

## DOCUMENTACIÓN: CONTROLADOR DE DATOS DE DISPOSITIVOS

### DESCRIPCIÓN GENERAL

Este módulo actúa como controlador para la gestión y consulta de datos relacionados con dispositivos IoT del sistema. Su función principal es proporcionar una interfaz API RESTful que permite acceder a información en tiempo real, histórica y estadística de todos los dispositivos conectados, incluyendo gateways, endpoints y sensores.

### FUNCIONALIDAD PRINCIPAL

El controlador maneja las operaciones de consulta y gestión de datos de dispositivos, sirviendo como intermediario entre las solicitudes HTTP del cliente frontend y los servicios de datos que interactúan con las bases de datos. Implementa un patrón consistente para el manejo de respuestas y errores.

### CONSULTA DE DISPOSITIVOS

#### LISTADO COMPLETO DE DISPOSITIVOS

Provee un endpoint que retorna todos los dispositivos registrados en el sistema. Incluye mecanismos de robustez para asegurar que siempre se retorne un array válido, incluso en casos de error en la capa de servicio. La respuesta incluye conteo total de dispositivos para facilitar la paginación frontend.

#### INFORMACIÓN ESPECÍFICA POR DISPOSITIVO

Ofrece acceso detallado a un dispositivo individual mediante su identificador único. Combina información básica del dispositivo con estadísticas operacionales en una respuesta consolidada. Incluye validación de parámetros y manejo específico para casos de dispositivos no encontrados.

#### DATOS DE SENORES

#### LECTURAS EN TIEMPO REAL

Provee acceso a las mediciones más recientes de los sensores asociados a un dispositivo específico. Permite limitar el número de registros retornados mediante parámetros de consulta, optimizando el rendimiento para casos con grandes volúmenes de datos.

#### DATOS HISTÓRICOS

Accede a información histórica almacenada en InfluxDB, especializada en series de tiempo. Soporta diferentes rangos temporales y límites de registros, permitiendo análisis de tendencias y comportamiento histórico de los sensores.

#### MANEJO DE ERRORES

Implementa un sistema de manejo de errores estructurado que diferencia entre errores de cliente (como dispositivos no encontrados) y errores internos del servidor.

Proporciona respuestas HTTP apropiadas con códigos de estado específicos para cada escenario.

## VALIDACIÓN DE PARÁMETROS

Incluye validación básica de parámetros requeridos, asegurando que los identificadores de dispositivos estén presentes antes de proceder con las consultas. Aplica transformación de tipos de datos para parámetros opcionales como límites de registros.

## ESTRUCTURA DE RESPUESTAS

Mantiene una estructura consistente en todas las respuestas, incluyendo indicador de éxito, datos solicitados, metadatos como conteos y fuentes de datos. Esta estandarización facilita el consumo de la API por parte de aplicaciones cliente.

## INTEGRACIÓN CON SERVICIOS

Delega toda la lógica de negocio y acceso a datos a servicios especializados, manteniendo una clara separación de responsabilidades. Los controladores se enfocan exclusivamente en manejar la comunicación HTTP mientras los servicios gestionan la complejidad del acceso a datos.

## FLUJO DE DATOS

Cada solicitud sigue un patrón predecible: validación de entrada, delegación al servicio correspondiente, transformación de la respuesta, y manejo de excepciones. Este flujo asegura confiabilidad y consistencia en todas las operaciones del controlador.

## ROBUSTEZ Y TOLERANCIA A FALLOS

Incluye mecanismos para manejar casos edge como respuestas inesperadas de los servicios, asegurando que el controlador siempre retorne respuestas válidas incluso cuando los servicios subyacentes encuentren problemas.

## AUDITORÍA Y LOGGING

Registra errores y actividades importantes para facilitar el diagnóstico de problemas y el monitoreo del comportamiento del sistema, mientras mantiene respuestas genéricas al cliente para preservar la seguridad de la información.

