



**PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL
FORMATO ENTREGA DE EVIDENCIAS**

**CREACIÓN DE BASES DE DATOS
NOSQL**

Presentado a: Alexandra muñoz
Por Aprendiz: David Santiago viveros mera
Ficha: 2874057
Competencia:
Resultado de Aprendizaje:

Tecnólogo en Análisis y Desarrollo de Software
Servicio Nacional de Aprendizaje SENA
Centro de Teleinformática y Producción Industrial
Regional Cauca

Popayán, día 19 de febrero del año 2024



**PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL
FORMATO ENTREGA DE EVIDENCIAS**

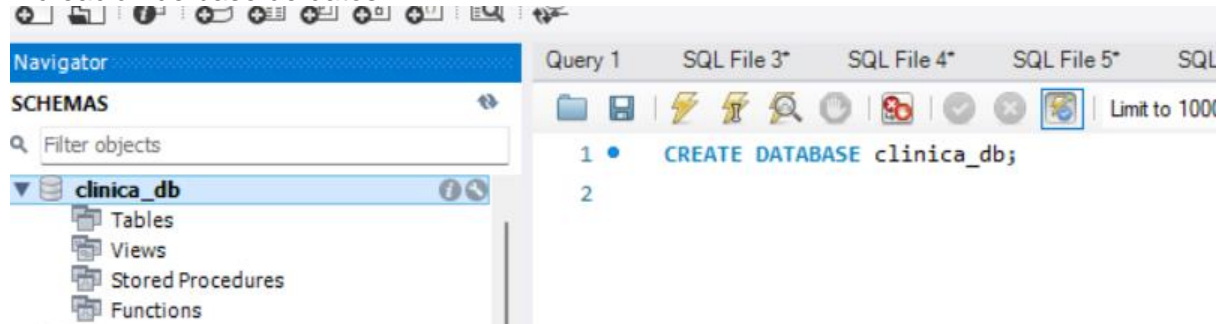
Tabla de Contenido



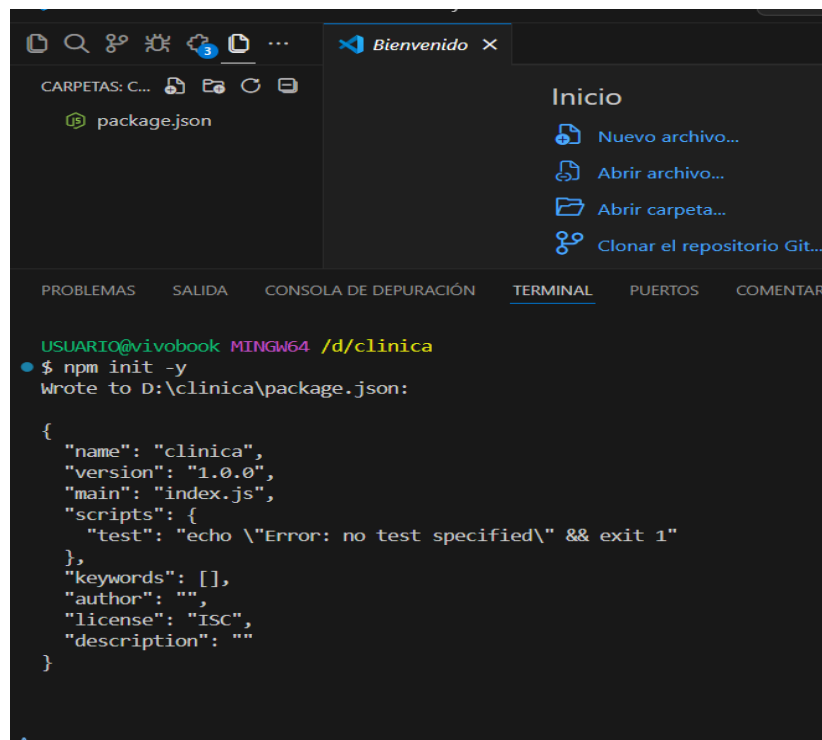
PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Configuración del proyecto

