Un dibujo de un perro

Descripción generada automáticamente con confianza media

Proyecto Final Backend JS

API para comprar automóviles seminuevos.

**Integrantes:**

* Asael Hernández ([asael.dev@gmail.com](mailto:asael.dev@gmail.com))
* Diego Mendoza ([diegomer4@gmail.com](mailto:diegomer4@gmail.com))
* Ricardo Hernández ([ricardo.hernandez.martinez96@gmail.com](mailto:ricardo.hernandez.martinez96@gmail.com))
* Salvador Nieves ([ing.salvador@hotmail.com](mailto:ing.salvador@hotmail.com))

**Objetivos del proyecto:**

La aplicación a desarrollar tiene como objetivo el poder vender automóviles usados a nuestros distintos clientes, llevar a cabo la venta, gestión de las compras, administración de vendedores y sus ventas para potencializar el negocio y su escalabilidad.

Funcionalidades

* Login de usuarios
* Visualización de autos a elegir
* Compra de autos
* Registro de ventas
* Administración de vendedores y ventas
* Funcionalidades CRUD sobre las tablas establecidas

**PLANEACIÓN: Tablero TRELLO, definición y asignación de actividades**

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

**Definición de plataformas o herramientas a utilizar**

Texto

Descripción generada automáticamente con confianza mediaSe anexa documento Excel en repositorio para mejor visualización

**Épica e Historias de usuario**

|  |  |  |
| --- | --- | --- |
| Diseño Proyecto API para venta de automóviles seminuevos | | |
| Épica | Característica | Historia de usuario |
| API | Generación de información base de datos | 1 |
| Implementación de base datos Postgres sobre Heroku | 2 |
| Generación de modelo base del proyecto | 3 |
| Generación de CRUD | 4 |
| Definición de servicio de obtención de datos basados en selección | 5 |
| Definición de routers | 6 |
| Identificación y protección de información sensible | 7 |
| Gestión de accesos para usuarios autorizados | 8 |
| Deploy de proyecto | 9 |
| Generación de documentación de servicios | 10 |

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamenteCalendario

Descripción generada automáticamente con confianza mediaSe anexa documento Excel en repositorio para mejor visualización

**Tipos de usuarios**

Los usuarios de la API pueden ser administradores o clientes. A continuación, se detalla que permisos tiene cada tipo de usuario dependiendo de su rol:

* Rol de administrador (administrator):

1. Crear, editar y eliminar autos.
2. Obtener, crear, actualizar y eliminar otros usuarios administradores. Debe iniciar sesión para poder realizar estas acciones.
3. Crear, actualizar, eliminar autos. Debe iniciar sesión para poder realizar estas acciones.
4. Actualizar el estado de las compras realizadas por los clientes. Los estados válidos son:
   1. Received – Orden nueva creada por cliente y registrada en el sistema.
   2. Processed – Orden ha sido procesada.
   3. Shipped – El auto de la orden ha sido enviado.
   4. Delivered – El auto de la orden ha sido entregado al cliente.

* Rol de cliente (customer):

1) Ver y filtrar listado de autos.

2)  Comprar autos y dar seguimiento a sus compras.

**Entidades del proyecto**

**Entidades generadas para la definición de base de datos**

* **Users**. Tabla para almacenar los datos generales de los usuarios de la API, esto incluye:
* **Cars**. Tabla para almacenar datos de los automóviles semi-nuevos que están en venta.
* **Sales**. Tabla para almacenar los datos de una compra de automóvil realizada por un usuario con el rol *customer*.

