ADSO
(Análisis y Desarrollo de Software)

Servicio Nacional de Aprendizaje

TABLA DE **CONTENIDO**

NODE.JS

Servicio Nacional de Aprendizaje

NODE.JS - SQLite

Manejo de SQLite con Node.js

Servicio Nacional de Aprendizaje

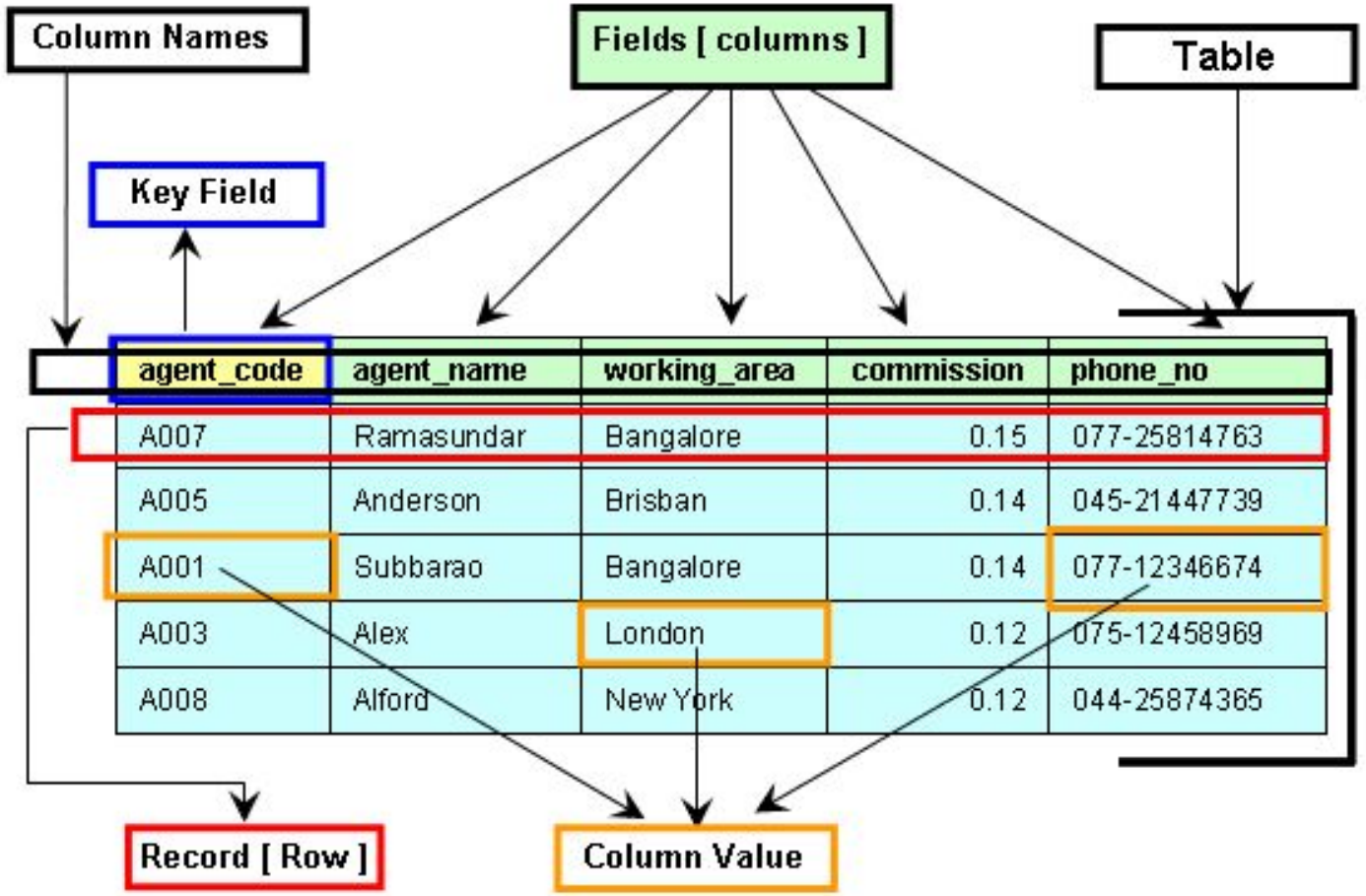
NODE.JS - SQLite

Manejo de SQLite con Node.js

Comenzaremos con una descripción general de SQLite y Node.js, luego aprenderemos a instalar los módulos necesarios y conectarnos a una base de datos SQLite. Finalmente, crearemos un ejemplo de aplicación CRUD (Crear, Leer, Actualizar, Eliminar) usando SQLite y Node.js.

