

## DOCUMENTACIÓN: ARCHIVO PACKAGE.JSON

### DESCRIPCIÓN GENERAL

Este archivo define la configuración fundamental del proyecto backend, incluyendo metadatos, dependencias y scripts de ejecución. Su función principal es gestionar el ecosistema de paquetes Node.js requeridos para el funcionamiento de la aplicación IoT, estableciendo las versiones compatibles y la estructura del proyecto.

### CONFIGURACIÓN DEL PROYECTO

#### METADATOS BÁSICOS

Identifica el proyecto como el backend del prototipo C, especificando la versión actual y el punto de entrada principal del servidor. Esta información es esencial para herramientas de despliegue y gestión de paquetes.

#### MÓDULOS ECMASCRIPT

Configura el proyecto para utilizar el sistema de módulos ES6 nativo de Node.js, permitiendo sintaxis de import/export moderna en lugar del CommonJS tradicional. Esta configuración mejora la compatibilidad con herramientas modernas y facilita el tree-shaking.

### DEPENDENCIAS PRINCIPALES

#### EXPRESS.JS

Framework web minimalista y flexible para Node.js que proporciona la base del servidor HTTP, manejo de rutas, middleware y gestión de solicitudes/respuestas. Es el núcleo sobre el cual se construye toda la API REST.

#### SEGURIDAD Y PROTECCIÓN

Helmet: Middleware de seguridad que protege la aplicación de vulnerabilidades web comunes mediante la configuración apropiada de cabeceras HTTP.

CORS: Habilita el intercambio de recursos entre diferentes orígenes, esencial para que el frontend se comuniquen con el backend.

Cookie Parser: Middleware para analizar y manejar cookies HTTP, utilizado potencialmente para autenticación y sesiones.

#### AUTENTICACIÓN Y AUTORIZACIÓN

JOSE: Implementación completa de estándares JWT, JWS, JWE para autenticación basada en tokens con algoritmos criptográficos modernos.

Joi: Biblioteca de validación de datos utilizada para validar variables de entorno y esquemas de entrada.

## BASES DE DATOS

MariaDB: Cliente oficial para conectarse a bases de datos MariaDB/MySQL, utilizado para almacenamiento relacional de datos de dispositivos y usuarios.

Influx: Cliente para InfluxDB, base de datos de series de tiempo optimizada para métricas y datos de sensores IoT.

## COMUNICACIÓN EN TIEMPO REAL

MQTT: Implementación del protocolo MQTT para comunicación con el broker Mosquitto, esencial para recibir datos de dispositivos IoT.

WS: Biblioteca WebSocket para Node.js que permite comunicación bidireccional en tiempo real con clientes frontend.

## UTILIDADES DE DESARROLLO

Dotenv: Carga variables de entorno desde archivos .env, facilitando la configuración por entorno.

Morgan: Middleware de logging HTTP para registrar solicitudes y respuestas, útil para desarrollo y debugging.

## SCRIPTS DE EJECUCIÓN

### INICIO DE LA APLICACIÓN

Define un script simple que inicia el servidor utilizando Node.js directamente, apuntando al archivo principal del servidor. Este enfoque minimalista evoca complejidad innecesaria en el proceso de arranque.

## DEPENDENCIAS DE DESARROLLO

### VARIABLES DE ENTORNO CRUZADAS

Incluye Cross-env para establecer variables de entorno de manera compatible entre diferentes sistemas operativos, asegurando consistencia en los entornos de desarrollo.

## ARQUITECTURA IMPLÍCITA

La selección de dependencias refleja una arquitectura moderna y escalable:

API RESTful con Express.js

Autenticación JWT con estándares seguros

Comunicación bidireccional via MQTT y WebSockets

Almacenamiento híbrido con MariaDB (datos relacionales) e InfluxDB (series de tiempo)

Seguridad por defecto con Helmet y validaciones

## GESTIÓN DE VERSIONES

Las versiones especificadas utilizan el versionado semántico, balanceando estabilidad con acceso a características modernas. Las dependencias están seleccionadas para ser compatibles entre sí y con versiones recientes de Node.js.

## AUSENCIAS NOTABLES

La configuración no incluye dependencias para testing, transpilación o construcción, sugiriendo un enfoque minimalista donde el código se ejecuta directamente en Node.js sin pasos de build intermedios.

## COMPATIBILIDAD Y REQUISITOS

El conjunto de dependencias implica requisitos específicos:

Node.js versión que soporte módulos ES6

Servidor MariaDB/MySQL para datos relacionales

Instancia InfluxDB para métricas temporales

Broker MQTT (Mosquitto) para comunicación IoT

Esta configuración proporciona una base sólida y moderna para una aplicación backend IoT, equilibrando funcionalidad, seguridad y mantenibilidad.