**Definición de diagrama Entidad-Relación**

Diagrama, Tabla

Descripción generada automáticamente

**Creación de proyecto y DB Posrgres sobre Heruku**

Interfaz de usuario gráfica, Texto, Aplicación, Teams

Descripción generada automáticamente

**Conexión a la DB Postgres Heroku a través de PgAdmin4**

Interfaz de usuario gráfica, Texto, Aplicación

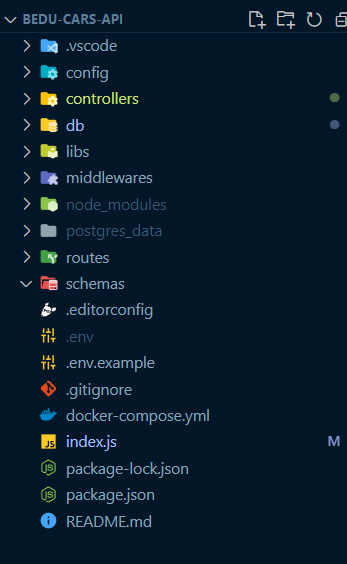
Descripción generada automáticamente

**Uso de Mockaroo para poblar base de datos**

Tabla

Descripción generada automáticamente

**Estructura del proyecto**



**Dependencias del proyecto**

Texto

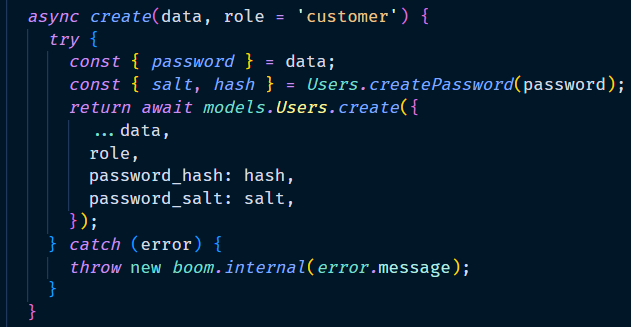
Descripción generada automáticamente

**Definición de conexión a la base de datos usando Sequelize**

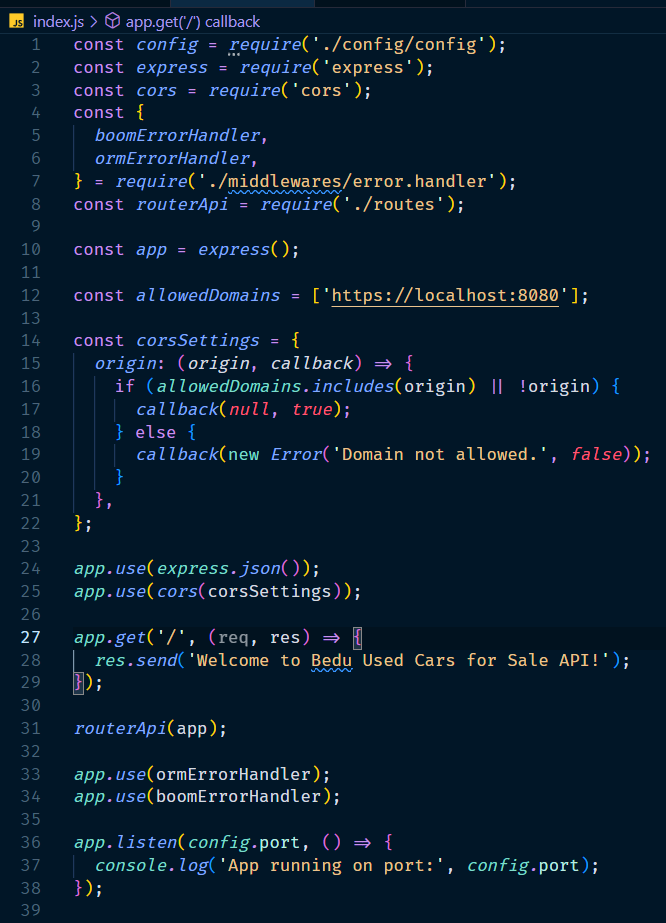
