UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

API COMPRESORES DE REFRIGERADORES

EQUIPO DESARROLLADOR

RODRIGUEZ GONZALEZ JULIO CESAR	RG01136120
MOZO CRUZ, JUDITH ABIGAIL	MC01136470
ZABALETA CERÉN JOSE CRUZ	ZC01135718
SANTOS GARCIA NATALY GEORGINA	SG01135335
LOPEZ BELTRAN EMERSON BLADIMIR	LB01136352

CÁTEDRA: NUEVAS TENDENCIAS DE PROGRAMACIÓN

HORARIO: VIERNES 7:00 PM - 8:40 PM

SÁBADO DE 7:00 AM - 8:40 AM

FACILITADOR: LIC. IRWIN GUARDADO

SAN SALVADOR 27 DE OCTUBRE DE 2022

Contenido

INTRODUCCIÓN

Se sabe que los equipos de refrigeración, aire acondicionado, entre otros emplean el ciclo frigorífico de compresión para producir frío, y prácticamente todos ellos tienen un motocompresor hermético. Esta máquina se conecta directamente al condensador y al evaporador formando un circuito cerrado. Su funcionamiento principal consiste en incorporar el mecanismo del compresor y del motor para su accionamiento dentro de un dispositivo hermético.

Tanto los compresores alternativos como los centrífugos se encuentran en el mercado formando equipos herméticos que incluyen el motor. Estos motores cerrados son de un tipo distinto al convencional porque están enfriados por el mismo líquido o vapor refrigerante, a temperaturas mucho más bajas que el aire empleado para enfriar los motores abiertos. Es decir, su función principal consiste en forzar mecánicamente la circulación de un fluido en un circuito cerrado, creando zonas de alta y baja presión con el propósito de que el fluido absorba calor en un lugar y lo disipe en el otro.

Dicho lo anterior, en el presente proyecto se presenta y se describe la importancia que un compresor tiene, de acuerdo a su utilidad en los diferentes aparatos electrónicos. Por lo tanto, en la API implementada con las mejores herramientas de programación, pretende almacenar información técnica y capacidad en HP (potencia de trabajo) de cada uno de los compresores utilizados en la refrigeración.

1. ANÁLISIS DEL PROBLEMA

API COMPRESORES DE REFRIGERADORES

1.1 PROBLEMÁTICA A RESOLVER

Con la creación de api de compresores de refrigeradores, se pretende dar a conocer el HP (caballos de fuerza) de los compresores de acuerdo al modelo y marca de este, debido a que cada empresa productora de compresores le da un código, combinación de letras y números y modelo diferente a cada producto y ese código equivale a las medidas más usadas en HP en los compresores de refrigeradores como lo es ½, ¼, ⅕, etc.

Debido a que este dato técnico no viene especificado en la viñeta o ficha técnica del compresor se busca en los catálogos que las mismas empresas proveen, pero esto solo es proporcionado a los distribuidores o empleados de los productores, técnicos y personas dedicadas a la refrigeración consulta en almacenes de distribución de compresores, o se consulta en foros de internet que a veces no contribuyen mucho.

Con el desarrollo de la API de Compresores de Refrigeradores se pretende guardar la información de la ficha técnica del compresor de cada una de las refrigeradores y guardar la capacidad en HP(caballos de fuerza) previamente validado por eso está el campo estado donde también se puede guardar como pendiente, el usuario podrá ingresar al sistema, previamente registrado y él podrá consultar de acuerdo a la marca del motor o código del modelo, en donde el sistema será capaz de mostrar los datos de esos modelos y mostrar la capacidad en HP, lo más importante para un técnico o cliente.

Esta información estará en un solo lugar conociendo el nombre de dominio, el usuario puede ingresar y consultar el código “modelo que busca”.

1.3 OBJETIVOS

1.3.1 Objetivo general:

Desarrollar una API con los datos técnicos de los compresores de refrigeradores y su respectiva validación de la capacidad en HP.

1.3.2 Objetivos específicos:

- Estructurar el código JS de acuerdo a la funcionalidad de la API.
- Diseñar una interfaz amigable e intuitiva para los usuarios.
- Elaborar documentación correspondiente del desarrollo de la API, Documento Científico, Manual de programador y Manual de usuario.

1.4 ANÁLISIS DE BENEFICIOS DEL PROYECTO

La ficha técnica es aliada en la búsqueda de la estandarización de productos, en el caso de los compresores se manejan muchos datos técnicos, pero el de HP(Caballos de Fuerza) es muy raro que las empresas productoras lo especifica en la ficha técnica la mayoría de veces viene oculta en el código los técnicos las obtienen aplicando diversas fórmulas para conocer el hp "unidad de potencia", es la potencia del compresor cuando trabaja de acuerdo a eso es la longitud de la tubería para realizar el ciclo de la refrigeración.

Por ende, la API de compresores de refrigeradores beneficiara a todas aquellas personas que interrelación con los compresores de refrigeradores como técnicos en refrigeración, ingenieros en refrigeración, distribuidores de compresores, consumidor final del compresor

Técnicos en refrigeración: Todas aquellas personas que se dedican a la reparación de las refrigeradoras, ellos podrán consultar en la api la información de algún modelo "código del motor" así realizar las modificaciones que necesita el refrigerador, tales como cambios de motores ya sea nuevo o de segunda.

Ingenieros en refrigeración: Personas que se dedican al diseño de nuevos aparatos de refrigeración, necesitan ver las especificaciones de la competencia o de los compresores antiguos.

Distribuidores de compresores: Empresas que se dedican a la distribución de los diferentes marcas y modelos de compresores que están a la venta en el mercado. Podrán consultar la información de la API o incluso colaborar con los administradores para realizar las verificaciones de los modelos.

Consumidor final del compresor: Persona que se le daña el refrigerador y llama al técnico, con dudas de que el compresor es de esa capacidad mayor, sucede cuando es de segunda (refiriéndose a otro compresor de otra refrigeradora que se encuentra en buen estado), él puede ingresar en línea y buscar la información y así estar más seguro de lo que realiza el técnico.

1.5 JUSTIFICACIÓN

Debido al crecimiento e implementación de nuevas tecnologías de trabajo en el área de la industria de compresores de refrigeradores, hay empresas que utilizan nuevas especificaciones técnicas, nuevos códigos y modelos, y en muchas ocasiones estas empresas solo dan las especificaciones técnicas a sus empleados o empresas distribuidoras, dejando fuera a la gran mayoría de personas que se dedican al mantenimiento de aparatos refrigerantes, y como ya es conocido en casi todos los compresores es extraño que contengan la capacidad en hp una medida necesaria para cambiar el compresor o realizar modificaciones al aparato refrigerante como el condensador.

Para identificar el HP de los compresores hay unas fórmulas como lo es el LRA = Locked Rotor Amps en español Amperaje de rotor bloqueado (motor detenido) y RLA = Running Load Amps o Rated Load Amps en español Amperaje de carga nominal (motor en marcha normal). Esta medida electrónica puede ser de 5.6,6.8,7.0 amperios si sube más el compresor es de más capacidad en HP, pero esto va de acuerdo a la experiencia de los técnicos en refrigeración.

El desarrollo de la API, esa información se encuentra en línea, los usuarios conectados a internet podrán ingresar y consultar el HP del compresor, generalmente mediante el modelo y la marca de forma ágil debido a que usamos la tecnología moderna en programación y base de datos.

La creación de la API de compresores de refrigeradores beneficiará: Técnicos en refrigeración, Ingenieros en refrigeración, Distribuidores de compresores y al Consumidor final

2. METODOLOGÍA Y CICLO DE VIDA UTILIZADOS

Ciclo de vida del software

En este proyecto se ha decidido desarrollar una metodología que cumpla con las características específicas del sistema de compresores, dentro de ellas se encuentran: usa una estructura clara que determina el objetivo rápidamente y transmite bien la información, es por esa razón que seleccionamos este tipo de ciclo de vida en cascada.

En este proceso se llevaron a cabo las siguientes actividades:

Recolección de requerimientos necesarios para el sistema, estos requerimientos se recolectaron mediante entrevistas dirigidas hacia el usuario, para validar la información que debía contener el sistema de acuerdo a las necesidades y la problemática a resolver; lo cual nos permitió como equipo de trabajo, tener una visión completa y compartida del sistema a desarrollar.

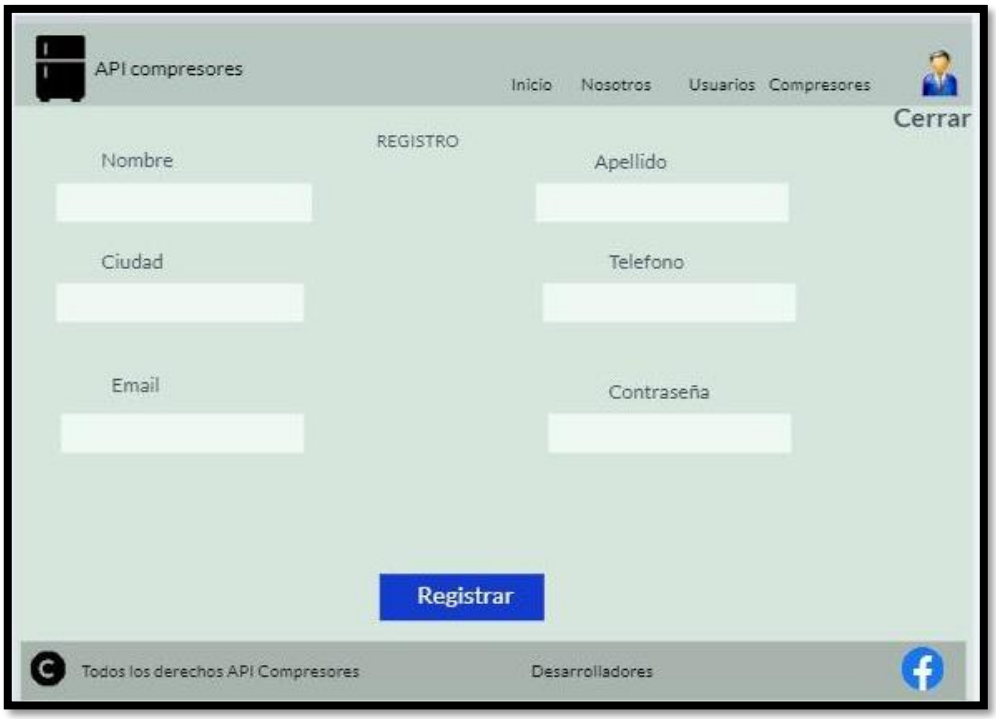
Modelo en cascada

Es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez.