1 creación de base de datos

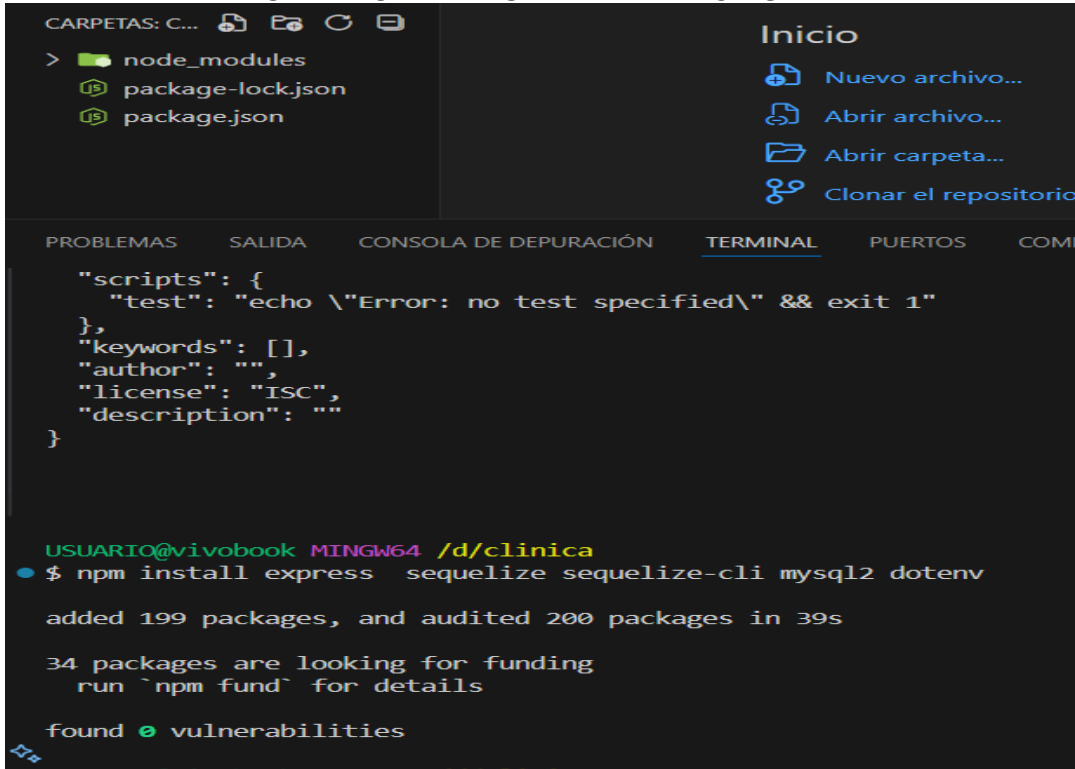


Configuración del proyecto





PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS



```

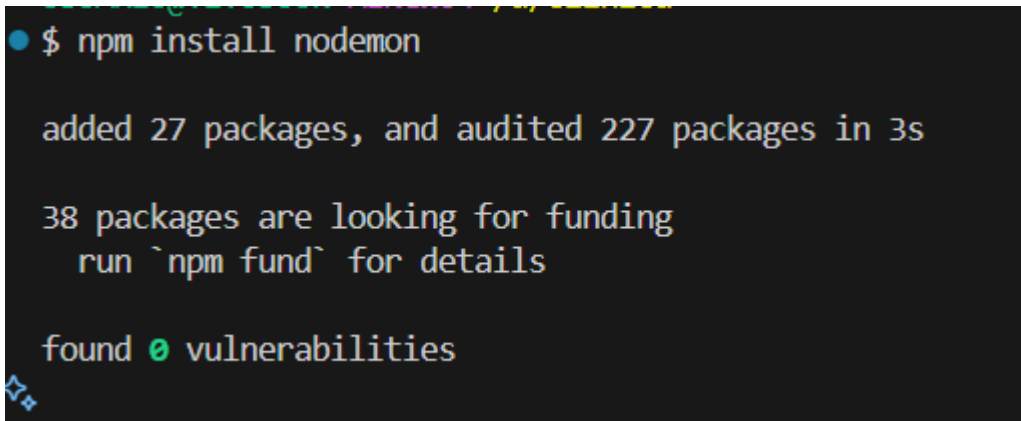
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC",
"description": ""
}

USUARIO@vivobook MINGW64 /d/clinica
● $ npm install express sequelize sequelize-cli mysql2 dotenv

added 199 packages, and audited 200 packages in 39s

34 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



