

SENA

Centro Nacional de hotelería, turismo y alimentos

Ficha:

2627076

Trabajo:

Desarrollar software a partir de la integración de sus módulos componentes -
BackEnd

Aprendices:

Jorge Orlando Rincón Ortiz

Miguel Ángel Torres Díaz

María Camila Carreño Maldonado

Talía Yurani López Gualcala

Milton Herrera

Instructor

Juan Manuel

Servicio Nacional de Aprendizaje

2024

Introducción

En el presente documento presentamos la creación de un proyecto **CRUD (Create, Read, Update, Delete)** Utilizando el framework **Express.js** para **Node.js**. El objetivo principal es demostrar las capacidades de **Express.js** para crear aplicaciones web dinámicas y escalables con una arquitectura **MCV (modelo vista controlador)**

Se desarrollo una aplicación CRUD completa y funcionar, por parte del backend que será complementada en una evidencia próxima con el frontend de la misma para obtener ambas partes del proyecto, tanto en el lado del servidor, como en el lado del cliente.

Herramientas empleadas para la creación del proyecto

Node.js

Node.js es un entorno de ejecución de JavaScript construido sobre el motor v8 de Chrome, que permite ejecutar JavaScript en el lado del servidor. Node.js proporciona una arquitectura de eventos asincrónicos y no bloqueantes, lo que lo hace ideal para el desarrollo de aplicaciones web escalables y de alto rendimiento.

Express.js

Es un marco de aplicación web para **Node.js** que simplifica el desarrollo de aplicaciones web, mediante la provisión de una capa de abstracción sobre **Node.js**, facilitando la gestión de rutas, middleware y peticiones **HTTP**.

MongoDB Atlas

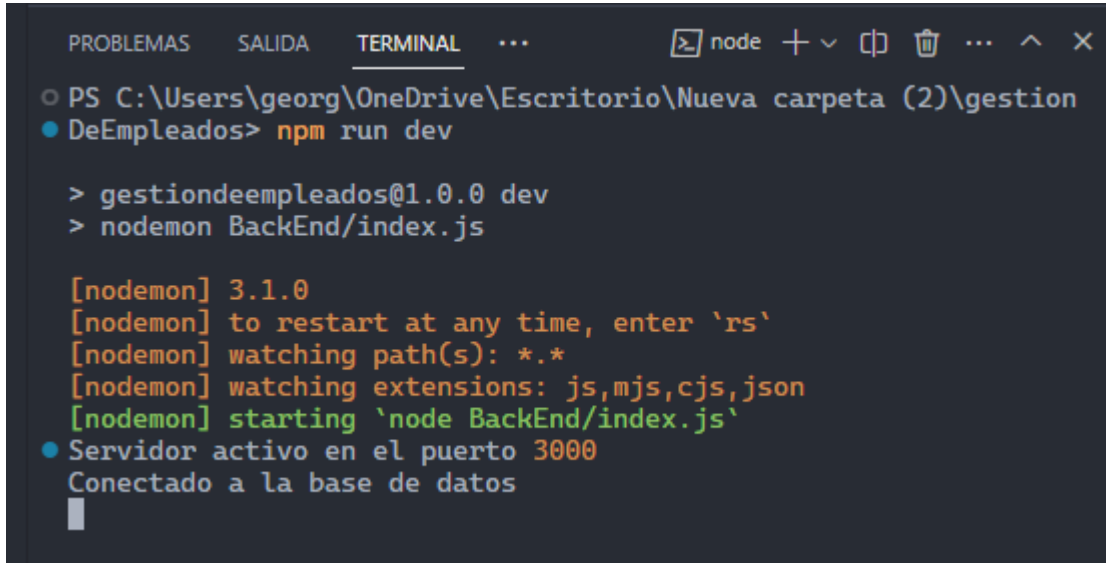
Es un servicio de datos en la nube totalmente gestionado, ofrecido por **MongoDB Inc.** Proporciona una plataforma escalable y segura para alojar y administrar bases de datos no relacionales **MongoDB**.