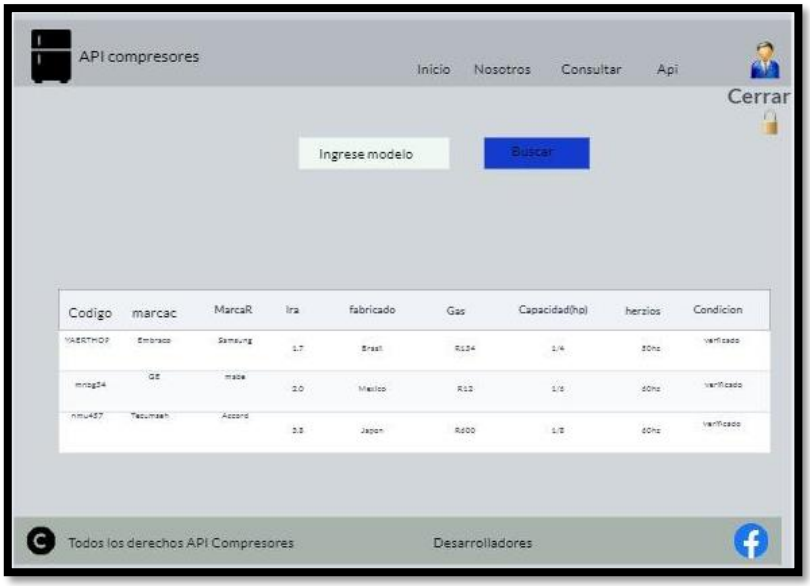
3. DISEÑO DEL SISTEMA

Se muestran imágenes de la interfaz de la API realizada.

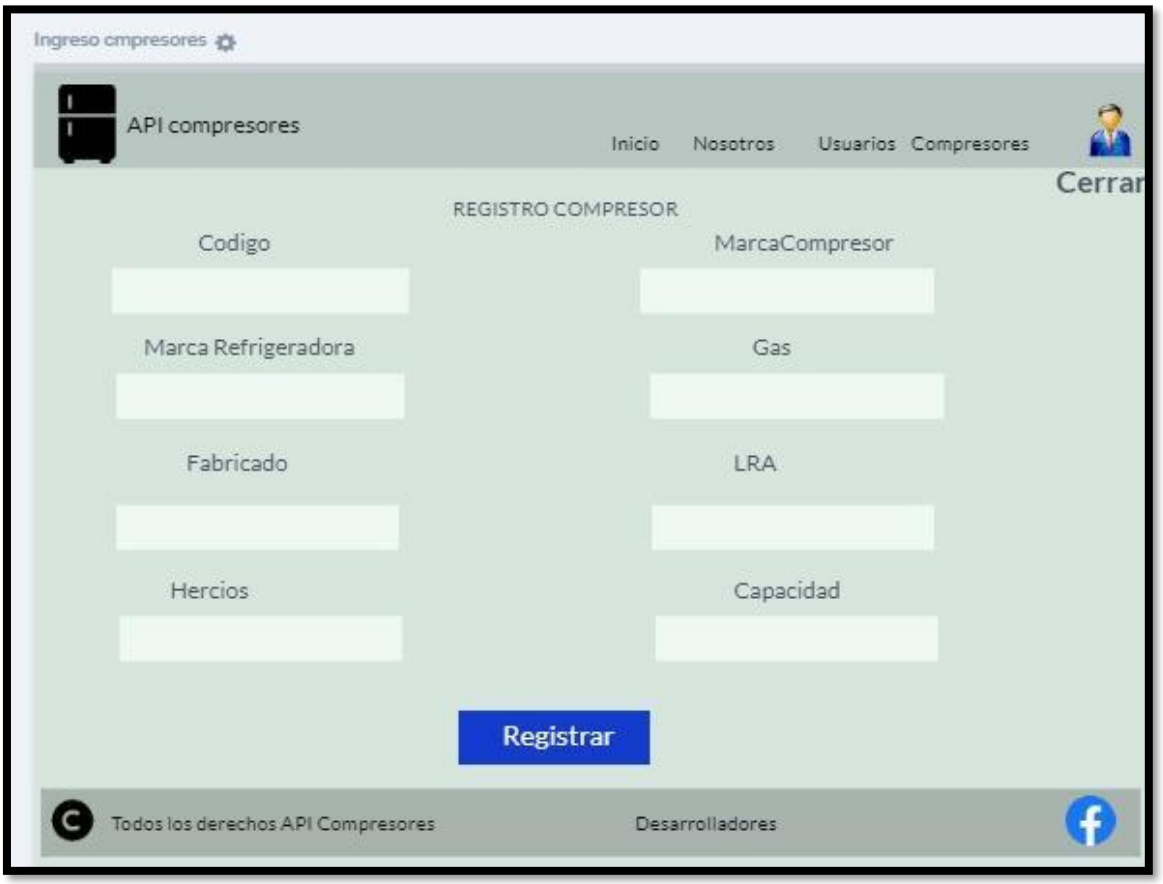
- Login de usuario.



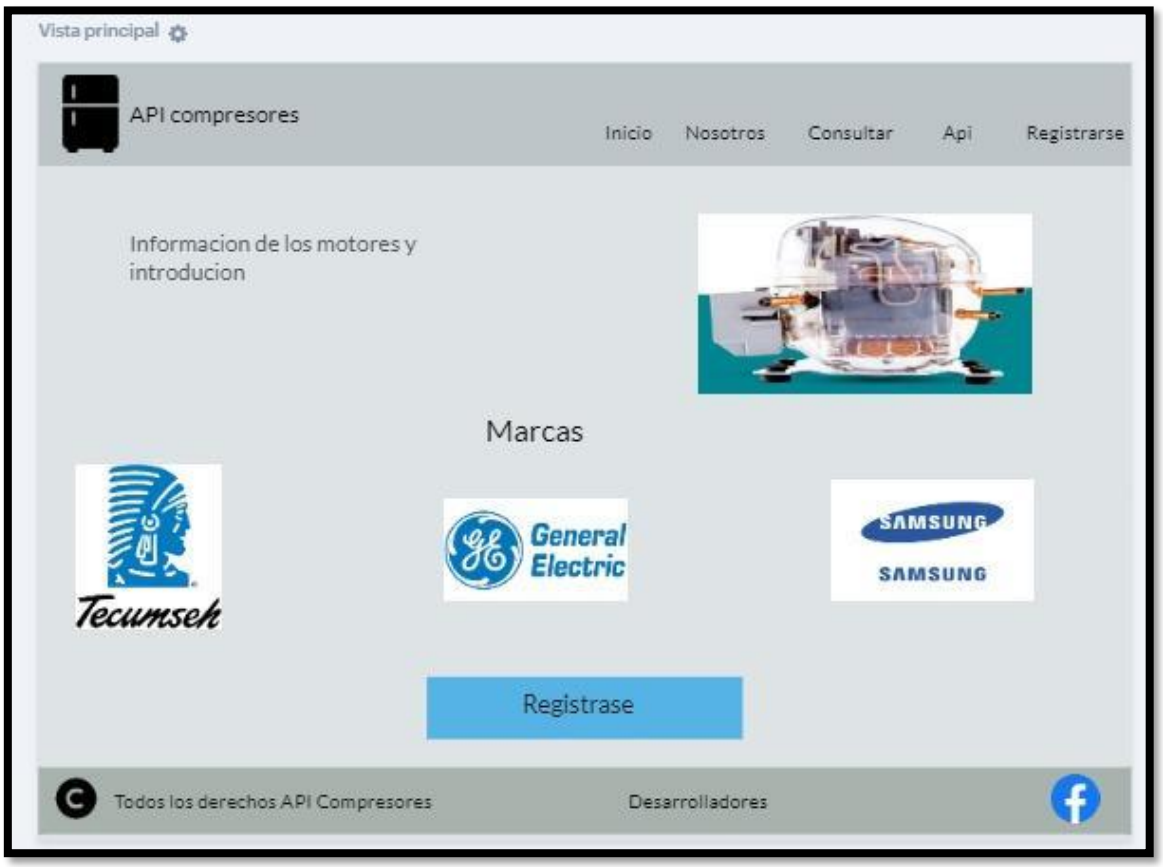
- Página de inicio.



- Registro de compresores.

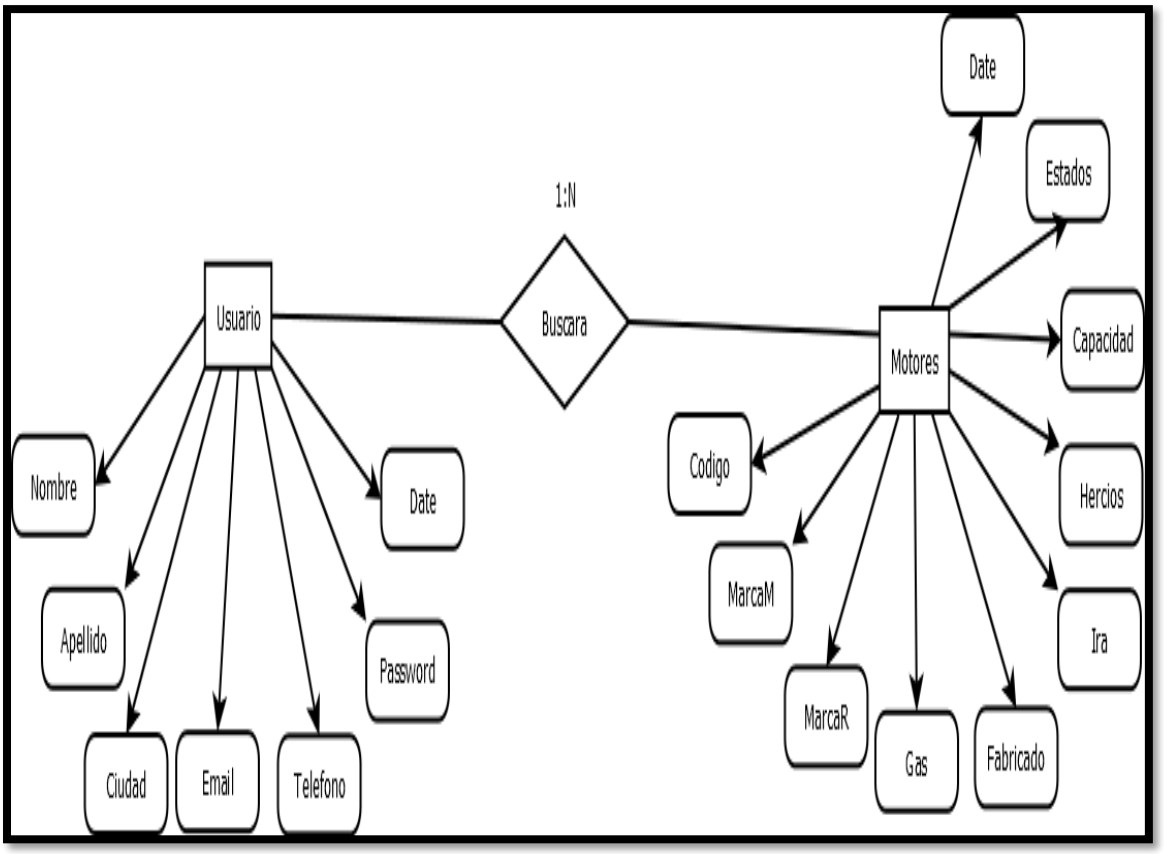


- Información de los motores.



4. DIAGRAMA ENTIDAD RELACIÓN DE LA API-COMPRESORES

El siguiente Diagrama Entidad Relación, muestra el funcionamiento paso a paso de la API. El usuario podrá realizar diferentes acciones, todo esto de acuerdo a su consulta.



