Logotipo

Descripción generada automáticamenteLogotipo

Descripción generada automáticamente

**INSTUTO TECNOLÓGICO DE NUEVO LAREDO**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**PROGRAMACIÓN MULTIPARADIGMA**

UNIDAD 2

**PRACTICAS**

Docente

Ing. Luis Daniel Castillo García

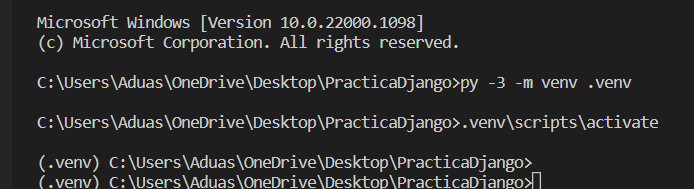
**INTEGRANTES:**

* JUAN ESTEBAN BALTIERREZ TOBON 18100152
* JOSE ANGEL CABRERA MORALES 18100155
* ANTONIO DE LA CRUZ RIVERA 18100163

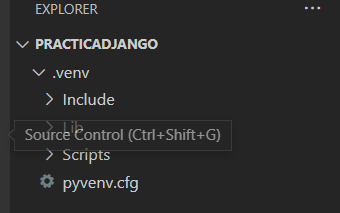
30 de octubre del 2022

**Practica #2  
Realizar una aplicación utilizando el Framework DJANGO y que contenga lo siguiente**

**Instalación del entorno virtual.**

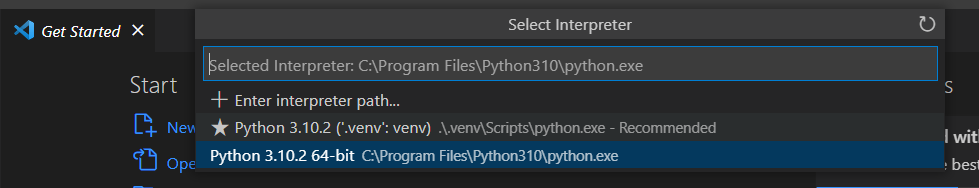


Una vez que se instala, se activa el .venv y nos aparecerá una carpeta.



Posterior a ello, vamos a seleccionar el interpretador, el cual será:

Ctrl + shft + p



Una vez que activamos el interpretador, ejecutaremos primero el siguiente comando:

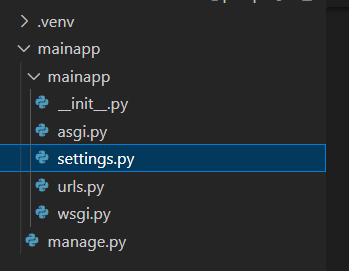
python -m pip install --upgrade pip

Posterior a ello, se ejecuta el comando para instalar django en nuestro entorno.

python -m pip install django

Para los siguientes pasos, vamos a crear la carpeta principal mainapp (la cual es la que se encargara de ejecutar todo entre nuestro entorno virtual y la base de datos).

Una vez que se cree nuestra carpeta de mainapp



**1-. Conexión a base de datos postgresql**

Vamos a settings a configurar nuestra base de datos.

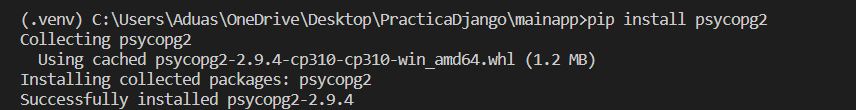
En postgres creamos la base de datos: transportebd

En la carpeta de settings, pasamos a cambiar el apartado de la conexión a la base de datos.



Una vez realizado esto pasaremos a realizar la migración.

Antes de esto, tenemos que instalar psycopg2.



**2. Utilizar al menos 4 entidades**

Viajes Rutas Chofer Ciudad

idViaje idRuta idChofer idCiudad

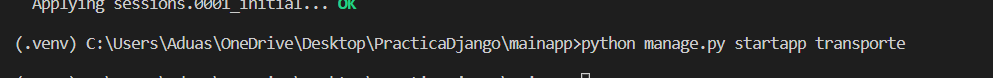
idRuta nombreRuta nombreChofer nombreciudad

idChofer distanciaRuta domicilioChofer

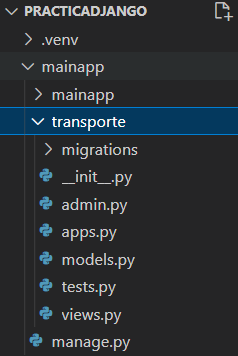
fechaViaje Edad

observaciones idCiudadNacimiento

Una vez que tenemos dise;adas las tablas a utilizar, procedemos a crear dentro de mainapp, nuestra app para esta bd.



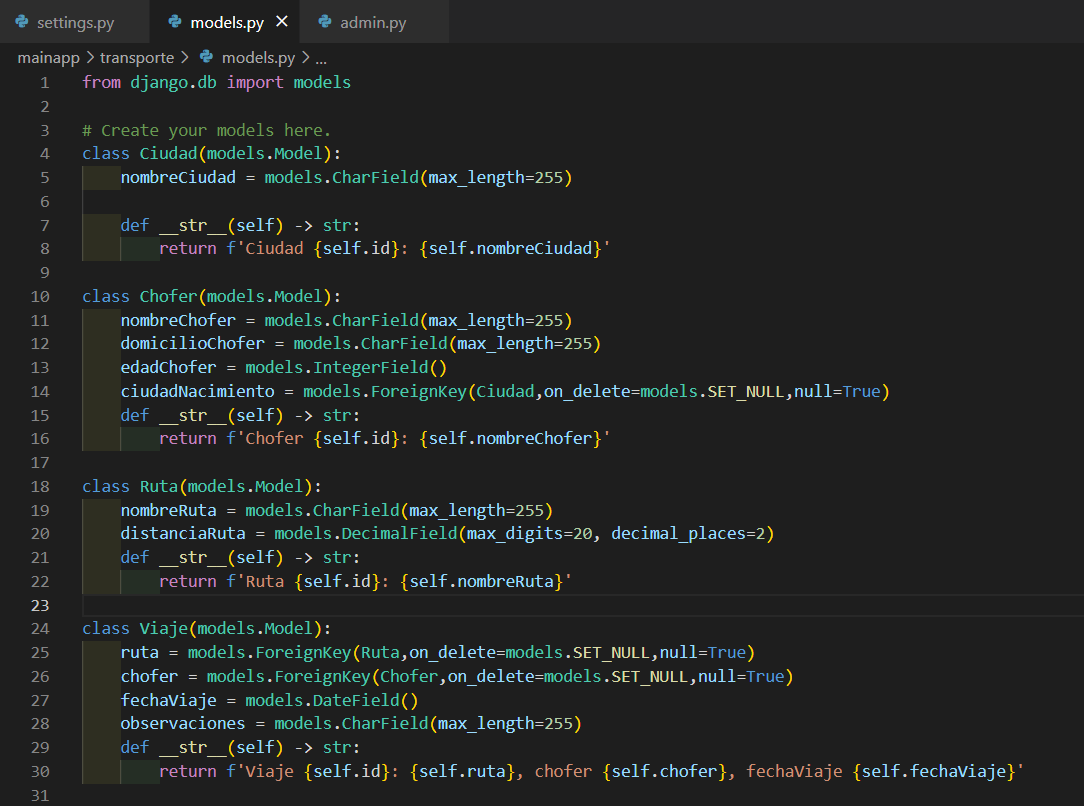
Y nos crea una carpeta, en al cual tenemos actualmente las condiciones necesarias para poder realizar los modelos de nuestras entidades.



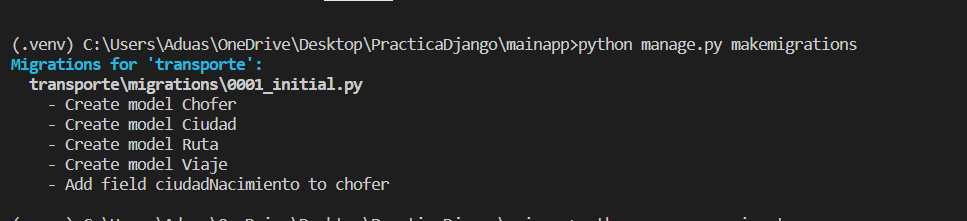
Nos vamos al apartado de models, y crearemos los modelos.

