



SERVICIO NACIONAL DE APRENDIZAJE SENA REGIONAL CAUCA

MÓDULOS DE SOFTWARE CODIFICADOS Y PROBADOS

GA7-220501096-AA2-EV02

**PRESENTA:
JOANA ANDREA SILVA MACIAS
CC: 1125181139**

**INSTRUCTOR:
AREA TECNICA**

TECNOLOGIA EN ANALISIS Y DESARROLLO DE SOFTWARE

FICHA: 2977349

BOGOTA 15/06/2025



INTRODUCCION

Realizaremos la codificación a partir del programa o lenguaje seleccionado (JavaScript) que incluya nombramiento de variable, clases, método y funciones, se definió que se va a utilizar el estándar de nombramiento (`camelCase`) además, conexión a la base de datos MYSQL, realización CRUD (INSERCIÓN, ACTUALIZAR, ELIMINAR Y CONSULTAR) la base de datos.

Herramientas a utilizar:

- MySQL Workbench (DB)
- Visual Studio Code
- Express.js
- Node.js

Agrego URL : <https://github.com/Andreasilva-code/prados2v2.git>



CONCEPTOS CLAVES

- **Variables**

Las variables en JavaScript permiten almacenar datos que pueden cambiar durante la ejecución del programa. Se deben seguir las convenciones de ECMAScript y utilizar `let` y `const` en lugar de `var` para una mejor gestión del alcance y la mutabilidad.

- **Clases**

Las clases en JavaScript permiten estructurar el código en objetos reutilizables. Se definen con la palabra clave **class** y pueden contener atributos y métodos para gestionar datos y comportamientos de una entidad.

- **Métodos**

Métodos Los métodos en JavaScript son funciones definidas dentro de una clase y permiten operar sobre los datos de los objetos. Pueden ser estáticos (**static**) o de instancia.

- **Funciones**

Las funciones en JavaScript permiten encapsular lógica reutilizable dentro de un programa. Pueden definirse como funciones tradicionales o como funciones de flecha (`=>`) para mayor simplicidad y legibilidad.

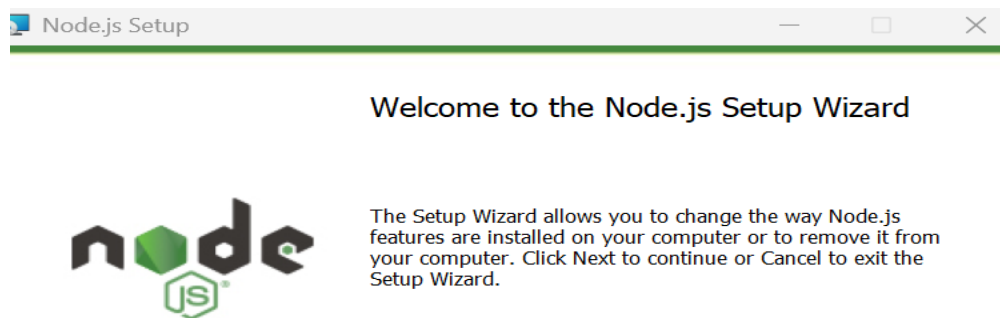


DESARROLLO

Node.js

Es un entorno de ejecución de JavaScript de código abierto y multiplataforma que se utiliza para construir aplicaciones del lado del servidor y de red. Permite ejecutar código JavaScript fuera de un navegador web, lo que lo hace útil para crear servidores, aplicaciones web, herramientas de línea de comandos y scripts. Node.js es conocido por su eficiencia y escalabilidad, especialmente en aplicaciones que manejan grandes cantidades de datos y conexiones simultáneas.

Concebido como un entorno de ejecución JavaScript asíncrono basado en eventos, Node.js está diseñado para construir aplicaciones de red escalables. En el siguiente ejemplo de 'hola mundo', se pueden manejar muchas conexiones de manera concurrente. Por cada conexión el callback sera ejecutado, sin embargo si no hay trabajo que hacer, Node.js entrará en reposo.



VISUAL STUDIO CODE

Es un editor de código fuente ligero pero eficaz que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux.



Gratuito y de código abierto. Git integrado, depuración y extensiones.


