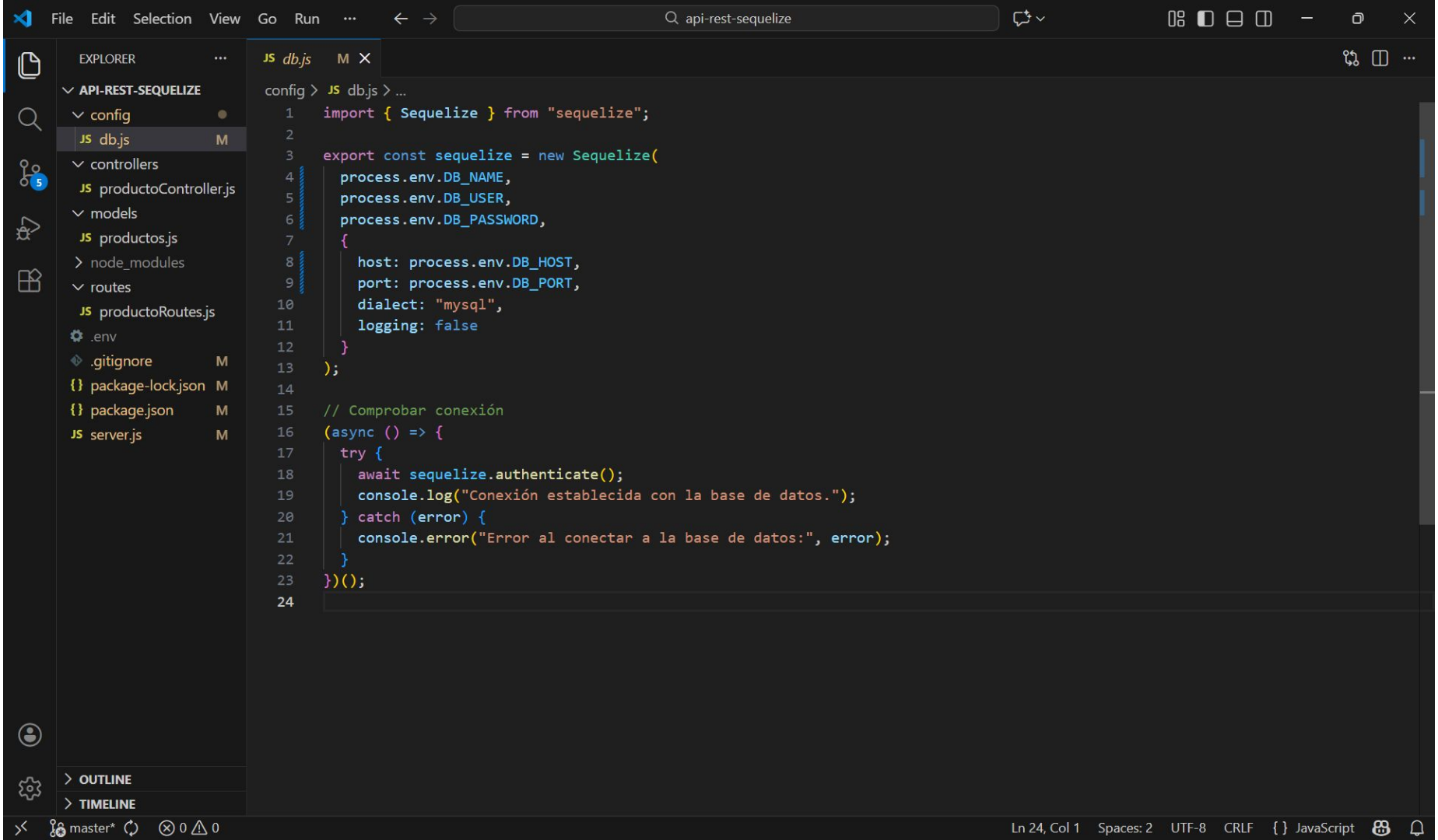
RA1-5. Taller express.js

2ºDAM

Ricardo Boza Villar

2026-01-22


File Edit Selection View Go Run ···
api-rest-sequelize
EXPLORER
API-REST-SEQUELIZE
config
db.js M
controllers
productoController.js
models
productos.js
node_modules
routes
productoRoutes.js
.env
.gitignore M
package-lock.json M
package.json M
server.js M
db.js M
config > db.js > ...
import { Sequelize } from "sequelize";
export const sequelize = new Sequelize(
process.env.DB_NAME,
process.env.DB_USER,
process.env.DB_PASSWORD,
{
host: process.env.DB_HOST,
port: process.env.DB_PORT,
dialect: "mysql",
logging: false
}
);
// Comprobar conexión
(async () => {
try {
await sequelize.authenticate();
console.log("Conexión establecida con la base de datos.");
} catch (error) {
console.error("Error al conectar a la base de datos:", error);
}
})();
OUTLINE
TIMELINE
master*
0 0
Ln 24, Col 1
Spaces: 2
UTF-8
CRLF
{ } JavaScript

File Edit Selection View Go Run ··· api-rest-sequelize

EXPLORER

API-R...

