

Contenido

1. Antecedentes.....	1
2. Descripción de la actividad.....	1
3. Materiales requeridos.....	2
4. Desarrollo de la actividad.....	3
5. Entregables (subir a GitHub).....	3
6. Criterios de evaluación (100 pts).....	3
7. Fecha de entrega	4
8. Recomendaciones.....	4

Proyecto Final Capstone — Consumo y visualización de datos desde una API en contenedor

1. Antecedentes

El presente proyecto final integrador surge como culminación del trayecto académico en la asignatura Arquitectura y Sistemas Operativos, cuyo propósito es brindar una comprensión sólida sobre el diseño, funcionamiento e interacción de los componentes fundamentales de un sistema computacional. A lo largo del curso se han abordado temáticas clave como la administración de procesos, gestión de memoria, mecanismos de seguridad y control de recursos, redes y sistemas distribuidos, y tecnologías modernas como la virtualización y los contenedores.

En este contexto, se han explorado en profundidad herramientas actuales como Docker y Node-RED, permitiendo al estudiante experimentar con la creación, despliegue y orquestación de servicios en entornos virtualizados y distribuidos.

El trabajo propuesto sintetiza estos conocimientos en una solución concreta orientada al tratamiento de datos reales generados por dispositivos IoT. La propuesta contempla el uso de contenedores para el despliegue de servicios, la manipulación de flujos de datos en tiempo real, y la visualización de información en un entorno accesible y funcional, representando así una aplicación práctica y significativa de los contenidos teóricos y prácticos desarrollados durante la cursada.

2. Descripción de la actividad

1. Consumir la información desde el siguiente endpoint público, correspondiente a un dispositivo IoT operativo:

<https://callback-iot.up.railway.app/data>

2. Configurar un flujo en Node-RED (u otra plataforma de tu elección) que:

- Realice una consulta periódica al endpoint (ej. cada 10 segundos)
- Extraiga y visualice los dos registros más recientes usando widgets como Gauge, Chart, Text, etc.

3. Contenerización con Docker (OBLIGATORIO):

La solución completa debe estar empaquetada dentro de uno o más contenedores Docker. Se debe entregar un archivo docker-compose.yml para levantar el/los contenedores fácilmente. No se acepta ninguna solución que no use Docker para ejecutar el sistema.

4. Investigar y explicar brevemente (dentro del informe) los siguientes conceptos aplicados en el proyecto:

- ¿Qué es Docker y para qué se utiliza?
- ¿Qué es una API REST y cómo se consulta?
- ¿Qué características generales tiene un sistema IoT y qué tipo de datos genera?
- ¿Qué papel cumple una plataforma como Node-RED en este tipo de arquitectura?

5. Incluir un diagrama de bloques del sistema IoT desarrollado, donde se representen claramente:

(como vimos en clase.....)

- El origen de los datos (dispositivo IoT)
- El transporte y formato (API REST / JSON)
- El procesamiento intermedio (flujo en Node-RED u otra plataforma)
- Y la capa de visualización final

3. Materiales requeridos

- Plataforma de visualización: Node-RED (recomendado) u otra alternativa web
- Acceso a internet para consumir el endpoint
- Docker para contenerizar la solución (opcional pero recomendado)

- Navegador web para pruebas e interacción

4. Desarrollo de la actividad

1. Analizar el formato de respuesta del endpoint y revisar la estructura del JSON.

2. Crear un flujo en Node-RED que:

- Consulte el endpoint cada cierto intervalo
- Ordene los datos por fecha/hora
- Extraiga los dos más recientes
- Los visualice en tiempo real mediante widgets adecuados

3. Realizar pruebas de funcionamiento y documentar el proceso

4. Incluir en el informe:

- Una sección breve de investigación técnica sobre los temas mencionados
- Un diagrama de bloques claro del sistema completo

5. Entregables (subir a GitHub)

- Capturas del flujo en Node-RED (o plataforma usada)
- Capturas de los datos visualizados (gauge, gráfico, etc.)
- Informe técnico en PDF que incluya:
 - Ejemplo del JSON recibido
 - Descripción del flujo implementado
 - Explicaciones técnicas de Docker, APIs, IoT y Node-RED
 - Diagrama de bloques del sistema IoT completo
 - Reflexión sobre dificultades y aprendizajes
- Video breve (máx. 3 minutos) mostrando el sistema en funcionamiento
- docker-compose.yml (contenedores)

6. Criterios de evaluación (100 pts)

- Flujo funcional que consume y procesa datos correctamente – 35 pts
- Visualización clara y en tiempo real – 10 pts
- Informe técnico completo (estructura + explicaciones + diagrama) – 35 pts

- Video demostrativo – 20 pts

Promoción directa: 60 puntos o más.

7. Fecha de entrega

Viernes 1 de agosto de 2025 – 23:59 h (UTC-3, hora Buenos Aires)

8. Recomendaciones

- Usá el nodo http request en Node-RED para consumir el endpoint.
- Aplicá nodos function o sort si necesitás manipular los datos.
- Mostrá la información con widgets útiles y bien organizados.
- Si lo deseás, usá Docker para contenerizar tu solución y compartila fácilmente.
- Incluí un diagrama de bloques con nombres claros y flechas bien direccionadas.
- Citá las fuentes utilizadas para la parte de investigación si tomás definiciones externas.