

Taller ORM

**PRESENTADO POR:
DARWIN STEVEN GOMEZ
DIRIGIDO A:
ALEXANDRA**

**CENTRO DE TELEINFORMATICA Y PRODUCCION INDUSTRIAL
ANALISIS Y DESARROLLO DE SOFTWARE
SENA REGIONAL CAUCA
POPAYAN CAUCA
2025**

SE ejecuta los comandos para la instalación de las librerías y la estructura

```
PS C:\Users\Sena\Desktop\examenalexandraexamen> npx sequelize-cli init
Unable to resolve sequelize package in C:\Users\Sena\Desktop\examenalexandraexamen
PS C:\Users\Sena\Desktop\examenalexandraexamen> npm init -y
wrote to C:\Users\Sena\Desktop\examenalexandraexamen\package.json:

  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen> npm install sequelize
added 21 packages, and audited 22 packages in 20s

1 package is looking for funding
  run `npm fund` for details

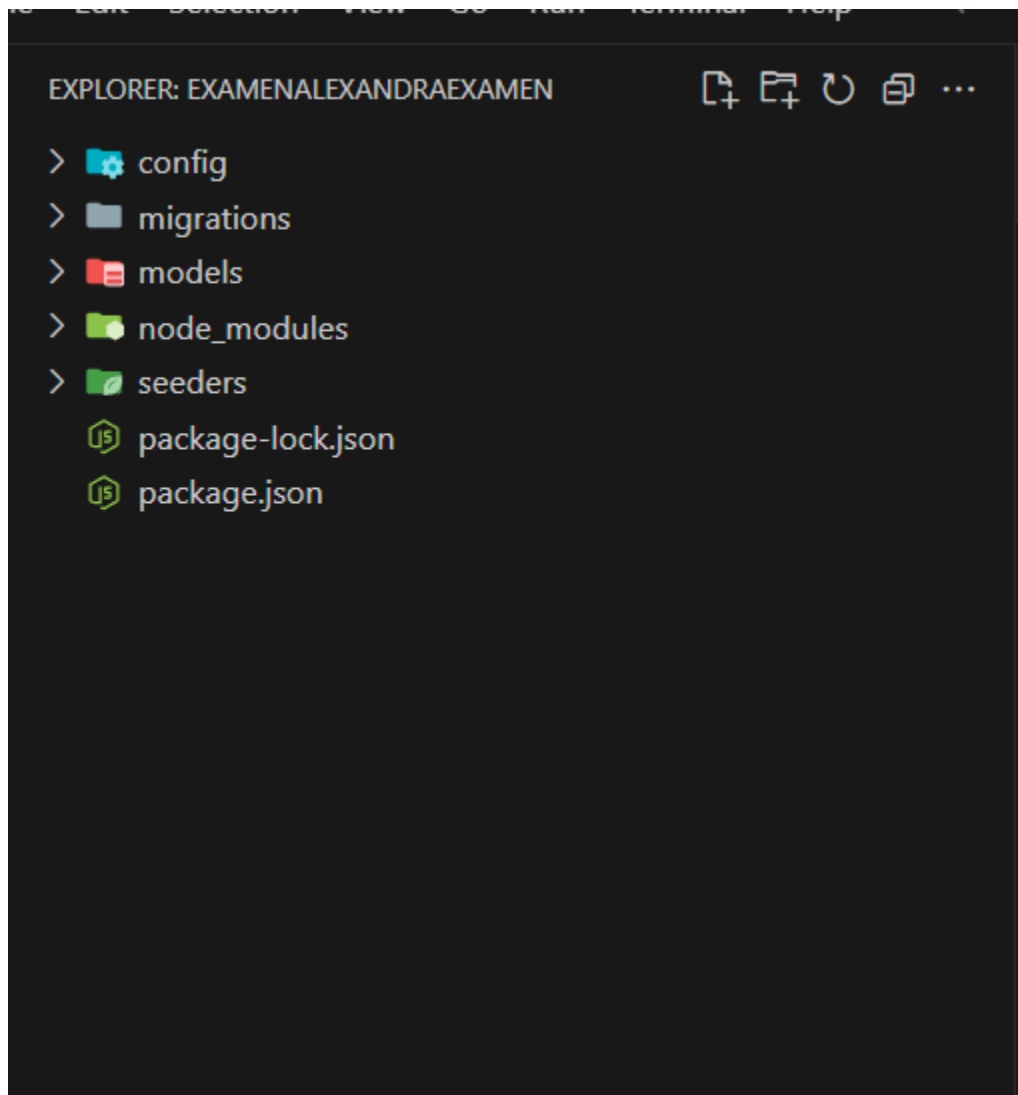
found 0 vulnerabilities
PS C:\Users\Sena\Desktop\examenalexandraexamen> npm install sequelize-cli
npm ERR! code ERR_SOCKET_TIMEOUT
npm ERR! errno ERR_SOCKET_TIMEOUT
npm ERR! network Invalid response body while trying to fetch https://registry.npmjs.org/supports-preserve-symlinks-flag: Socket timeout
npm ERR! network This is a problem related to network connectivity.
npm ERR! network In most cases you are behind a proxy or have bad network settings.
npm ERR! network
npm ERR! network If you are behind a proxy, please make sure that the
npm ERR! network 'proxy' config is set properly. See: 'npm help config'

npm ERR! A complete log of this run can be found in: C:\Users\Sena\AppData\Local\npm-cache\_logs\2025-03-03T22_37_49_352Z-debug-0.log
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen> npm install sequelize-clinpx sequelize-cli init
¿Desea terminar el trabajo por lotes (S/N)?
```

```
npm ERR! A complete log of this run can be found in: C:\Users\Sena\AppData\Local\npm-cache\_logs\2025-03-03T22_37_49_352Z-debug-0.log
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen>
PS C:\Users\Sena\Desktop\examenalexandraexamen> npm install sequelize-clinpx sequelize-cli init
¿Desea terminar el trabajo por lotes (S/N)?
^C
PS C:\Users\Sena\Desktop\examenalexandraexamen> npx sequelize-cli init

Sequelize CLI [Node: 18.17.1, CLI: 6.6.2, ORM: 6.37.5]

Created "config\config.json"
Successfully created models folder at "C:\Users\Sena\Desktop\examenalexandraexamen\models".
Successfully created migrations folder at "C:\Users\Sena\Desktop\examenalexandraexamen\migrations".
Successfully created seeders folder at "C:\Users\Sena\Desktop\examenalexandraexamen\seeders".
PS C:\Users\Sena\Desktop\examenalexandraexamen>
```