```

● $ npm install nodemon

added 27 packages, and audited 227 packages in 3s

38 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
CARPETAS: C:\...
> config
> migrations
> models
> node_modules
> seeders
package-lock.json
package.json

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS COMENTARIOS

found 0 vulnerabilities

USUARIO@vivobook MINGW64 /d/clinica
$ npx sequelize-cli init

Sequelize CLI [Node: 22.14.0, CLI: 6.6.2, ORM: 6.37.5]

Created "config\config.json"
Successfully created models folder at "D:\clinica\models".
Successfully created migrations folder at "D:\clinica\migrations".
Successfully created seeders folder at "D:\clinica\seeders".

USUARIO@vivobook MINGW64 /d/clinica

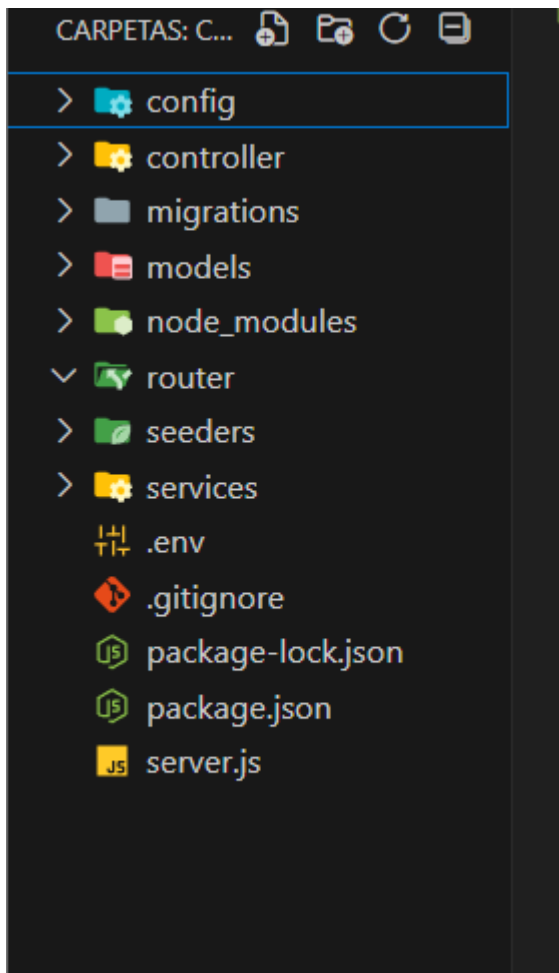
package.json > ...
1 {
2   "name": "dd",
3   "version": "1.0.0",
4   "main": "server.js",
5   "scripts": {
6     "start": "nodemon index.js",
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "description": "",
13  "dependencies": {
14    "dotenv": "^16.4.7",
15    "express": "^4.21.2",
16    "mysql2": "^3.12.0",
17    "nodemon": "^3.1.9",
18    "sequelize": "^6.37.5",
19    "sequelize-cli": "^6.6.2"
20  }
21 }
22
```



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
1 {
2   "development": {
3     "username": "root",
4     "password": "1234",
5     "database": "clinica_db",
6     "host": "127.0.0.1",
7     "dialect": "mysql"
8   },
9   "test": {
10    "username": "root",
11    "password": "1234",
12    "database": "clinica_db",
13    "host": "127.0.0.1",
14    "dialect": "mysql"
15  },
16  "production": {
17    "username": "root",
18    "password": "1234",
19    "database": "clinica_db",
20    "host": "127.0.0.1",
21    "dialect": "mysql"
22  }
23 }
```