5. DESARROLLO DEL PROYECTO

5.1 TÉCNICAS DE PROGRAMACIÓN APLICADAS

El sistema presentado a continuación fue evaluado y sometido a diferentes pruebas, para comprobar que su función sea requerida y aceptable. Habiendo solucionado los posibles errores que el sistema podría generar, refiriéndonos de esta manera que el sistema sea funcional y exitoso para el usuario. Las pruebas realizadas son estrictamente aplicadas para que la API sea apta en su funcionamiento concreto. Dichas pruebas van desde las herramientas que le permitirán al usuario el manejo del sistema, hasta la conexión con la base de datos con MongoDB, y las consultas que este pueda generar.

Por otro lado, el proyecto se realizó con modelo de arquitectura de software VMC o mejor conocido como Modelo vista controlador, donde se separan los datos de una aplicación, la interfaz de usuario y la lógica de control en tres diferentes módulos.

5.2 HERRAMIENTAS DE DESARROLLO UTILIZADAS

Las herramientas utilizadas para la programación de la API Compresores de refrigeradores, se describen a continuación con sus respectivas características:

Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft. La versión utilizada es la 1.72.2 la más reciente para el sistema de Linux.

Características: El código fuente combina la interfaz de usuario optimizada del editor moderno con asistencia y navegación de código enriquecido y una experiencia de depuración integrada, sin la necesidad de un IDE completo. Visual Studio Code cuenta con herramientas de Debug hasta opciones para actualización en tiempo real del código fuente con vista en navegador y el lenguaje en vivo como los css, además de las extensiones que ofrece otro tema o configurarlo al gusto. Visual Studio Code es un editor

de código fuente compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado. Algunas de las características de este no están disponibles en el menú o la interfaz de usuario.

- Colores de sintaxis: algunos elementos de los archivos de código y marcado están coloreadas de forma distinta para distinguirlos. Ejemplo: using en C# y import son de un color y los console y uri de otro color.
- Marca de errores y advertencias: al agregar código y compilar solución aparecen subrayados ondulados en color rojo esto indica error en el texto.
- Coincidencia de llaves: la inserción es colocada en una llave de apertura en el archivo de código, esta característica permite visualizar si han sido abiertas o cerradas correctamente o si hay algún error
- Visualizador de estructura: las líneas de punto conectan a las llaves que coinciden en los archivos de código, lo cual permite que sea más fácil ver las llaves de apertura y cierre.
- Número de línea: se muestran los números de línea en el margen ubicado en el margen izquierdo de la ventana del código.
- Seguimientos de cambios: el color del margen izquierdo permite realizar un seguimiento de los cambios realizados en el archivo. Los cambios realizados que no han sido guardados lo indica en barra amarilla en el lado izquierdo.
- Selección de código y texto: seleccione texto en el modo de flujo continuo estándar o continúo cuadrado, selecciona una parte rectangular del texto en lugar de un conjunto de líneas.

NODE.Js

Node.js fue creado por los desarrolladores originales de JavaScript. Lo transformaron de algo que solo podía ejecutarse en el navegador en algo que se podría ejecutar en los ordenadores cómo si de aplicaciones independientes se tratara. Gracias a Node.js se puede ir un paso más allá en la programación con JavaScript, no solo creando sitios web interactivos, sino teniendo la capacidad de hacer cosas que otros lenguajes de secuencia de comandos cómo Python pueden crear.

Tanto JavaScript cómo Node.js se ejecutan en el motor de tiempo de ejecución JavaScript V8 (V8 es el nombre del motor de JavaScript que alimenta Google Chrome). Este motor coge el código JavaScript y lo convierte en un código de máquina más rápido. El código de máquina es de nivel más bajo que la computadora

puede ejecutar sin necesidad de interpretarlo primero, ignorando la compilación y por lo tanto aumentando su velocidad.

Mongodb Compass

Es una aplicación multiplataforma que permite explorar la estructura de documentos de las distintas colecciones que componen una base de datos MongoDB, de manera fácil e intuitiva.

Características:

- Está disponible para sistemas operativos Linux, Mac y Windows.
- Tiene soporte para las últimas versiones de MongoDB.
- Interfaz intuitiva que permite visualizar y trabajar con datos fácilmente.
- Permite realizar algunas consultas (insertar, modificar y eliminar datos) de forma visual sin necesidad de saber la sintaxis de MongoDB.
- Permite ver el rendimiento de las consultas de manera visual.
- Muestra estadísticas del servidor en tiempo real que permiten ver información sobre el estado del servidor.
- Permite escribir reglas de validación en formato JSON.
- Permite exportar datos a formato JSON y CSV

Mongodb

Es una base de datos de documentos que ofrece una gran escalabilidad y flexibilidad y un modelo de consultas e indexación avanzado, este proporciona a los desarrolladores todas las funcionalidades que necesitan para satisfacer los requisitos más complejos a cualquier escala.

Insomnia: Es un cliente api REST multiplataforma con un interfaz clara y sencilla. Con funcionalidades que nos va a facilitar el trabajo las principales funcionalidades son:

Personalización, atajos de teclado, importar y exportar, configuración proxy, admite complementos, ayudas para la autenticación

Nodemoon

Es una utilidad de interfaz de línea de comandos (CLI) desarrollado por @rem que envuelve su aplicación Node, vigila el sistema de archivos y reinicia automáticamente el proceso

Dotenv

Es una forma muy eficiente de manejar las variables de entorno, usando una herramienta externa.

Express

Express es un Framework de aplicaciones web extremadamente ligero y minimalista que presenta grandes ventajas, pero también, para poder exprimir todo su potencial, necesitaremos añadir bibliotecas y características de terceros. Es un marco de desarrollo minimalista para Node.js que permite estructurar una aplicación de una manera ágil, nos proporciona funcionalidades como el enrutamiento, opciones para gestionar sesiones y cookies, etc.

Bcrypt

Una función utilizada en nuestra API de hashing de contraseñas, que lleva incorporada un valor llamado Salt que es un fragmento aleatorio que se usará para generar el hash asociado a la password y se juntará junto con ella a la base de datos, para evitar que dos contraseñas iguales generen el mismo hash.

Body-parser

Es una librería que contiene información desde una petición tipo POST cuando el cliente desea crear una nueva identidad, registro o actualizar uno existente mediante el método PUT. Se debe instalar body-parser y habilitar json

Cors

Es un intercambio de recursos de origen cruzado, es una característica de seguridad del navegador que restringe las solicitudes HTTP de origen cruzado que se inician desde secuencias de comandos que se ejecutan en el navegador.

5.3 LENGUAJE DE PROGRAMACIÓN

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

La versión utilizada es ECMAScript 2021. Por si solo es bastante compacto, aunque muy flexible, y los desarrolladores han escrito gran cantidad de herramientas encima del núcleo del lenguaje de JavaScript, desbloqueando una gran cantidad de funcionalidad adicional con un mínimo esfuerzo.

Una **función** en **JavaScript** es similar a un procedimiento, un conjunto de instrucciones que realiza una tarea o calcula un valor, pero para que un procedimiento califique como **función**, debe tomar alguna entrada y devolver una salida donde hay alguna relación obvia entre la entrada y la salida.

Características de JavaScript

- Lenguaje del lado del cliente
- Lenguaje orientado a objetos
- De tipado débil o no tipado
- De alto nivel
- Lenguaje interpretado
- Muy utilizado por desarrolladores

MONGODB

MongoDB (del inglés humongous, "enorme") es un sistema de base de datos NoSQL orientado a documentos de código abierto y escrito en C++, que en lugar de guardar los datos en tablas lo hace en estructuras de datos BSON (similar a JSON) con un esquema dinámico. Al ser un proyecto de código abierto, sus binarios están disponibles para los sistemas operativos Windows, GNU/Linux, OS X y Solaris y es usado en múltiples proyectos o implementaciones en empresas como MTV Network, Craigslist, BCI o Foursquare.

La razón de esto es que MongoDB, al estar escrito en C + +, cuenta con una más que notoria capacidad para aprovechar los recursos de la máquina y, al estar licenciado bajo una licencia GNU AGPL 3.0, es posible adaptarlo a nuestras necesidades.

MongoDB es una base de datos orientada a documentos. Esto quiere decir que en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON.

Características principales:

- Consultas ad hoc. Con Mongodb podemos realizar todo tipo de consultas. Podemos hacer búsqueda por campos, consultas de rangos y expresiones regulares. Además, estas consultas pueden devolver un campo específico del documento, pero también puede ser una función JavaScript definida por el usuario.

- Indexación. El concepto de índices en MongoDB es similar al empleado en bases de datos relacionales, con la diferencia de que cualquier campo documentado puede ser indexado y añadir múltiples índices secundarios.

- Replicación. Del mismo modo, la replicación es un proceso básico en la gestión de bases de datos. MongoDB soporta el tipo de replicación primario-secundario. De este modo, mientras podemos realizar consultas con el primario, el secundario actúa como réplica de datos en solo lectura a modo copia de seguridad con la particularidad de que los nodos secundarios tienen la habilidad de poder elegir un nuevo primario en caso de que el primario actual deje de responder.

- Balanceo de carga. Resulta muy interesante cómo MongoDB puede escalar la carga de trabajo. MongoDB tiene la capacidad de ejecutarse de manera simultánea en múltiples servidores, ofreciendo un balance de carga o servicio de replicación de datos, de modo que podemos mantener el sistema funcionando en caso de un fallo del hardware.

- Almacenamiento de archivos. Aprovechando la capacidad de MongoDB para el balanceo de carga y la replicación de datos, Mongo puede ser utilizado también como un sistema de archivos. Esta funcionalidad, llamada GridFS e incluida en la distribución oficial, permite manipular archivos y contenido.

- Ejecución de JavaScript del lado del servidor. MongoDB tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutadas.

VENTAJAS

- Validación de documentos
- Motores de almacenamiento integrado
- Menor tiempo de recuperación ante fallos

DESVENTAJAS

- No es una solución adecuada para aplicaciones con transacciones complejas
- No tiene un reemplazo para las soluciones de herencia
- Aún es una tecnología joven

6. GLOSARIO

- **PH:** El caballo de fuerza (en inglés mechanical horsepower o imperial horsepower, HP). Se define como la potencia necesaria para elevar verticalmente a la velocidad de 1 pie/minuto un peso de 33 000 libras. Son la verdadera potencia del motor eléctrico, dado que esta es la fuerza que se mantendrá durante el uso normal del compresor, pueden ser de ½, ¼, ⅕, ⅙, ⅛, etc.

- **El compresor:** es el corazón de un frigorífico o refrigeradora. Su función es la de hacer circular el refrigerante por todo el sistema y añadir presión a la parte calefactada del circuito para calentar el refrigerante.

- **BTU:** quieren decir “British Thermal Unit”. Un BTU mide la cantidad de calor que una unidad de aire acondicionado puede extraer de una habitación.