↓ Ventanas
Windows 10, 11

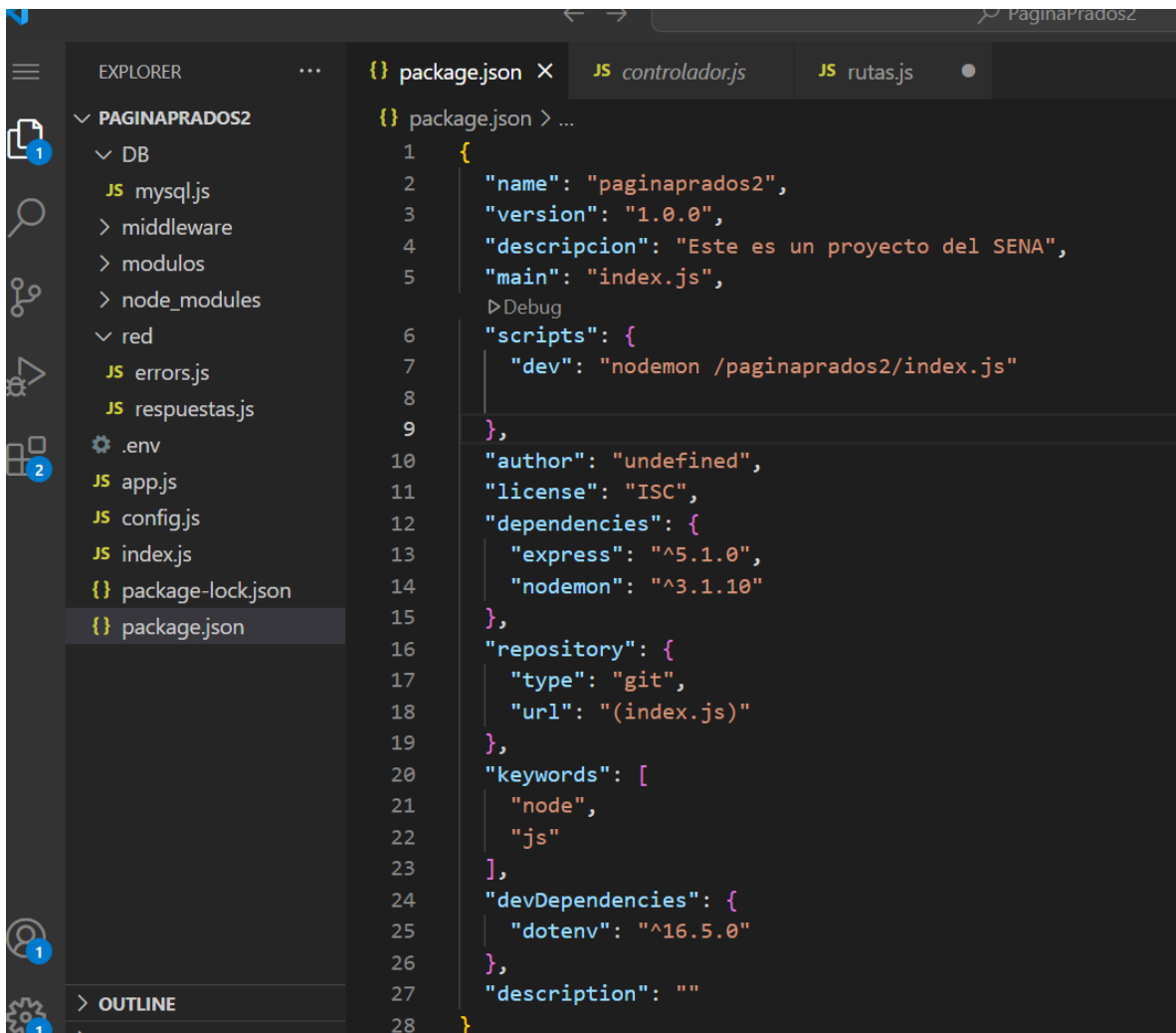
Instalador de usuario [x64](#) [Brazo64](#)
Instalador del sistema [x64](#) [Brazo64](#)
.cremallera [x64](#) [Brazo64](#)
CLI [x64](#) [Brazo64](#)


↓ .debutante
Debian, Ubuntu
↓ .rpm
Red Hat, Fedora, SUSE

.debutante [x64](#) [Brazo32](#) [Brazo64](#)
.rpm [x64](#) [Brazo32](#) [Brazo64](#)
.tar.gz [x64](#) [Brazo32](#) [Brazo64](#)
Quebrar [Tienda Snap](#)
CLI [x64](#) [Brazo32](#) [Brazo64](#)

