

Práctica 4: Node-RED

Sistemas Distribuidos

Curso: 2021/2022

Índice

1. Información general de la práctica	2
2. Contexto y MQTT	2
3. API REST	3
4. Bot de Telegram	3
5. Evaluación	3

1. Información general de la práctica

- La práctica se puede realizar de forma conjunta con un compañero/a. Ambos integrantes deben pertenecer al mismo grupo de prácticas.
- Se admiten trabajos individuales, pero no de tres estudiantes.
- En el Campus Virtual se encuentra habilitada una tarea para realizar la entrega.
- La fecha de entrega establecida debe consultarse en dicha tarea. Atención, puede ser diferente para cada grupo de prácticas.
- No se admitirán entregas fuera de plazo.
- La entrega debe realizarla solo uno de los integrantes del grupo, en caso de realizar la práctica por parejas.
- La entrega se compone de tres archivos:
 - un archivo comprimido que contendrá los flujos (**sin crear subdirectorios**). El comprimido deberá nombrarse como `code.zip`.
 - un vídeo de máximo 5 minutos mostrando la funcionalidad del bot del ejercicio 3.
 - un archivo de texto, denominado INTEGRANTES.txt, donde se incluyan los nombres y apellidos de los integrantes del grupo, en el caso de realizarse la práctica por parejas.
- **Se ruega encarecidamente** que comprobéis que los flujos entregados están completos. Si se recibe un flujo incompleto, el ejercicio se evaluará con un 0.

2. Contexto y MQTT

Crear un flujo con Node-RED que publique de forma periódica, cada 10 segundos, un número entero aleatorio¹ entre 10 y 100 (intervalo cerrado) a través de un evento MQTT. Otro flujo deberá recibir este evento y almacenar en un acumulador local al nodo la sumatoria de los números pares recibidos. Este último flujo, mostrará cada vez que reciba un evento, independientemente de si el número es par o impar, el número recibido y el valor actual del acumulador en la terminal de depuración de Node-RED.

Como broker de MQTT se puede instalar Mosquitto de forma local, tal y como se expuso en el Seminario 6.

Para este ejercicio se deberán entregar dos flujos, nombrados como `ex1_publisher.json` y `ex1_subscriber.json`.

¹<https://flows.nodered.org/node/node-red-node-random>

3. API REST

Crear un flujo en Node-RED que implemente un endpoint POST, denominado, `/reverse`. Este endpoint recibirá un JSON con una cadena de texto, bajo la clave `“str”`, la cual devolverá al cliente invertida bajo la clave `“reversed_str”`.

Por ejemplo, si el cliente envía la siguiente petición:

```
1 {  
2   "str": "I can only show you the door. You're the one that has to walk through it."  
3 }
```

El endpoint, devolverá:

```
1 {  
2   "reversed_str": ".ti hguorht klaw ot sah taht eno eht er'uoY .rood eht uoy wohs ylno nac I"  
3 }
```

Para este ejercicio, se deberá entregar el flujo implementado con el nombre `ex2_api.json`.

4. Bot de Telegram

Se deberá crear un flujo de Node-RED que implemente un bot de Telegram. Dicho bot deberá ofrecer, al menos, tres interacciones diferentes con el usuario. La naturaleza de las interacciones y la funcionalidad del bot quedarán a juicio del estudiante, valorándose la originalidad de estas. Pueden incorporarse tantos nodos de Node-RED contribuidos como se desee o consultas a API externas (p. ej., <https://openweathermap.org/>)

Para este ejercicio, se deberá entregar el flujo del bot implementado, bajo el nombre `ex3_bot.json`. Adicionalmente, se deberá entregar un vídeo de **máximo 5 minutos**, mostrando la funcionalidad del bot.

En cada grupo de prácticas, a partir de los vídeos, cada estudiante votará los tres bots más interesantes desde su punto de vista. Los tres con mayor número de votos recibirán un punto extra en la nota de esta práctica (Práctica 4). No serán válidos los autovotos, y, en caso de empate, será el criterio del profesorado el que determine las mejores implementaciones.

5. Evaluación

La nota de la práctica se divide de la siguiente forma:

- Contexto y MQTT: 25 %
- API REST: 25 %
- Bot de Telegram: 50 % + 10 %

Bibliografía

- Node-RED. Connect to an MQTT Broker. <https://cookbook.nodered.org/mqtt/connect-to-broker>