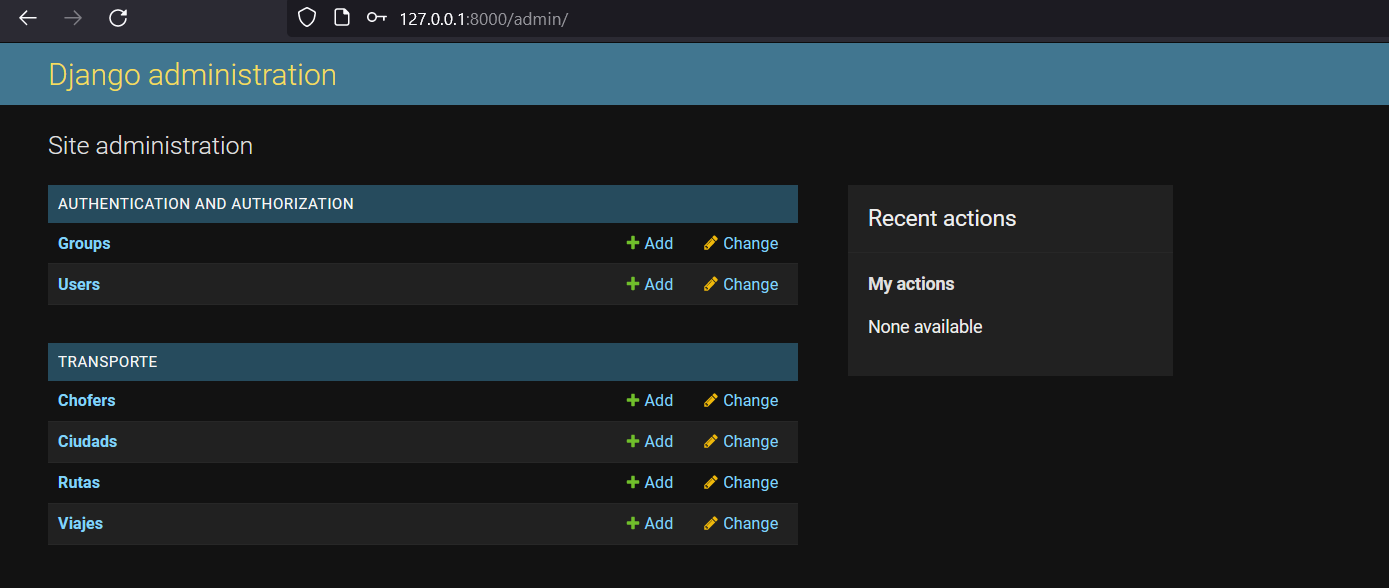
En este caso se creada de manera inversa, debido a que Python es un lenguaje que va de manera secuencial.

**3. Una de las entidades debe relacionarse con otra**

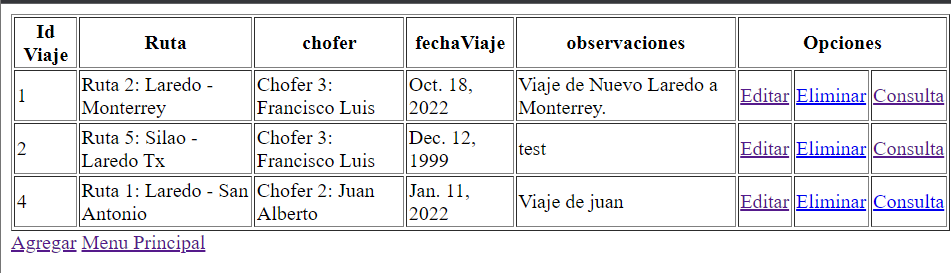
 **4. Realizar migraciones**

Se realiza la migración de las tablas de los modelos anteriores.



**  
5. CRUD de las 4 entidades utilizando templates**

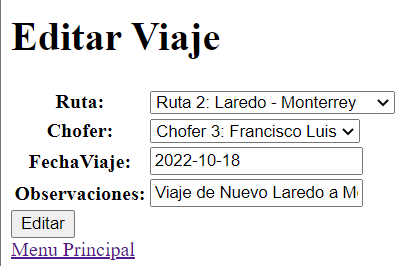
**Viaje.**



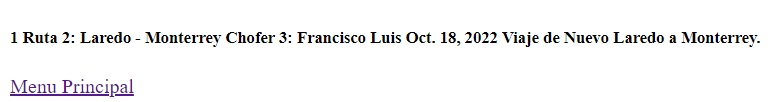
Viaje Agregar



Viaje Editar



Viaje Consulta.

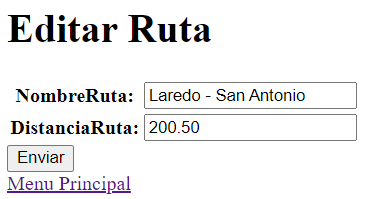


**Ruta**

**Ruta Agregar**



**Ruta Editar**

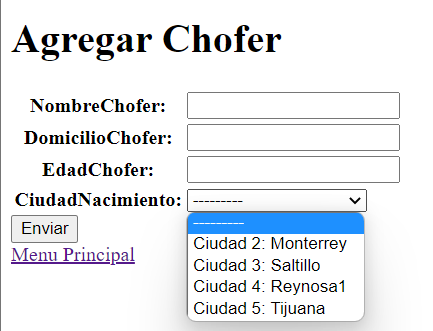


**Ruta Consulta**

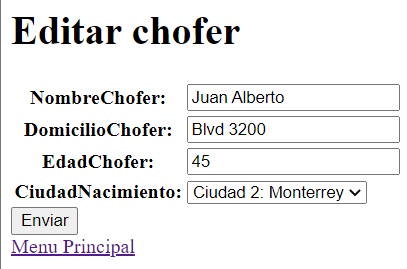


**Chofer**

**Chofer Agregar**



**Chofer Editar**



**Chofer consulta**

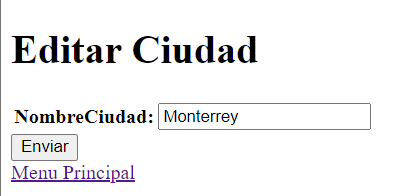


**Ciudad**

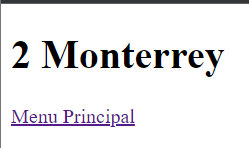
**Ciudad Agregar**



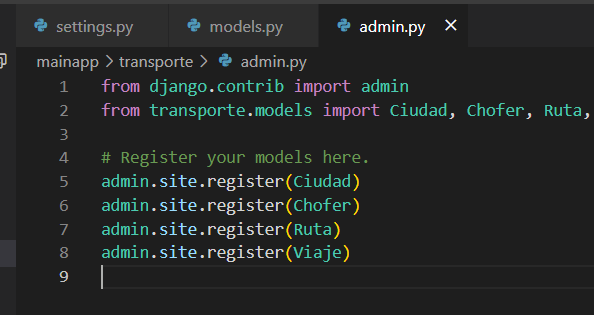
**Ciudad Editar**



**Ciudad Consulta**

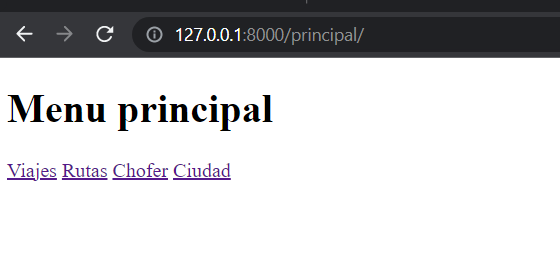
 **6. Agregar las 4 entidades a la página de administración**

Se agregan las páginas a admin.