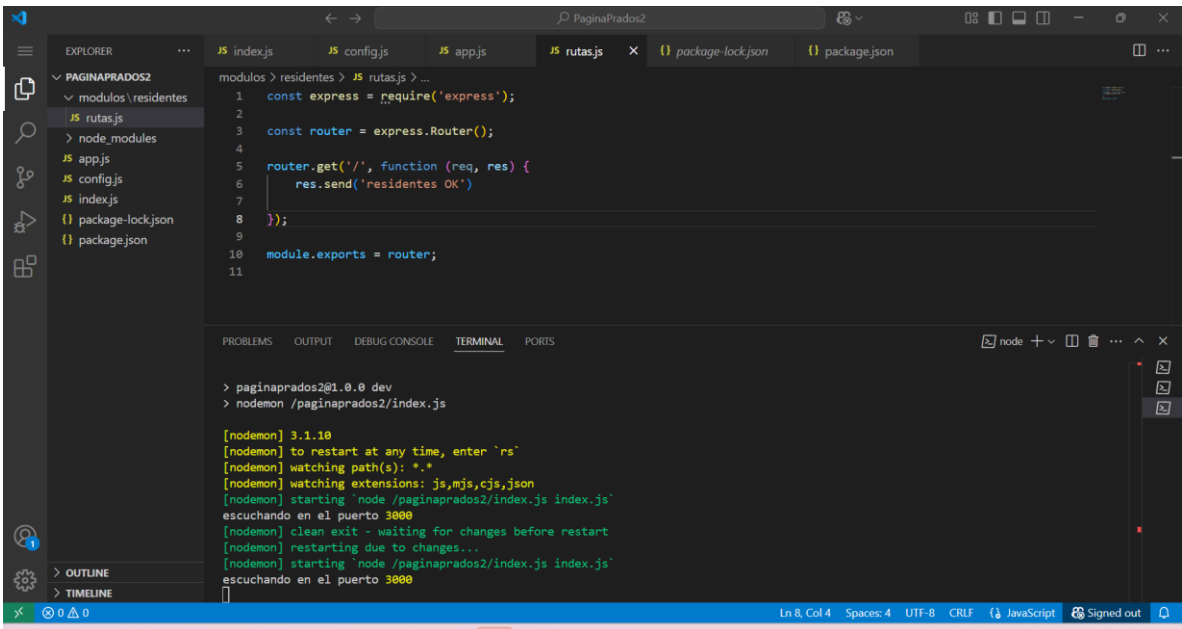
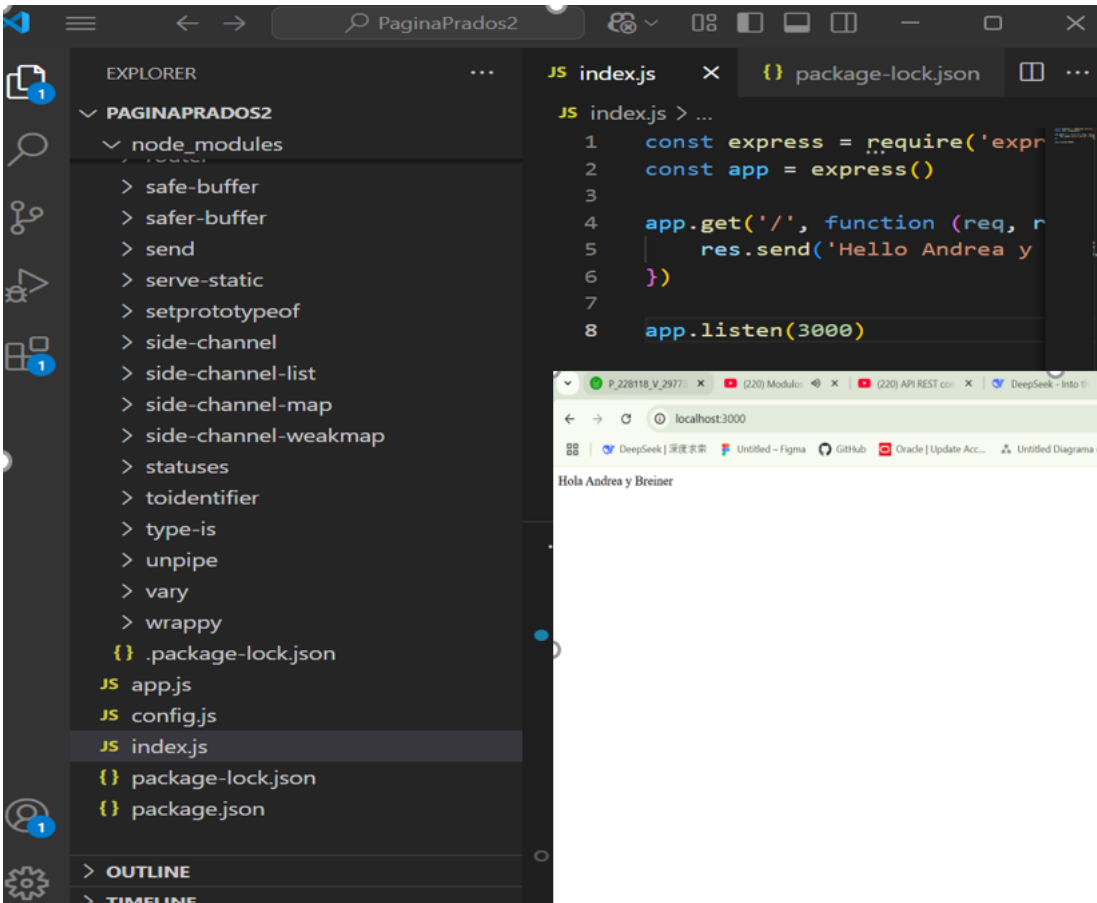

↓ Impermeable
macOS 10.15+

.cremallera [Chip Intel](#) [silicio de Apple](#) [Universal](#)
CLI [Chip Intel](#) [silicio de Apple](#)



The screenshot shows the Visual Studio Code editor interface. On the left, the Explorer sidebar displays the project structure for 'PAGINAPRADOS2', including folders like 'DB', 'middleware', 'modulos', 'node_modules', and 'red', and files like 'mysql.js', 'errors.js', 'respuestas.js', '.env', 'app.js', 'config.js', 'index.js', 'package-lock.json', and 'package.json'. The 'package.json' file is selected and its content is displayed in the main editor area. The code defines the project name as 'paginaprados2', version '1.0.0', and includes dependencies for 'express' and 'nodemon'. It also specifies a development environment with 'dotenv'.

```
{} package.json > ...
1  {
2    "name": "paginaprados2",
3    "version": "1.0.0",
4    "descripcion": "Este es un proyecto del SENA",
5    "main": "index.js",
6    "scripts": {
7      "dev": "nodemon /paginaprados2/index.js"
8    },
9  },
10  "author": "undefined",
11  "license": "ISC",
12  "dependencies": {
13    "express": "^5.1.0",
14    "nodemon": "^3.1.10"
15  },
16  "repository": {
17    "type": "git",
18    "url": "(index.js)"
19  },
20  "keywords": [
21    "node",
22    "js"
23  ],
24  "devDependencies": {
25    "dotenv": "^16.5.0"
26  },
27  "description": ""
28 }
```



CREAR



Postman interface showing a REST client request to `http://localhost:3000/api/usuarios` using the POST method. The request body is a JSON object:

```
{  "idUsuarios": 0,  "nombreUsuario": "mateo cruz",  "clave": "cruz123",  "fechaCreacion": "2025-03-10 14:30:00",  "correo": "materocruz@gmail.com"}
```

The response status is **201 Created** (19 ms, 88 B). The response body is a JSON object:

```
{  "error": false,  "status": 201,  "body": {    "mensaje": "Usuario creado con éxito",    "datos": true  }  }
```

MySQL Workbench interface showing the database structure and query results.

