Documentación Técnica - Visualización de Datos en Tiempo Real

# 1. Resumen

Esta aplicación móvil permite la visualización en tiempo real de datos provenientes de sensores conectados a través de un broker MQTT. La solución está construida en React Native y está diseñada para monitorear variables como temperatura y humedad, así como recibir notificaciones desde dispositivos IoT.

# 2. Arquitectura del Sistema

El sistema está compuesto por tres componentes principales:  
- Dispositivos IoT (ej. sensores de temperatura y humedad)  
- Broker MQTT (intermediario para la transmisión de mensajes)  
- Aplicación móvil en React Native que consume y visualiza los datos.

El flujo de datos es: Sensor → Broker MQTT → Aplicación Móvil

# 3. Configuración del Entorno

Requisitos:  
- Node.js  
- npm o yarn  
- React Native CLI o Expo  
- Librería MQTT (paho-mqtt)  
  
Pasos:  
1. Clonar el repositorio.  
2. Instalar dependencias con `npm install` o `yarn install`.  
3. Ejecutar con `npm start` o usar Expo.

# 4. Conexión al Broker MQTT

La conexión al broker MQTT se realiza utilizando la librería `paho-mqtt`. Se configura la conexión estableciendo el host, puerto y callbacks para manejar la conexión y los mensajes entrantes. En caso de error, se reintenta la conexión o se muestra una alerta.

# 5. Suscripción a Topics

La app se suscribe a los siguientes topics:  
- sensor/temperatura  
- sistema/notificaciones  
- actuador/ventilador  
- actuador/bombillo  
  
Cada mensaje recibido es interpretado y utilizado para actualizar el estado de la interfaz.

# 6. Visualización de Datos

Los datos se visualizan en tiempo real usando el componente `LineChart` de `react-native-chart-kit`. Se mantiene un arreglo de valores que se actualiza cada vez que se recibe un nuevo dato. Los gráficos muestran la evolución de temperatura y humedad con etiquetas de tiempo.

# 7. Manejo de Notificaciones

La app también recibe mensajes en el topic `sistema/notificaciones`, los cuales son almacenados en un arreglo y mostrados como una lista de alertas o notificaciones informativas dentro de la interfaz de usuario.

# 8. Anexos

El proyecto contiene los siguientes archivos y carpetas principales:  
- `mqttClient.js`: Configura la conexión MQTT  
- `RealTimeScreen.jsx`: Pantalla principal de visualización de datos  
- `App.js`: Configuración base de la app