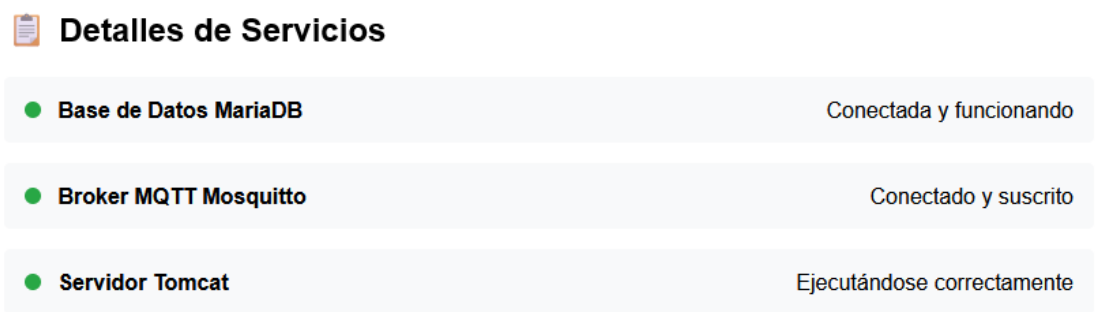
PECL2

Creación de infraestructura para el soporte de los dispositivos y aplicaciones de control del sistema de gestión urbano.

Una vez desarrollados los dispositivos que distribuiremos por las calles de la ciudad para la obtención de los datos, necesitamos crear la infraestructura necesaria para soportar nuestro sistema en producción.

El sistema contará con 3 servicios interconectados entre sí:

- Base de datos donde el sistema almacena la información recibida.
- Bróker MQTT Mosquitto el cual permite la conexión entre la red de sensores y el servidor de forma bidireccional.
- Servidor Tomcat.



Las estaciones se comunicarán con el sistema por medio del Broker MQTT, con el cual intercambiarán información en formato JSON (el mismo que era requerido para la PECL1). A continuación, se muestra la estructura básica de mensajes aceptada por el servidor:

```
{
  "sensor_id":"WS_BAR_01",
  "sensor_type":"XXXX",
  "street_id":"ST_0478",
  "timestamp":"2025-09-18T09:37:28.225646",
  "location":{
    "latitude":40.45409486,
    "longitude":-3.60391176,
    "altitude_meters":646.19,
    "district":"Barajas",
    "neighborhood":"Barajas"
  },
  "data":{ ... }
}
```

Dado que existen distintas tipologías de sensores, la etiqueta “sensor_type” especificará el tipo de red de sensores que está transmitiendo la información. Además, se ha incluido en el mensaje anterior la etiqueta **data** que contendrá información específica de cada tipología.

A continuación, se muestra cada una de las tipologías de datos en mayor detalle.

Estaciones meteorológicas

El objetivo principal de este tipo de estaciones de sensores es monitorizar las variables meteorológicas de la zona. Entre los valores recopilados se encuentran la temperatura, la humedad o la presión atmosférica entre otros.

Para poder enviar información al sistema será necesario especificar en el JSON base el valor de la variable sensor_type como "weather". Además, la estructura de la etiqueta data seguirá la siguiente estructura:

```
{
  .....
 "sensor_type":"weather",
 "data":{
   "temperature_celsius":17.9,
   "humidity_percent":64.9,
   "air_quality_index":79,
   "wind_speed_kmh":8.0,
   "wind_direction_degrees":17,
   "atmospheric_pressure_hpa":992.7,
   "uv_index":2
 }
```

Contadores de tráfico

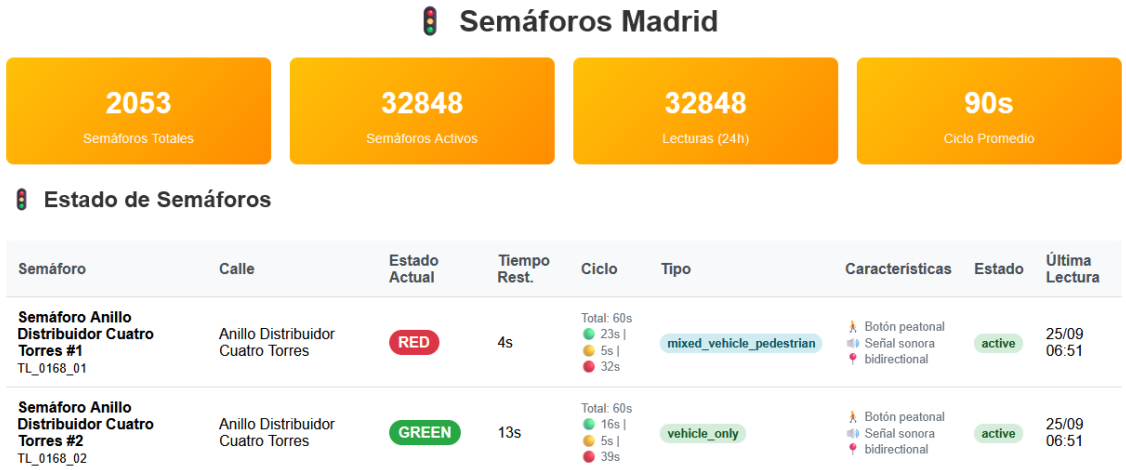
Este tipo de red de sensores permite hacer un seguimiento del tráfico de la calle.

