Guía completa para configurar un proyecto MERN con estructura MVC

---

## 1️⃣ Estructura recomendada

```
mern-mvc-app/
│
├── client/                # App React (frontend)
│   ├── package.json       # Dependencias del frontend
│   ├── package-lock.json  # Lockfile de npm
│   └── src/
│
├── server/                # App Node/Express (backend)
│   ├── index.js           # Punto de entrada
│   ├── config/            # Configuración de la base de datos
│   ├── controllers/
│   ├── models/
│   ├── routes/
│   └── .env               # Variables de entorno del backend
│
├── package.json           # Dependencias raíz + scripts globales
├── package-lock.json      # Lockfile del backend
├── node_modules/          # Dependencias instaladas en la raíz
└── README.md
```

---

## 2️⃣ package.json raíz

```
{
  "name": "mern-mvc-app",
  "version": "1.0.0",
  "main": "server/index.js",
  "scripts": {
    "server": "nodemon server/index.js",
    "client": "npm start --prefix client",
    "dev": "concurrently \"npm run server\" \"npm run client\""
  },
  "dependencies": {
    "express": "^4.21.2",
    "mongoose": "^8.19.1",
    "bcryptjs": "^3.0.2",
    "jsonwebtoken": "^9.0.2",
    "dotenv": "^17.2.3",
```

```
    "cors": "^2.8.5"
  },
  "devDependencies": {
    "nodemon": "^3.0.1",
    "concurrently": "^8.2.2"
  }
}
```

---

## 3️⃣ package.json del client

```
{
  "name": "client",
  "version": "1.0.0",
  "private": true,
  "dependencies": {
    "axios": "^1.12.2",
    "react": "^19.2.0",
    "react-dom": "^19.2.0",
    "react-router-dom": "^7.9.4",
    "react-scripts": "5.0.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  }
}
```

---

## 4️⃣ .env del backend (/server/.env)

```
PORT=5000
MONGO_URI=mongodb+srv://usuario:password@cluster.mongodb.net/miBase
JWT_SECRET=supersecreto123
```

---

## 5️⃣ Configuración de base de datos (/server/config/db.js)

```
import mongoose from "mongoose";

const connectDB = async () => {
  try {
    const conn = await mongoose.connect(process.env.MONGO_URI, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log(`✅ MongoDB conectado: ${conn.connection.host}`);
  } catch (error) {
    console.error(`❌ Error de conexión: ${error.message}`);
    process.exit(1);
  }
};

export default connectDB;
```

---

## 6️⃣ index.js del backend (/server/index.js)

```
import express from "express";
import dotenv from "dotenv";
import connectDB from "./config/db.js";
import cors from "cors";

// Cargar variables de entorno
dotenv.config();

// Conectar a MongoDB
connectDB();

// Inicializar Express
const app = express();

// Middlewares
app.use(cors());
app.use(express.json());

// Rutas de ejemplo
app.get("/", (req, res) => {
  res.send("Servidor funcionando correctamente ✅");
});
```

```
// Puerto
const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`✅ Servidor corriendo en puerto ${PORT}`));
```

---

## 7️⃣ Instalación de dependencias

### Backend (raíz)

```
cd mern-mvc-app
npm install express mongoose dotenv cors bcryptjs jsonwebtoken
npm install --save-dev nodemon concurrently
```

### Frontend (client)

```
cd client
npm install
npm install axios react-router-dom
```

### Ejecutar ambos simultáneamente

```
cd ..
npm run dev
```

---

## 8️⃣ node_modules y package-lock.json

- **node_modules/** en la raíz → backend y scripts globales
- **node_modules/** en `/client` → frontend React
- **package-lock.json** en la raíz → bloquea versiones del backend
- **package-lock.json** en `/client` → bloquea versiones del frontend

**Gitignore recomendado:**

```
node_modules
client/node_modules
server/.env
```

---

Esta guía asegura un proyecto MERN MVC organizado y funcional.