Query 1: `SELECT * FROM prados2dbv3.usuarios;`

Result Grid:

	idUsuarios	nombreUsuario	clave	fechaCreacion	correo
1	00001	montola123	montola123	2025-03-10 14:30:00	juannrut@gmail.com
2	00002	montola123	montola123	2025-03-10 14:30:00	juannrut@gmail.com
3	00003	montola123	montola123	2025-03-10 14:30:00	juannrut@gmail.com
4	00004	cruz123	cruz123	2025-03-10 14:30:00	materocruz@gmail.com
5	00005	cruz123	cruz123	2025-03-10 14:30:00	materocruz@gmail.com



```
DeepSeek - Into the Unkn... x P_228118_V_2977349_R_1 x localhost3000/api/usuario x WhatsApp x (58) #05 - Rutas para us...
localhost3000/api/usuarios
Impresión con formato estilístico
{
  "error": falso,
  "estado": 200,
  "cuerpo": [
    {
      "idUsuarios": 1,
      "nombreUsuario": null,
      "clave": "montolla123",
      "fechaCreacion": "2025-03-10T19:30:00.000Z",
      "correo": "juanruit@gmail.com"
    },
    {
      "idUsuarios": 2,
      "nombreUsuario": null,
      "clave": "montolla123",
      "fechaCreacion": "2025-03-10T19:30:00.000Z",
      "correo": "juanruit@gmail.com"
    },
    {
      "idUsuarios": 3,
      "nombreUsuario": null,
      "clave": "montolla123",
      "fechaCreacion": "2025-03-10T19:30:00.000Z",
      "correo": "juanruit@gmail.com"
    },
    {
      "idUsuarios": 4,
      "nombreUsuario": null,
      "clave": "cruz123",
      "fechaCreacion": "2025-03-10T19:30:00.000Z",
      "correo": "materocruz@gmail.com"
    }
  ]
}
```

AGREGAR

```

  DB > JS mysql.js > [0] bdconfig
  33
  34 conMySQL();
  35
  36 function todos(tabla){
  37   return new Promise( (resolve, reject) => {
  38     conexion.query('SELECT * FROM ${tabla}', (error, result) => {
  39       return error ? reject(error) : resolve(result);
  40     });
  41   });
  42 }
  43
  44 function uno(tabla, id){
  45   return new Promise( (resolve, reject) => {
  46     conexion.query('SELECT * FROM ${tabla} WHERE id=${id}', (error, result) => {
  47       return error ? reject(error) : resolve(result);
  48     });
  49   });
  50 }
  51 function insertar(tabla, data){
  52   return new Promise( (resolve, reject) => {
  53     conexion.query('INSERT INTO ${tabla} SET ?', data, (error, result) => {
  54       return error ? reject(error) : resolve(result);
  55     });
  56   });
  57 }
```




Application File Edit View Window Tools Help

Search. Ctrl + P

My Collection Run

Base Environment
Add Cookies
Add Certificates

Filter

residentes
consultar
eliminar
agregar

POST http://localhost:3000/api/residentes Send

201 Created 25 ms 69 B Just Now

Params Body Auth Headers 4 Scripts Docs

JSON

```
1 {  
2   "idCedulaResidente": 0,  
3   "nombre": "pepito",  
4   "apellido": "ruiz",  
5   "direccion": "usme centro",  
6   "correo": "ruiz@gmail.com",  
7   "celular": "3209876546",  
8   "estado": 1  
9 }  
10  
11
```

Preview

```
1 {  
2   "error": false,  
3   "status": 201,  
4   "body": "residente guardado con exito"  
5 }
```

Beautiful JSON

\$.store.books[*].author

Online Made with Kong

DeepSeek - Into the... JSON Online Validato... P_228118_V_297734... localhost:3000/api/re... (1) WhatsApp... (56) #04 | 2 - Conexi... porque me ale este... operador logico dife...

localhost:3000/api/residentes

DeepSeek | 深度求索 | Untitled - Figma | GitHub | Oracle | Update Acc... | Untitled Diagrama d...