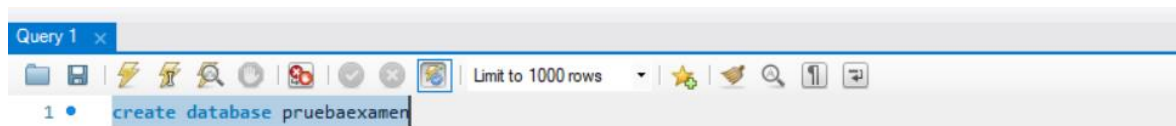


Creamos el modelo

```
PS C:\Users\Sena\Desktop\examenalexandraexamen> npx sequelize-cli model:generate --name Paciente --attributes nombre:string,apellido:string,direccion:string
Sequelize CLI [Node: 18.17.1, CLI: 6.6.2, ORM: 6.37.5]

New model was created at C:\Users\Sena\Desktop\examenalexandraexamen\models\paciente.js .
New migration was created at C:\Users\Sena\Desktop\examenalexandraexamen\migrations\20250303224120-create-paciente.js .
PS C:\Users\Sena\Desktop\examenalexandraexamen>
```

Configuramos para la base de datos



Output			
Action Output			
#	Time	Action	Message
1	17:42:38	create database pruebaexamen	1 row(s) affected

Configuramos en config

```
config.json X
config > config.json > {} production > database
1 {
2   "development": {
3     "username": "root",
4     "password": null,
5     "database": "pruebaexamen",
6     "host": "127.0.0.1",
7     "dialect": "mysql"
8   },
9   "test": {
10    "username": "root",
11    "password": null,
12    "database": "pruebaexamen",
13    "host": "127.0.0.1",
14    "dialect": "mysql"
15  },
16  "production": {
17    "username": "root",
18    "password": null,
19    "database": "pruebaexamen",
20    "host": "127.0.0.1",
21    "dialect": "mysql"
22  }
23 }
24
```

Creamos el servidor con la api de pacientes

```
examenalexandraexamen
config.json  pacientesServices.js  server.js  X
server.js > ...
1  const express = require('express');
2  const pacienteRoute = require('./router/pacienteRoute');
3  const dotenv = require('dotenv');
4
5  dotenv.config();
6
7  const app = express();
8  app.use(express.json());
9
10 app.use("/api", pacienteRoute);
11
12 const PORT = process.env.PORT || 5000;
13 app.listen(PORT, () => {
14   console.log(`Servidor corriendo en http://localhost:${PORT}`);
15 });
16
```

Definimos el controlador:

Creamos la clase

```
pacienteController.js  PacienteController  eliminar paciente  error
const PacienteService = require("../services/pacientesServices");

class PacienteController {
  static async listarPacientes(req, res) {
    try {
      let lista = await PacienteService.obtenerPacientes();
      res.json(lista);
    } catch (e) {
      res.status(500).json({ error: "Error al obtener pacientes" });
    }
  }
}
```

Crear paciente

```

static async crearPaciente(req, res) {
  try {
    const { nombre, apellido, direccion } = req.body;
    let paciente = await PacienteService.crearPaciente(nombre, apellido, direccion);
    res.json(paciente);
  } catch (e) {
    res.status(500).json({ error: "Error al crear paciente" });
  }
}

