

Trabajo Practico N2

Alumno: Carlos Emanuel Balcazar

Fecha de entrega: 02/05/2018

Repositorio del proyecto: <https://github.com/emanuelbalcazar/TNT-node-red>

1. Descargue la imagen de nginx y mqtt. Cree un proyecto con la herramienta docker-compose dónde utilice a:

- **nginx como proxy inverso 12 dónde el tráfico se dirija a la aplicación express del práctico anterior.**
- **node agregue node-red al package.json 3**
- **mqtt configurado con transporte websockets.**

Se descargo nginx y node-red para realizar el trabajo, quedando mqtt pendiente para los proximos avances. En el *Dockerfile* de node-red y storage-api se definieron los comandos que permiten levantar cada aplicación utilizando el CMD.

2. Defina una serie de endpoints REST para dar soporte a <https://github.com/node-red/node-red/tree/master/red/runtime/storage> Defina qué mecanismo de Express la permitirán autenticar los requerimientos HTTP.

Para este ejercicio se desarrollo un plugin llamado *http-storage.js* en donde se implementan el guardado y la recuperacion de los flows utilizados en node-red. Estos realizan el correspondiente GET y POST HTTP a un backend implementado en NodeJS y ExpressJS llamado storage-api que expone los servicios REST asociados a los flows.

3. Agregue registro de cambios a los flows definidos en el punto 2. Implemente la especificación. Cada cambios en los flujos deberá estar asociado con un autor, una fecha y un estado previo.

Por el momento, los flows se guardan dentro de un objeto que contiene como adicional un atributo llamado *version* en donde se coloca la fecha actual momentos antes de ser enviado a persistir, esto permite tener versionado por fechas los diferentes flows y al momento de recuperar siempre se obtienen los ultimos flows almacenados.

```
[{ flows: [array of flows], version: 'date', id: " }]
```

Queda pendiente agregar el autor y el estado previo.

4. Modifique la configuración de Node-RED dentro del contenedor para que utilice los cambios implementados en el punto 3. Para esto deberá escribir una backend y modificar la configuración de su proyecto para que lo utilice.

La configuracion del proyecto se adapto para ser ejecutada desde un contenedor, para ello se tiene un *docker-compose.yml* en donde se especifica que elementos se utilizaran, que puertos se exponen y quienes dependen de otros contenedores. La conexión a la base de datos y el host a donde se realizan las peticiones HTTP tambien se modificaron en cada componente para permitir la conexión de los mismos dentro del contenedor.