**(EVIDENCIA ASAEL)**

**Definir consultas usando los métodos de Sequelize**





**Definición y configuración de Express**



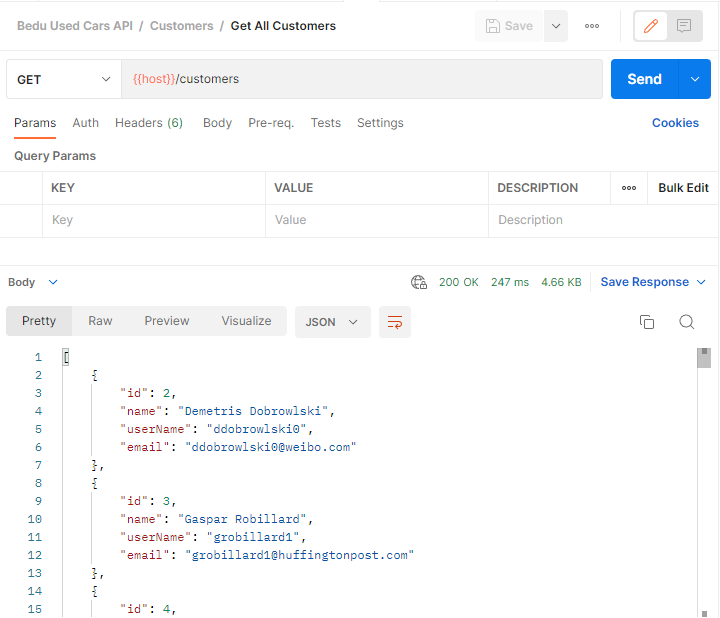
**Validación y petición GET a URL**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

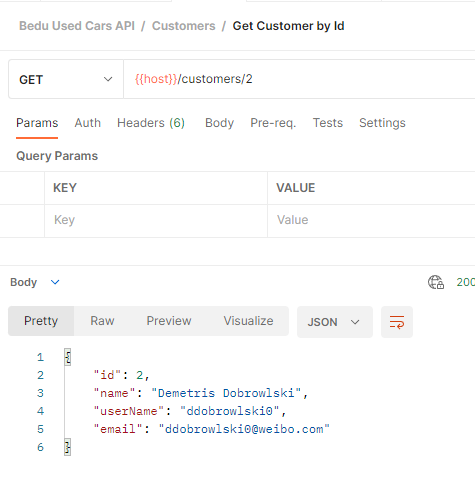
Descripción generada automáticamente

**Validación de servicios CRUD**

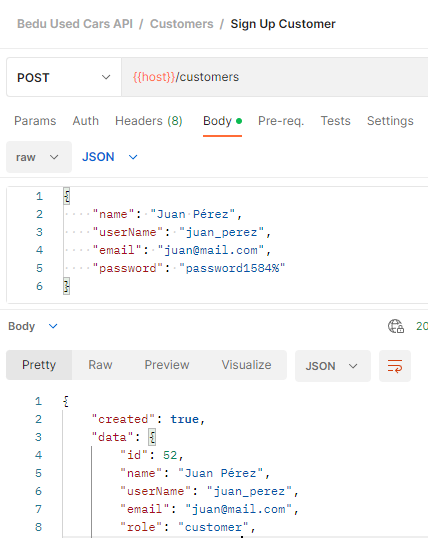
**Get All Customers**



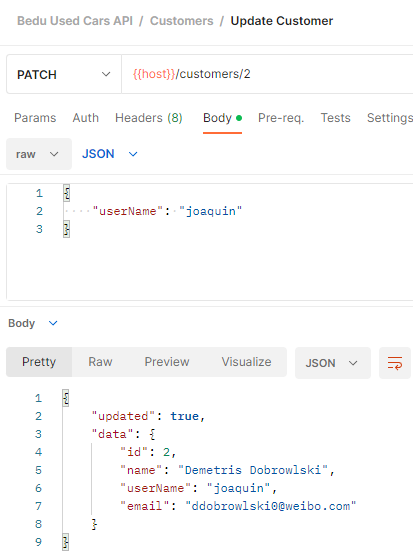
**Get Customer by Id**



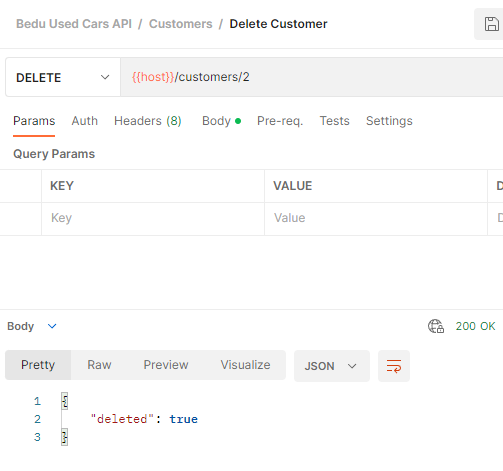
**Sign Up Customer (Create Customer)**



**Update Customer**



**Delete Customer**



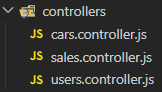
**Definición y validación de servicios por campo**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