Hacemos la configuración del proyecto creando las carpetas services,router,controller y los archivos .env, gitignore,server.js





PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

```
> config
> controller
migrations
  20250303225808-create-paci...
models
  index.js
  paciente.js
node_modules
router
seeders
services
.env
.gitignore
package-lock.json
package.json
server.js

1 'use strict';
2 const {
3   Model
4 } = require('sequelize');
5 module.exports = (sequelize, DataTypes) => {
6   class Paciente extends Model {
7     /**
8      * Helper method for defining associations.
9      * This method is not a part of Sequelize lifecycle.
10      * The `models/index` file will call this method automatically.
11      */
12     static associate(models) {
13       // define association here
14     }
15   }
16   Paciente.init({
17     nombre: DataTypes.STRING,
18     apellido: DataTypes.STRING,
19     direccion: DataTypes.STRING
20   }, {
21     sequelize,
22     modelName: 'Paciente',
23   });
24   return Paciente;
25 };

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS COMENTARIOS
bash + ~

USUARIO@vivobook MINGW64 /d/clinica
$ npx sequelize-cli model:generate --name Paciente --attributes nombre:string,apellido:string,direccion:string

Sequelize CLI [Node: 22.14.0, CLI: 6.6.2, ORM: 6.37.5]

New model was created at D:\clinica\models\paciente.js .
New migration was created at D:\clinica\migrations\20250303225808-create-paciente.js .

USUARIO@vivobook MINGW64 /d/clinica
$

USUARIO@vivobook MINGW64 /d/clinica
$ npx sequelize-cli db:migrate

Sequelize CLI [Node: 22.14.0, CLI: 6.6.2, ORM: 6.37.5]

Loaded configuration file "config\config.json".
Using environment "development".
== 20250303225808-create-paciente: migrating =====
== 20250303225808-create-paciente: migrated (0.033s)

USUARIO@vivobook MINGW64 /d/clinica
$
```



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Paciente controller

```

CARPETAS: CLINICA
├── config
│   └── config.json
├── controller
│   └── pacienteController.js M
├── migrations
│   └── 20250303225808-create-pacient...
├── models
│   ├── index.js M
│   └── paciente.js M
├── node_modules
├── router
│   ├── pacienteRoutes.js
│   └── seeders
├── services
│   └── pacienteServices.js M
├── .env M
├── .gitignore
├── package-lock.json
├── package.json
└── server.js M

controller > pacienteController.js > pacienteController > eliminarpacientes
1  const { pacientes } = require('../models');
2  const pacienteServices = require('../services/pacienteServices');
3
4  class pacienteController {
5      Tabnine | Edit | Test | Explain | Document
6      static async listarpacientes(req, res) {
7          try {
8              const pacientes = await pacienteServices.listarpacientes();
9              res.json(pacientes);
10         } catch (error) {
11             console.log(error);
12         }
13     }
14     Tabnine | Edit | Test | Explain | Document
15     static async createPaciente(req, res) {
16         try {
17             const nuevoPaciente = await pacienteServices.crearpacientes(req.body);
18             res.json(nuevoPaciente);
19         } catch (error) {
20             console.log(error);
21         }
22     }
23     Tabnine | Edit | Test | Explain | Document
24     static async actualizarpacientes(req, res) {
25         try {
26             const { id } = req.params;
27             const pacienteActualizado = await pacienteServices.actualizarpacientes(id, req.body);
28             res.json(pacienteActualizado);
29         } catch (error) {
30             console.log(error);
31         }
32     }
33     Tabnine | Edit | Test | Explain | Document
34     static async eliminarpacientes(req, res) {
35         try {
36             const { id } = req.params;
37             await pacienteServices.eliminarpacientes(id);
38             res.json({ message: 'Paciente eliminado correctamente' });
39         } catch (error) {
40             console.log(error);
41         }
42     }
43 }
44
45 module.exports = pacienteController

