

## DOCUMENTACIÓN: SERVIDOR EXPRESS PRINCIPAL

### DESCRIPCIÓN GENERAL

Este módulo es el punto de entrada y configuración principal del servidor backend. Su función principal es inicializar y configurar todas las componentes del sistema: servidor Express, middleware de seguridad, rutas API, servidor HTTP, WebSockets y conexión MQTT. Actúa como el orquestador central que integra todos los subsistemas de la aplicación.

### FUNCIONALIDAD PRINCIPAL

El módulo construye y configura un servidor web completo que combina capacidades RESTful API con comunicación en tiempo real via WebSockets, proporcionando una base sólida y segura para la aplicación IoT.

### CONFIGURACIÓN DE EXPRESS

#### MIDDLEWARE DE SEGURIDAD

Aplica Helmet.js para proteger la aplicación de vulnerabilidades web comunes mediante cabeceras HTTP seguras, con configuración específica para delegar políticas de seguridad de contenido al proxy reverso.

#### LOGGING CON MORGAN

Implementa logging de solicitudes HTTP con formato adaptativo según el entorno (desarrollo/producción), proporcionando visibilidad sobre el tráfico y ayudando en el diagnóstico de problemas.

#### MANEJO DE CORS

Configura políticas de Cross-Origin Resource Sharing basadas en variables de entorno, permitiendo integración con frontends en diferentes dominios mientras mantiene control sobre los orígenes permitidos.

#### PARSERS DE DATOS

Incluye middleware para procesar cuerpos de solicitud JSON y cookies, estableciendo límites de tamaño apropiados para prevenir ataques de denegación de servicio.

#### ARQUITECTURA DE RUTAS

#### ORGANIZACIÓN MODULAR

Estructura las rutas en módulos especializados según su funcionalidad: autenticación, configuración, datos, temperatura, CO2 y gateways. Esta separación facilita el mantenimiento y escalamiento.

#### ENDPOINT DE SALUD

Provee un endpoint simple de verificación de salud para monitoreo y balanceadores de carga, posicionado estratégicamente para evitar conflictos con otras rutas.

## SERVIDOR HTTP Y WEBSOCKETS

### SERVIDOR HTTP UNIFICADO

Crea un servidor HTTP base sobre Express que sirve tanto la API RESTful como el servidor WebSocket, optimizando el uso de recursos y simplificando el despliegue.

### INTEGRACIÓN WEBSOCKET

Inicializa el servidor WebSocket en el mismo puerto HTTP, permitiendo comunicación bidireccional en tiempo real para actualizaciones de datos IoT sin necesidad de conexiones separadas.

### INICIALIZACIÓN DE SERVICIOS

### CONEXIÓN MQTT AUTOMÁTICA

Inicia la conexión con el broker MQTT inmediatamente al arrancar el servidor, asegurando que el sistema esté listo para recibir datos de dispositivos IoT desde el inicio.

### MANEJO DE ERRORES EN INICIALIZACIÓN

Captura y maneja errores durante la conexión MQTT inicial, previniendo que fallos en servicios externos impidan el arranque del servidor principal.

### CONFIGURACIÓN POR ENTORNO

### VARIABLES DE ENTORNO

Utiliza configuración cargada desde variables de entorno para todos los aspectos del servidor: puerto, niveles de log, orígenes CORS y parámetros MQTT.

### MENSAJES INFORMATIVOS DE INICIO

Provee logging detallado durante el arranque mostrando configuración crítica como puerto, broker MQTT y tópicos suscritos, facilitando la verificación de la configuración.

### INTEGRACIÓN DE COMPONENTES

El módulo actúa como el punto de integración central donde todos los subsistemas se conectan: rutas API, servicio MQTT, WebSockets y configuración de seguridad. Esta integración asegura coherencia y comunicación adecuada entre componentes.

### ESCALABILIDAD Y MANTENIBILIDAD

### ESTRUCTURA MODULAR

La organización en módulos independientes permite actualizar o reemplazar componentes específicos sin afectar el sistema completo.

#### CONFIGURACIÓN CENTRALIZADA

Toda la configuración se maneja desde un punto único, simplificando la gestión de diferentes entornos (desarrollo, testing, producción).

#### ROBUSTEZ Y TOLERANCIA A FALLOS

#### INICIALIZACIÓN GRADUAL

Los componentes se inicializan en secuencia con manejo de errores individual, permitiendo que el servidor arranque incluso si algunos servicios auxiliares tienen problemas temporales.

#### LOGGING COMPREHENSIVO

Provee información detallada durante el arranque y operación, facilitando la identificación y resolución de problemas en todos los entornos.