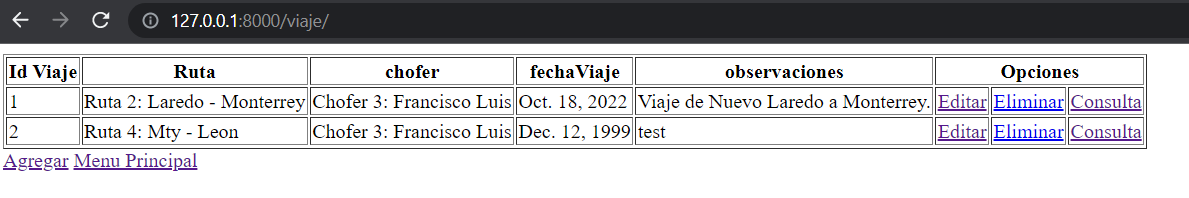
Resultados.

Se crea menú principal.



Cada uno de ellos nos envía a las entidades correspondientes. En cada uno se puede agregar, eliminar, editar y consultar.

Si nos dirigimos al de viaje:

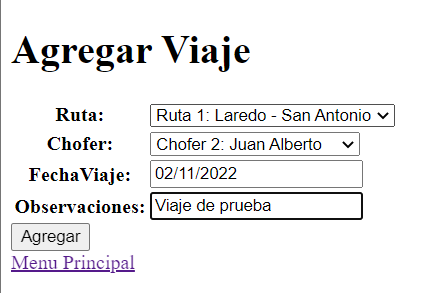


Se ve los viajes agregados, y la información que contiene.

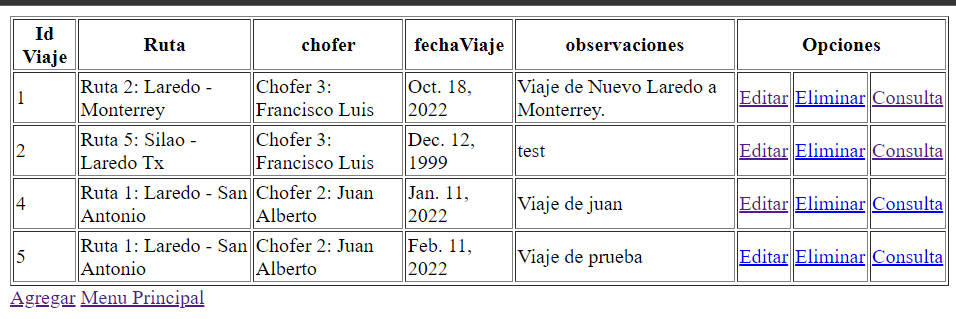
Si nosotros damos clic en Agregar, nos mandara a su formulario:

.

Este form contiene las relaciones con Ruta, y Chofer, por ende, las otras entidades actualmente tienen la funcionalidad de Agregar, Editar, Eliminar y consultar.

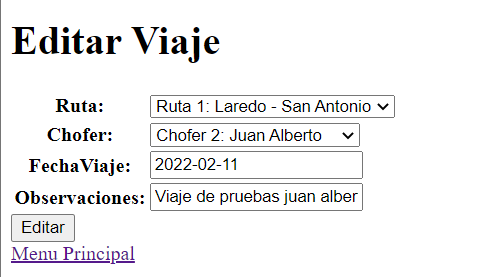


Al momento darle agregar, nos mandara a la pantalla principal de viajes.

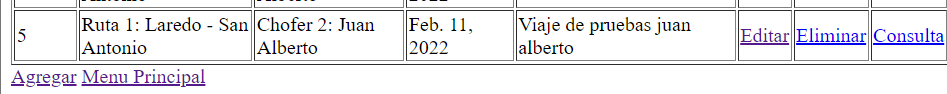


Y se actualiza al instante.

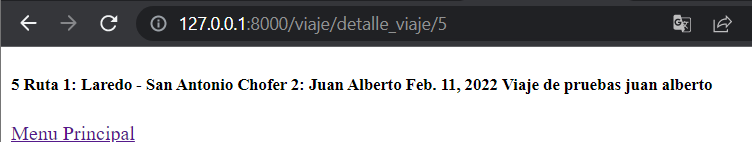
Si nosotros damos clic en el botón de editar.



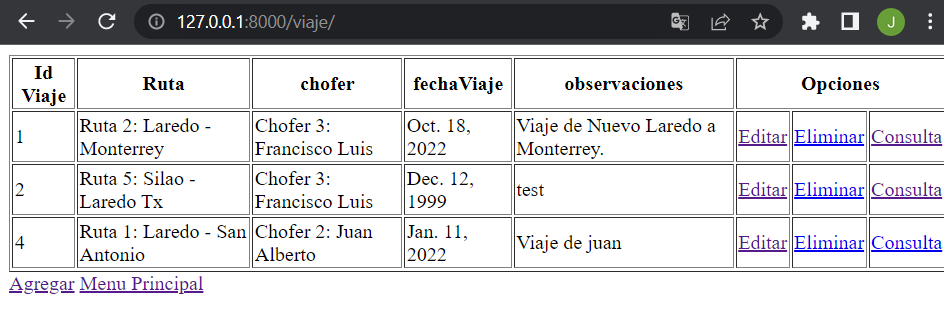
De igual manera, al momento de darle en editar, se edita al instante.



Consulta:



Y si le damos en eliminar, solo nos aparecerán los viajes actuales.



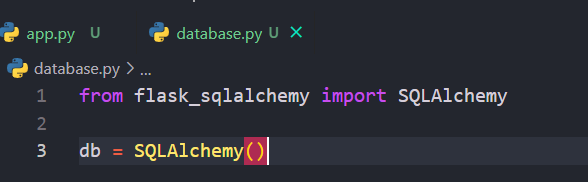
**Practica #3 Realizar una aplicación utilizando Framework Flask**

1. Conexión a base de datos postgresql con SQLAlchemy
2. Al menos un formulario con estilo css
3. Utilizar app logging
4. Utilizar al menos 3 entidades
5. CRUD con pantallas de 2 entidades
6. CRUD de al menos una entidad utilizando peticiones HTTP
7. Utilizar migraciones