```

Paciente modelo

```

CARPETAS: CLINICA
├── config
│   └── config.json
├── controller
│   └── pacienteController.js M
├── migrations
│   └── 20250303225808-create-pacient...
├── models
│   ├── index.js M
│   └── paciente.js M
├── node_modules
├── router
│   ├── pacienteRoutes.js
│   └── seeders
├── services
│   └── pacienteServices.js M
├── .env M
├── .gitignore
├── package-lock.json
├── package.json
└── server.js M

models > paciente.js > ...
1
2  'use strict';
3  const { Model } = require('sequelize');
4
5  Tabnine | Edit | Test | Explain | Document
6  module.exports = (sequelize, DataTypes) => {
7      class Paciente extends Model {
8          /**
9           * Helper method for defining associations.
10          * This method is not a part of Sequelize lifecycle.
11          * The `models/index` file will call this method automatically.
12          */
13          static associate(models) {
14              // define association here
15          }
16      }
17
18      Paciente.init({
19          nombre: DataTypes.STRING,
20          apellido: DataTypes.STRING,
21          direccion: DataTypes.STRING
22      }, {
23          sequelize,
24          modelName: 'Pacientes',
25      });
26
27      return Paciente;
28  };

```




PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Paciente router

```
router > pacienteRoutes.js > ...
1  const express = require('express');
2  const pacienteController = require('../controller/pacienteController');
3  const router = express.Router();
4
5
6  router.get('/', pacienteController.listarpacientes);
7  router.post('/', pacienteController.createPaciente);
8  router.put('/:id', pacienteController.actualizarpacientes);
9  router.delete('/:id', pacienteController.eliminarpacientes);
10
11 module.exports = router;
12
```



