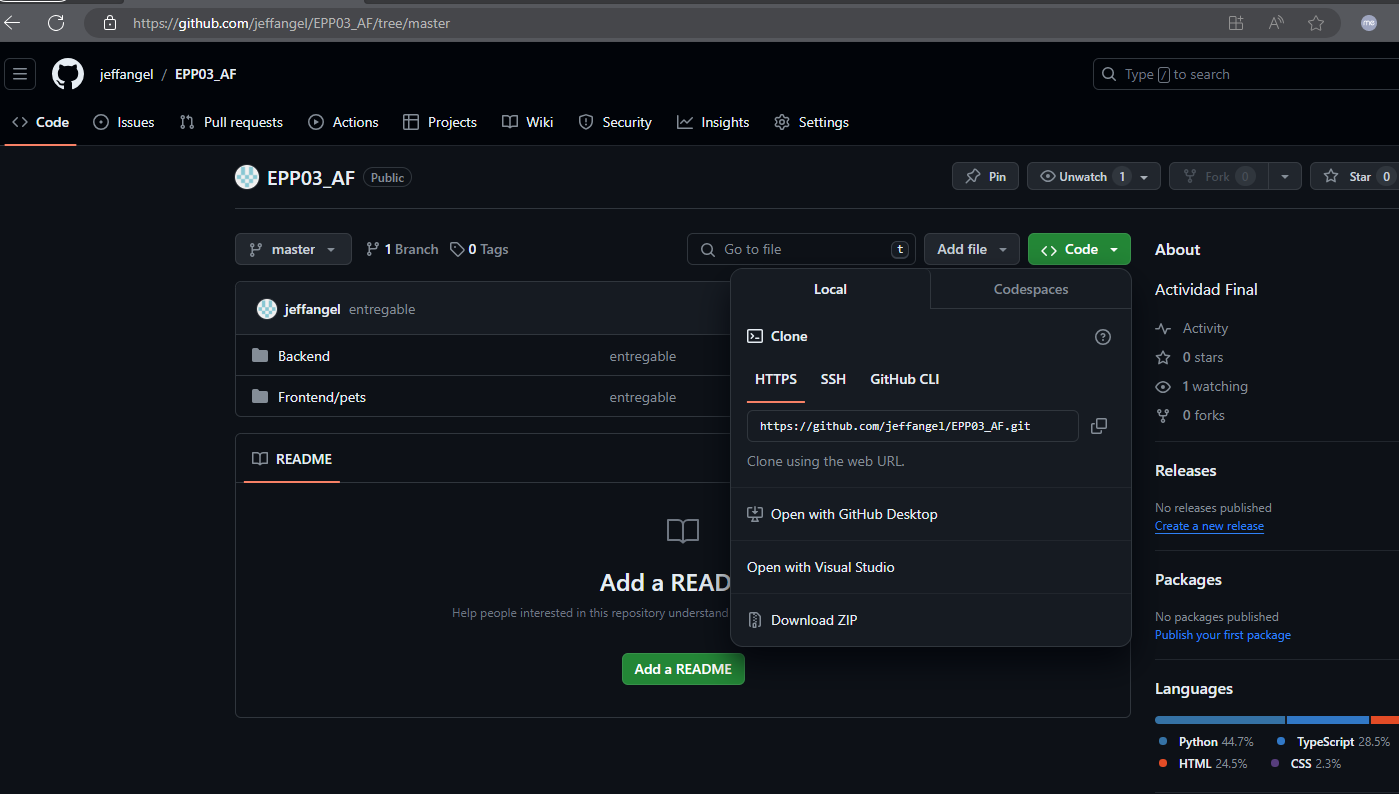
# Requisitos:

**Clonar el proyecto público:** [**https://github.com/jeffangel/EPP03\_AF.git**](https://github.com/jeffangel/EPP03_AF.git)

