

Propósito y Funcionalidad

Componente canvas optimizado para visualización de streams de datos en tiempo real mediante suscripción a tópicos MQTT/WebSocket.

Características Técnicas

Arquitectura de Renderizado

```
javascript

// Buffer circular eficiente

const data = [];

function pushPoint(v) {

    if (data.length >= maxPoints) data.shift();

    data.push(v);

    dirty = true; // Flag para repintado

}

// Loop con requestAnimationFrame

function loop() {

    draw();

    requestAnimationFrame(loop); // 60 FPS optimizado

}
```

Sistema de Escalas Automático

javascript

```
function draw() {

    const min = Math.min(...data);

    const max = Math.max(...data);

    const pad = (max - min) * 0.1 || 10; // 10% padding o mínimo 10

    const ymin = Math.floor(min - pad);

    const ymax = Math.ceil(max + pad);

    // Escalado de coordenadas

    const x = (i / (maxPoints - 1)) * (W - 20) + 10;

    const y = H - 20 - ((v - ymin) / (ymax - ymin)) * (H - 40);

}
```

Gestión de Suscripciones MQTT

javascript

```
const setupTopicSubscription = async () => {

    // Búsqueda de tópicos por tipo

    if (topicType) {

        const topicsByType = mqttTopicsService.getTopicsByType(topicType);

        if (topicsByType.length > 0) {

            currentTopic = topicsByType[0].nombre;

        }

    }

}
```

```

}

// Suscripción con cleanup

unsubscribe = rtClient.subscribe(currentTopic, (msg) => {

  const val = Number(msg?.payload?.value);

  if (!Number.isFinite(val)) return; // Validación

  pushPoint(val);

});

};

```

Auto-limpiado con MutationObserver

```

javascript

const observer = new MutationObserver(() => {

  if (!document.body.contains(root)) {

    if (unsubscribe) unsubscribe(); // Limpiar suscripción

    observer.disconnect(); // Limpiar observer

  }

});

observer.observe(document.body, { childList: true, subtree: true });

```

API Completa

```

javascript

chartWidget({

```

```
title: "Tiempo real",           // String - Título display

topic: "metrics/demo",         // String - Tópico MQTT por defecto

maxPoints: 120,                // Number - Capacidad buffer circular

topicType: "temperature"      // String - Filtro tipo tópico

})
```

Flujo de Datos

MQTT Broker → WebSocket Client → Message Handler →

Data Validation → Circular Buffer → RAF Loop →

Canvas Context → Visual Rendering