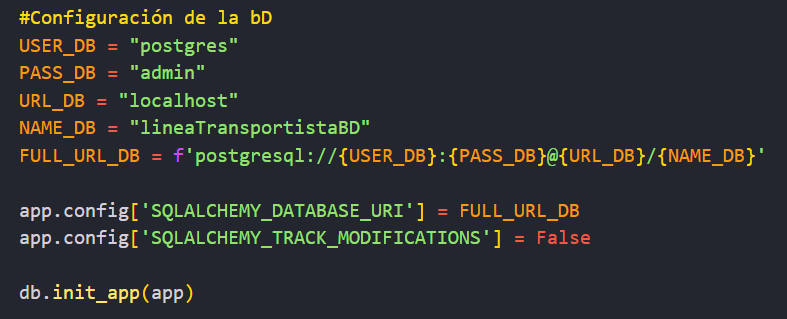
**Comprobación**

1. Conexión a BD postgresql con SQLAlchemy

Conexión a la BD

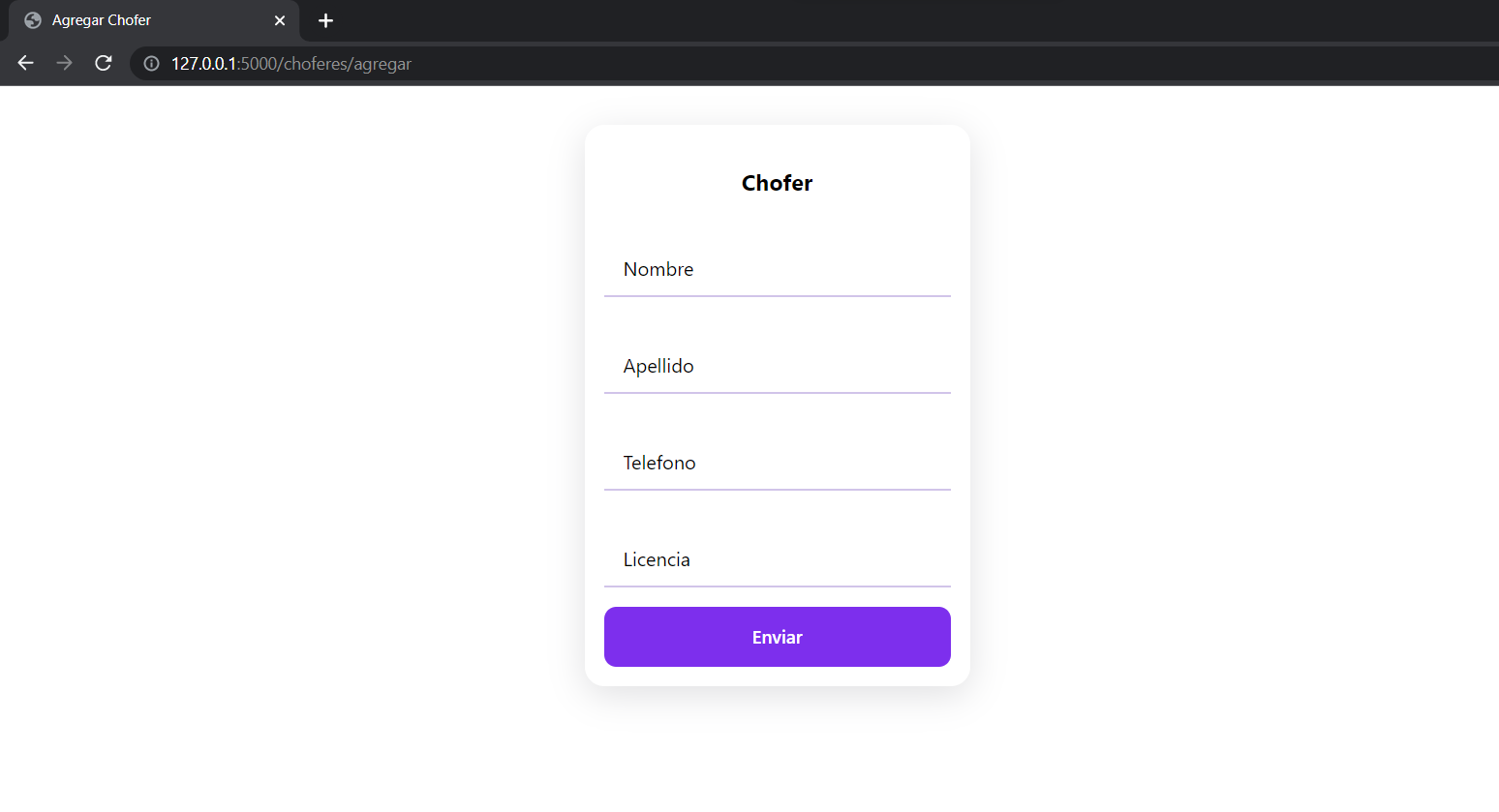


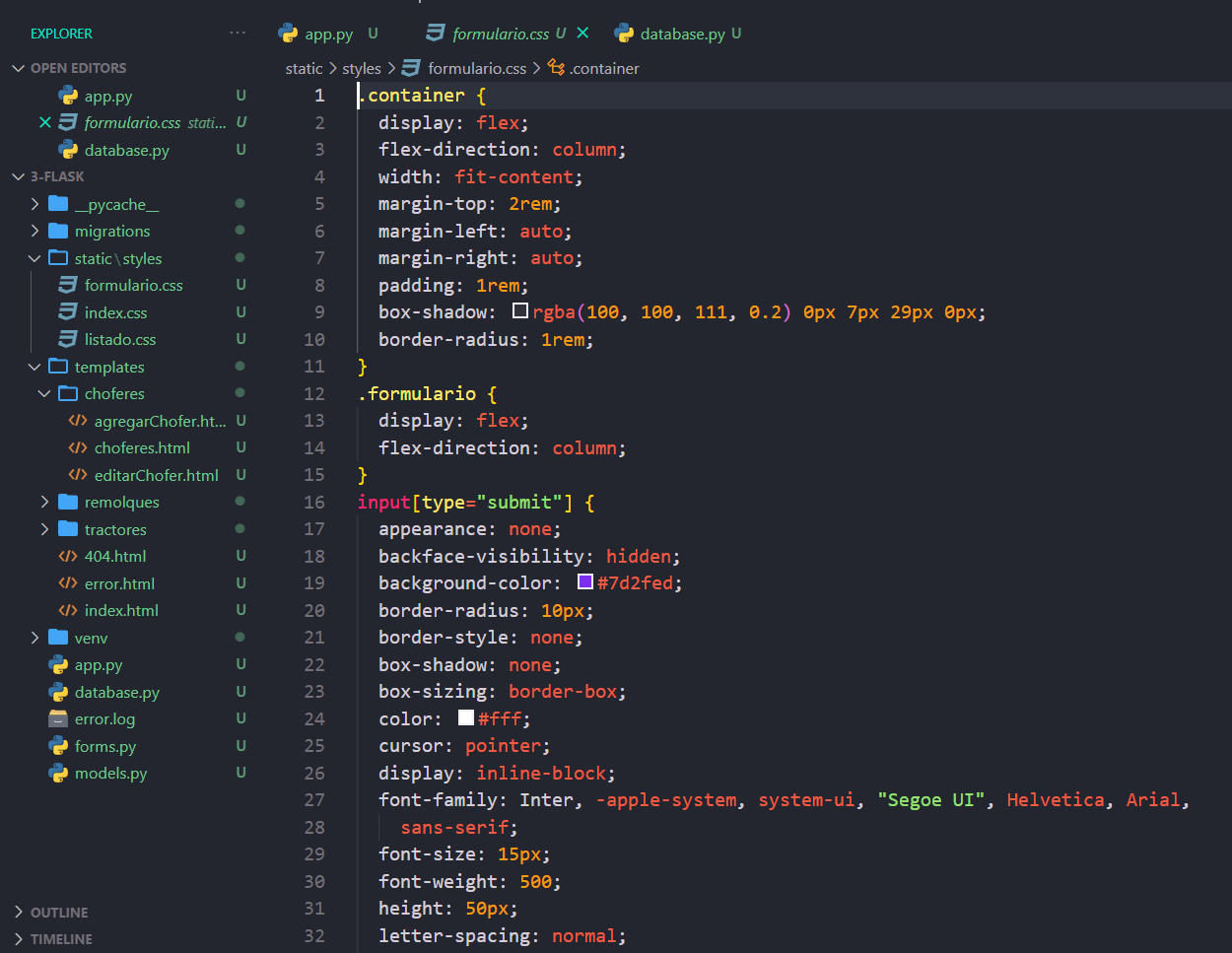
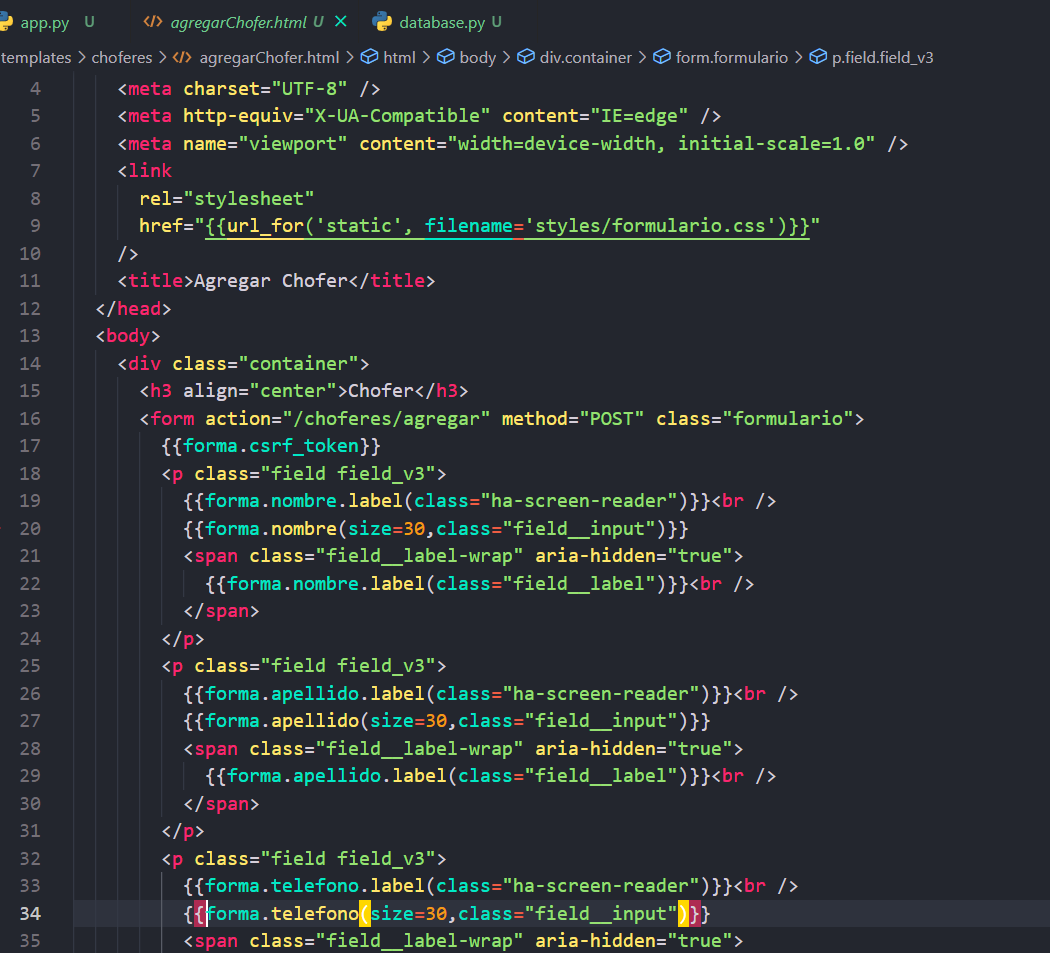
Configuración para la conexión a la BD



