

Propósito: Panel principal de monitoreo en tiempo real con múltiples widgets.

Arquitectura de Widgets:

javascript

```
// Widgets principales del dashboard

const generalStatus = await generalStatusWidget();

const systemStatus = await systemStatusWidget();

const hierarchyWidget = createHierarchyWidget();

const mqttLogs = mqttLogsWidget();

const alertsWidget = alertWidget();

const deviceSelector = await deviceSelectorWidget();
```

Funcionalidades Específicas por Dispositivo:

Para Sensores Individuales

javascript

```
async function loadSensorData(device) {

    const sensorData = await
deviceService.getDeviceSensorData(device.id_dispositivo, 50);

    const deviceDetails = await
deviceService.getDeviceDetails(device.id_dispositivo);

    // Renderiza información específica del sensor
}
```

Para Endpoints (con múltiples sensores)

javascript

```
async function loadEndpointSensorData(device) {
```

```
    const endpointData = await
GatewayAPI.getEndpointById(device.id_dispositivo);

    const { endpoint, sensors } = endpointData.data;

    // Renderiza tabla con todos los sensores del endpoint
}
```

Sistema de Gráficos en Tiempo Real

javascript

```
async function updateTemperatureChart(device) {

    // Configuración dinámica basada en perfil

    const config = configService.getVisualizationConfig();

    const maxPoints = config.chartPoints || 60;

    return await temperatureChartWidget({
        title: `Temperatura - ${device.tipo} ${device.id_dispositivo}`,
        maxPoints: maxPoints,
        deviceId: device.id_dispositivo,
        deviceType: device.tipo
    });
}
```

Características Avanzadas:

- Suscripción a cambios de configuración
- Reconexión automática WebSocket
- Gestión de estado con Redux-like store
- Notificaciones del navegador