Impresión con formato estilístico

```
{  
  "error": false,  
  "status": 200,  
  "body": [  
    {  
      "idCedulaResidente": 0,  
      "nombre": "pepito",  
      "apellido": "ruiz",  
      "direccion": "usme centro",  
      "correo": "ruiz@gmail.com",  
      "celular": "3209876546",  
      "estado": 1  
    },  
    {  
      "idCedulaResidente": 6666666,  
      "nombre": "pepito",  
      "apellido": "ruiz",  
      "direccion": "usme centro",  
      "correo": "ruiz@gmail.com",  
      "celular": "3209876546",  
      "estado": 1  
    },  
    {  
      "idCedulaResidente": 7777777,  
      "nombre": "pepito",  
      "apellido": "ruiz",  
      "direccion": "usme centro",  
      "correo": "ruiz@gmail.com",  
      "celular": "3209876546",  
      "estado": 1  
    },  
    {  
      "idCedulaResidente": 8888888,  
      "nombre": "pepito",  
      "apellido": "ruiz",  
      "direccion": "usme centro",  
      "correo": "ruiz@gmail.com",  
      "celular": "3209876546",  
      "estado": 1  
    },  
    {  
      "idCedulaResidente": 999999999,  
      "nombre": "pepito",  
      "apellido": "perez",  
      "direccion": "usme pueblo",  
      "correo": "pepito@gmail.com",  
      "celular": "3209876545",  
      "estado": 1  
    },  
    {  
      "idCedulaResidente": 1019048241,  
      "nombre": "Juan",  
      "apellido": "Rui",  
      "direccion": "Cra 11",  
      "correo": "juanruit@gmail.com",  
      "celular": "3123532126",  
      "estado": 1  
    },  
    {  
      "idCedulaResidente": 1125181139,  
      "nombre": "Andrea",  
      "apellido": "silva",  
      "direccion": "Cra 12",  
      "correo": "andrea.silva.macias@gmail.com",  
      "celular": "3154609161",  
      "estado": 1  
    }  
  ]  
}
```



MySQL Workbench

Query 1: `SELECT * FROM prados2db.residentes;`

idCedulaResidente	nombre	apellido	direccion	correo	celular	estado
0	carlos	cruz	Cra 213	carlocruz@gmail.com	3209876544	2
6666666	pepito	ruiz	usme centro	ruiz@gmail.com	3209876546	1
7777777	pepito	ruiz	usme centro	ruiz@gmail.com	3209876546	1
8888888	pepito	ruiz	usme centro	ruiz@gmail.com	3209876546	1
9999999	pepito	perez	usme pueblo	pepito@gmail.com	3209876546	1
1019048241	Juan	Ruiz	Cra 11	juanruiz@gmail.com	3123532128	1
1125181139	andrea	silva	Cra 12	andrea.silva.macias@gmail.com	31254609191	1

Table: **residentes**

Columns:

- idCedulaResidente (int PK)
- nombre (varchar(4))
- apellido (varchar(4))
- direccion (varchar(4))
- correo (varchar(4))
- celular (varchar(4))
- estado (tinyint)

Output: 6 19:58:05 SELECT * FROM prados2db.residentes LIMIT 0, 1000. Message: 2 row(s) returned

CONSULTAR

Insomnia

Application File Edit View Window Tools Help

GET `http://localhost:3000/api/residentes`

Params Body Auth Headers (4) Scripts Docs

JSON

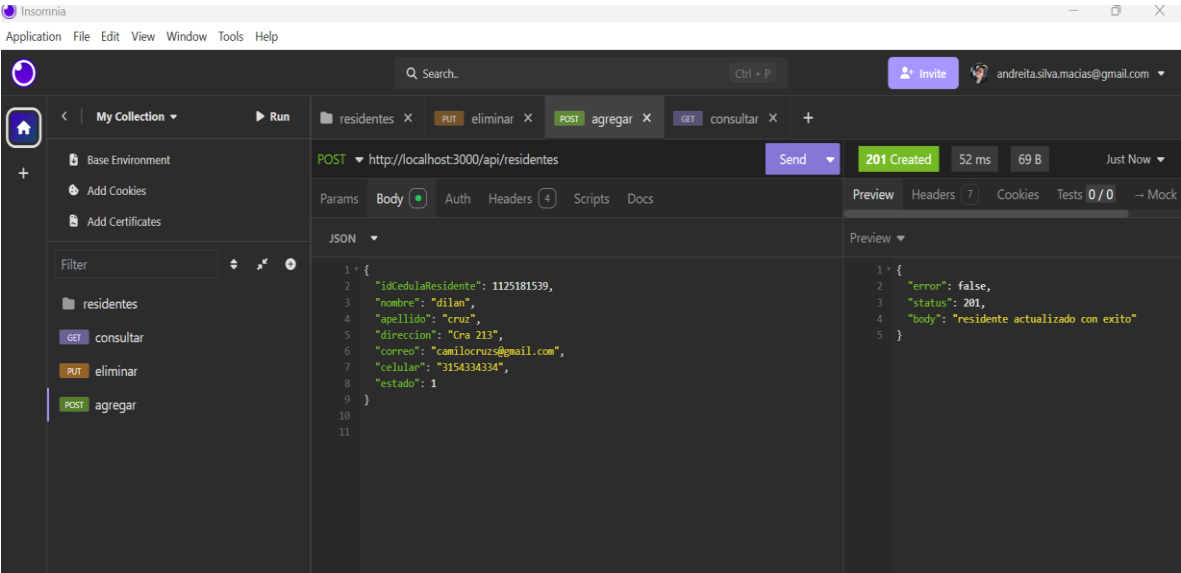
```
1 {
2   "error": false,
3   "status": 200,
4   "body": [
5     {
6       "idCedulaResidente": 0,
7       "nombre": "pepito",
8       "apellido": "ruiz",
9       "direccion": "usme centro",
10      "correo": "ruiz@gmail.com",
11      "celular": "3209876546",
12      "estado": 1
13    },
14    {
15      "idCedulaResidente": 6666666,
16      "nombre": "pepito",
17      "apellido": "ruiz",
18      "direccion": "usme centro",
19      "correo": "ruiz@gmail.com",
20      "celular": "3209876546",
21      "estado": 1
22    },
23  ]
24 }
```

