## **🧪 Trabajo Práctico: API REST de Biblioteca con Prisma y PostgreSQL en Docker**

### **📁 Estructura del Proyecto**

biblioteca-api/

├── prisma/

│ └── schema.prisma

├── src/

│ ├── controllers/

│ ├── routes/

│ ├── services/

│ ├── models/

│ ├── app.ts

│ └── server.ts

├── .env

├── docker-compose.yml

├── package.json

├── tsconfig.json

└── README.md

## **🔧 Paso a paso**

### **1. Inicializar el proyecto**

npm init -y

npm install express typescript @prisma/client bcrypt dotenv

npm install ts-node-dev @types/node @types/express @types/bcrypt rimraf prisma --save-dev

npx tsc --init --outdir dist- --rootDir src

### **1.5 configurar tsc**

en el archivo tsconfig.json agregar:

“exclude”:[“node\_modules”,”dist”],

“include”:[“src”],

### **2. Crear base de datos PostgreSQL con Docker**

#### **docker-compose.yml**

version: "3.8"

services:

postgres:

image: postgres:latest

restart: always

environment:

POSTGRES\_USER: ${POSTGRES\_USER}

POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD}

POSTGRES\_DB: ${POSTGRES\_DB}

ports:

- "5432:5432"

volumes:

- ./postgres:/var/lib/postgresql/data

docker-compose up -d

### **3. Configurar Prisma**

npx prisma init

#### **.env**

DATABASE\_URL="postgresql://postgres:postgres@localhost:5432/biblioteca"

#### **prisma/schema.prisma**

generator client {

provider = "prisma-client-js"

}

datasource db {

provider = "postgresql"

url = env("DATABASE\_URL")

}

model Usuario {

id String @id @default(uuid())

nombre String

email String @unique

password String

}

### **4. Crear y migrar base de datos**

npx prisma generate

npx prisma migrate dev --name init

### **5. Crear Endpoints**

**Endpoints Requeridos**

👤 **Users**

* **GET /usuarios**: Obtener todos los usuarios.
* **GET /usuarios/:id**: Obtener un usuario por ID.
* **POST /usuarios/register**: registrar un usuario.
* **PUT /usuarios/:id**: Editar un usuario.
* **DELETE /usuarios/:id**: Eliminar un usuario.

### **6. Scripts útiles en package.json**

"scripts": {

"dev": "tsnd --respawn --clear src/server.ts",

"prisma:generate": "npx prisma generate",

"prisma:migrate": "npx prisma migrate dev"

}