7. BIBLIOGRAFÍA

cloudfare. (sf de sf de sf). *cloudfare*. Obtenido de cloudfare: https://www.cloudflare.com/es-es/learning/security/api/what-is-api-endpoint/

openwibenars. (12 de octubre de 2019). Obtenido de openwibenars: https://openwebinars.net/blog/que-es-mongodb/

openwibernars. (04 de septiembre de 2019). *openwibenars*. Obtenido de openwibenars: https://openwebinars.net/blog/que-es-nodejs/

Vega, L. (2022). *mundohvacr.* Recuperado el 18 de Octubre de 2022, de mundohvacr: https://www.mundohvacr.com.mx/2007/11/el-compresor-parte-fundamental-en-los-sistemas-de-refrigeracion/

8. ANEXOS

MANUAL DE USUARIO

UNIVERSIDAD LUTERANA SALVADOREÑA

FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

CÁTEDRA NUEVAS TENDENCIAS DE PROGRAMACIÓN

MANUAL DE USUARIO API COMPRESORES DE REFRIGERADORES

EQUIPO DESARROLLADOR

RODRIGUEZ GONZALEZ JULIO CESAR
RG01136120

MOZO CRUZ, JUDITH ABIGAIL
MC01136470

ZABALETA CERÉN JOSE CRUZ
ZC01135718

SANTOS GARCIA NATALY GEORGINA
SG01135335

LOPEZ BELTRAN EMERSON BLADIMIR
LB01136352

FACILITADOR: LIC. IRWIN GUARDADO

INTRODUCCIÓN

En el presente manual se muestra el proceso de programación de la API COMPRESORES DE REFRIGERADORES. Dicha API pretende almacenar información técnica de los diferentes compresores de refrigeración. Por lo tanto, en este documento se muestra paso a paso las principales funciones de la misma, desde su lógica pensada, hasta la idea plasmada. Cada captura expuesta muestra el proceso realizado con su respectiva explicación para futuras consultas.

OBJETIVOS

Objetivo general:
Conocer las principales funciones de la API, permitiéndole al usuario manipularla efectivamente.

Objetivos específicos:

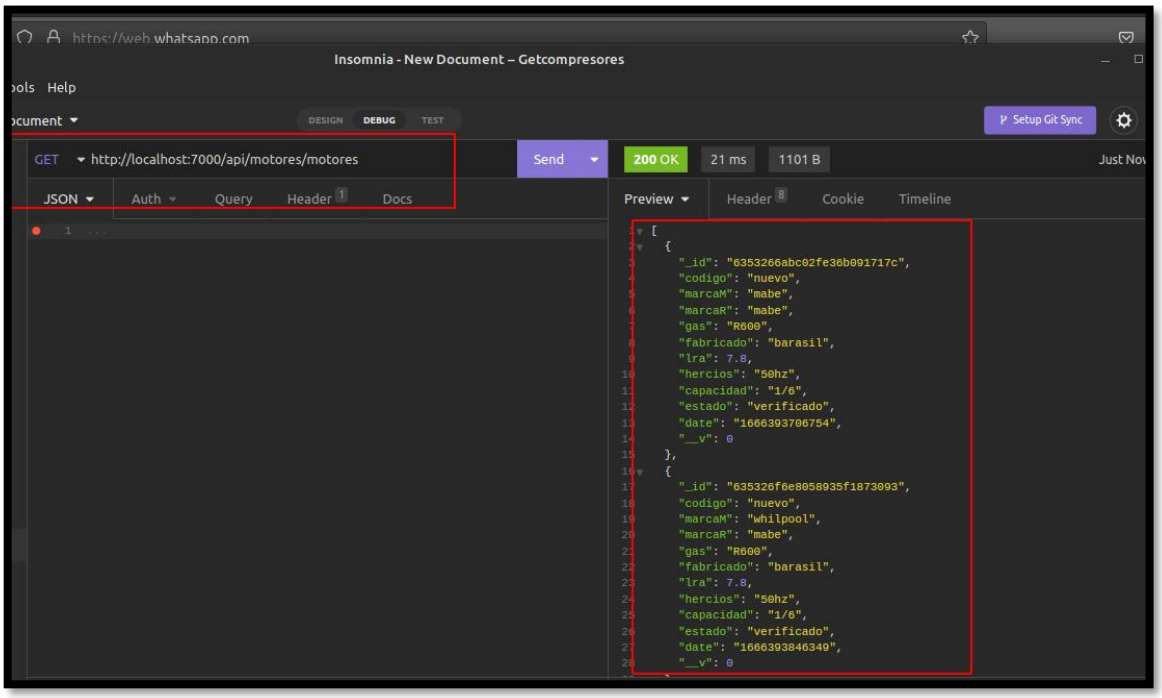
- Que el presente manual sea de beneficio para el usuario o desarrollador, en cuanto al uso y manejo de la API.
- Que el usuario adquiera conocimiento de la API a través de la información plasmada en este documento.

DESARROLLO

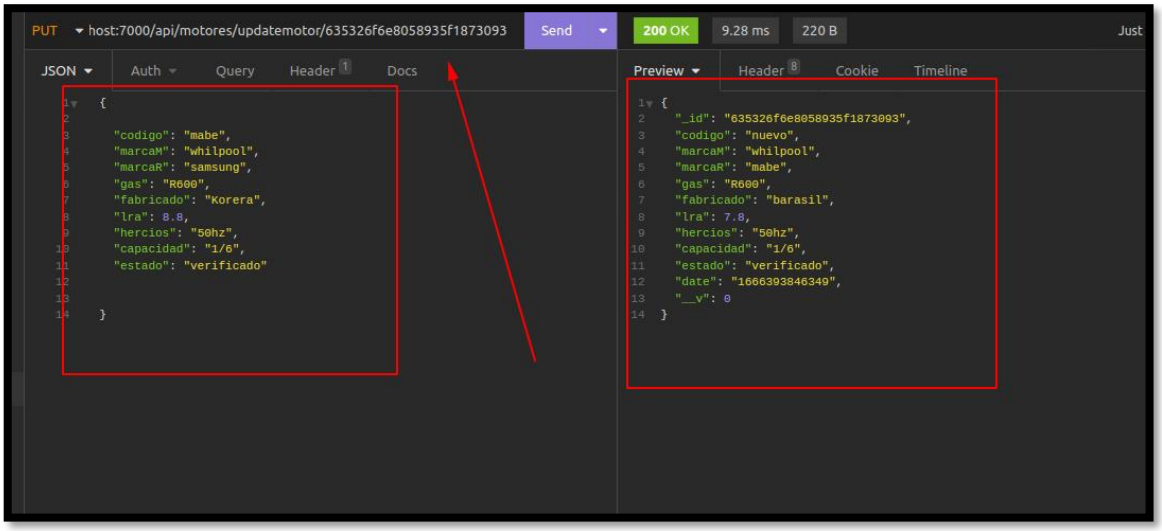
En las siguientes imágenes se muestra el funcionamiento de los EndPoint, los cuales cumplen la función de ser el punto final de la conexión donde se recibe una llamada API.

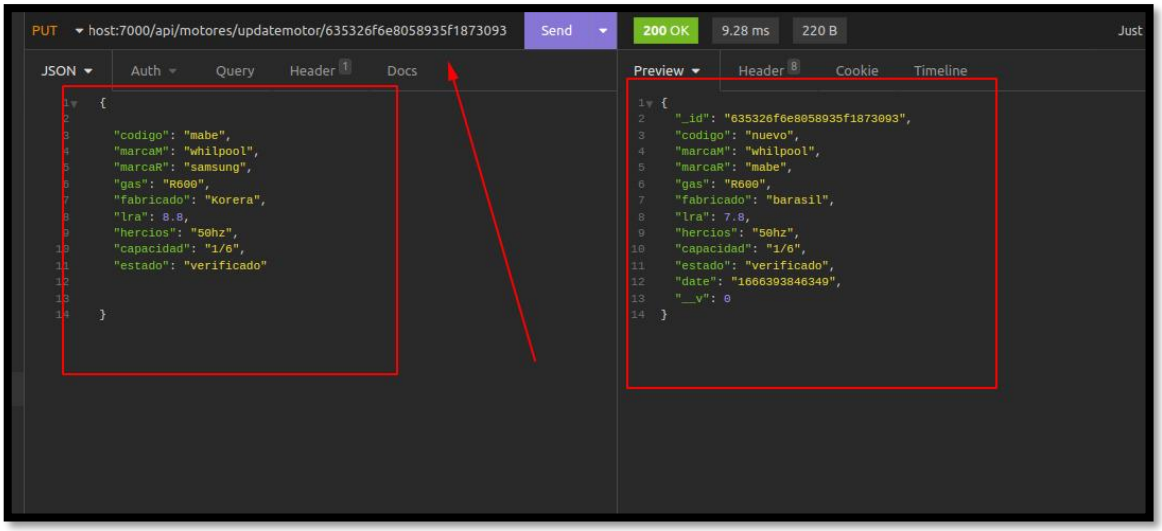
ENDPOINT MOTORES

El método GET, hace la función de enviar los datos al servidor. En la imagen se muestran los datos obtenidos en el JSON, en los campos definidos anteriormente.

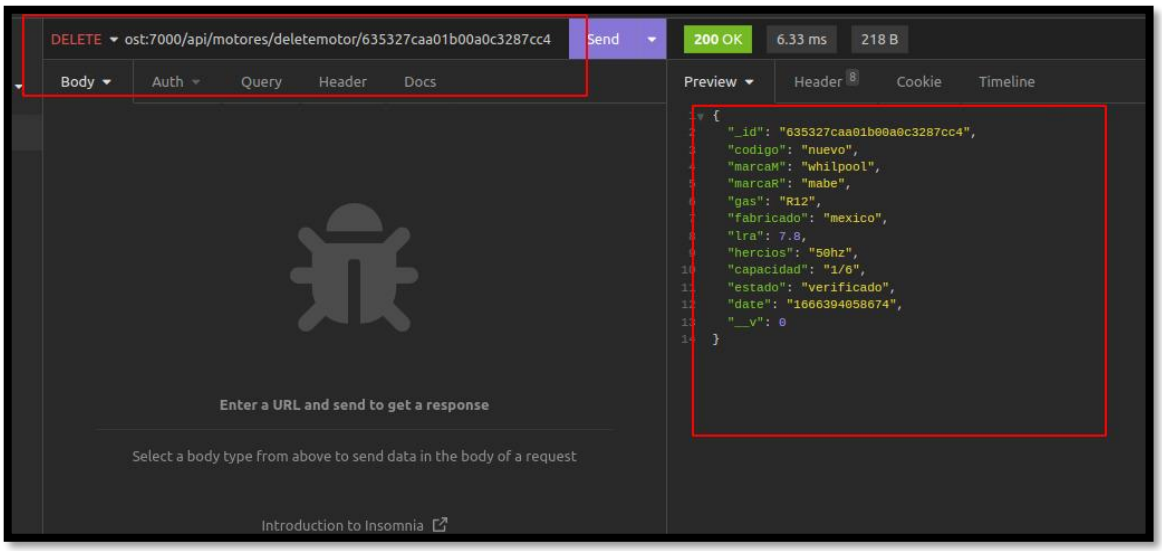


Se muestra el método PUT, previo actualizar y con sus respectivos cambios.



