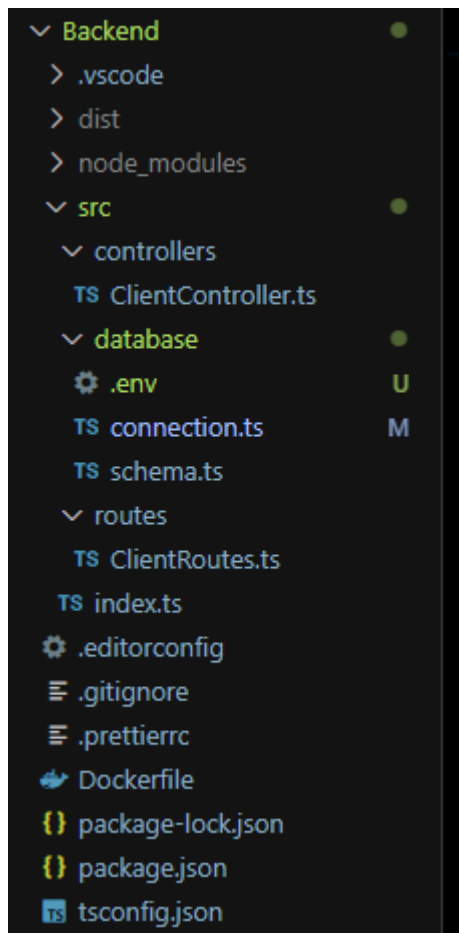


Documentación de Scaffolding de Backend

Inicialización del Proyecto

1. **Estructura del Proyecto:** Establecer la estructura básica del proyecto, incluyendo directorios y archivos necesarios.
2. **Control de versiones:** Inicializar un repositorio de Git para el control de versiones.

Directorios inicial del Backend



Gestión de Configuración

1. **Archivos de Configuración:** Crear archivos de configuración (por ejemplo, para variables de entorno).
2. **Configuración para Diferentes Entornos:** Configurar diferentes entornos (desarrollo, pruebas, producción).

Gestión de Paquetes y Dependencias

1. **Inicialización del Gestor de Paquetes:** Configurar para Node.js.
2. **Instalación de Dependencias Esenciales:** Instalar dependencias como para Node.js.

```
JavaScript
// src/package.json
{
  "name": "backend",
  "version": "1.0.0",
  "main": "index.js",
  "type": "module",
  "scripts": {
    "start": "node dist/index.js",
    "prebuild": "rimraf dist",
    "build": "tsc",
    "predev": "npm run build",
    "dev": "tsc -w & nodemon dist/index.js"
  },
  "keywords": [],
  "author": "Saul Castellanos",
  "license": "ISC",
  "description": "",
  "dependencies": {
    "drizzle-orm": "^0.32.2",
    "express": "^4.19.2",
    "mysql2": "^3.11.0"
  },
  "devDependencies": {
    "@types/express": "^4.17.21",
    "@types/node": "^22.1.0",
```

```
"nodemon": "^3.1.4",  
"prettier": "^3.3.3",  
"rimraf": "^6.0.1"  
}  
}
```

Enrutamiento Básico y Controladores

1. **Configuración de Rutas Básicas:** Establecer rutas y puntos de entrada básicos para la API.
2. **Implementación de Controladores Iniciales:** Crear controladores iniciales para manejar las solicitudes.

JavaScript

// src/routes/clientroutes.ts

```
import express from 'express';  
import ClientController from '../controllers/ClientController.js';  
import { db } from '../database/connection.js';  
  
const clientRouter = express.Router();  
const clientController = new ClientController(db);  
  
clientRouter.get('/all', clientController.getClients);  
clientRouter.get('/:id', clientController.getClient);  
clientRouter.post('/create', clientController.createClient);  
clientRouter.put('/:id', clientController.updateClient);  
  
export default clientRouter;
```

Configuración de la Base de Datos

1. **Configuración de Conexiones a la Base de Datos:** Configurar la conexión a la base de datos.
2. **Esquemas Iniciales de la Base de Datos y Migraciones:** Establecer esquemas iniciales y migraciones.

```
JavaScript
// src/database/connection.ts
import { drizzle } from 'drizzle-orm/mysql2';
import mysql from 'mysql2/promise';

const connection = await mysql.createConnection({
  host: process.env.DB_HOST || 'localhost',
  port: Number(process.env.DB_PORT) || 3307,
  user: process.env.DB_USER,
  password: process.env.DB_PASS,
  database: process.env.DB_DATABASE,
});

export const db = drizzle(connection);
```

Documentación y Pruebas

1. **Agregar Documentación Básica:** Incluir archivos README, documentación de la API.
2. **Configuración de Frameworks de Pruebas y Pruebas Básicas:** Configurar frameworks de pruebas y escribir pruebas iniciales.

Configuración de Compilación y Despliegue

1. **Configuración de Scripts de Compilación:** Configurar scripts de compilación y herramientas de automatización.

2. Configuración de Pipelines de Despliegue y CI/CD: Establecer pipelines de despliegue y configuraciones de CI/CD.

Dockerfile

```
JavaScript
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
EXPOSE 5000
CMD ["npm", "run", "start"]
```

Este proceso establece una estructura organizada y estandarizada, facilitando la colaboración entre desarrolladores y sirviendo como base sólida para el desarrollo futuro del proyecto. La estructura inicial incluye la configuración del proyecto, la gestión de dependencias, la configuración de rutas y controladores básicos, la configuración de la base de datos, la implementación de middleware y medidas de seguridad, la documentación y pruebas iniciales, y la configuración de compilación y despliegue.