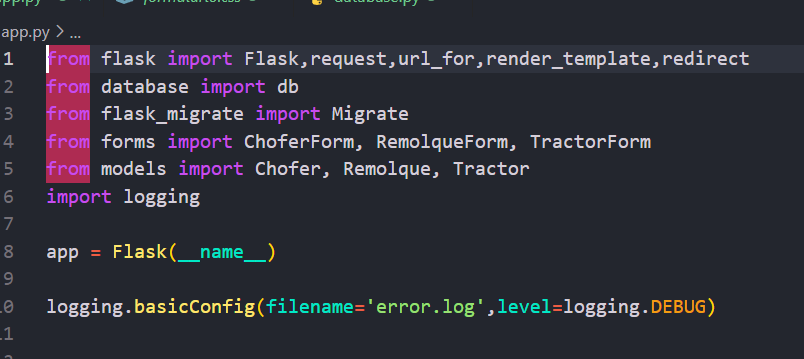
1. Formulario con CSS

Se creo un archivo css llamado “formulario” en el cual se definieron los estilos para todos los formularios que se utilizaron para la práctica.

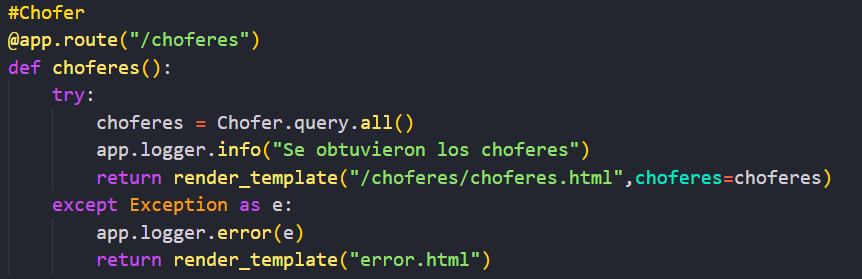




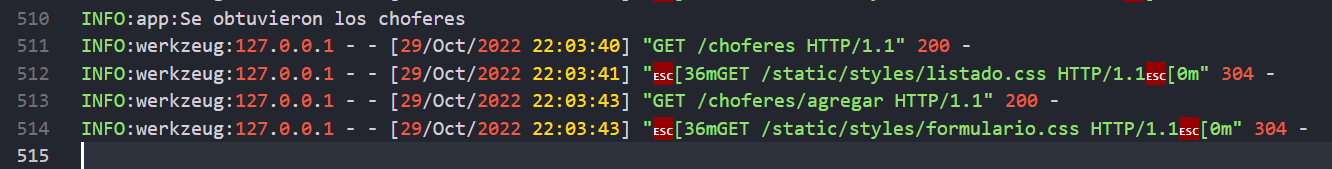
1. App Logging



Importación y configuración del logger, se le dio el nombre de “error” al archivo de log.

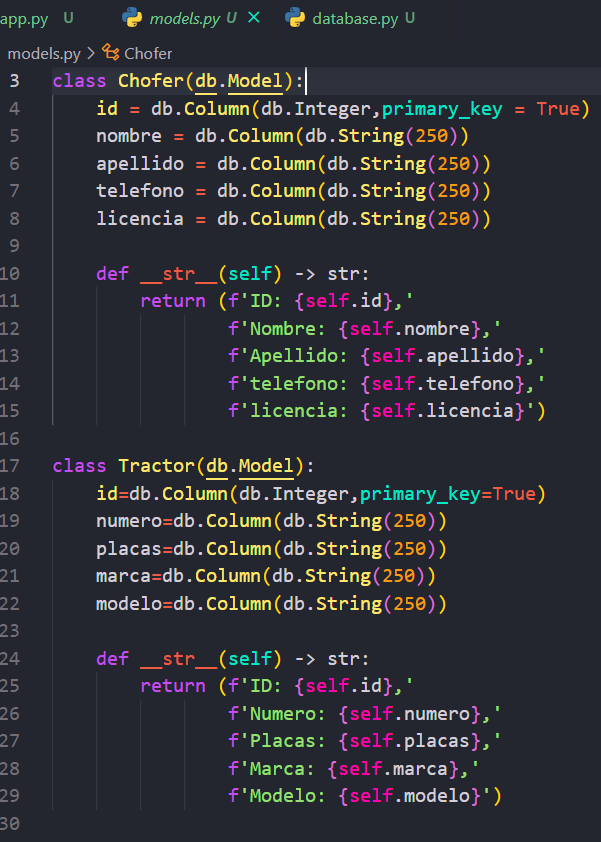
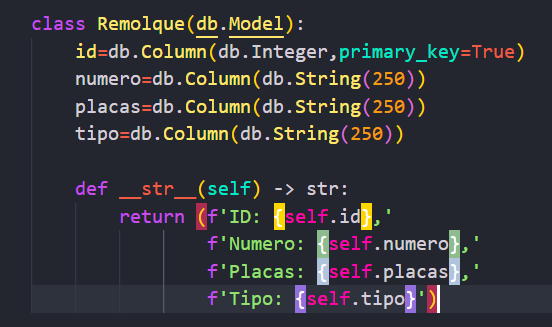


Se agregaron en cada endpoints, para registrar las acciones en la aplicación y los errores que pudiesen producirse.