- config
  - db.js M
- controllers
  - productoController.js
- models
  - productos.js
- node_modules
- routes
  - productoRoutes.js
- .env
- .gitignore M
- package-lock.json M
- package.json M
- server.js M

OUTLINE

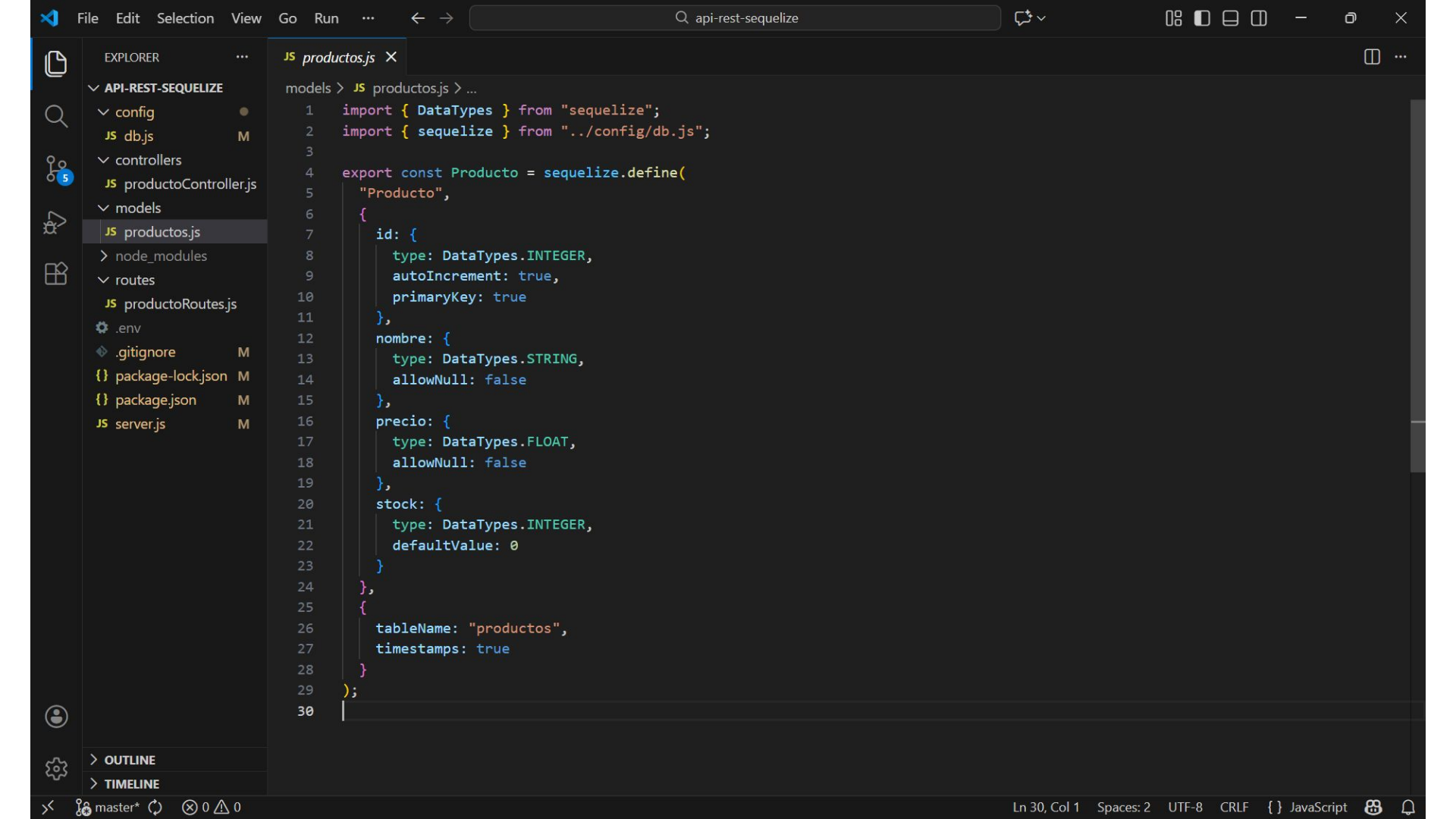
TIMELINE

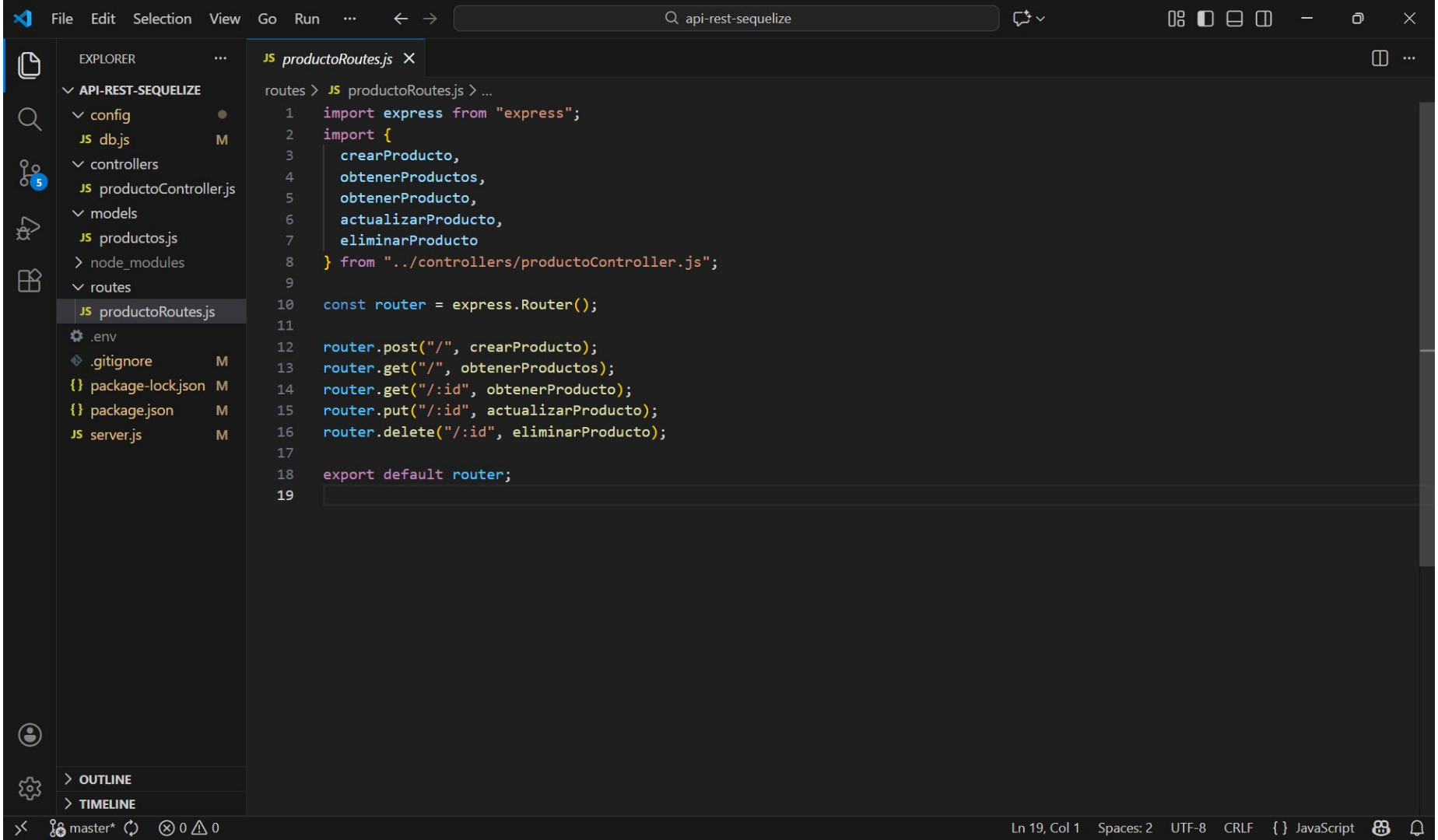
productoController.js

controllers > productoController.js > ...

```
1  import { Producto } from "../models/productos.js";
2
3  // CREATE
4  export const crearProducto = async (req, res) => {
5    try {
6      const nuevoProducto = await Producto.create(req.body);
7      res.status(201).json(nuevoProducto);
8    } catch (error) {
9      res.status(500).json({ mensaje: "Error al crear producto", error });
10   }
11 };
12
13 // READ (todos)
14 export const obtenerProductos = async (req, res) => {
15   try {
16     const productos = await Producto.findAll();
17     res.json(productos);
18   } catch (error) {
19     res.status(500).json({ mensaje: "Error al obtener productos", error });
20   }
21 };
22
23 // READ (uno)
24 export const obtenerProducto = async (req, res) => {
25   try {
26     const producto = await Producto.findByPk(req.params.id);