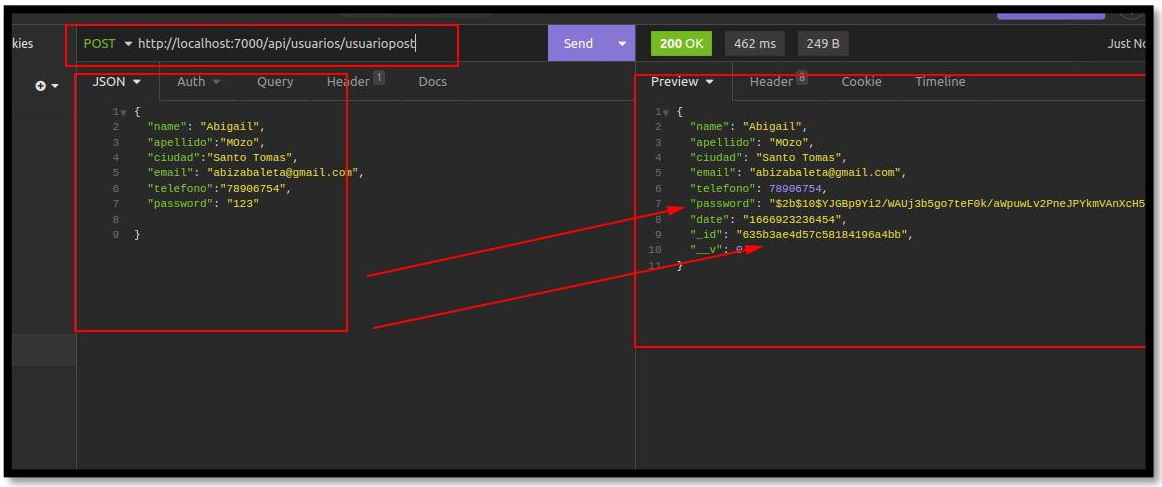


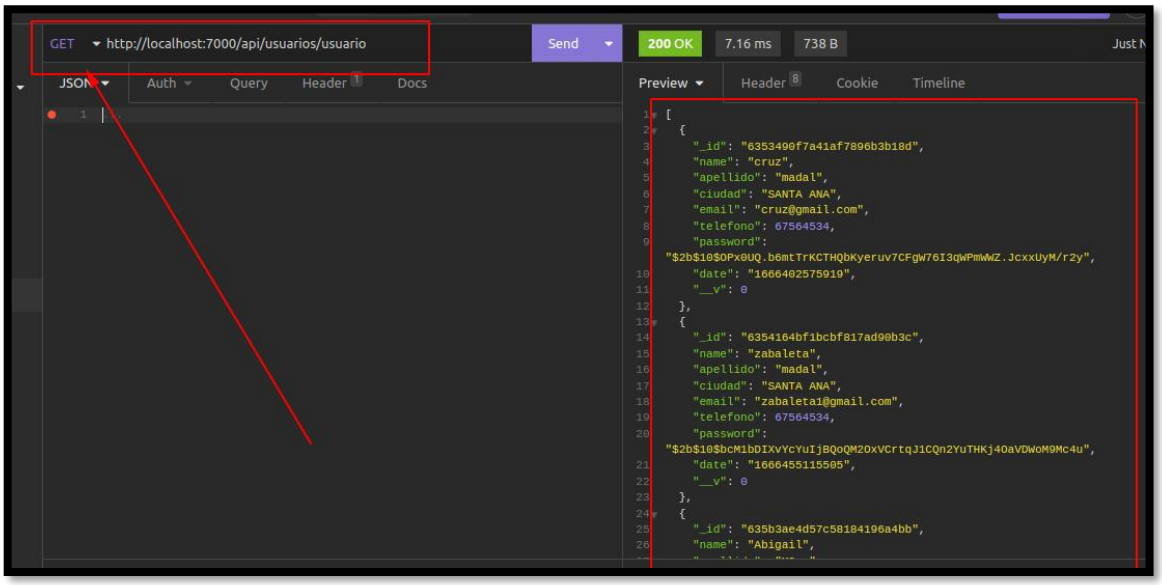
Se borran datos con el método DELETE



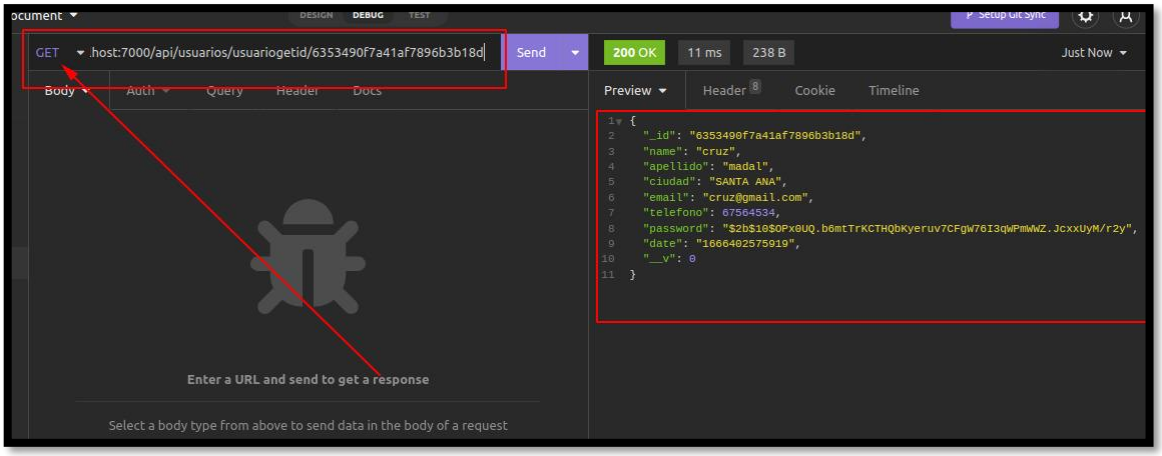
ENDPOINT USUARIOS

Se muestran los datos ingresados en los campos establecidos anteriormente, los cuales son recibidos con éxito.

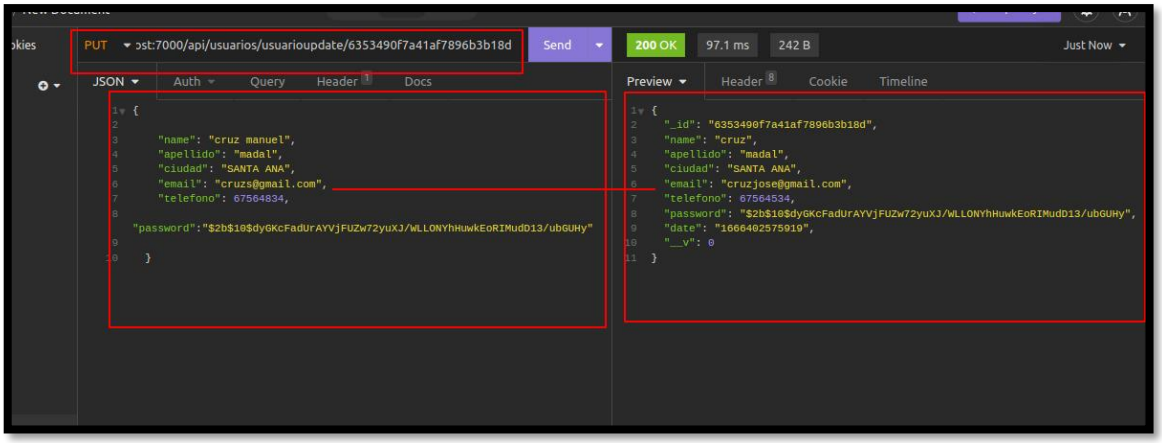




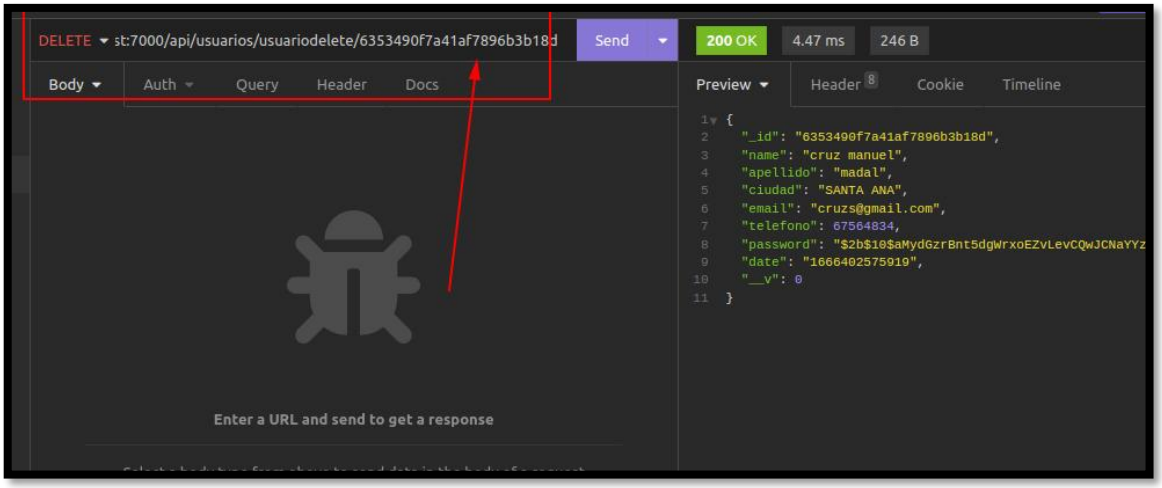
A través del id del usuario, se concreta mandar a llamarlo.



Si el usuario desea actualizar su registro, puede hacerlo siguiendo la siguiente ruta.