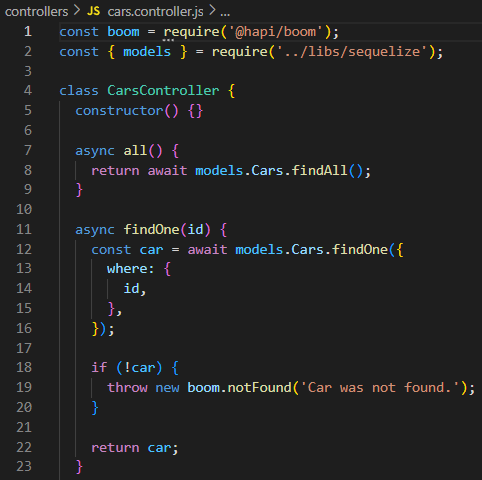
Descripción generada automáticamente

**Controladores, Modelos y Routers por entidad**

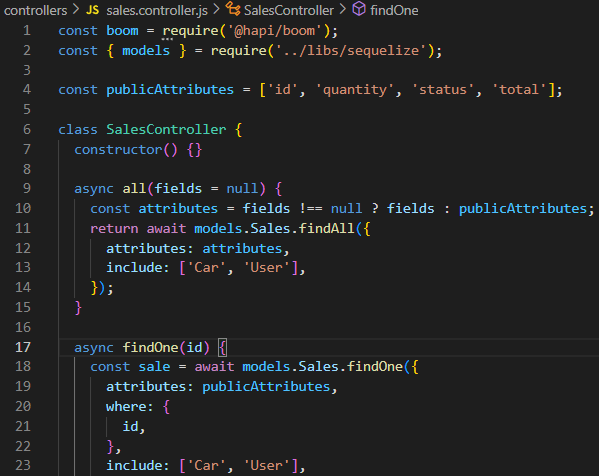
* **Controladores**



* + Cars.controller.js



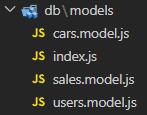
* + Sales.controller.js



* + Users.controller.js



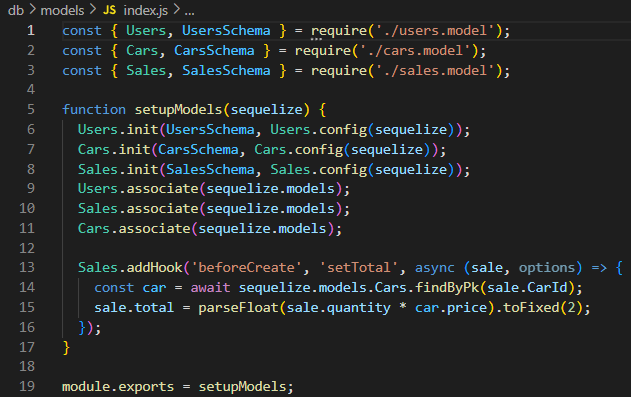
* **Modelos**



* + Cars.model.js



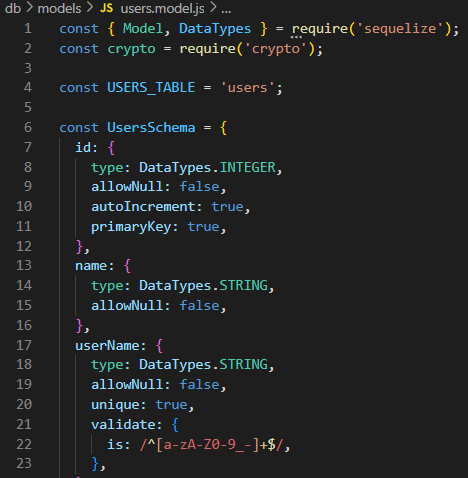