27     if (!producto)
28       return res.status(404).json({ mensaje: "No encontrado" });
29     res.json(producto);
30   } catch (error) {
31     res.status(500).json({ mensaje: "Error al obtener producto", error });
32   }
33 };
```

master* 0 0 Ln 1, Col 1 Spaces: 2 UTF-8 CRLF { } JavaScript


File Edit Selection View Go Run ···
api-rest-sequelize
EXPLORER
API-REST-SEQUELIZE
config
db.js M
controllers
productoController.js
models
productos.js
node_modules
routes
productoRoutes.js
.env
.gitignore M
package-lock.json M
package.json M
server.js M
productos.js
models > productos.js > ...
1 import { DataTypes } from "sequelize";
2 import { sequelize } from "../config/db.js";
3
4 export const Producto = sequelize.define(
5 "Producto",
6 {
7 id: {
8 type: DataTypes.INTEGER,
9 autoIncrement: true,
10 primaryKey: true
11 },
12 nombre: {
13 type: DataTypes.STRING,
14 allowNull: false
15 },
16 precio: {
17 type: DataTypes.FLOAT,
18 allowNull: false
19 },
20 stock: {
21 type: DataTypes.INTEGER,
22 defaultValue: 0
23 }
24 },
25 {
26 tableName: "productos",
27 timestamps: true
28 }
29);
30
OUTLINE
TIMELINE
master*
0 0
Ln 30, Col 1
Spaces: 2
UTF-8
CRLF
