

Propósito: Vista general jerárquica de dispositivos IoT (Gateway → Endpoints → Sensores).

Arquitectura de Datos:

```
javascript
```

```
    // Estructura jerárquica

    {

        gateways: [],

        endpointsByGateway: {},

        sensorsByEndpoint: {}

    }
```

Funcionalidades Principales:

Carga de Datos con Fallback

```
javascript
```

```
    async function loadDevicesFromDB() {

        // Primero intenta desde MariaDB

        const response = await DevicesAPI.getAllDevices();

        if (response.success) return response.data;

        // Fallback a estado del sistema MQTT

        return await loadDevicesFromSystemStatus();

    }
```

Sistema de Históricos Dual

javascript

```
async function renderHistoryForDevice(deviceId, containerNode) {  
  
    // MariaDB: datos estructurados de sensores  
  
    const mdb = await DevicesAPI.getDeviceSensorData(deviceId, 20);  
  
  
    // InfluxDB: mensajes MQTT en tiempo real  
  
    const influxData = await DevicesAPI.getHistoricalData(deviceId, 50,  
"24h");  
  
  
    // Interfaz de tabs para alternar entre fuentes  
  
}
```

Visualización Jerárquica Mejorada

javascript

```
function organizeDevicesHierarchy(devices) {  
  
    const gateways = devices.filter(d => d.tipo === 'gateway');  
  
    const endpointsByGateway = {};  
  
    const sensorsByEndpoint = {};  
  
  
    // Lógica de agrupamiento con indicadores visuales  
  
    // y manejo de dispositivos huérfanos  
  
}
```

Características Visuales:

- Indicadores de estado con colores y animaciones

- Estructura en árbol con líneas conectivas
- Cards interactivas con hover effects
- Auto-refresh cada 10 minutos
- Estadísticas del sistema en tiempo real