NODE.JS - SQLite

¿Qué es SQLite?



NODE.JS - SQLite

¿Qué es SQLite?

- Una base de datos relacional liviana y autónoma
- No requiere un servidor separado para ejecutarse
- Almacena datos en un archivo de base de datos único
- Fácil de usar y configurar
- Ideal para aplicaciones móviles y web

NODE.JS - SQLite

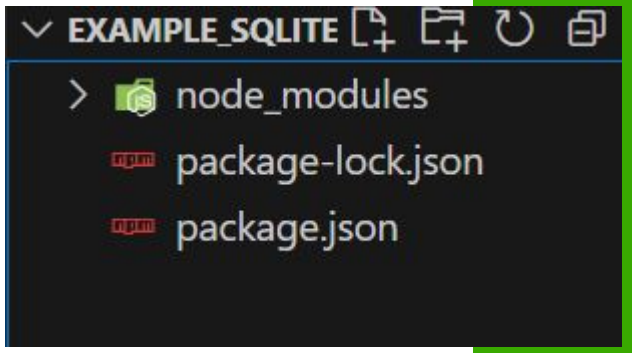
Instalación de módulos

- Creamos una carpeta llamada EXAMPLE_SQlite
- Acceder a la carpeta EXAMPLE_SQlite
- Para usar SQLite con Node.js, primero debemos instalar el módulo sqlite3.
- Podemos hacer esto usando el siguiente comando en la terminal: npm install sqlite3