```

Eliminar paciente

```

static async eliminarPaciente(req, res) {
  try {
    const { id } = req.params;
    if (isNaN(id)) {
      return res.status(400).json({ error: "id inválido" });
    }
    let resultado = await PacienteService.eliminarPaciente(id);
    if (!resultado) {
      return res.status(404).json({ error: "Paciente no hay " });
    }
    res.json({ mensaje: "Paciente se elimino correctamente" });
  } catch (e) {
    res.status(500).json({ error: "Error al eliminar " });
  }
}

```

Actualizar paciente

```

static async actualizarPaciente(req, res) {
  try {
    const { id } = req.params;
    const { nombre, apellido, direccion } = req.body;
    if (isNaN(id)) {
      return res.status(400).json({ error: "id inválido" });
    }
    let resultado = await PacienteService.actualizarPaciente(id, { nombre, apellido, direccion });
    if (!resultado[0]) {
      return res.status(404).json({ error: "Paciente no hay" });
    }
    res.json({ mensaje: "Paciente actualizado correctamente" });
  } catch (e) {
    res.status(500).json({ error: "Error al actualizar paciente" });
  }
}

```

Obtener los pacientes

```
static async obtenerPacientePorId(req, res) {
  try {
    const { id } = req.params;
    if (isNaN(id)) {
      return res.status(400).json({ error: "id invalido" });
    }
    let paciente = await PacienteService.buscarPaciente(id);
    if (!paciente) {
      return res.status(404).json({ error: "Paciente no hay" });
    }
    res.json(paciente);
  } catch (e) {
    res.status(500).json({ error: "Error al obtener paciente" });
  }
}
```

Ahora creamos las ruta

Importamos el controlador la clase y los métodos del controlador

```
pacienteRoute.js X
routes > JS pacienteRoute.js > ...
1  const express = require('express');
2  const PacienteController = require('../controllers/pacienteControllers');
3  const router = express.Router();
4
5  router.get('/pacientes', PacienteController.listarPacientes);
6  router.post('/pacientes', PacienteController.crearPaciente);
7  router.delete('/pacientes/:id', PacienteController.eliminarPaciente);
8  router.put('/pacientes/:id', PacienteController.actualizarPaciente);
9  router.get('/pacientes/:id', PacienteController.obtenerPacientePorId);
10
11  module.exports = router;
```

Ahora creamos el servicio para pacientes

Creamos la clase pacientes

```
const { Paciente } = require('../models');

class PacienteService {
  static async obtenerPacientes() {
    try {
      return await Paciente.findAll();
    } catch (e) {
      console.log("Error al obtener pacientes:", e);
    }
  }
}
```

Crear paciente

```
static async crearPaciente(nombre, apellido, direccion) {
  try {
    return await Paciente.create({ nombre, apellido, direccion });
  } catch (e) {
    console.log("Error al crear el paciente:", e);
  }
}
```

Eliminar paciente

```
static async eliminarPaciente(id) {
  try {
    let paciente = await Paciente.findByPk(id);
    if (paciente) {
      await paciente.destroy();
      return true;
    }
    return false;
  } catch (e) {
    console.log("Error al eliminar el paciente :", e);
  }
}
```


Actualizar paciente

```
static async actualizarPaciente(id, datos) {  
  try {  
    let resultado = await Paciente.update(datos, { where: { id } });  
    return resultado;  
  } catch (e) {  
    console.log("Error al actualizar el paciente:", e);  
  }  
}
```

Buscar paciente

```
static async buscarPaciente(id) {  
  try {  
    return await Paciente.findByPk(id);  
  } catch (e) {  
    console.log("Error al buscar el paciente:", e);  
  }  
}  
}
```

Hacemos la migración con el comando

`npx sequelize-cli db:migrate`

```
PS C:\Users\Sena\Desktop\examenalexandraexamen> npx sequelize-cli db:migrate  
  
Sequelize CLI [Node: 18.17.1, CLI: 6.6.2, ORM: 6.37.5]  
  
Loaded configuration file "config\config.json".  
Using environment "development".  
  
ERROR: Please install mysql2 package manually  
  
PS C:\Users\Sena\Desktop\examenalexandraexamen> |
```

```
config.json X
config > {} config.json > ...
1  {
2    "development": {
3      "username": "root",
4      "password": null,
5      "database": "database_development",
6      "host": "127.0.0.1",
7      "dialect": "mysql"
8    },
9    "test": {
10     "username": "root",
11     "password": null,
12     "database": "database_test",
13     "host": "127.0.0.1",
14     "dialect": "mysql"
15   },
16   "production": {
17     "username": "root",
18     "password": null,
19     "database": "database_production",
20     "host": "127.0.0.1",
21     "dialect": "mysql"
22   }
23 }
24
```

Creamos la base de datos

```
PS C:\Users\Sena\Desktop\examenalexandraexamen> npx sequelize-cli db:migrate

Sequelize CLI [Node: 18.17.1, CLI: 6.6.2, ORM: 6.37.5]

Loaded configuration file "config\config.json".
Using environment "development".

ERROR: Please install mysql2 package manually

PS C:\Users\Sena\Desktop\examenalexandraexamen> 
```

Hay queda la estructura