Postman

Es una herramienta de software de código abierto muy utilizada para el desarrollo de aplicaciones web y móviles que se comunican con servicios web a través de **API**. Permite enviar y recibir solicitudes HTTP a la API como lo son las solicitudes **GET, POST, PUT, DELETE** y otros mas

Muestra de pruebas con solicitudes HTTP

Para hacer las pruebas por los métodos HTTP primero debemos poner a correr nuestro sistema desde la terminal del editor de texto, en nuestro caso cuando este arranca nos muestra dos mensajes, el primero indica que el servidor esta activo y el puerto en el que esta activo, el segundo nos indica que estamos conectados a la base de datos, después de inicializar el servidor este se ve así:



```
PROBLEMAS  SALIDA  TERMINAL  ...  node + v  [icon]  [icon]  ...  ^  x
PS C:\Users\georg\OneDrive\Escritorio\Nueva carpeta (2)\gestion
DeEmpleados> npm run dev

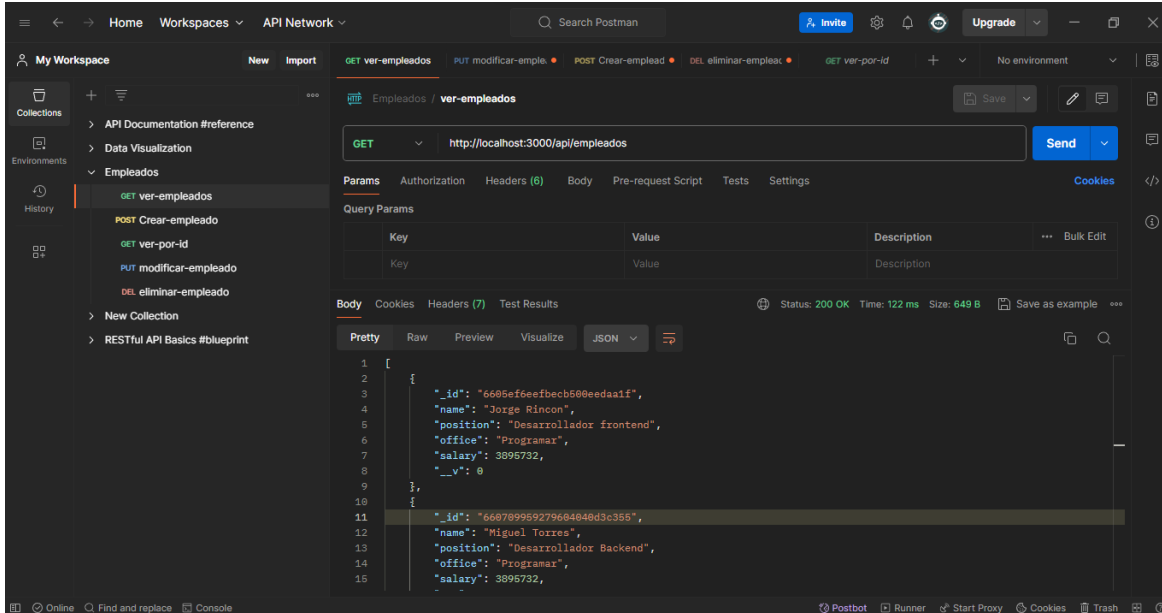
> gestiondeempleados@1.0.0 dev
> nodemon BackEnd/index.js

[nodemon] 3.1.0
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node BackEnd/index.js'
● Servidor activo en el puerto 3000
  Conectado a la base de datos
█
```