* + Index.js



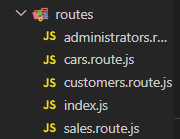
* + Sales.model.js



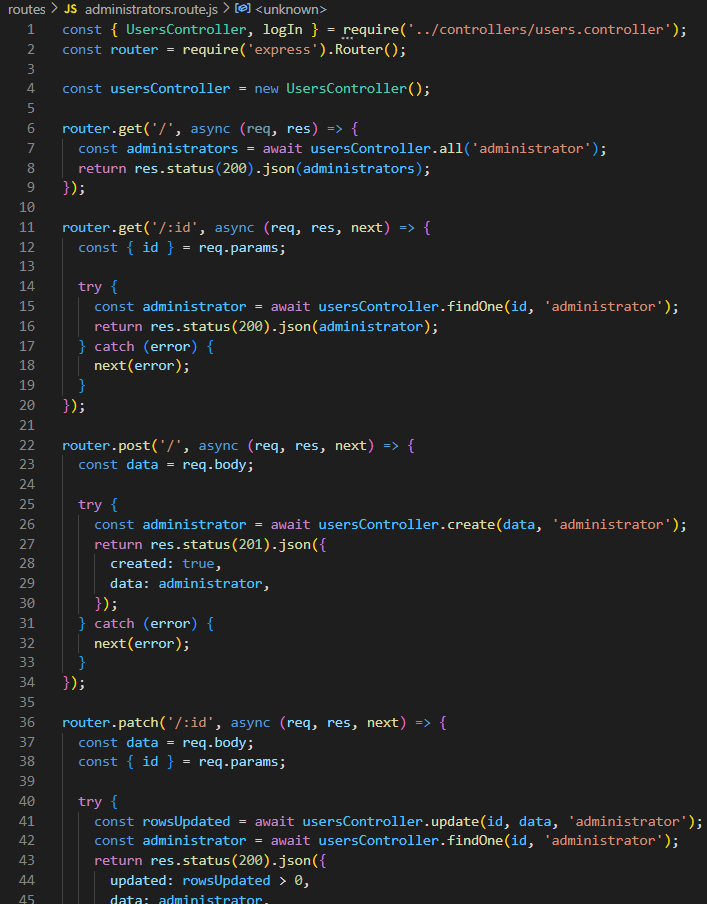
* + Users.model.js



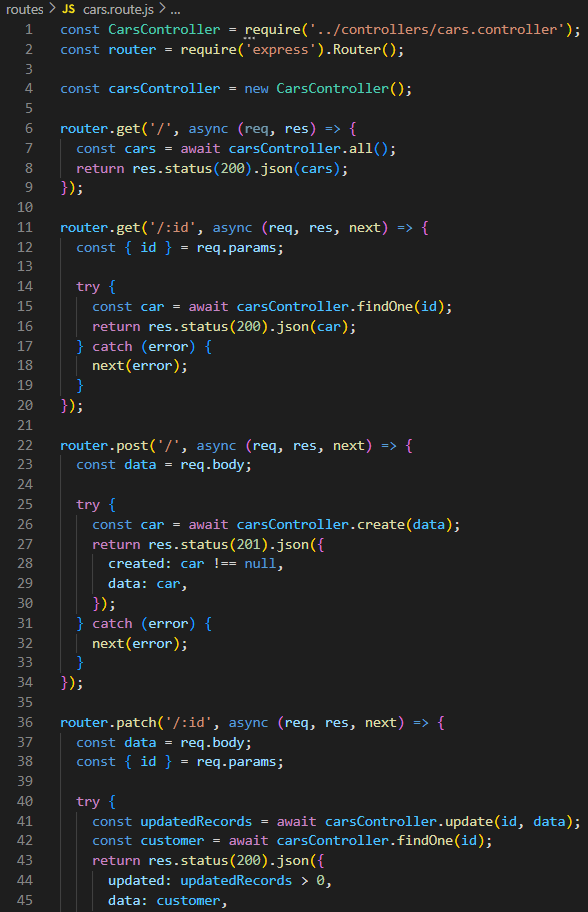
* **Routers**



* + Adminstrators.route.js



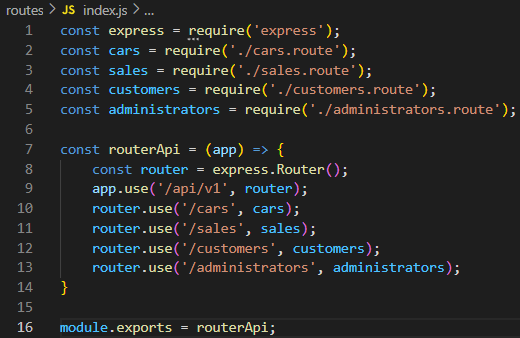
* + Cars.route.js



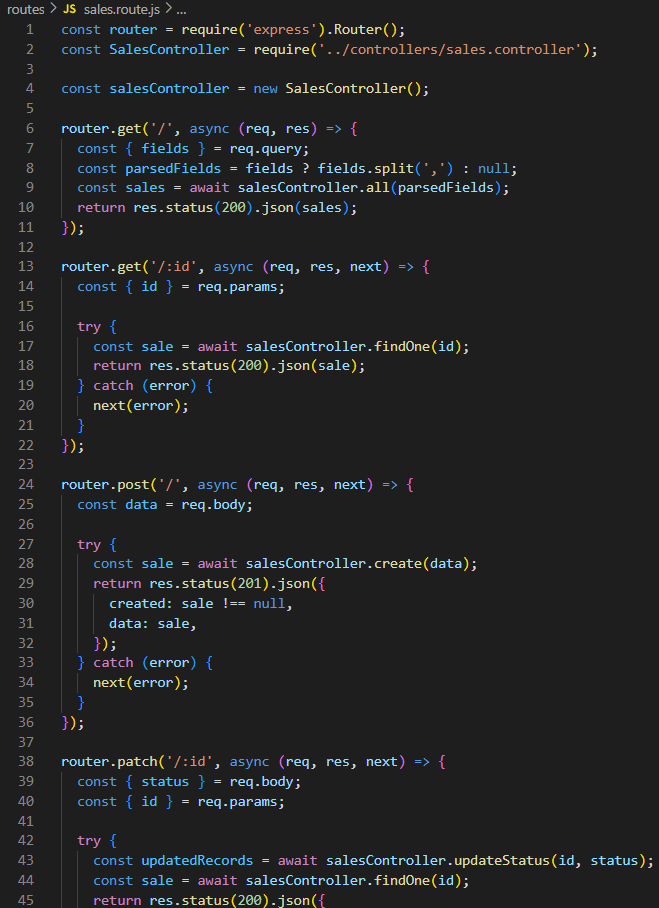
* + Customers.route.js



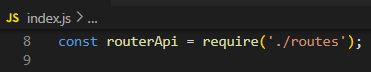
* + Index.js

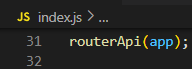


* + Sales.route.js



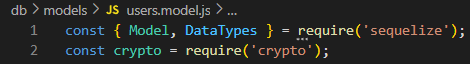
**Configuración de Router**

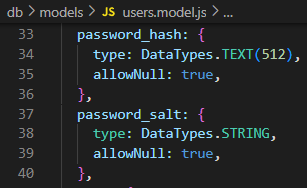


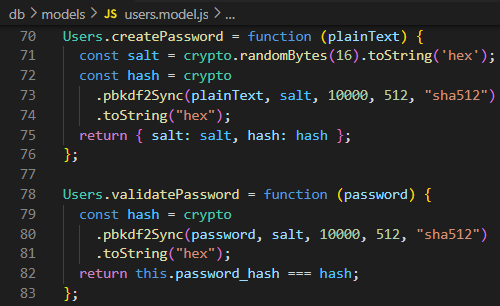


**Definición y configuración de protección de información a través de plataforma Cripto**

* Se identificó como datos sensibles las contraseñas de los *users* dentro de la API por lo que se realizó una configuración de protección de datos.