Para poder enviar información al sistema, la estructura de mensajes enviados será la siguiente:

```
{
  ......
  "sensor_type":""traffic_counter",
  "data":{
    "vehicle_count":46,
    "pedestrian_count":181,
    "bicycle_count":9,
    "direction":"east",
    "counter_type":"mixed",
    "technology":"magnetic",
    "average_speed_kmh":27.6,
    "occupancy_percentage":100,
    "traffic_density":"high"
  }
}
```

Semáforos

El sistema permite la monitorización y control de los semáforos. Esto permite la incorporación de agentes externos que permitan añadir inteligencia extra al funcionamiento en caso de ser necesario. Este tipo de redes de sensores permiten conocer el color del semáforo, el tiempo restante hasta el cambio de color, incluso la tipología del semáforo entre otros datos.



Para poder enviar información al sistema, la estructura de mensajes enviados será la siguiente:


```
{
  ......
 "sensor_type":""traffic_light",
 "data":{
   "current_state":"green",
   "cycle_position_seconds":7,
   "time_remaining_seconds":46,
   "cycle_duration_seconds":120,
   "traffic_light_type":"vehicle_only",
   "circulation_direction":"bidirectional",
   "pedestrian_waiting":false,
   "pedestrian_button_pressed":false,
   "malfunction_detected":false,
   "cycle_count":8,
   "state_changed":true,
   "last_state_change":"2025-09-18T09:50:39.549745"
 }
```

Pantallas de información

Por último, el sistema cuenta con una red de pantallas distribuidas por la ciudad con diferente propósito como puede ser informar del estado del tráfico o de las condiciones meteorológicas de la zona.

Para poder enviar información al sistema, la estructura de mensajes enviados será la siguiente:


```
{
  ......
 "sensor_type":""information_display",
 "data":{
   "display_status":"active",
   "current_message":"Desvío temporal",
   "content_type":"traffic",
   "brightness_level":78,
   "display_type":"lcd_panel",
   "display_size_inches":55.0,
   "supports_color":true,
   "temperature_celsius":41.0,
   "energy_consumption_watts":153.5,
   "last_content_update":"2025-09-18T09:53:28.874874"
 }
}
```

Trabajo a realizar

Se pide realizar una infraestructura de contenedores basada en Docker mediante la creación de los ficheros necesarios, Docker_compose.yml y Dockerfiles para levantar los 3 servicios y que los mismos en caso de fallo vuelvan a ponerse en marcha automáticamente.

Todos los servicios estarán en el puerto estándar de cada servicio: MQTT 1883, Base de datos (por ejemplo, MariaDb 3306) y Tomcat 8080, el servicio de la base de datos solo puede ser accesible desde dentro de la infraestructura de Docker y nunca desde el exterior, estando los 3 servicios dentro del mismo segmento de red y exponiendo al exterior solo lo necesario. Además, la conexión del proyecto desplegado en tomcat con la base de datos deberá realizarse mediante un pool de conexiones previamente configurado.

Cuando arranquen los servicios por primera vez deberán crear automáticamente la/s bases de datos necesarias y crear automáticamente las tablas e incorporar información previa, o en el caso de tomcat ya sea por inicio por primera vez o por un reinicio del servicio o cambio de la aplicación, deberá desplegar la aplicación automáticamente.

Todos los contenedores deberán tener almacenamiento persistente para que se mantengan los datos en el tiempo.

En cuanto al proyecto de Tomcat deberá contar con al menos un endpoint HTTP que permita realizar una consuta sobre alguno de los datos de la base de datos (por ejemplo, conocer los valores de los sensores en un día concreto). Este endpoint será utilizado por la aplicación Android desarrollada en la siguiente práctica. Además, deberá de mandar la información a MQTT necesaria para activar los actuadores desarrollados en la PECL1.

Entregable

Para que la entrega se considere válida se debe subir a la plataforma con fecha límite marcada en la Balckboard el código desarrollado junto con las carpetas y ficheros, en caso de que sea necesario, para la prueba y funcionamiento de la infraestructura, así como un archivo .txt con las instrucciones para levantar los servicios y pequeño vídeo donde se muestre el comportamiento en tiempo real, es decir, como se crean y despliegan todos los contenedores necesarios para el buen funcionamiento de la infraestructura propuesta.

Bibliografía:

- Presentación teórico-practica de la asignatura
- Documentacion Docker Docker Docs
- https://www.freecodecamp.org/espanol/news/guia-de-docker-para-principiantes-como-crear-tu-primera-aplicacion-docker/
- http://recetas-docker.readthedocs.io/es/latest/capitulo_1.html