{ } JavaScript


File Edit Selection View Go Run ···
api-rest-sequelize
EXPLORER
API-REST-SEQUELIZE
config
db.js M
controllers
productoController.js
models
productos.js
node_modules
routes
productoRoutes.js
.env
.gitignore M
package-lock.json M
package.json M
server.js M
productoRoutes.js
routes > productoRoutes.js > ...
1 import express from "express";
2 import {
3 crearProducto,
4 obtenerProductos,
5 obtenerProducto,
6 actualizarProducto,
7 eliminarProducto
8 } from "../controllers/productoController.js";
9
10 const router = express.Router();
11
12 router.post("/", crearProducto);
13 router.get("/", obtenerProductos);
14 router.get("/:id", obtenerProducto);
15 router.put("/:id", actualizarProducto);
16 router.delete("/:id", eliminarProducto);
17
18 export default router;
19
OUTLINE
TIMELINE
master*
0 0
Ln 19, Col 1
Spaces: 2
UTF-8
CRLF
{ } JavaScript

File Edit Selection View Go Run ··· api-rest-sequelize

EXPLORER

API-REST-SEQUELIZE

- config
  - db.js M
- controllers
  - productoController.js
- models
  - productos.js
- node_modules
- routes
  - productoRoutes.js
- .env
- .gitignore M
- package-lock.json M
- package.json M
- server.js M