Ya inicializado el servidor podemos empezar con las solicitudes HTTP

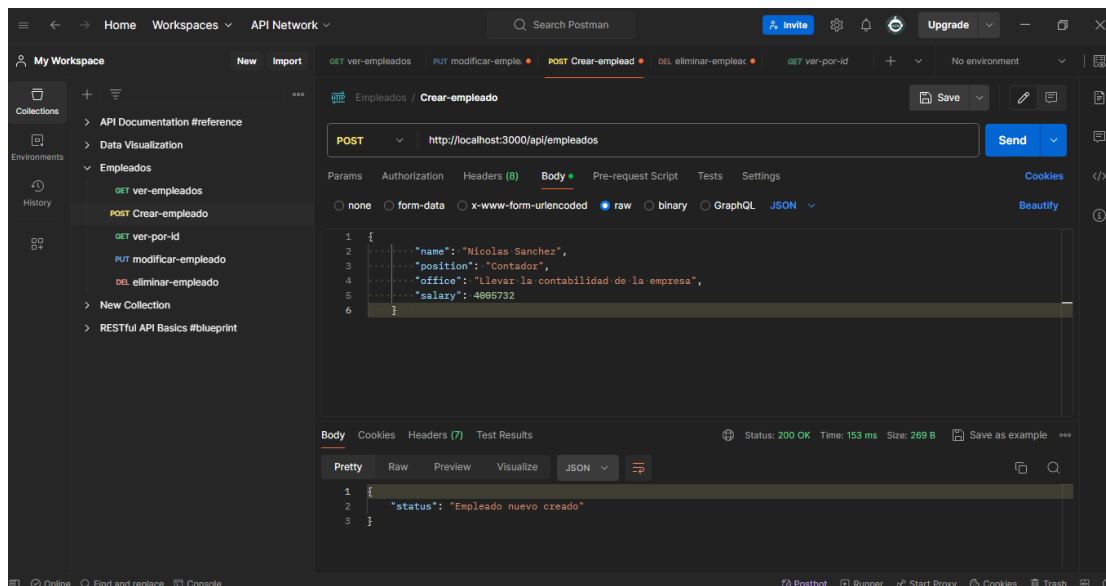
GET

Esta la empleamos para hacer en este caso dos tipos de solicitud a nuestro sistema desde **Postman**, la empleamos para realizar la solicitud de todos los empleados registrados, así mismo podemos solicitar uno solo por **ID**