Preview Headers (7) Cookies Tests 0/0 Mock Console

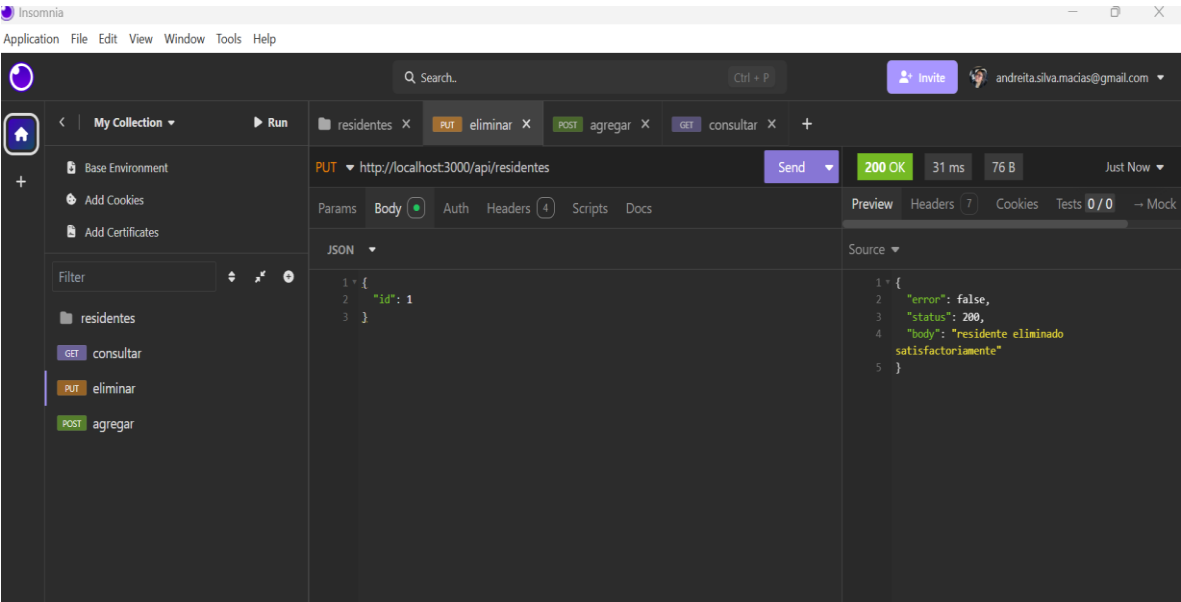
Beautify JSON

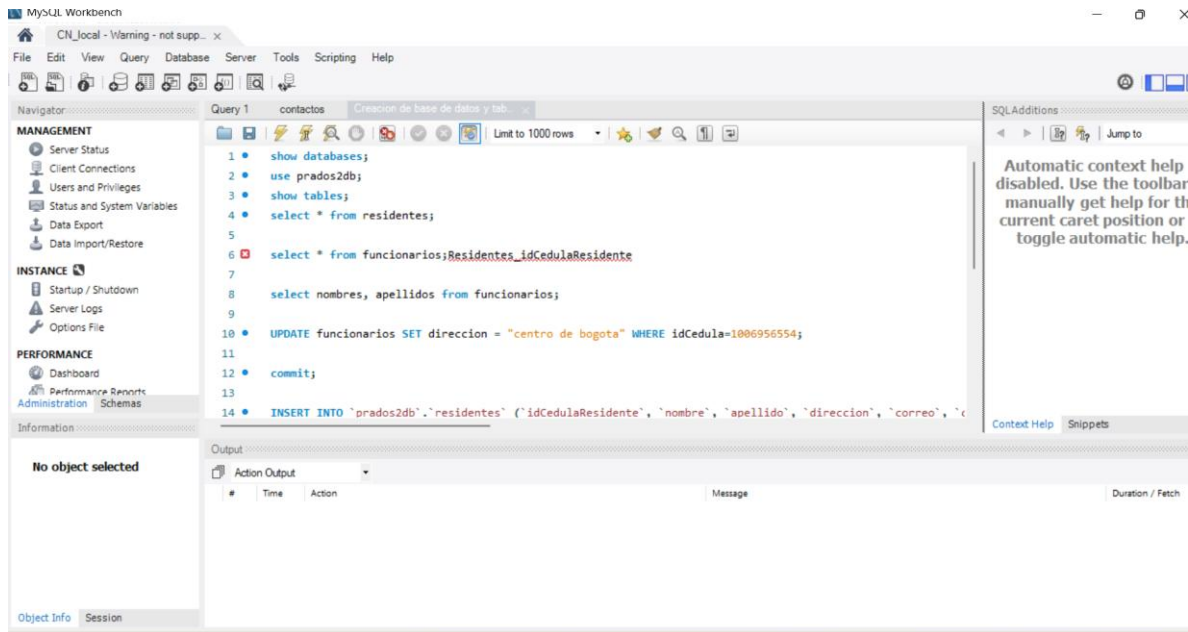
\$.store.books[*].author

ACTUALIZAR

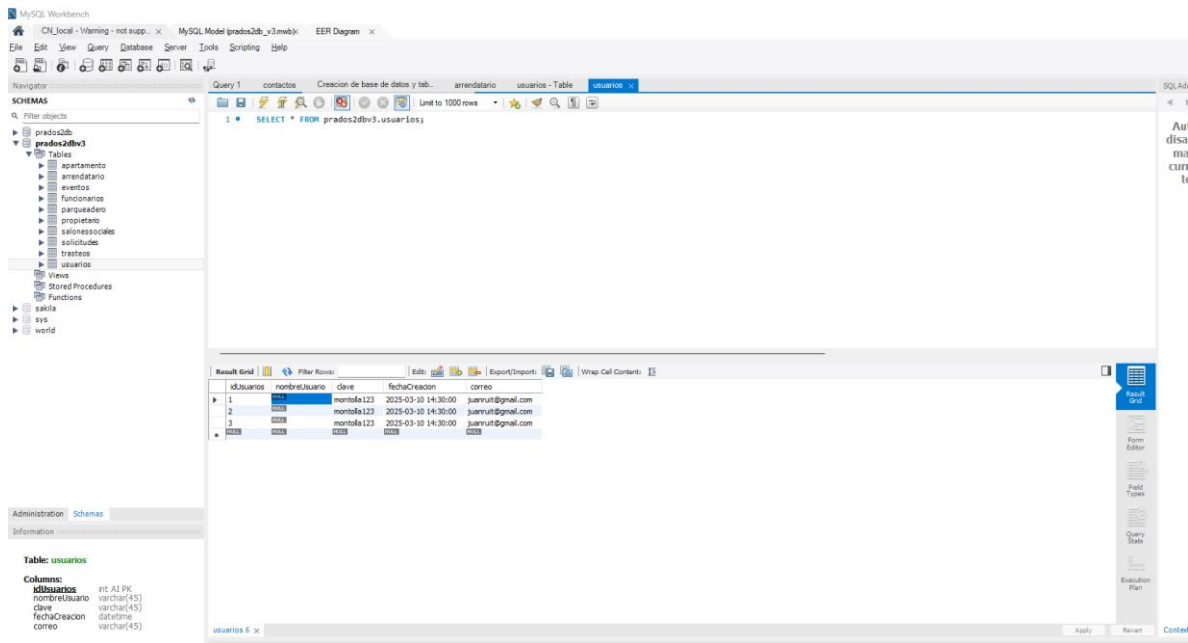


ELIMINAR

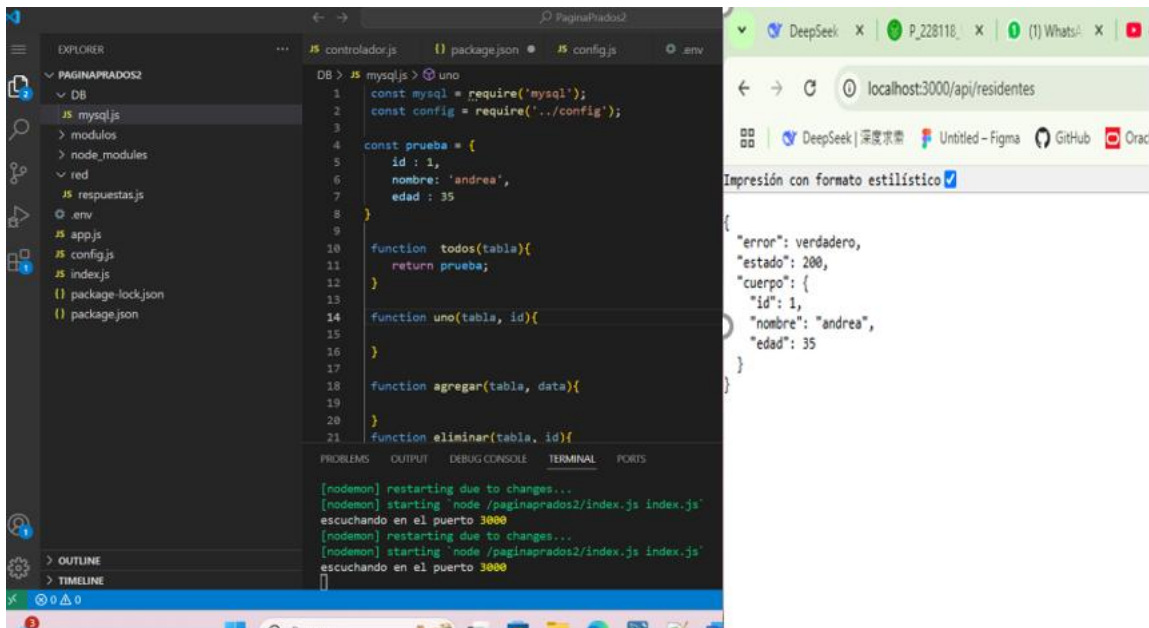




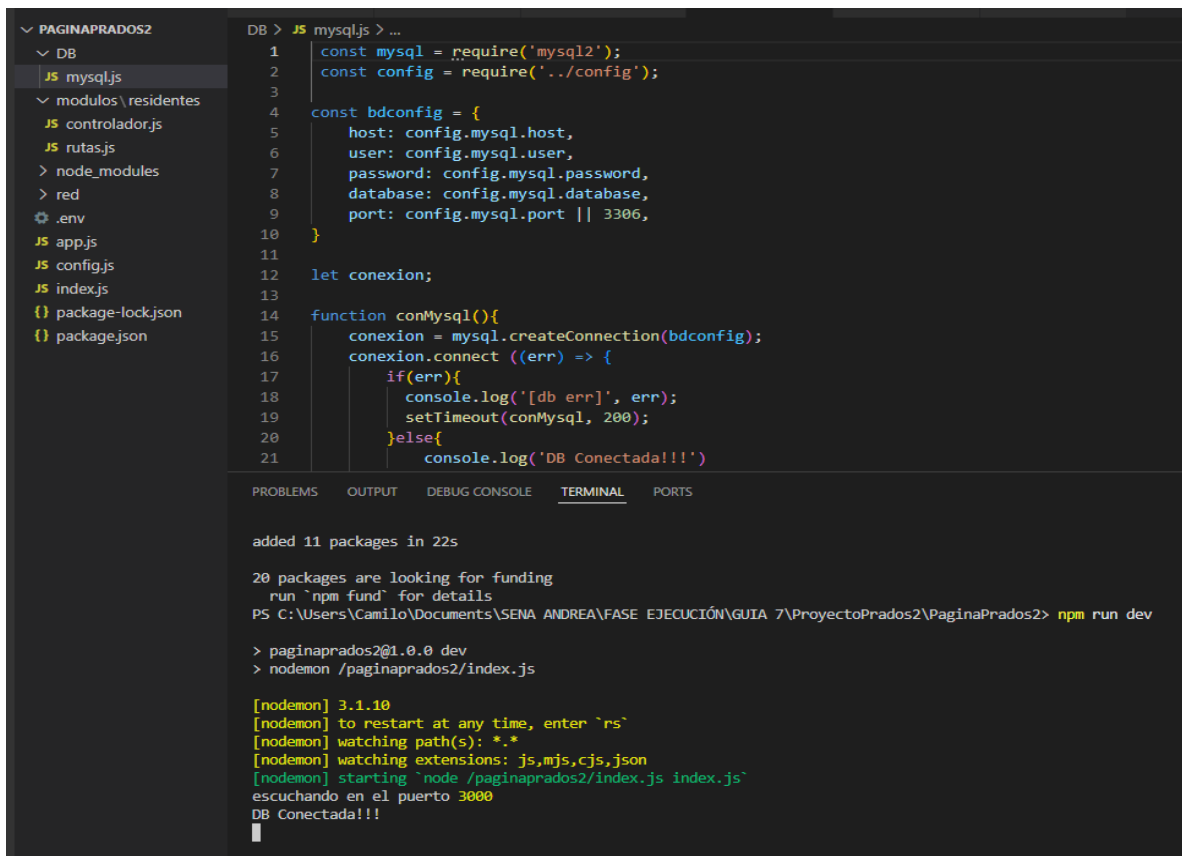
MySQL Workbench



Configuration MYSQL



Node js conectado a mysql



HERRAMIENTA DE VERSIONAMIENTO GITHUB



DeepSeek - Into the Uni...
P_228118_V_2977348_R...
localhost:3000/api/vsua...
WhatsApp
(56) 805 - Rutas para usu...
GitHub
Subir archivos - Andread...
Anja2 que es - Buscar co...
+
-
X

github.com

DeepSeek | 深度学习
Untitled - Figma
GitHub
Oracle | Update Acc...
Untitled Diagrama d...

Panel

Escribe para buscar

+
-
X