OUTLINE

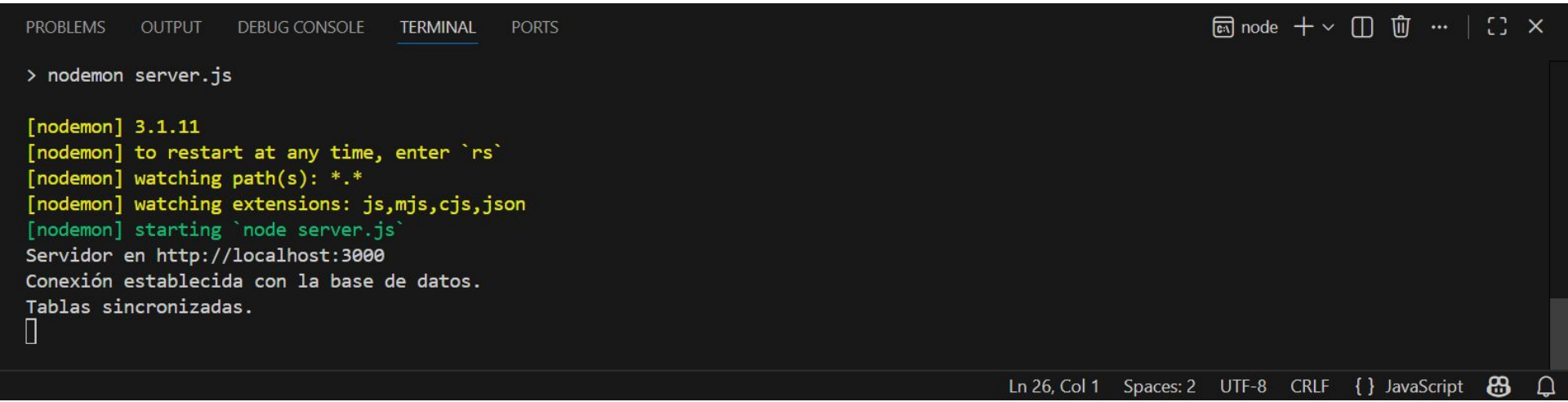
TIMELINE

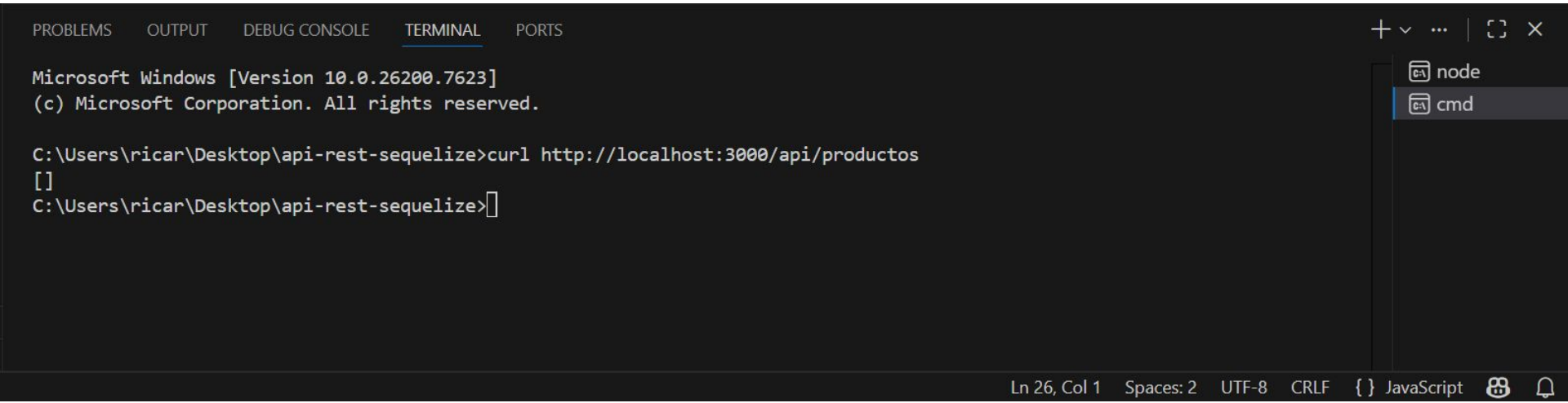
server.js M

server.js > ...

```
1  import 'dotenv/config';
2  import express from "express";
3  import productoRoutes from "./routes/productoRoutes.js";
4  import { sequelize } from "./config/db.js";
5
6  const app = express();
7  app.use(express.json());
8
9  // Rutas
10 app.use("/api/productos", productoRoutes);
11
12 // Sincronizar base de datos
13 (async () => {
14   try {
15     await sequelize.sync({ alter: true });
16     console.log("Tablas sincronizadas.");
17   } catch (error) {
18     console.error("Error al sincronizar las tablas:", error);
19   }
20 })();
21
22 const PORT = 3000;
23 app.listen(PORT, () =>
24   console.log(`Servidor en http://localhost:${PORT}`)
25 );
26
```

master* 0 0 Ln 26, Col 1 Spaces: 2 UTF-8 CRLF { } JavaScript


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
> nodemon server.js
[nodemon] 3.1.11
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
Servidor en http://localhost:3000
Conexión establecida con la base de datos.
Tablas sincronizadas.
Ln 26, Col 1
Spaces: 2
UTF-8
CRLF
{ } JavaScript