**Definición y configuración de Passport**

**(Evidencia Ricardo)**

**Definición de accesos por servicio**

**(Evidencia Ricardo)**

**Implementación de control de acceso**

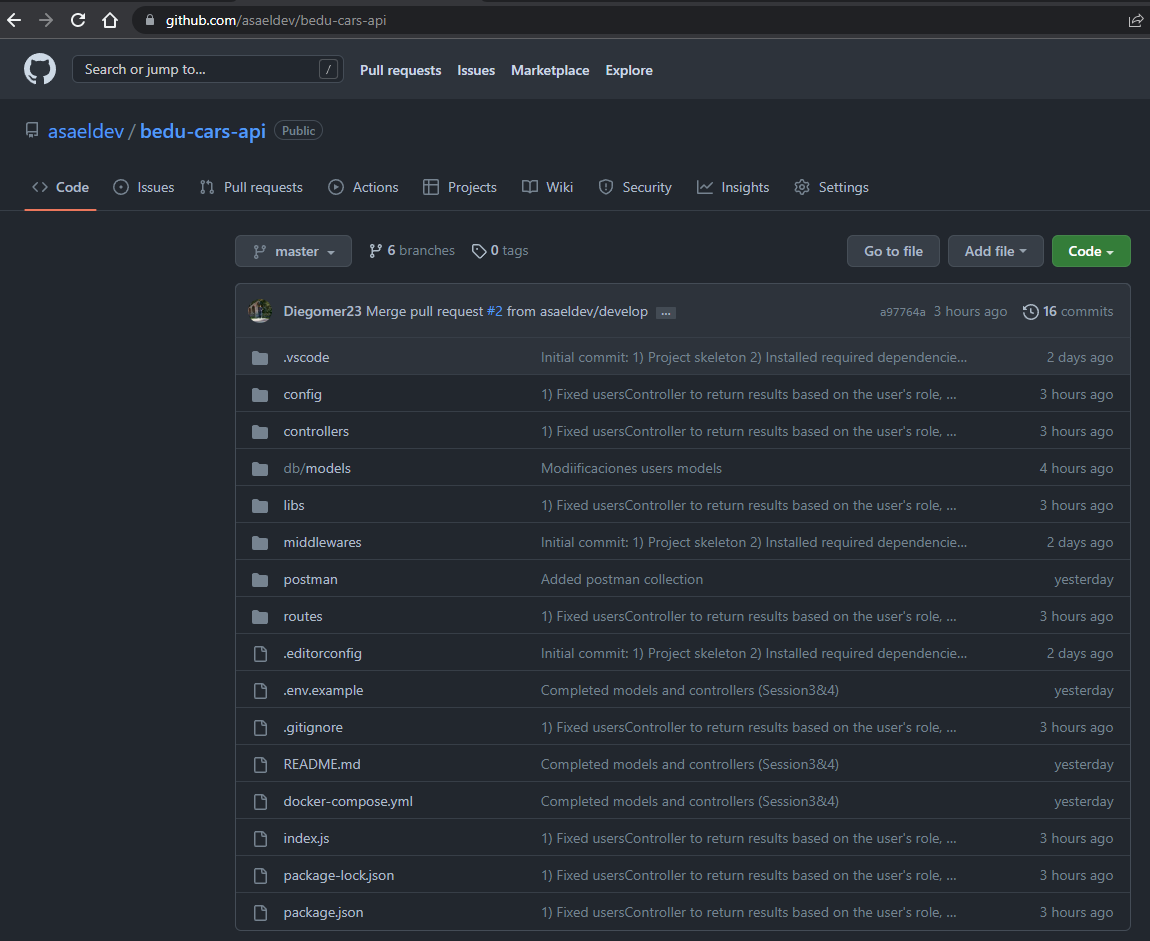
**(Evidencia Ricardo)**

**Definición de variables de entorno (ocultar la información confidencial)**

**(Evidencia Ricardo)**

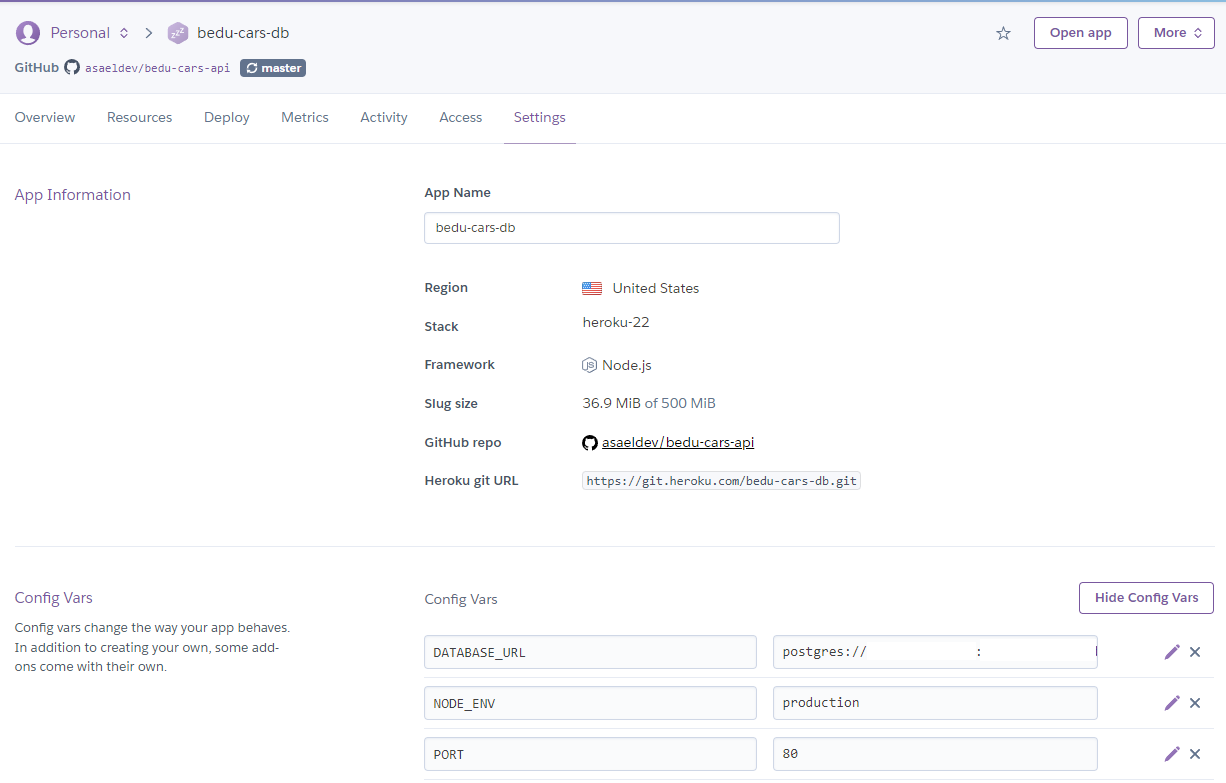
**Generación de repositorio de GitHub con código de App**

* URL del repositorio: <https://github.com/asaeldev/bedu-cars-api>



**Despliegue de aplicación sobre Heroku**

* API URL: <https://bedu-cars-db.herokuapp.com>



**Validación de servicios Producción**



**Definición de documentación de la aplicación**

**(Evidencia Ricardo)**

**Anexos**

**Conexión a base de datos Heroku**

**Datos de conexión a la base de datos de heroku:**

**Host**: [ec2-44-209-186-51.compute-1.amazonaws.com](http://ec2-44-209-186-51.compute-1.amazonaws.com/)

**Database**: dbp7j7qktqs438

**User**: wvdrdomdthklqb

**Port**: 5432

**Password**:885820e889dba3db8abc229924f089833b6577c9c27699ba1bb392fa8e4223de

**URI**:postgres://wvdrdomdthklqb:[885820e889dba3db8abc229924f089833b6577c9c27699ba1bb392fa8e4223de@ec2-44-209-186-51.compute-1.amazonaws.com](mailto:885820e889dba3db8abc229924f089833b6577c9c27699ba1bb392fa8e4223de@ec2-44-209-186-51.compute-1.amazonaws.com):5432/dbp7j7qktqs438

**Heroku CLI**

heroku pg:psql postgresql-encircled-74502 --app bedu-cars-db

**Tablero TRELLO**

https://trello.com/invite/b/QxZxFLRV/807fead925816ae7d4ba303e54b90464/js-project-backend-4