NODE.JS - SQLite

Instalación de módulos

- Validamos los módulos creados

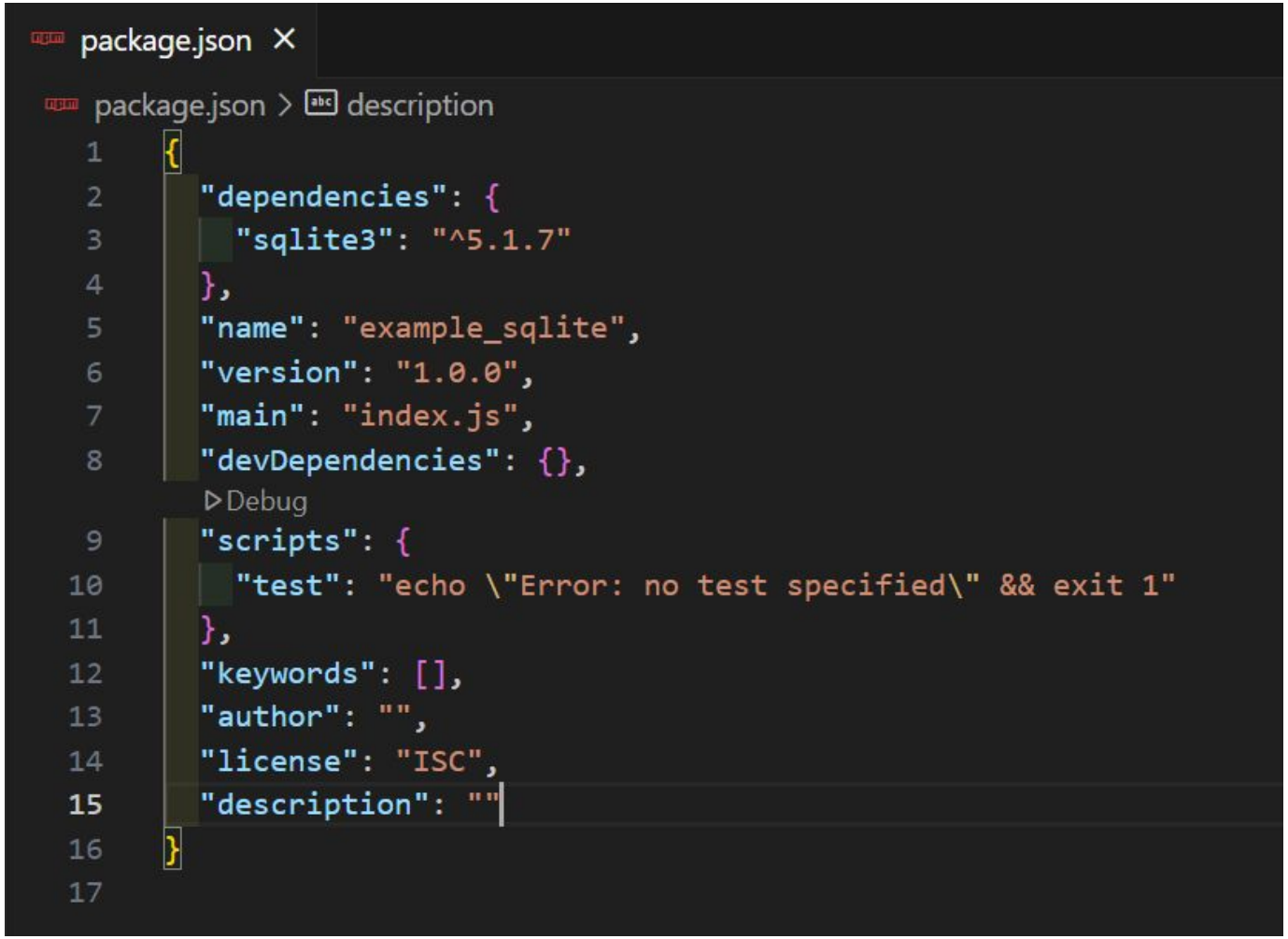


- Inicializamos el proyecto con el siguiente comando

```
npm init -y
```

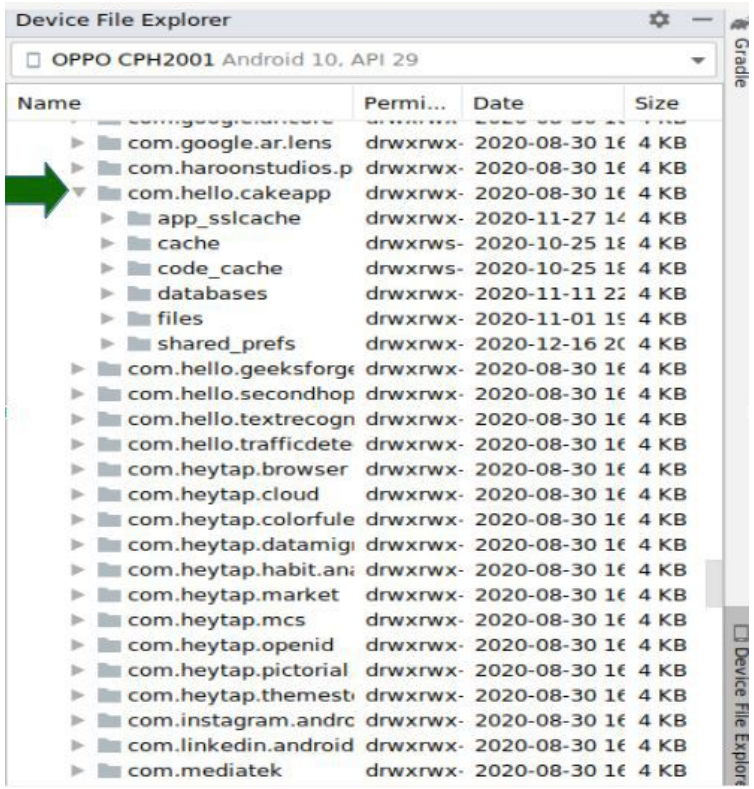
NODE.JS - SQLite

Validar el package.json



NODE.JS - SQLite

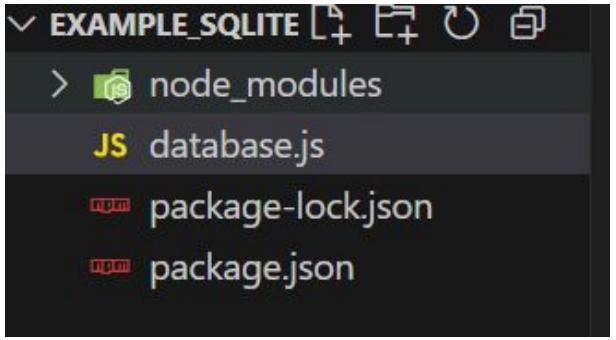
Conexión a una base de datos SQLite



NODE.JS - SQLite

Conexión a una base de datos SQLite

- Para conectarnos a una base de datos SQLite, usaremos el módulo sqlite3.
- **Crear un archivo database.js,** este archivo va a ser un módulo para trabajar elemento de la base de datos



NODE.JS - SQLite

Conexión a una base de datos SQLite

database.js, este archivo va a ser un módulo para trabajar elemento de la base de datos

JS database.js

```
JS database.js > ...
1   const sqlite3 = require("sqlite3").verbose();
2   const db = new sqlite3.Database("database.db");
3
4   db.serialize(() => {
5       db.run("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT, email TEXT)");
6   });
7
8   module.exports = db;
```