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
cmd
Microsoft Windows [Version 10.0.26200.7623]
(c) Microsoft Corporation. All rights reserved.
C:\Users\ricar\Desktop\api-rest-sequelize>curl http://localhost:3000/api/productos
[]
C:\Users\ricar\Desktop\api-rest-sequelize>
Ln 26, Col 1
Spaces: 2
UTF-8
CRLF
{ } JavaScript

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node
cmd

```
C:\Users\ricar\Desktop\api-rest-sequelize>curl -X POST http://localhost:3000/api/productos -H "Content-Type: application/json" -d "{\"nombre\":\"Producto 1\",\"precio\":10}"
{"stock":0,"id":1,"nombre":"Producto 1","precio":10,"updatedAt":"2026-01-22T12:21:07.378Z","createdAt":"2026-01-22T12:21:07.378Z"}
C:\Users\ricar\Desktop\api-rest-sequelize>
```

Ln 26, Col 1 Spaces: 2 UTF-8 CRLF { } JavaScript

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node
cmd

```
C:\Users\ricar\Desktop\api-rest-sequelize>curl http://localhost:3000/api/productos/1
{"id":1,"nombre":"Producto 1","precio":10,"stock":0,"createdAt":"2026-01-22T12:21:07.000Z","updatedAt":"2026-01-22T12:21:07
.000Z"}
C:\Users\ricar\Desktop\api-rest-sequelize>
```

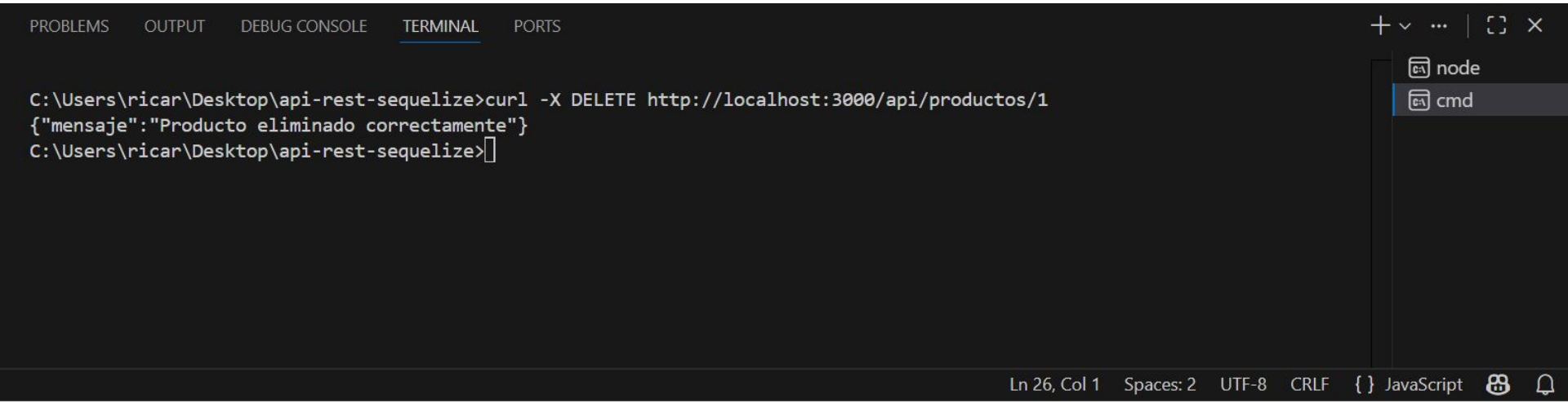
Ln 26, Col 1 Spaces: 2 UTF-8 CRLF { } JavaScript

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node
cmd

```
C:\Users\ricar\Desktop\api-rest-sequelize>curl -X PUT http://localhost:3000/api/productos/1 -H "Content-Type: application/j
son" -d "{\"nombre\":\"Producto actualizado\",\"precio\":15}"
{"id":1,"nombre":"Producto actualizado","precio":15,"stock":0,"createdAt":"2026-01-22T12:21:07.000Z","updatedAt":"2026-01-2
2T12:22:50.593Z"}
C:\Users\ricar\Desktop\api-rest-sequelize>
```

Ln 26, Col 1 Spaces: 2 UTF-8 CRLF { } JavaScript


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
cmd
C:\Users\ricar\Desktop\api-rest-sequelize>curl -X DELETE http://localhost:3000/api/productos/1
{"mensaje":"Producto eliminado correctamente"}
C:\Users\ricar\Desktop\api-rest-sequelize>
Ln 26, Col 1
Spaces: 2
UTF-8
CRLF
{ } JavaScript

2026-01-22