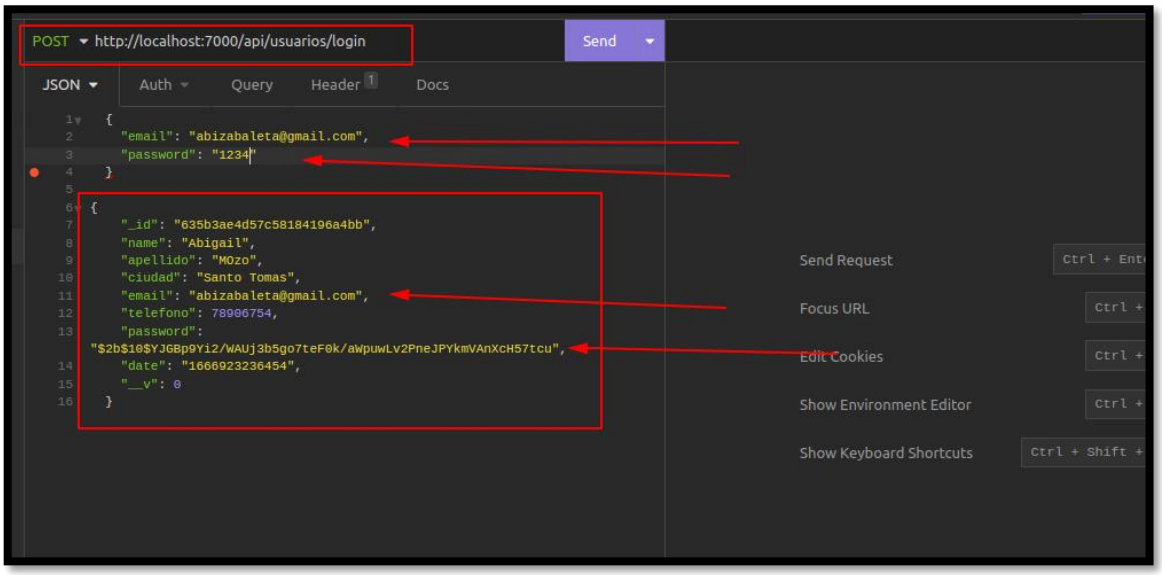


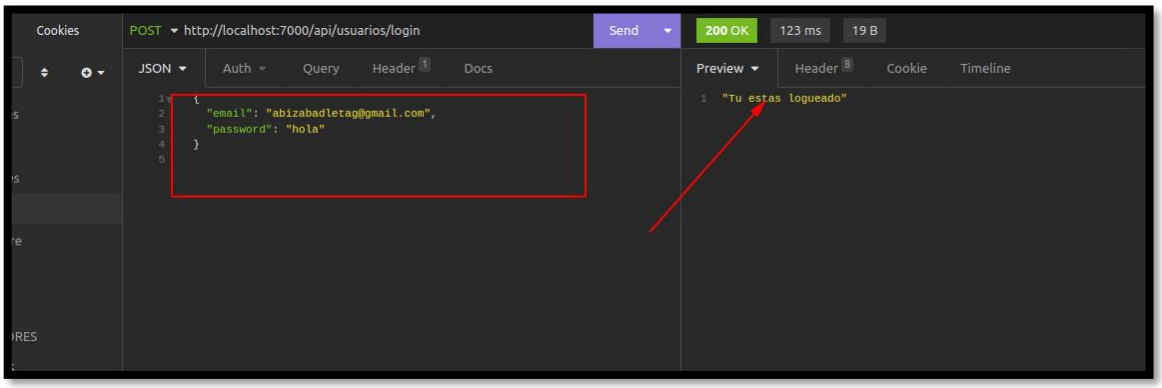
De igual forma, si desea eliminar un registro.



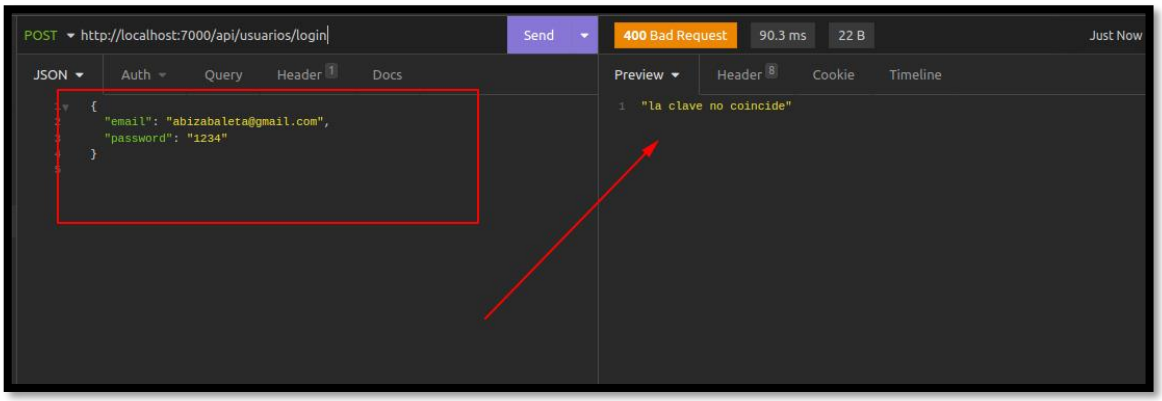
ENDPOINT LOGIN

Con el usuario ingresado anteriormente y sus credenciales correctas, el mismo podrá ingresar a la API.





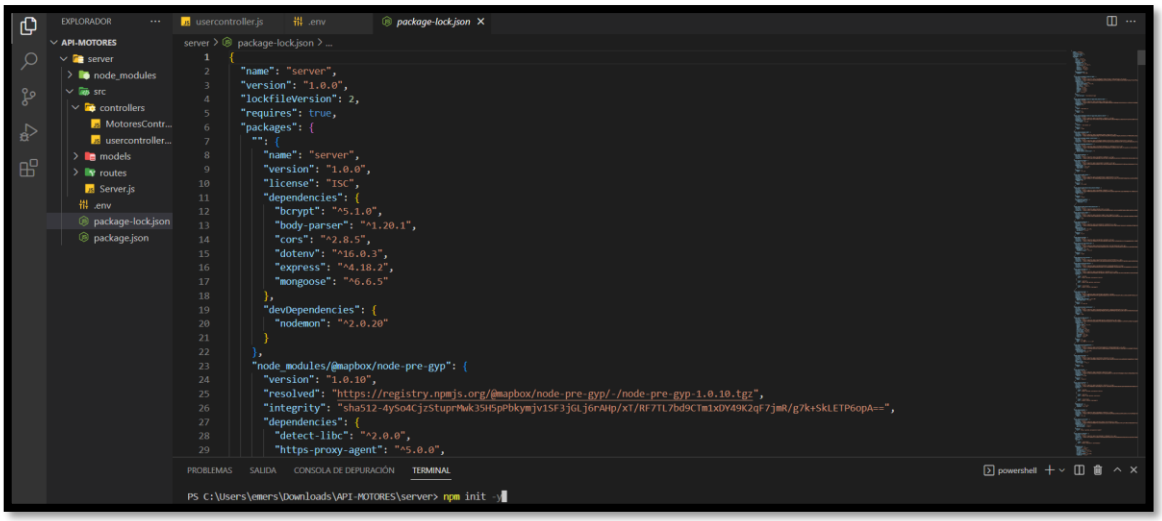
Si las credenciales son incorrectas, la API mandará un mensaje de error.



MANUAL DE DESARROLLADOR

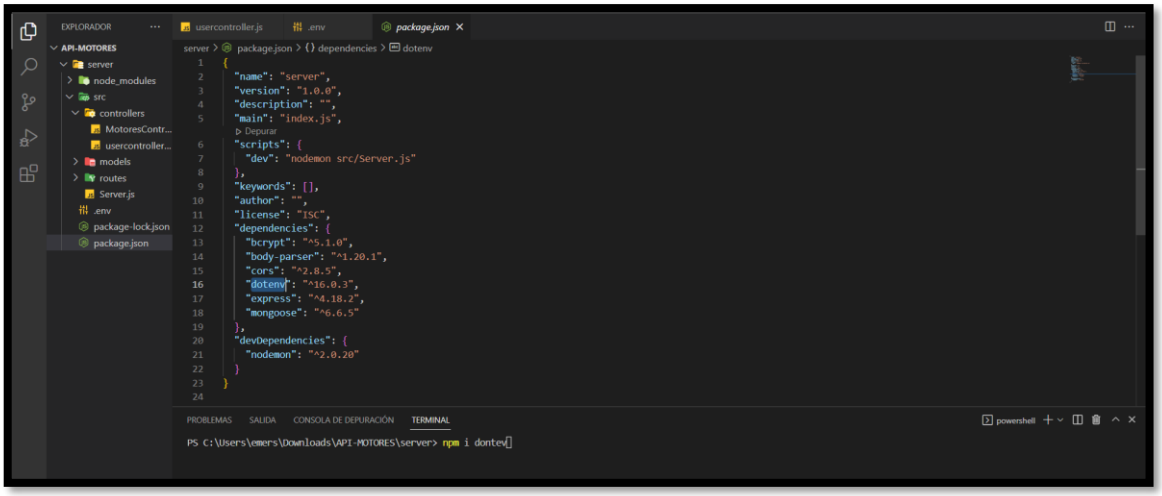
DESARROLLO

Instalación de los paquetes. JSON, con el comando desde la terminal “**npm init -y”.**



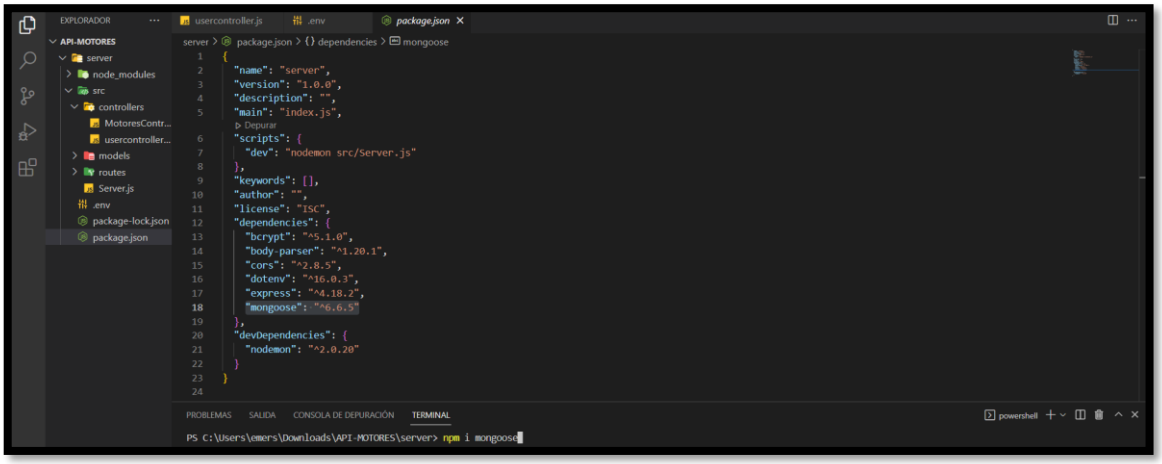
Instalación de la librería de express con el comando **“npm i express”.**

Instalación de dotenv con el comando **“npm i dontenv”.**

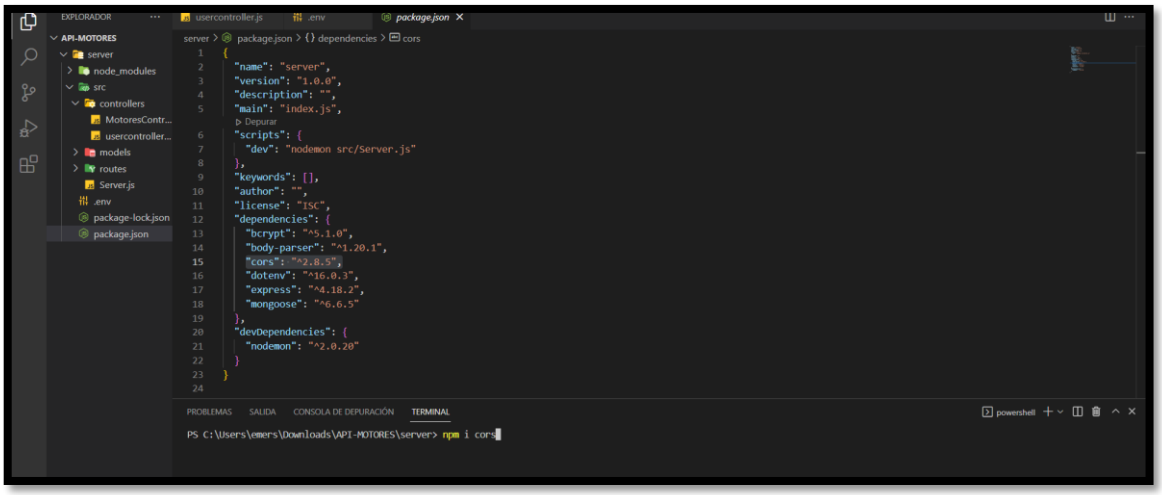


Instalación de dotenv con el comando **“npm i bcrypt”**.