NODE.JS - SQLite

Crear CRUD

- **Crear el archivo index.js,** este archivo contendrá la lógica para realizar el crud.
- **Cargar el módulo de la librería express**
  - npm install express

```
EXAMPLE_SQlite> npm install express
```

NODE.JS - SQLite

Crear CRUD

- **Archivo index.js**

```
JS index.js > ...
1   const express = require("express");
2   const db = require("./database");
3   const app = express();
4   const port = 3000;
5
6   app.use(express.json());
```

NODE.JS - SQLite

Crear CRUD

- **Archivo index.js =>Insert**

```
// User Creation
// POST /users
app.post("/users", (req, res) => {
    const { name, email } = req.body;
    db.run("INSERT INTO users (name, email) VALUES (?, ?)", [name, email], function(err) {
        if (err) return res.status(500).json({ error: err.message });
        res.json({ id: this.lastID, name, email });
    });
});
```

NODE.JS - SQLite

Crear CRUD

- **Archivo index.js =>Select all**

```
18  // Get data users
19  app.get("/users", (req, res) => {
20      db.all("SELECT * FROM users", [], (err, rows) => {
21          if (err) return res.status(500).json({ error: err.message });
22          res.json(rows);
23      });
24  });
25
```

NODE.JS - SQLite

Crear CRUD

- **Archivo index.js =>Update**

```
34  // update user
35  app.put("/users/:id", (req, res) => {
36      const { name, email } = req.body;
37      db.run("UPDATE users SET name = ?, email = ? WHERE id = ?", [name, email, req.params.id], function(err) {
38          if (err) return res.status(500).json({ error: err.message });
39          res.json({ updated: this.changes });
40      });
41  });
```

NODE.JS - SQLite

Crear CRUD

- **Archivo index.js**

```
51  app.listen(port, () => {
52      console.log(`Server running http://localhost:${port}`);
53  });
54
```

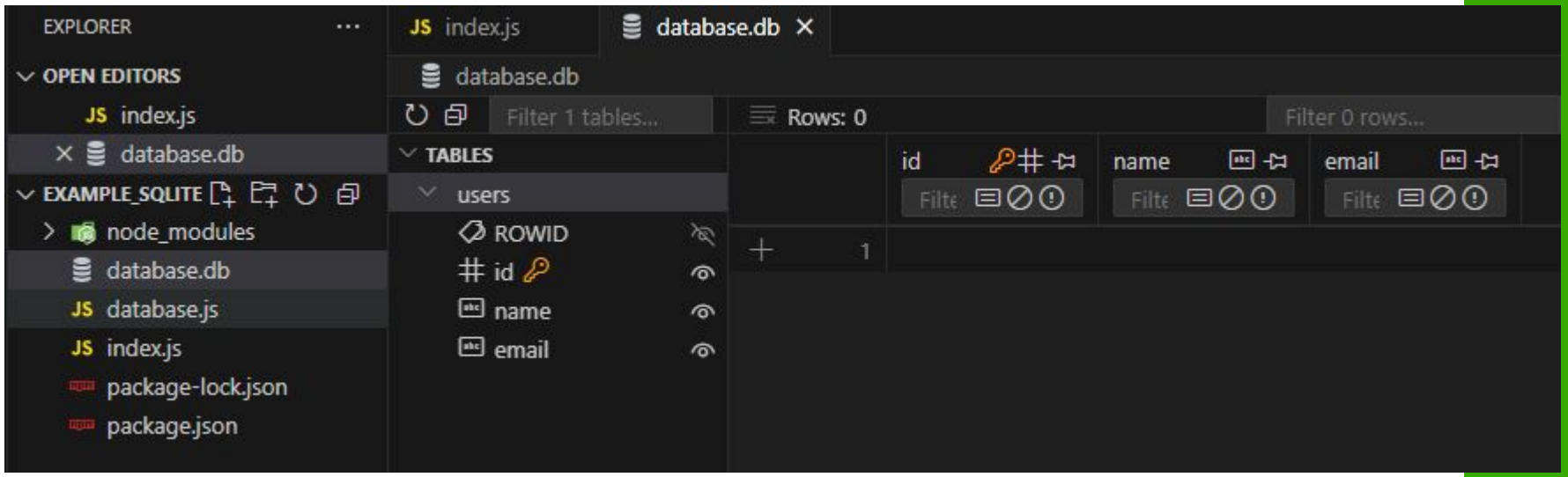
NODE.JS - SQLite

Ejecutar el proyecto

- Subimos el servicio y ejecutamos el proyecto con la siguiente instrucción

```
PS C:\xampp\htdocs\SENA\NodeJS\EXAMPLE_SQlite> node index.js
Server running http://localhost:3000
```