1. Entidades

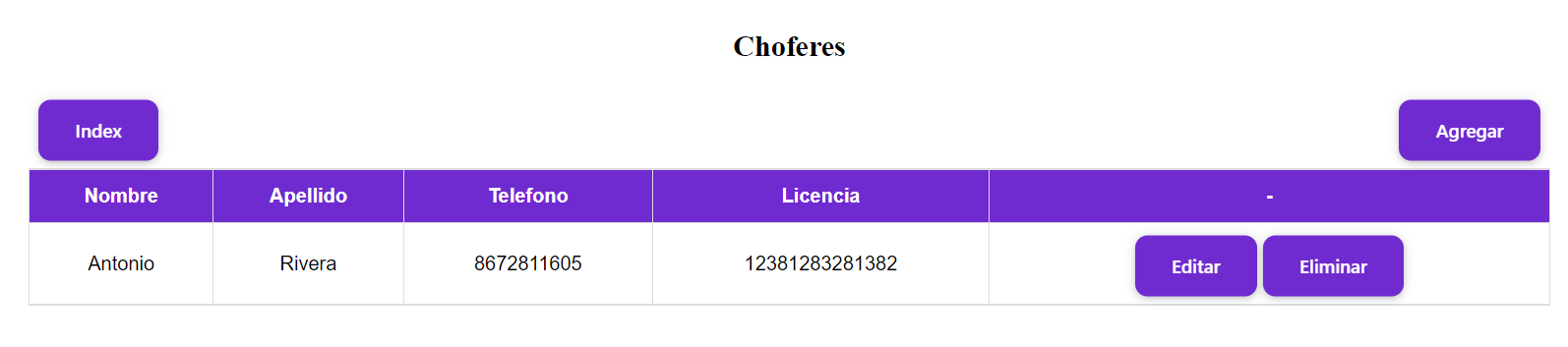
Se utilizaron 3 entidades, Chofer, Tractor y Remolques, para cada una se creó el Modelo correspondiente.

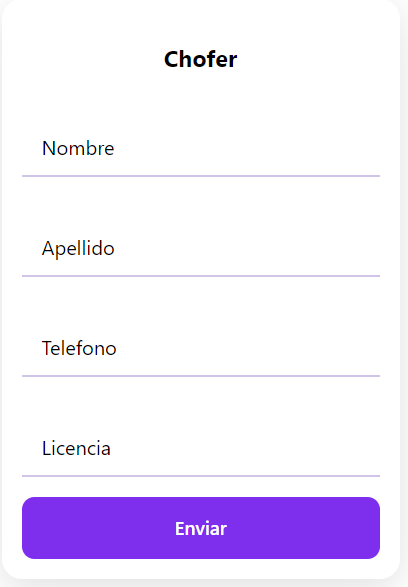
 

1. CRUD de entidades

Para las 3 entidades se creo una pantalla de listado, además de las pantallas para agregar y editar registros de cada una de las entidades.

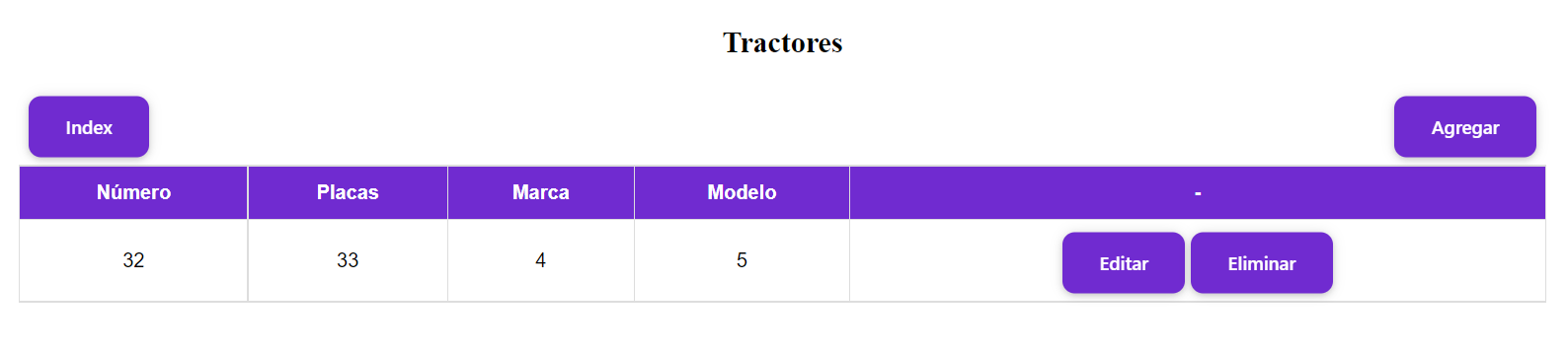
Pantallas de la entidad Chofer:

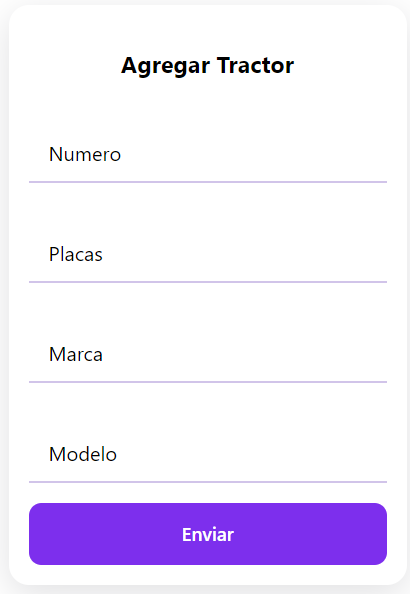




Se utilizo el mismo diseño para las pantallas para agregar y editar un chofer

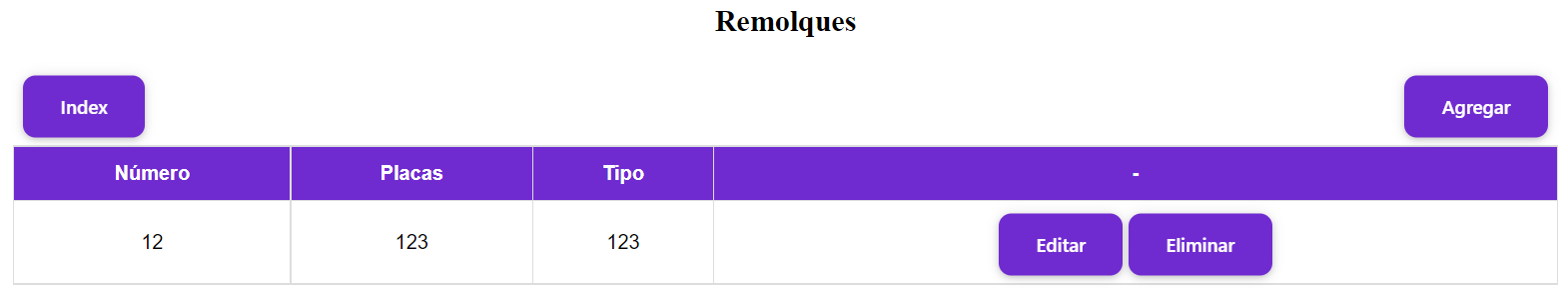
Pantallas de la entidad Tractor





Se utilizo el mismo diseño para la pantalla de agregar y editar un tractor.

Pantallas de la entidad Remolque



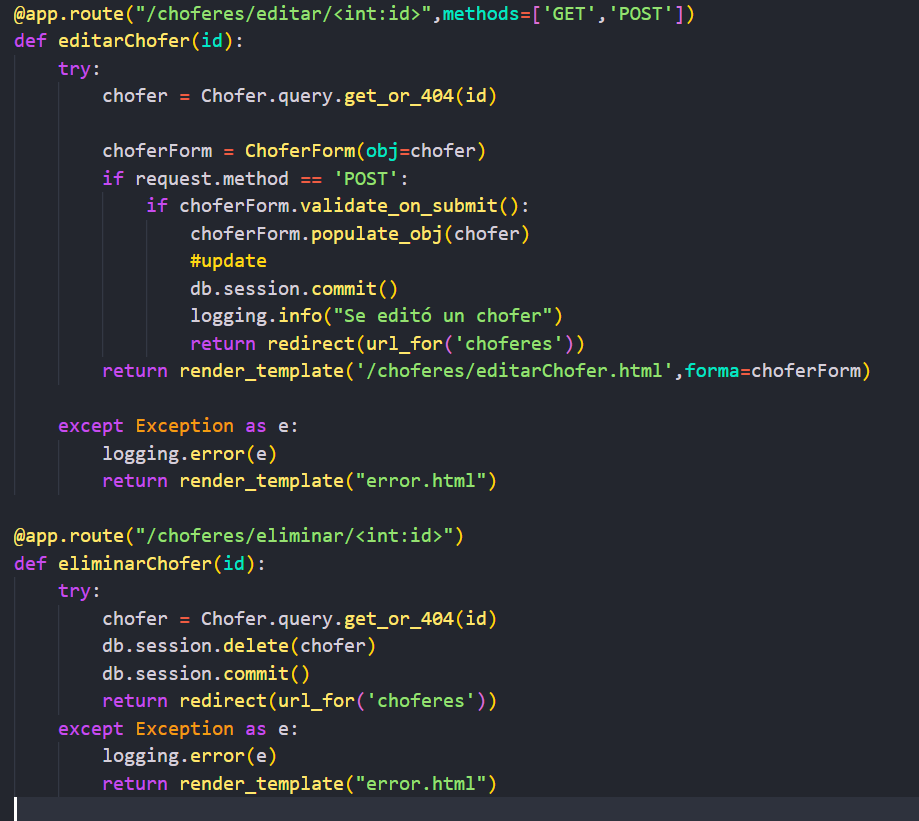


Se utilizo el mismo diseño para las pantallas para agregar y editar un remolque.

