bambú
tech services

Prueba Técnica TL

Actualizado: 15 Dic 2025

Back End

Objetivo

Desarrollar una REST API para la gestión de tareas (To-Do App) utilizando Node.js.
El objetivo de esta prueba es evaluar tus habilidades en el diseño, desarrollo y estructuración de servicios RESTful, manejo de autenticación y buenas prácticas de desarrollo backend.

1. **Autenticación:**
    a. **`POST /v1/auth/register`**
        Registra un nuevo usuario con los siguientes campos:
        i. **`nombre`** (string, requerido)
        ii. **`email`** (string, requerido, único)
        iii. **`password`** (string, requerido)
    b. **`POST /v1/auth/login`**
        Autentica a un usuario existente y genera un token JWT válido por 24 horas.
        i. Campos requeridos: **email, password**
    c. **`GET /v1/auth/me`**
        Devuelve la información del usuario actualmente autenticado.
        El usuario será identificado a partir del **token JWT** enviado en la solicitud.

2. **Módulo de Tareas (To-Do)**
    a. **`POST /v1/todo/create`**

    Crea una nueva tarea con los siguientes campos:

        i. **`nombre`** (string, requerido)
        ii. **`prioridad`** (enum: **`baja`, `media`, `alta`**, requerido)
        iii. **`finalizada`** (boolean, por defecto en **falso**)
        iv. **`fechaCreacion`** (date, generada automáticamente)
        v. **`fechaActualizacion`** (date, generada automáticamente)

    b. **`PATCH /v1/todo/update/:id`**

    Actualiza una tarea existente.

    Campos permitidos para actualizar:

        i. **`nombre`** (opcional)
        ii. **`prioridad`** (opcional)
        iii. **`finalizada`** (opcional)

        Además, la **`fechaActualizacion`** deberá actualizarse automáticamente.

    c. **`GET /v1/todo/list`**

    Obtiene la lista de tareas del usuario autenticado.

    Debe incluir:

        i. **Paginación** (`page`, `limit`)
        ii. **Filtro opcional por prioridad, finalizada** (opcional, si deseas agregarlo)

    d. **`GET /v1/todo/list/:id`**

    Obtiene una tarea específica por su id.

    Solo se permite acceder a tareas creadas por el usuario autenticado.

    e. **`DELETE /v1/todo/list/:id`**

    Elimina una tarea por su id.

    Solo se permite eliminar tareas creadas por el mismo usuario.

Requerimientos Técnicos

- Lenguaje: **Node.js** con **Typescript**
- Framework: **Express.js** (o **Fastify**)
- Base de datos: SQL obligatoria (PostgreSQL, MySQL, etc.)
- Autenticación: **JWT**
- ORM o query builder sugerido: **Prisma, Sequelize**, o **TypeORM**
- Manejo de variables de entorno con dotenv o equivalente.
- **Versionamiento de migraciones de base de datos** (por ejemplo, usando el sistema de migraciones de **Prisma, Sequelize o TypeORM**).
- Documentación básica (**README.md** con pasos de instalación) y documentación de la API mediante **Swagger / OpenAPI**.
- **Dockerfile** y **docker-compose.yml** para levantar el proyecto de forma automatizada y dockerizada.

Criterios de Evaluación

- Diseño y estructura del proyecto (claridad, organización y escalabilidad).
- Uso correcto de códigos HTTP.
- Validación de datos de entrada.
- Seguridad (manejo de contraseñas y autenticación)

bambú
tech services

Entregables

- El código fuente deberá entregarse en un **repositorio de GitHub público.**
- En caso de que el proyecto requiera variables de entorno, se deberá incluir un archivo de ejemplo .env.example indicando las claves necesarias.
- Adicionalmente, se deberá compartir por separado el archivo .env real utilizado para ejecutar el proyecto localmente, con los valores configurados.
- Todos los entregables, incluyendo archivos y enlaces, se deben adjuntar en la **respuesta del correo electrónico.**