Top repositories

Find a repository...

Andreasiva-code/prados-2
Andreasiva-code/programa
Andreasiva-code/Programa-git
Andreasiva-code/prados2v2

Hogar

Pregúntale al copiloto

Comience a usar GitHub
<> Aprenda a codificar
Crear una aplicación web
Chatea en mi idioma

Lista de reproducción

GitHub para principiantes en YouTube

Diseñado para ayudarte a dominar los conceptos básicos de GitHub, ya seas nuevo en codificación o busques mejorar tus habilidades de control de versiones.

Iniciar lista de reproducción

GitHub for Beginners
What is Git?

Empezando

2/3 completo



CONCLUSION

Esta evidencia es muy importante puesto que me permitió aprender a trabajar con JavaScript, lenguaje que, a pesar de ser antiguo, es muy utilizado actualmente por las empresas, no solo eso, sino que también pude aprender de otros programas como Insomnia: aplicación de escritorio multiplataforma, gratuita y de código abierto, que facilita el diseño, la depuración y las pruebas de Apis. Funciona como un cliente REST, permitiendo a los usuarios enviar solicitudes HTTP a Apis web para probar su funcionalidad, autenticación, y control de acceso, entre otras cosas. y formularios HTML con JSP, este código fue subido a GitHub.



REFERENCIAS

Valdivieso, C. (2019). Cómo crear un API REST con Express.js y Node.js.
Recuperado de <https://lavaldi.com/blog/como-crear-api-rest-express-node>

Llamas, L. (s.f.). Cómo usar una base de datos MySQL con Node.js.
Recuperado de <https://www.luisllamas.es/como-usar-mysql-con-nodejs/>