directorio, es decir, a "/" obtendremos una página de índice. También se accede a ella accediendo directamente a *index.html*.

[Obtener datos](#) [Detener servidor](#) [Parar](#) [Aumentar frecuencia de envío](#) [Reducir frecuencia de envío](#)

Este recurso será el punto de acceso a la API, el cual tendrá como enlaces los 4 que se muestran en la captura referentes a las distintas acciones que permite el servicio de información.

- Obtener datos : Se muestran los últimos datos de los sensores como se hacía en la práctica 5. El recurso que los genera es *info*.
- Detener servidor : Lanza un mensaje unicast que detiene el servidor. Este mensaje se implementó en la práctica 3. El recurso es *stop*.
- Aumentar frecuencia de envío : Se genera un mensaje de control unicast desde el cliente hacia el servidor para doblar el valor de la frecuencia de envío de datos. El recurso que lo genera es *speedup*.

- Reducir frecuencia de envío : La lógica es equivalente a la de aumentar la frecuencia, pero en este caso se reduce a la mitad, doblando el valor del *delay*. El recurso que lo realiza es *speeddown*.

