

# Trabajo Práctico Integrador - Python

Capacitación Semillero DISW - Agosto 2021

## Introducción

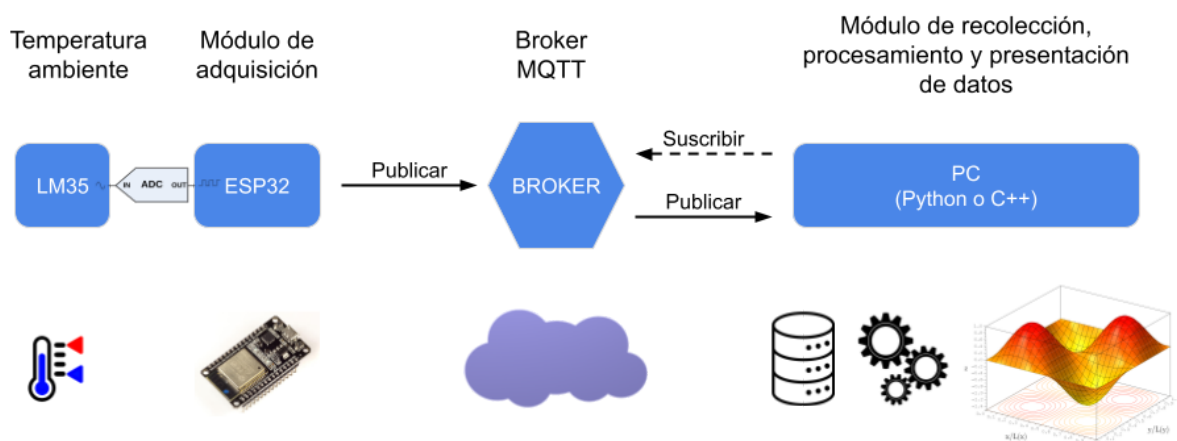
Este apartado del trabajo práctico integrador forma parte del curso de nivelación dictado por el Laboratorio de Sistemas Embebidos (LSE) de la Facultad de Ingeniería de la Universidad de Buenos Aires (FIUBA) por pedido de INVAP. El curso se lo denomina "Capacitación Semillero DISW". Las consignas siguientes en lo particular, tienen como objetivo que la/el alumno aplique los conocimientos teóricos del módulo de Python.

## Objetivos

Durante la realización de éste trabajo, se integrarán muchos de los conocimientos adquiridos en el módulo de Python, especialmente respecto de las buenas prácticas, estructura de código, documentación y testing. Se busca fomentar el trabajo en equipo y la resolución conjunta de problemas, así como también desarrollar la habilidad para exponer los resultados obtenidos luego de realizado un trabajo.

## Enunciado

Se debe desarrollar un sistema de adquisición, recolección y procesamiento de datos. En la siguiente imagen se observa un diagrama simplificado del sistema. El sistema completo se implementará a lo largo de todo el curso de nivelación.



Para el curso de Python, se desarrollarán únicamente los siguientes módulos:

- Módulo de recolección de datos: Se debe recolectar los datos de los módulos de adquisición (mediante una suscripción MQTT).
- Módulo de procesamiento de datos: Los datos recolectados se deben procesar para extraer algún tipo de información a partir de los mismos.

- Módulo de presentación de resultados: Una vez procesados los datos, se debe realizar un gráfico que permita observar los resultados: Los datos y la información extraída de los mismos.

## Módulo de recolección de datos

En el módulo de recolección de datos deben plantear una clase que permita recolectar los datos de todos los módulos mediante suscripción. Deben plantear métodos para la conexión con el broker, recolección del dato y almacenamiento. Es importante que puedan implementar algunas nociones de patrones de diseño y principios SOLID.

Para testear la conexión, pueden descargarse *mosquitto* a su sistema operativo, y desde la consola probar lo siguiente:

### Suscripción

```
$ mosquitto_sub -h 181.47.10.131 -p 1883 -t 'test/topic' -v
```

### Publicación

```
$ mosquitto_pub -h 181.47.10.131 -p 1883 -t 'test/topic' -m 'hello world'
```

Para realizar la conexión con el broker pueden utilizar la librería de Python PAHO. En la documentación, pueden encontrar referencias para suscribirse y publicar en un topic, event loops, etc.

La estructura de topics que vamos a usar es la siguiente: `'sensores/nodo_{i}'`, donde *i* es el id del sensor (único numérico). La clase de recolección de datos debe ser capaz de almacenar todos los datos de todos los sensores en un csv. Para eso se deben suscribir al topic padre (sensores). Pueden buscar documentación del protocolo MQTT para más información.

Cuando publican, siempre deben publicar usando el id de nodo. La estructura del mensaje será la siguiente:

```
message = {  
    "id": nodo_id,  
    "temp": valor_temperatura,  
    "timestamp": fecha_y_hora  
}
```

## Módulo de procesamiento de datos

En el módulo de procesamiento de datos, el objetivo es que, a partir de los datos almacenados, puedan realizar algunos análisis con Pandas y/o NumPy. En particular, queremos conocer:

- Estadísticas básicas de temperatura para cada nodo (mínimo, máximo, media, moda, desvío, etc.)
- Estadísticas básicas para todos los nodos en conjunto.
- Obtener los nodos que menos y más temperatura tienen, menos y más registros en general tienen, nodos más antiguos y más nuevos.

## Módulo de presentación de resultados

En el módulo de presentación de resultados, deben utilizar los análisis obtenidos del módulo anterior, para poder realizar gráficos de evolución de la temperatura de cada nodo, evolución del promedio de temperatura de todos los nodos, box-plots de cada nodo, usando preferentemente matplotlib o seaborn.