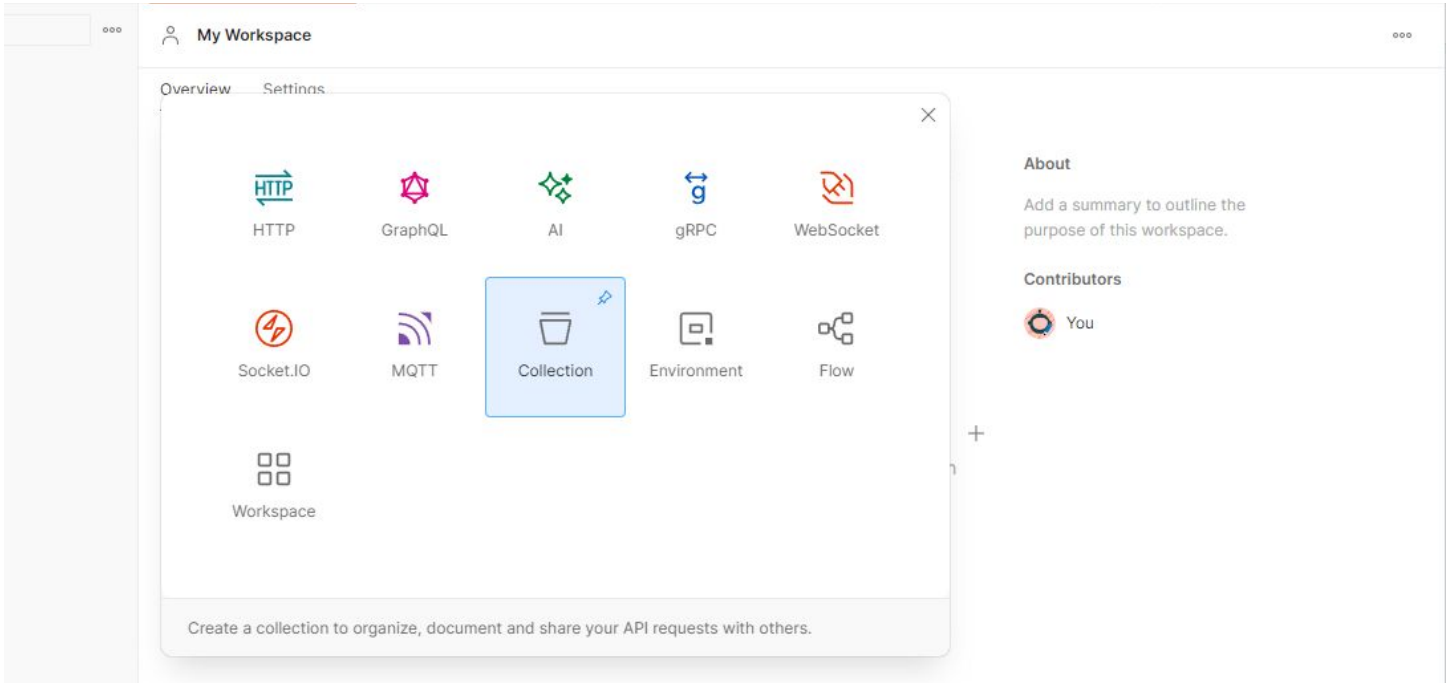
- Se crea la base de datos:



NODE.JS - SQLite

Ejecutar el proyecto

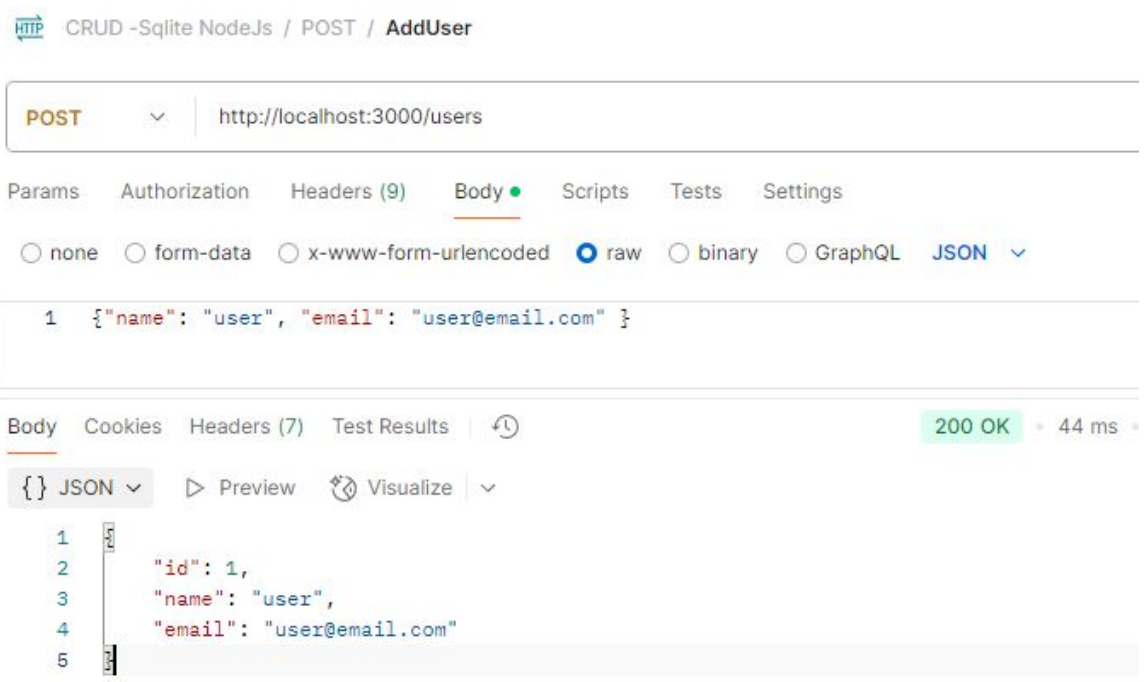
- Probamos el proyecto en postman
  - Se Crea una nueva colección



Servicio Nacional de Aprendizaje

NODE.JS - SQLite

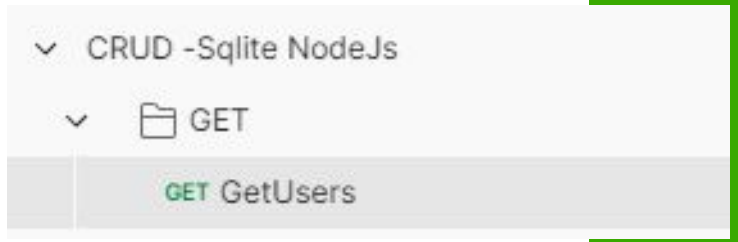
Ejecutar el proyecto

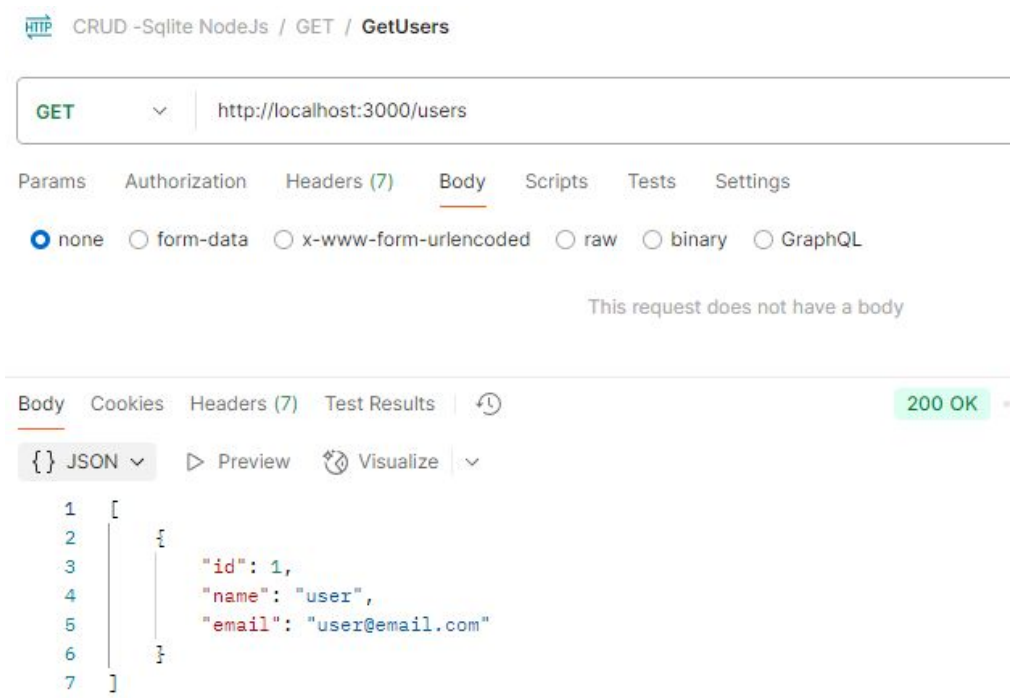
- Probamos el proyecto en postman
  - Ejecutar las consultas Post
    - 