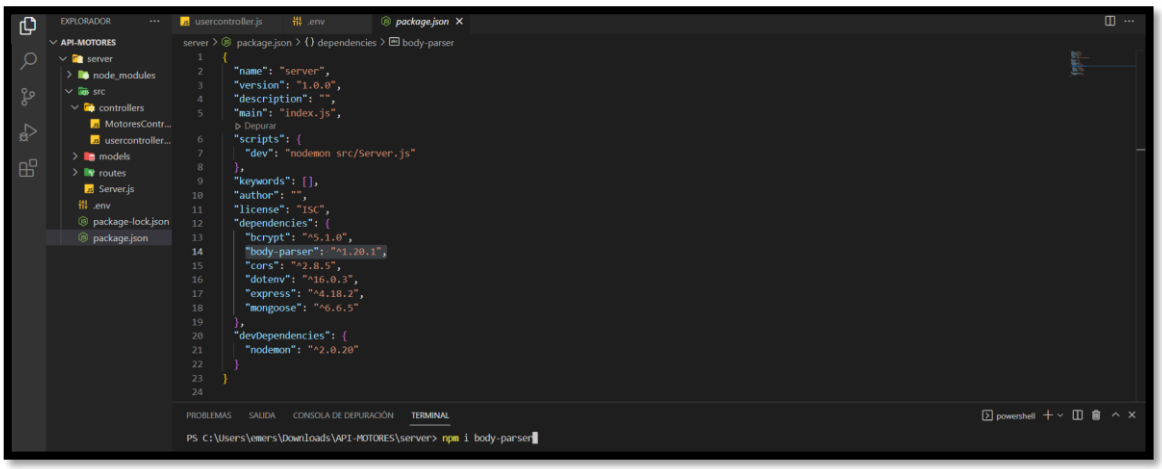
Instalación de Mongoose con el comando **“npm i mongoose”**.



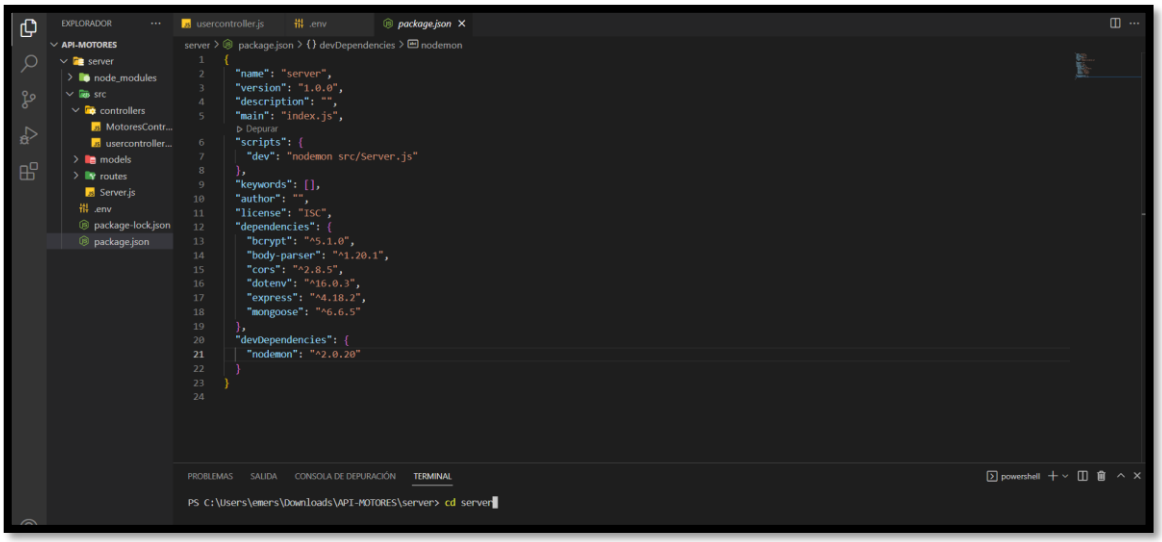
Instalación de cors con el comando **“npm i cors”**.



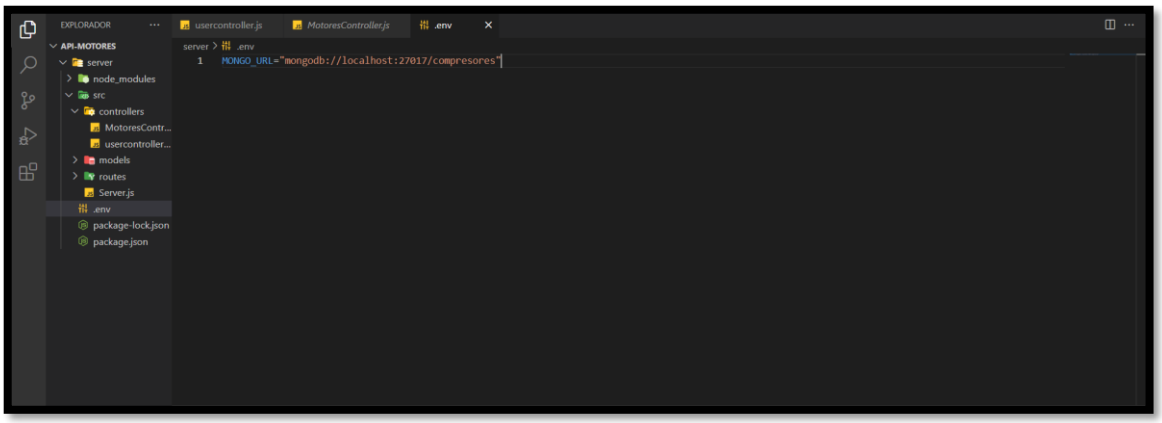
Instalación de boy-parser con el comando **“npm i body-parser”**.



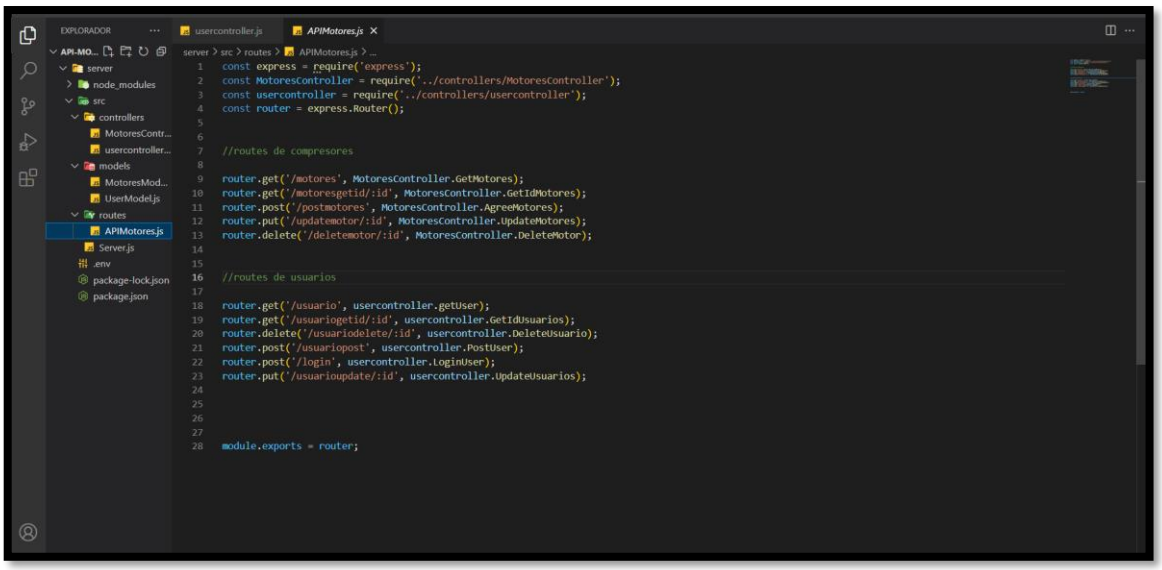
Para iniciar el proyecto, se navega en la carpeta **server** con el comando **“cd server”** y se ejecuta **“npm run dev”** para correr el servidor.



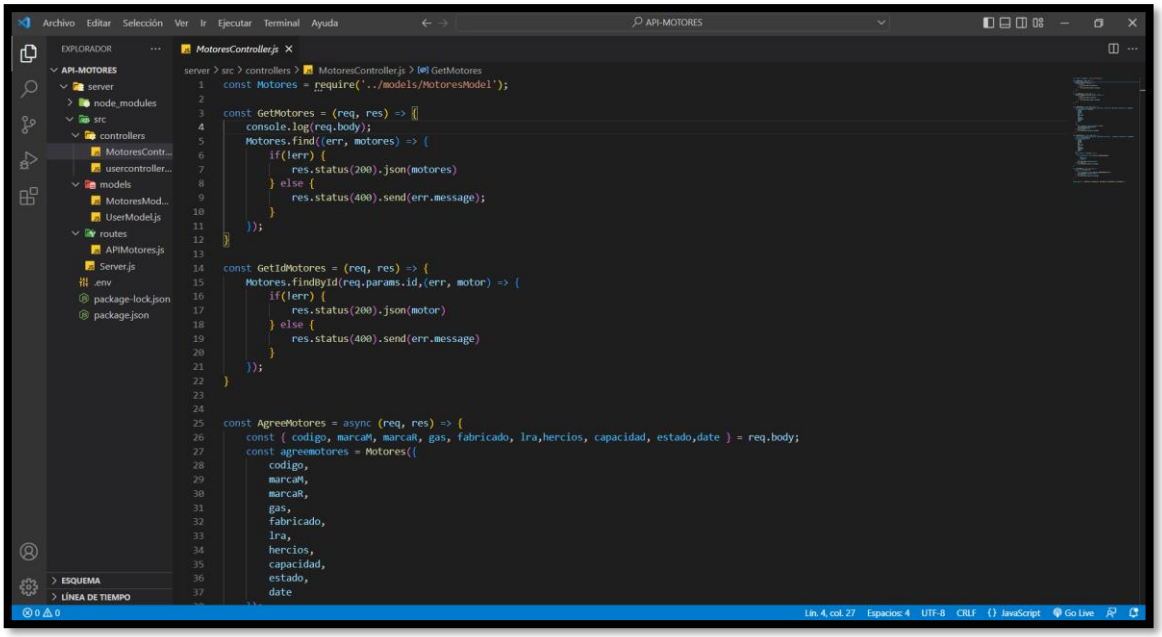
En el archivo. ENV se agrega la **URL** de la base de datos de mongo compas **“mongodb://localhost:27017/compresores”** la cual se crea automáticamente.



