

## Descripción General del Proyecto

Nombre: c-prototipo-backend

Versión: 1.0.0

Lockfile Version: 3 (indica que usa npm versión 7 o superior)

Node.js Requerido: Mínimo versión 16.0.0 (basado en las dependencias y engines especificados)

Este es un archivo package-lock.json que gestiona las dependencias de un proyecto backend construido con Node.js. El lockfile asegura que todas las instalaciones futuras generen exactamente la misma estructura de árbol de dependencias.

## Arquitectura Tecnológica del Proyecto

### Módulos de Seguridad y Protección

El proyecto implementa múltiples capas de seguridad:

Helmet (versión 7.1.0): Proporciona seguridad HTTP configurando apropiadamente los headers para prevenir ataques comunes como XSS, clickjacking, entre otros.

JOSE (versión 5.8.0): Biblioteca para trabajar con JWT (JSON Web Tokens) y operaciones criptográficas, utilizada para autenticación y autorización.

CORS (versión 2.8.5): Habilita el control de acceso entre diferentes dominios, esencial para APIs consumidas desde frontends en distintos orígenes.

Cookie-parser (versión 1.4.6): Middleware para parsear y firmar cookies, útil para manejar sesiones y autenticación basada en cookies.

### Servidor Web y APIs

El núcleo del backend está construido sobre:

Express (versión 4.19.2): Framework web minimalista y flexible para Node.js que proporciona un conjunto robusto de características para aplicaciones web y APIs.

Body-parser (versión 1.20.3): Middleware para analizar los cuerpos de las solicitudes entrantes, facilitando el trabajo con datos JSON, URL-encoded y otros formatos.

Morgan (versión 1.10.0): Middleware de logging para registrar las solicitudes HTTP, útil para debugging y monitoreo.

### Sistemas de Base de Datos

El proyecto utiliza dos tipos de bases de datos:

MariaDB (versión 3.4.5): Controlador para la base de datos relacional MariaDB/MySQL, utilizado para datos estructurados y transaccionales.

Influx (versión 5.11.0): Cliente para InfluxDB, una base de datos de series temporales especializada en métricas y datos con marca de tiempo, ideal para datos de sensores o métricas en tiempo real.

### Comunicación en Tiempo Real

Para capacidades de tiempo real, el proyecto incorpora:

MQTT (versión 5.14.1): Implementación del protocolo MQTT, ampliamente utilizado en IoT (Internet de las Cosas) para comunicación ligera entre dispositivos.

WebSockets (versión 8.18.0): Implementación de WebSockets para comunicación bidireccional en tiempo real entre el servidor y los clientes.

### Utilidades y Validación

Herramientas auxiliares para desarrollo y validación:

Joi (versión 17.13.3): Biblioteca de validación de esquemas para JavaScript, utilizada para validar datos de entrada, configuraciones y respuestas de API.

Dotenv (versión 16.4.5): Carga variables de entorno desde un archivo .env a process.env, fundamental para la configuración de entornos.

Cross-env (versión 10.0.0): Herramienta de desarrollo que permite establecer variables de entorno de manera compatible entre diferentes sistemas operativos.

### Dependencias Transitorias Importantes

El lockfile gestiona 150+ dependencias transitorias, algunas notables incluyen:

Debug: Utilizado ampliamente para logging de debugging

Bytes: Utilidades para conversión y formato de bytes

Qs: Parseador de cadenas de consulta (query strings)

Accept: Negociación de contenido para tipos MIME

Type-is: Determinación del tipo de contenido

Http-errors: Creación de errores HTTP

### Caso de Uso Inferido

Basado en las dependencias, este backend parece diseñado para un sistema de monitoreo IoT o industrial que:

Recibe datos de sensores/dispositivos a través de MQTT

Almacena métricas en tiempo real en InfluxDB

Guarda datos estructurados y de configuración en MariaDB

Proporciona APIs RESTful para frontends y aplicaciones móviles

Ofrece comunicación en tiempo real via WebSockets

Implementa autenticación JWT segura

Incluye logging y monitoreo de solicitudes

#### Especificaciones Técnicas

Engine de Node.js: Requiere versión 16 o superior

Sistema de Módulos: Utiliza el sistema de módulos CommonJS y ES6

Gestión de Dependencias: npm con lockfile version 3

Características: Soporte para async/await, promesas, y características modernas de JavaScript

#### Estructura de Archivos Implícita

El lockfile sugiere una estructura de proyecto que incluye:

Configuración de entorno (.env)

Múltiples bases de datos (SQL + Time-series)

APIs RESTful

Handlers para MQTT y WebSockets

Middlewares de seguridad y validación

Sistema de logging estructurado