NODE.JS - SQLite

Ejecutar el proyecto

- Probamos el proyecto en postman
  - Ejecutar las consultas Get
    -

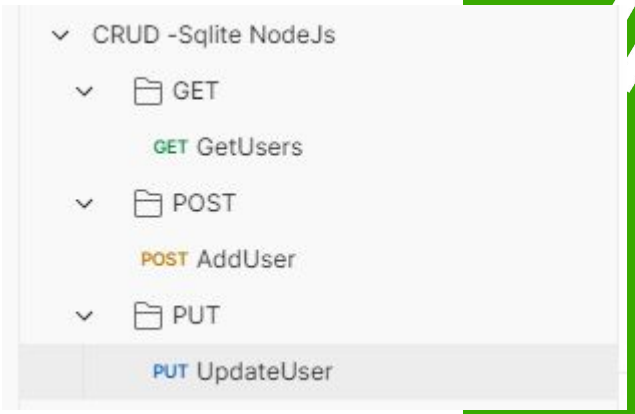


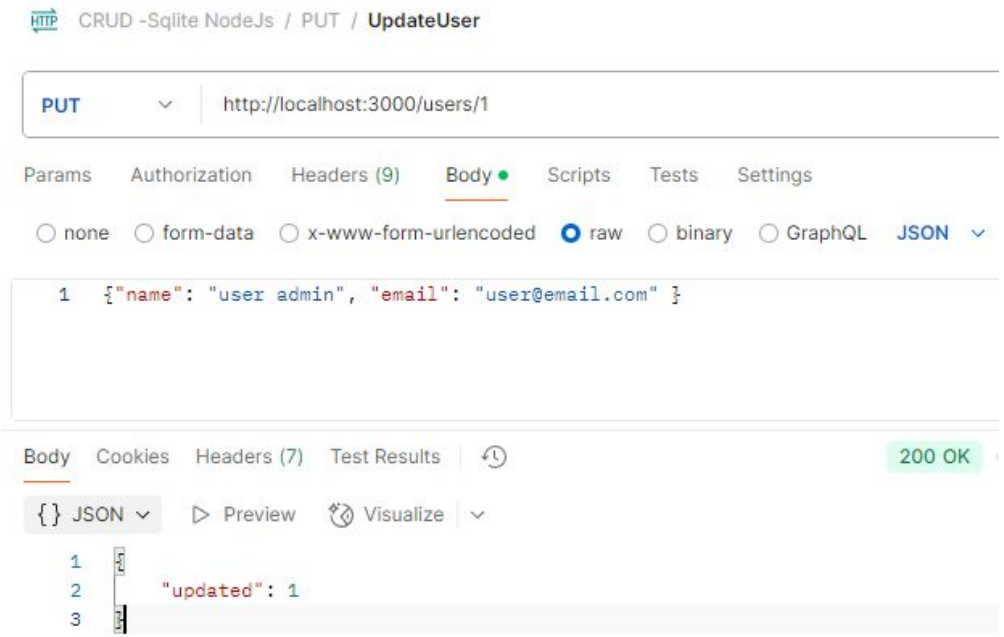


NODE.JS - SQLite

Ejecutar el proyecto

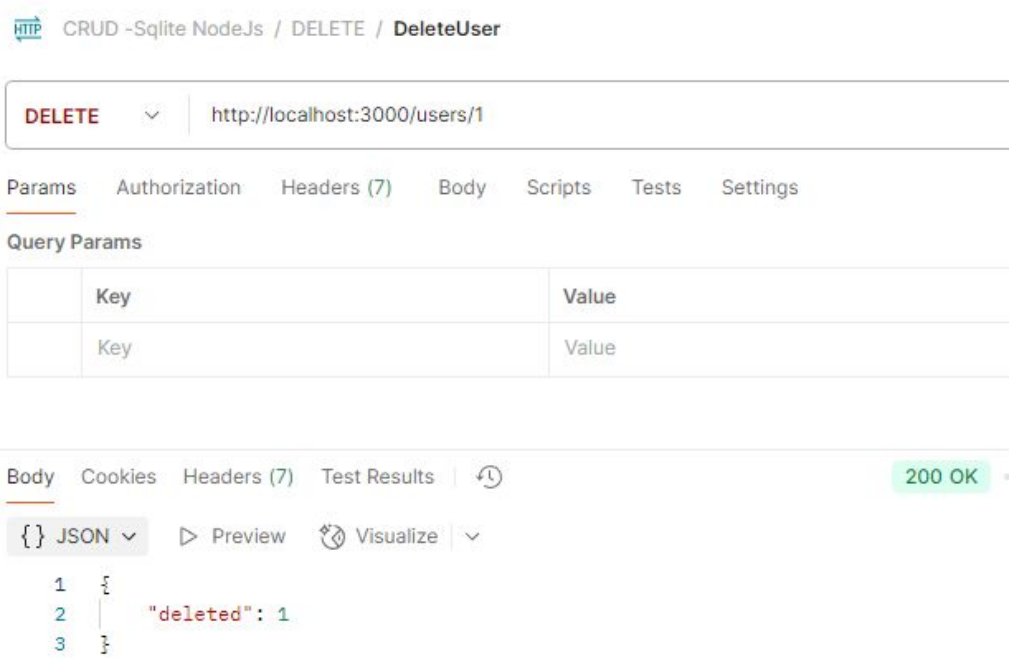
- Probamos el proyecto en postman
  - Ejecutar las consultas Put
    -