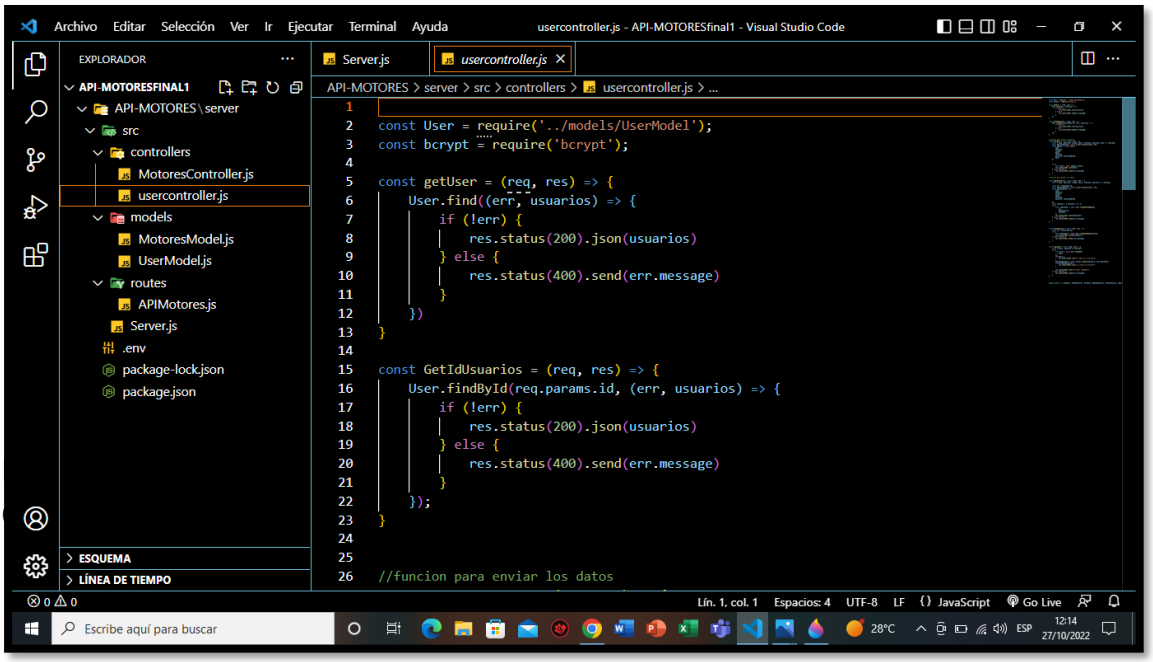
En el archivo **APIMOTORES.js**, se envía las rutas de los motores y del usuario con sus respectivas funciones.



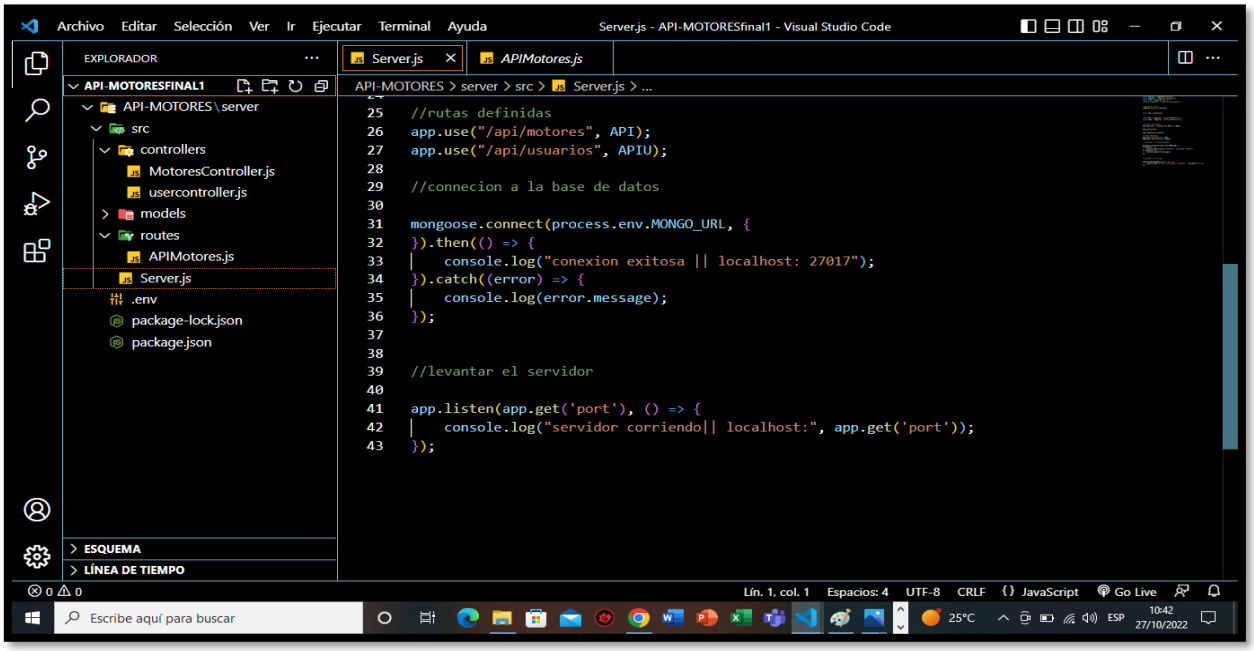
En el archivo **MotoresController.js** se realizan las funciones de los datos del motor: ver, agregar, actualizar y eliminar. De igual forma, se exportan cada una de ellas con **module. exports**.



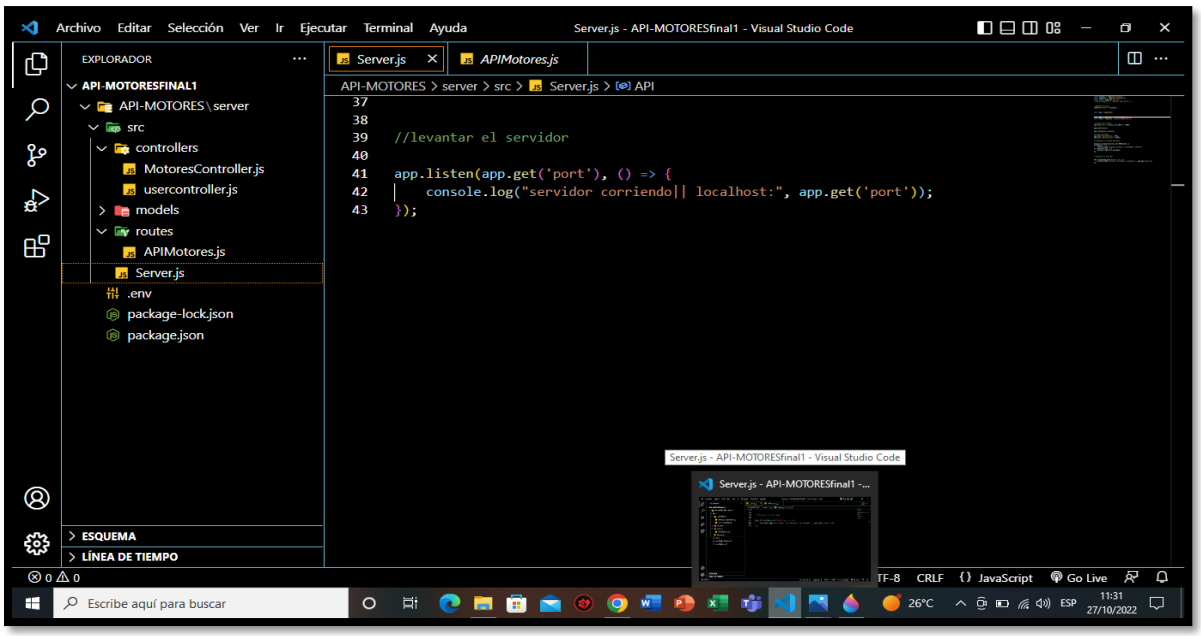
En el archivo **UserController.js** se realiza la validación del email, contraseña y usuario.



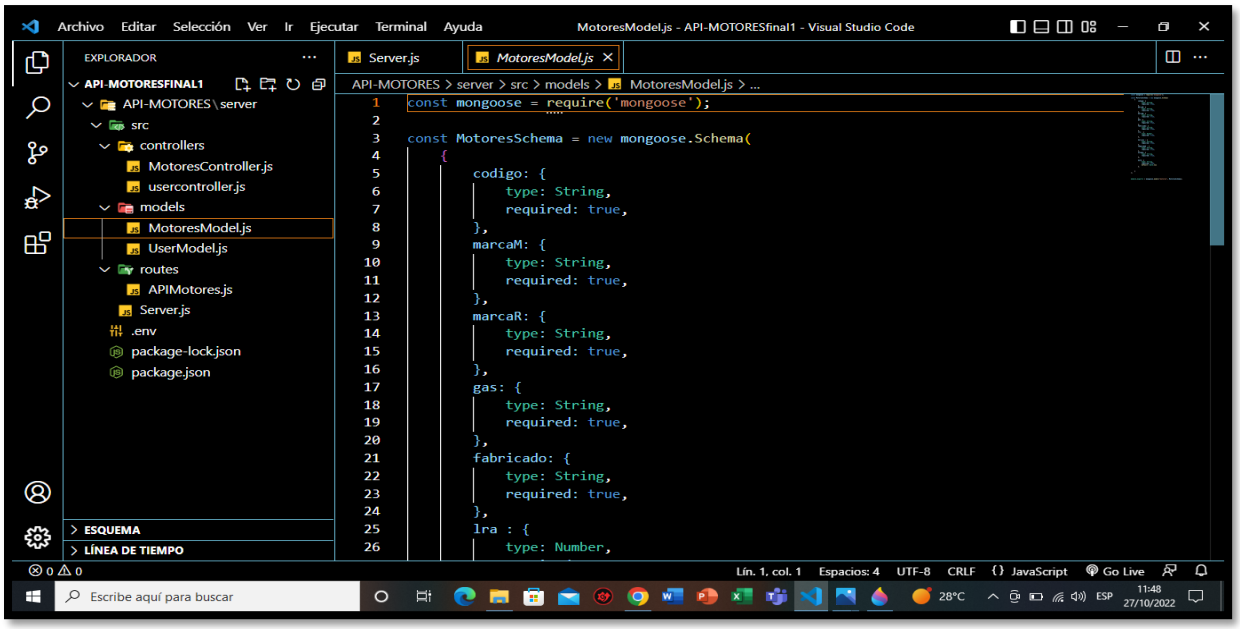
Se muestra la conexión a la base de datos con Mongoose



Se utiliza (app.listen) para visualizar en qué puerto se conecta, esto de forma local.



En el archivo **MotoresModel.js** Mongoose schema, se valida los datos de motores y user.



En el archivo **UserModel.js** se validan los datos del usuario.