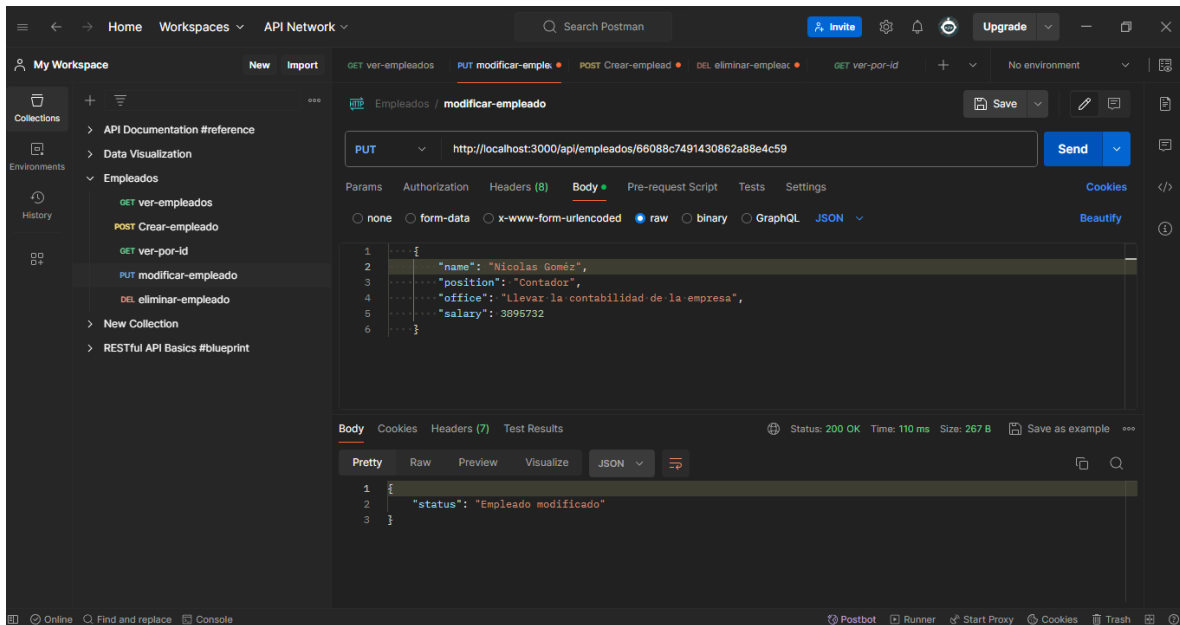
POST

Esta solicitud HTTP la empleamos para realizar la introducción de un nuevo empleado a la base de datos, desde la solicitud debemos colocar un json el cual en nuestro caso incluye nombre, posición, oficio y salario. Cuando hacemos la solicitud post, este nos responde con un mensaje de que el empleado fue creado



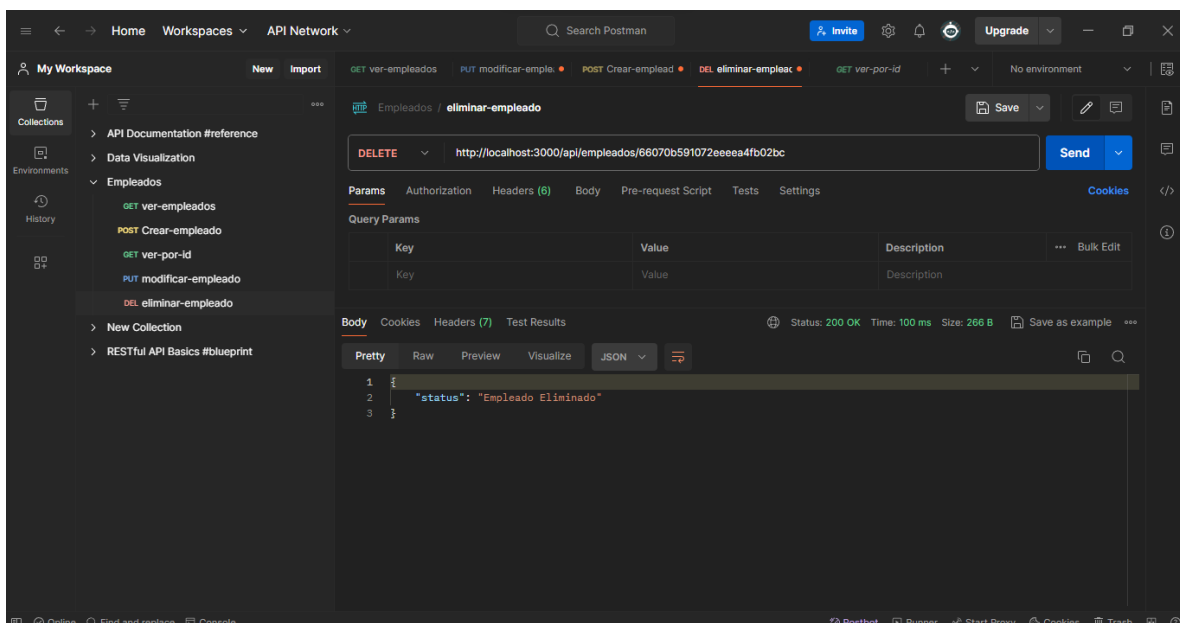
PUT

En esta solicitud HTTP lo que hacemos es editar un empleado, con esto logramos que, si a la hora de hacer un registro nuevo haya quedado algo mal, lo podamos modificar a nuestro antojo. Para hacer esta solicitud, debemos tener presente el id de la persona a modificar.



DELETE

Este básicamente lo empleamos para eliminar un empleado, se realiza también por id



Conclusión

Express.js es un framework poderoso y flexible para la creación de aplicaciones web. Su arquitectura MVC facilita la organización del código y la separación de responsabilidades. Además, la comunidad Express.js es activa y ofrece una gran cantidad de recursos y soporte.

La herramienta Postman en este proyecto nos fue de gran ayuda, ya que gracias a ella pudimos realizar todo tipo de solicitudes a nuestro proyecto.