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

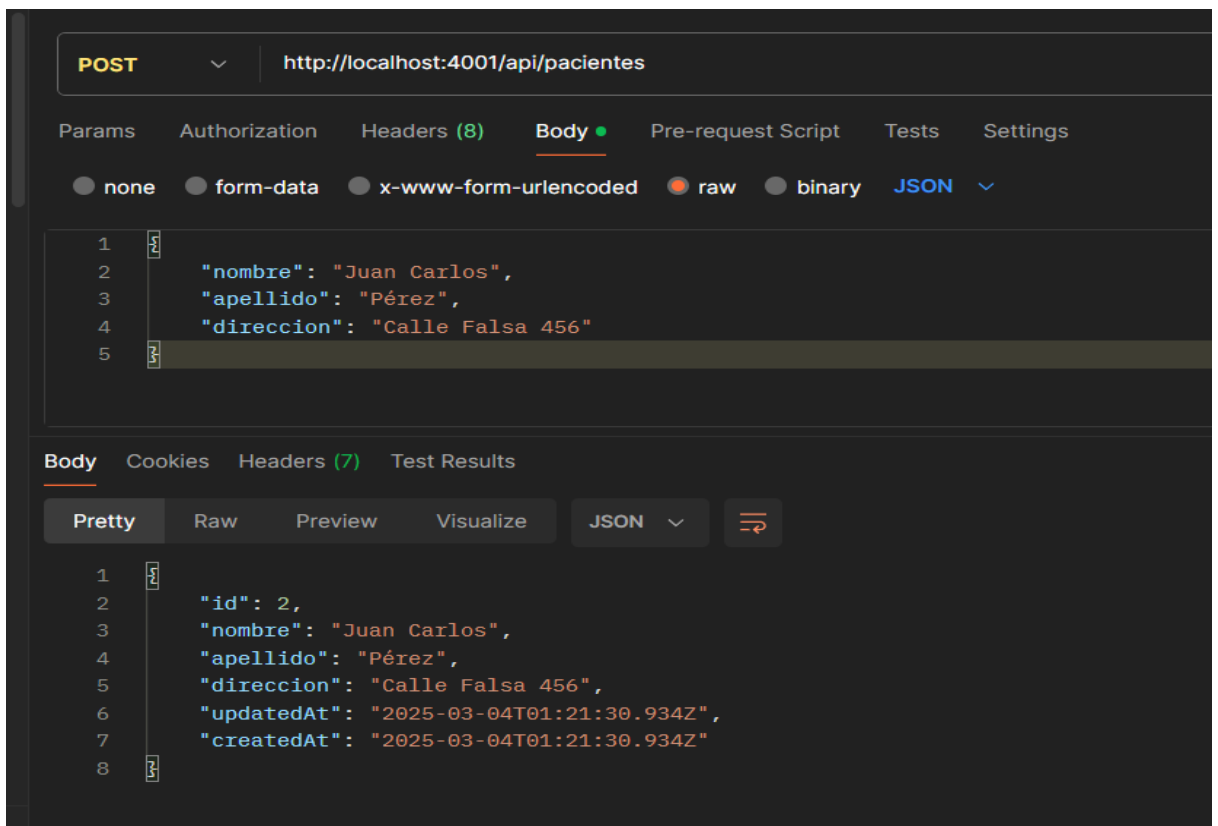
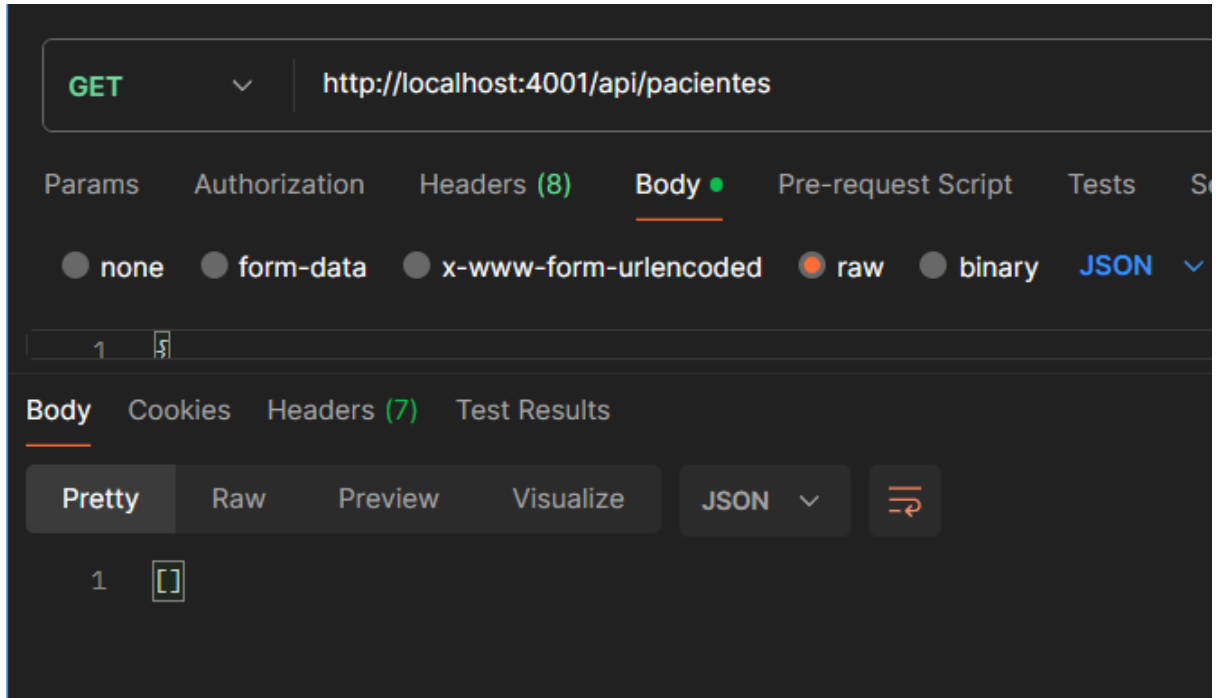
Paciente services

```
services > pacienteServices.js > pacienteServices > listarpacientes
1  const { Paciente } = require("../models");
2
3  class pacienteServices {
4      Tabnine | Edit | Test | Explain | Document
5      static async listarpacientes() {
6          try {
7              return await Paciente.findAll();
8          } catch (error) {
9              console.log(error);
10             throw error;
11         }
12     }
13
14     Tabnine | Edit | Test | Explain | Document
15     static async crearpacientes(data) {
16         try {
17             return await Paciente.create(data);
18         } catch (error) {
19             console.log(error);
20             throw error;
21         }
22     }
23
24     Tabnine | Edit | Test | Explain | Document
25     static async actualizarpacientes(id, data) {
26         try {
27             const pacienteEN = await Paciente.findByPk(id);
28             if (pacienteEN) {
29                 return await pacienteEN.update(data);
30             } else {
31                 throw new Error('Paciente not found');
32             }
33         } catch (error) {
34             console.log(error);
35             throw error;
36         }
37     }
38
39     Tabnine | Edit | Test | Explain | Document
40     static async eliminarpacientes(id) {
41         try {
42             const paciente = await Paciente.findByPk(id);
43             if (paciente) {
44                 return await paciente.destroy();
45             } else {
46                 throw new Error('Paciente not found');
47             }
48         } catch (error) {
49             console.log(error);
50             throw error;
51         }
52     }
53
54     module.exports = pacienteServices;
```