1. CRUD utilizando peticiones HTTP

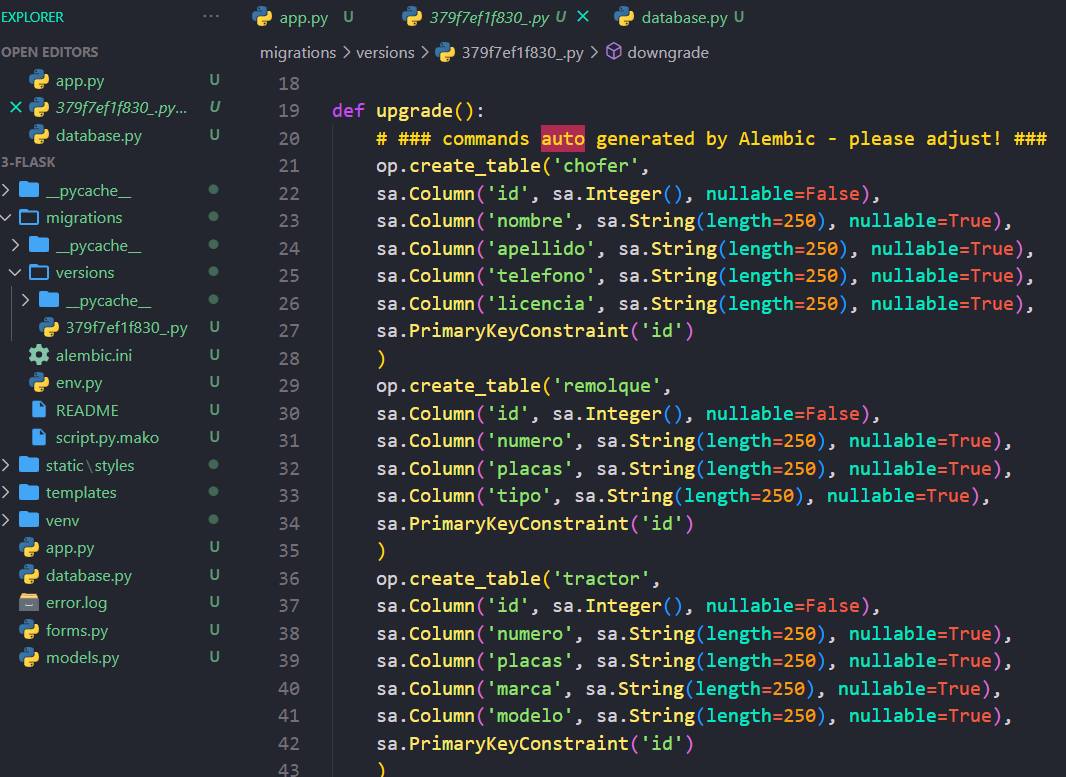
Para todas las entidades se utilizaron peticiones HTTP para obtener el listado de cada una de las entidades, además de agregar, editar y eliminar registros de cada entidad

Ejemplo de las peticiones HTTP definidas para la entidad Chofer.



1. Migraciones

Se hicieron las migraciones y esto se demuestra ya que se generó la carpeta migrations con los modelos correspondientes que antes habían sido definidos.



**Resultados**

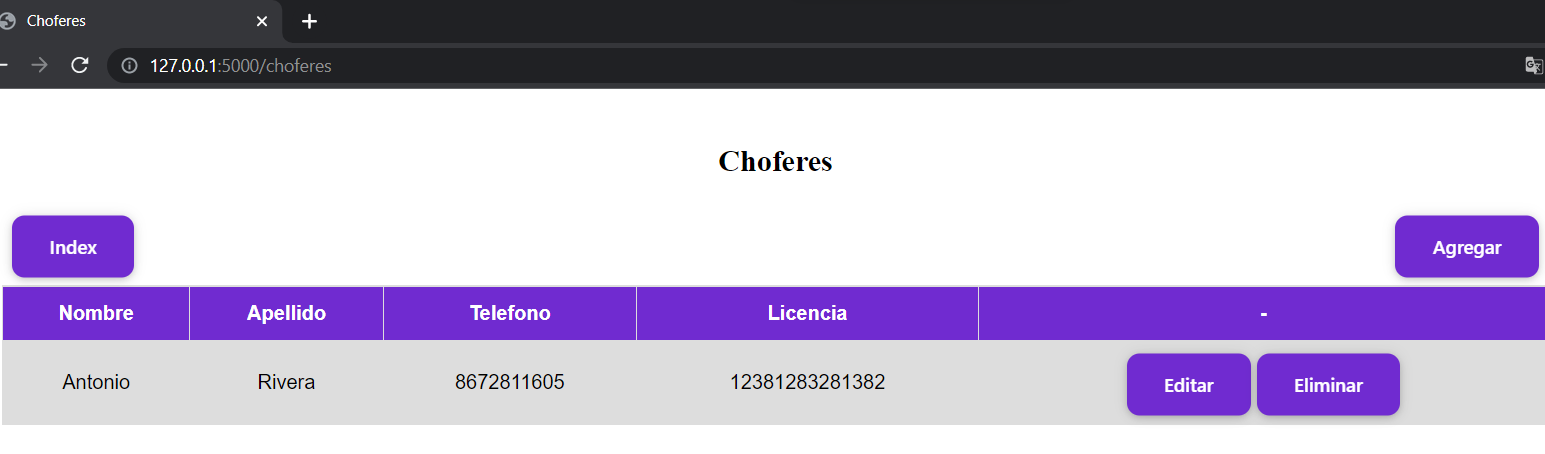
En la practica se hizo uso de tres entidades, para cada una de ellas se generó el modelo correspondiente, así como las pantallas, rutas y formulario correspondiente para así mostrar un listado de cada entidad, así como agregar, editar y eliminar registros. También se hizo la configuración del logger para posteriormente registrar las acciones que se daban en la aplicación y también para registrar los posibles errores. Ademas de esto se hizo la configuración de la conexión a la BD haciendo uso del ORM SQLAlchemy, posteriormente se hizo la migración de los modelos a la BD para así hacer las operaciones haciendo uso de la BD.

Ejemplo de funcionamiento de la aplicación con la entidad Chofer.