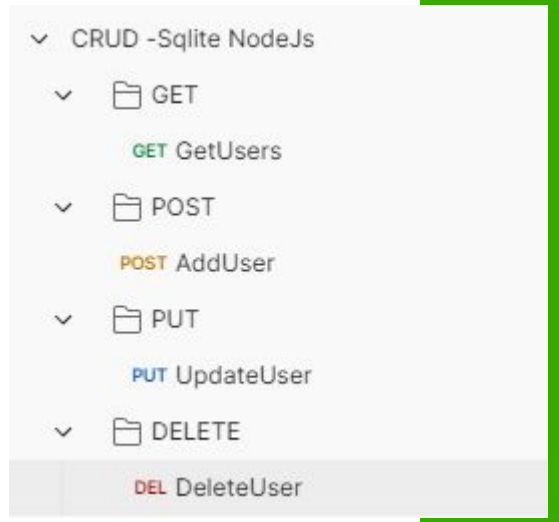




NODE.JS - SQLite

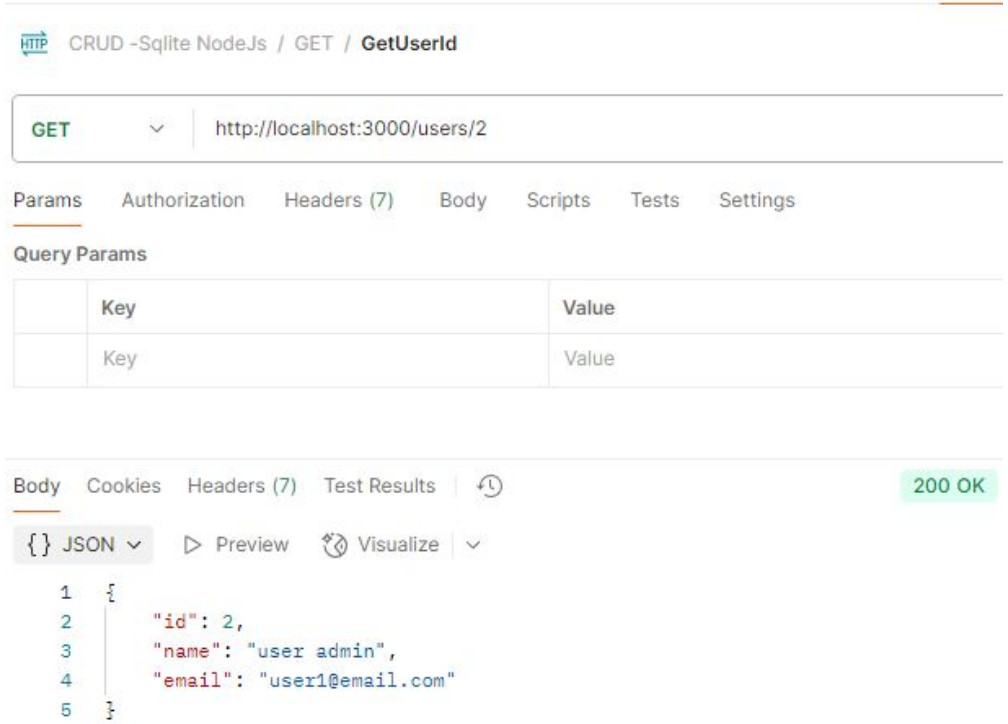
Ejecutar el proyecto

- Probamos el proyecto en postman
  - Ejecutar las consultas Delete
    - 



NODE.JS - SQLite

Ejecutar el proyecto

- Probamos el proyecto en postman
  - Ejecutar las consultas Get id
    - 

G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682

@SENAComunica

www.sena.edu.co