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Pruebas en postman





PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

PUT <http://localhost:4001/api/pacientes/2>

Params Authorization Headers (8) **Body** Pre-request Script Tests

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON**

```
1  {
2    "nombre": "Juan Carlos",
3    "apellido": "Pérez",
4    "direccion": "Calle correte 123"
5  }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize **JSON**

```
1  {
2    "id": 2,
3    "nombre": "Juan Carlos",
4    "apellido": "Pérez",
5    "direccion": "Calle correte 123",
6    "createdAt": "2025-03-04T01:21:30.000Z",
7    "updatedAt": "2025-03-04T01:22:08.617Z"
8  }
```

DELETE <http://localhost:4001/api/pacientes/2>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary **JSON**

```
1  {
2    "nombre": "Juan Carlos",
3    "apellido": "Pérez",
4    "direccion": "Calle correte 123"
5  }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize **JSON**

```
1  {
2    "message": "Paciente eliminado correctamente"
3  }
```



PROCESO DE GESTIÓN DE LA FORMACIÓN PROFESIONAL INTEGRAL FORMATO ENTREGA DE EVIDENCIAS

Base de datos de Mysql workbench

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'clinica_db' database with a table named 'pacientes'. The table's columns are listed: id (int AI PK), nombre (varchar(255)), apellido (varchar(255)), direccion (varchar(255)), createdAt (datetime), and updatedAt (datetime). The 'Information' pane below shows the table's structure.

The main editor shows a SQL query in 'Query 1':

```
1 • CREATE DATABASE clinica_db;
2 • select*from pacientes;
```

The 'Result Grid' shows the query results:

	id	nombre	apellido	direccion	createdAt	updatedAt
2	Juan C...	Pérez	Calle Falsa 456	2025-03-04 01:21:30	2025-03-04 01:21:30	

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	20:15:50	select*from pacientes LIMIT 0, 1000	1 row(s) ret