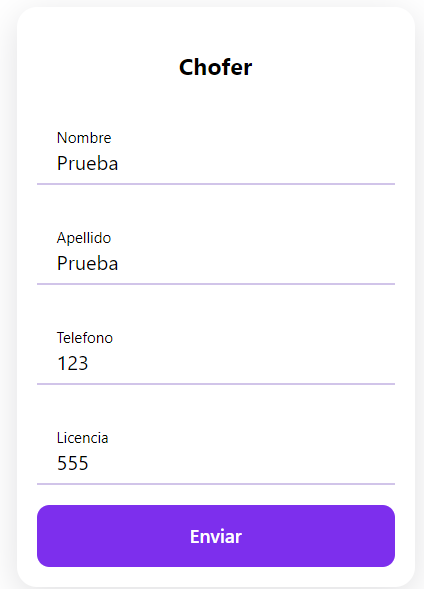
* Pantalla principal de la aplicación

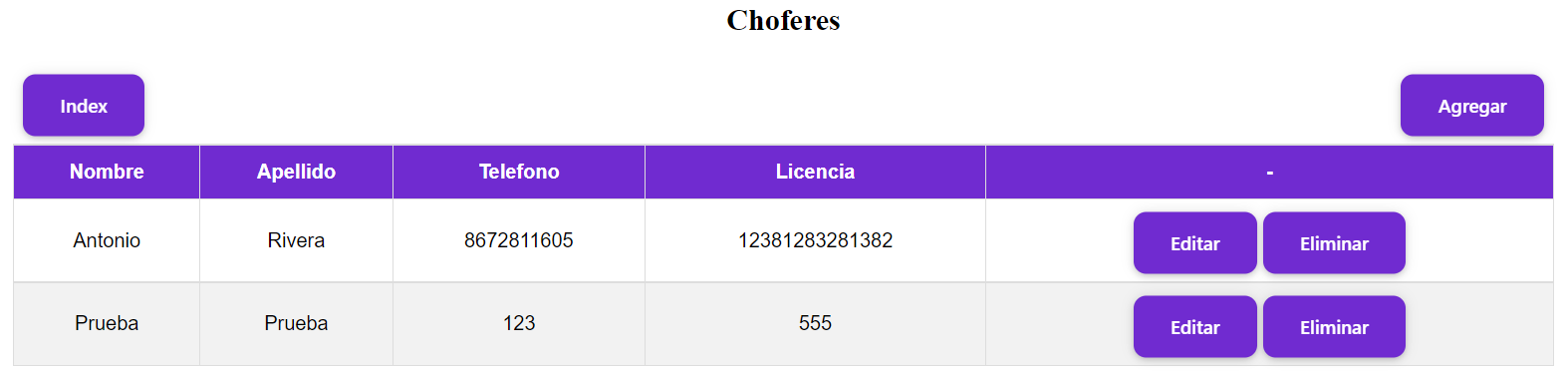


* Listado de Choferes

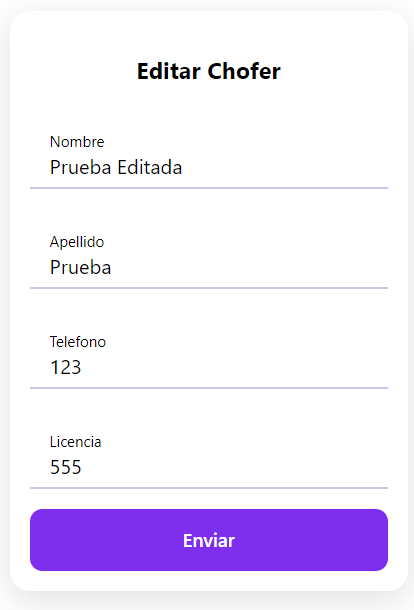


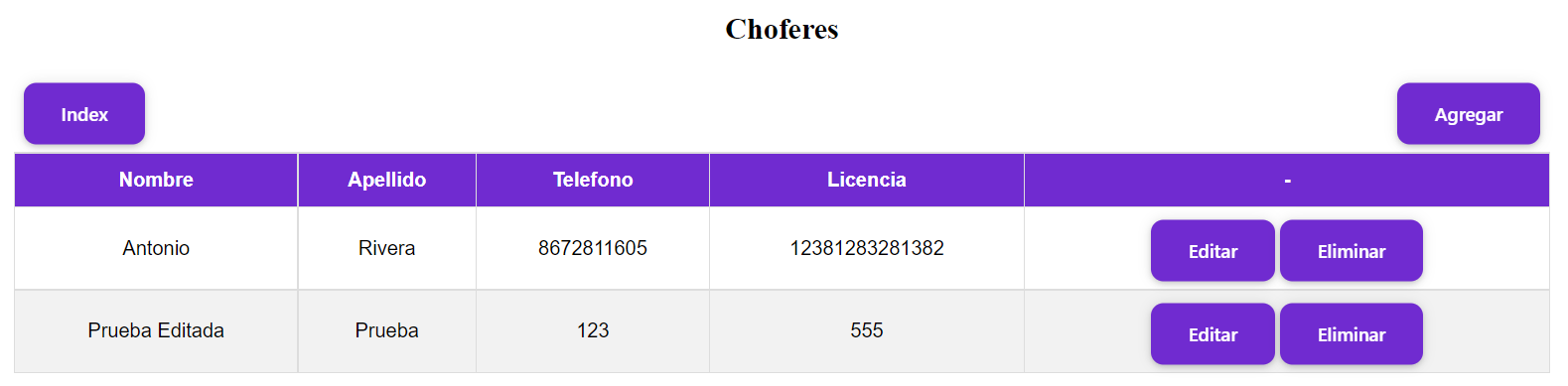
* Registro de un nuevo Chofer



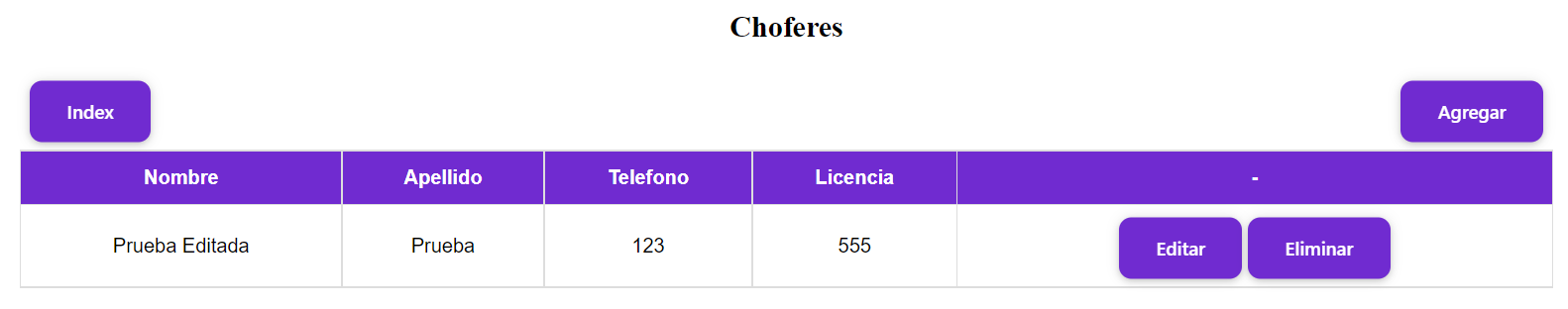


* Edición de un Chofer





* Eliminación de un Chofer



**Conclusiones y Resultados**

Durante esta unidad, se conocieron diferentes framework que para futuros desarrollos se pueden implementar, y sobre todo mejorar en un entorno web. Python a sido un lenguaje de programación que es practico y fácil de entender cada una de las cosas, y se pueden mejorar en cuestiones de lógica de programación, entendiendo los diferentes framework que se han utilizado. Algo a resaltar con las librerías que durante esta unidad han sido de gran ayuda, y que han sido fáciles de implementar.

Jose Ángel Cabrera Morales.

Esta unidad fue interesante y de mucha ayuda para conocer las diferentes formas con las que se puede trabajar con el lenguaje Python en un entorno web, ya sea sin el uso de frameworks o bien haciendo uso de alguno de ellos, los frameworks que se vieron (Django y flask) son bastante utilizados en el mundo laboral por lo que fue de gran ayuda el ver una introducción a estos frameworks para así tener una noción del funcionamiento y los posibles caso de uso que pueden tener, además de la utilización de librerías que pueden ser de mucha ayuda, especialmente la de psycopg2. Es por lo antes mencionado que esta unidad aporto mucho conocimiento nuevo que puede ser de ayuda en un futuro.

Antonio De la Cruz Rivera