Guía de Laboratorio: Control y Monitoreo con Node.js, MQTT y Gráficos en Tiempo Real

# 🎯 Objetivos

- Configurar un servidor con Node.js y Express.  
- Integrar el paquete MQTT para publicar y suscribirse a tópicos.  
- Crear un frontend responsivo en HTML con botones que publiquen mensajes MQTT.  
- Visualizar datos recibidos de un tópico MQTT en un gráfico en tiempo real estilo osciloscopio.

# 🛠️ Materiales

- PC con Node.js instalado (v14 o superior).  
- Editor de código (Visual Studio Code recomendado).  
- Paquetes npm: express, mqtt, socket.io.  
- Navegador web.  
- Un broker MQTT local o en la nube (ej. Mosquitto, HiveMQ, EMQX, etc.).

# 📌 Desarrollo

## 1. Inicializar proyecto Node.js

mkdir lab-mqtt  
cd lab-mqtt  
npm init -y  
npm install express mqtt socket.io

## 2. Crear servidor con Express y MQTT

Archivo server.js:  
- Configura un servidor Express que sirva un archivo index.html.  
- Conecta al broker MQTT.  
- Expone rutas /on y /off para que el frontend publique al tópico codigo/relay.  
- Se suscribe al tópico codigo/sensor y reenvía datos al navegador con Socket.io.

## 3. Crear carpeta public con index.html

Este archivo contiene:  
- Dos botones (ON y OFF) que hacen fetch al servidor.  
- Un campo <div> para mostrar el valor recibido del sensor.  
- Un gráfico con Chart.js en tiempo real que recibe los datos desde Socket.io.  
- Estilos CSS para hacerlo responsivo y con diseño moderno.

## 4. Integrar Chart.js

Se carga desde CDN:  
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>  
- Se configura un gráfico de tipo line.  
- Se fija el eje Y entre 0 y 1030.  
- Se limita a mostrar solo los últimos 20 puntos.

## 5. Responsividad y estilo

- Los botones tienen colores verde/rojo.  
- El gráfico ocupa el 90% de la pantalla vertical.  
- La interfaz se adapta en móviles y pantallas pequeñas.

# ▶️ Ejecución

1. Levantar el servidor:  
 node server.js  
  
2. Abrir navegador en:  
 http://localhost:3000  
  
3. Probar los botones ON y OFF → Se publican mensajes en el tópico codigo/relay.  
4. Publicar mensajes en el broker en el tópico codigo/sensor → Se visualizan en el gráfico en tiempo real.

# 📚 Resultados esperados

- Al presionar los botones se envían mensajes MQTT.  
- En el navegador se observa el valor actual del sensor en texto y en un gráfico de línea.  
- El gráfico se comporta como un osciloscopio simple, avanzando en el tiempo.

# 📝 Tarea para el estudiante

👉 Modificar el proyecto para que soporte:  
1. Cuatro botones, que publiquen en el mismo tópico codigo/relay con los siguientes valores:  
 - ON1, OFF1, ON2, OFF2.  
2. Dos gráficos independientes en la misma página, mostrando valores de:  
 - codigo/sensor1  
 - codigo/sensor2.  
  
💡 Reto adicional:  
- Hacer que cada gráfico tenga un color distinto (ej. azul y rojo).  
- Organizar los botones en una cuadrícula responsiva.