

Documento de Explicación: Proyecto IoT con Ionic y Angular

1. Descripción General

Este proyecto es una aplicación móvil creada con **Ionic** y **Angular** para la monitorización y control de un sistema de riego IoT. La aplicación permite a los usuarios ver una lista de sensores, consultar sus mediciones de humedad en tiempo real y accionar las electroválvulas de riego de forma remota.

El sistema completo se compone de un frontend (la app Ionic), un backend (un servidor en Node.js) y una base de datos (MySQL gestionada con XAMPP).

2. Tecnologías Utilizadas

- **Ionic**: Framework para el desarrollo de aplicaciones móviles multiplataforma usando tecnologías web.
- **Angular**: Framework que proporciona la estructura principal del frontend, gestionando componentes, servicios y navegación.
- **TypeScript**: Superconjunto de JavaScript que añade tipado estático para un código más robusto.
- **SCSS**: Preprocesador de CSS que permite usar variables, anidamiento y mixins para escribir estilos más mantenibles.
- **XAMPP**: Entorno de servidor local que incluye **MySQL** para la gestión de la base de datos.
- **Node.js**: Entorno de ejecución de JavaScript del lado del servidor, utilizado para crear la API que conecta la app con la base de datos.
- **MySQL**: Sistema de gestión de bases de datos relacional para almacenar toda la información de dispositivos, mediciones y logs de riego.

3. Guía de Instalación

Requisitos Previos

- **Node.js**: Esencial para manejar las dependencias y ejecutar tanto Ionic como el servidor backend.
- **Ionic CLI**: La interfaz de línea de comandos de Ionic para crear, construir y ejecutar la aplicación.
- **Visual Studio Code**: Editor de código recomendado.
- **XAMPP**: Para tener un servidor Apache y MySQL local.

Pasos de Instalación

1. **Clonar el Repositorio:** Descargar el código fuente desde el repositorio Git.
2. **Instalar Dependencias:** Ejecutar `npm install` en la raíz del proyecto para instalar las librerías necesarias para Ionic y Angular.
3. **Configurar la Base de Datos:**
 - Iniciar Apache y MySQL desde el panel de control de XAMPP.
 - Crear una base de datos llamada DAM en phpMyAdmin.
 - Importar el archivo `.sql` proporcionado para crear las tablas (Dispositivos, Mediciones, etc.) y cargar los datos iniciales.
4. **Ejecutar el Backend:**
 - Navegar a la carpeta del servidor (`backend/`).
 - Instalar sus dependencias con `npm install`.
 - Iniciar el servidor con `node server.js`. Este actuará como intermediario entre la app y la base de datos.
5. **Ejecutar la Aplicación:**
 - Volver a la raíz del proyecto.
 - Ejecutar `ionic serve` para lanzar la aplicación en un navegador web local. La app se conectará automáticamente al backend.

4. Estructura y Funcionalidad

Estructura del Proyecto

El código está organizado de forma modular para facilitar su mantenimiento:

- **src/app/pages/:** Cada página principal de la app (Home, Detalle, Historial) tiene su propia carpeta con su lógica (`.ts`), plantilla (`.html`) y estilos (`.scss`).
- **src/app/services/:** Contiene los servicios que manejan la lógica de negocio, como la comunicación con la API del backend.
- **src/app/components/:** Almacena componentes reutilizables, como el encabezado de la aplicación.
- **backend/:** Carpeta independiente con el código del servidor Node.js.

Funcionalidades Clave

1. **HomePage:**
 - **Descripción:** Es la pantalla de inicio. Muestra una lista de todos los dispositivos de riego registrados.
 - **Funcionalidad:** Realiza una petición a la API para obtener los dispositivos y los muestra en una lista. Cada elemento es clickable.
2. **DeviceDetailPage:**
 - **Descripción:** Muestra la información detallada de un solo dispositivo seleccionado.
 - **Funcionalidad:**
 - Muestra la última medición de humedad.
 - Tiene un botón para "Abrir/Cerrar Válvula" que envía una petición al backend para registrar la acción y obtener una nueva medición simulada.

- Permite navegar al historial de mediciones.

3. **HistorialPage:**

- **Descripción:** Presenta un registro histórico de todas las mediciones de un sensor.
- **Funcionalidad:** Obtiene y muestra una lista de todas las mediciones asociadas a un dispositivo, ordenadas por fecha.

4. **Interacción con la Base de Datos:**

- La app nunca se conecta directamente a la base de datos. Toda la comunicación se realiza a través de la API del backend.
- Cuando se acciona una válvula, la app envía una petición POST al servidor. El servidor inserta un nuevo registro en Log_Riegos y una nueva medición simulada en Mediciones, y luego devuelve la confirmación a la app.