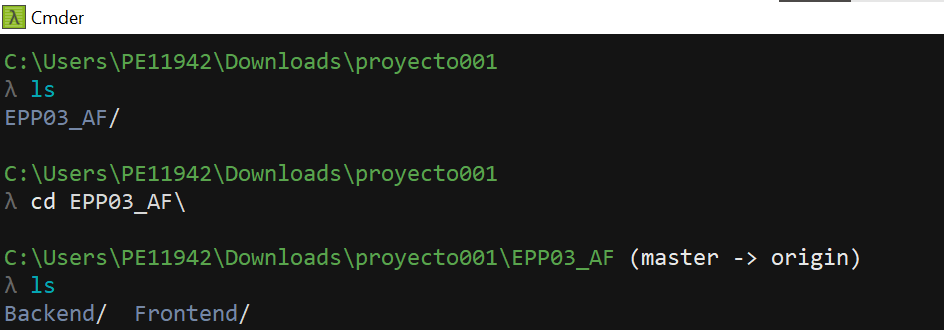
****

**Para mi caso he creado una carpeta en descargas llamada “proyecto001” sobre el que he clonado el repositorio. Dentro de esta se encontrará las 2 partes de la aplicación web que se componen de la siguiente manera:**

* **La parte de Front End que está soportada por angular para lo que se requerirá instalar Node JS en la versión 18.13.0, se puede descargar desde el siguiente enlace directo:**

[**https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj6-InGpNKEAxWYBbkGHc7IA94QFnoECA8QAQ&url=https%3A%2F%2Fnodejs.org%2Fdist%2Fv18.13.0%2Fnode-v18.13.0-x64.msi&usg=AOvVaw00HBlgoATZqmWy3Q4PsUBA&opi=89978449**](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj6-InGpNKEAxWYBbkGHc7IA94QFnoECA8QAQ&url=https%3A%2F%2Fnodejs.org%2Fdist%2Fv18.13.0%2Fnode-v18.13.0-x64.msi&usg=AOvVaw00HBlgoATZqmWy3Q4PsUBA&opi=89978449)

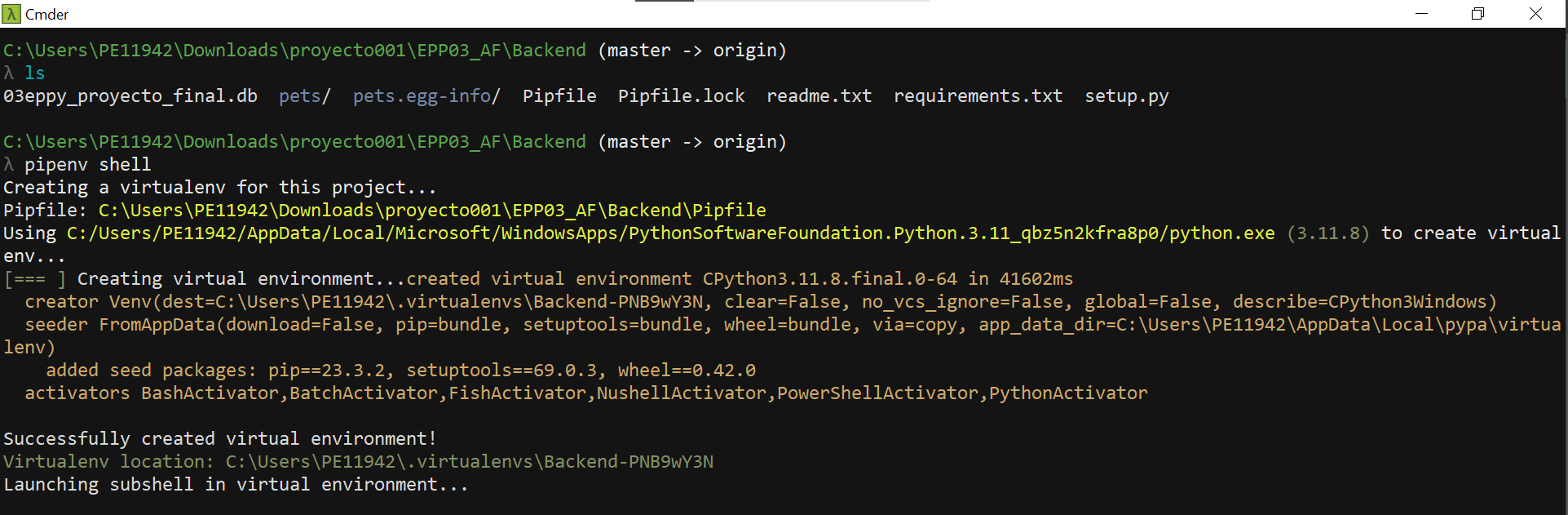
* **La parte de Back End que está soportada por Python, Flask es el servicio que se utilizará para interactuar con la base de datos que es SQLite, Preferentemente se requiere tener la versión Python 3.11, este puede ser descargado desde Microsoft Store o directamente desde el sitio web de Python.**

****

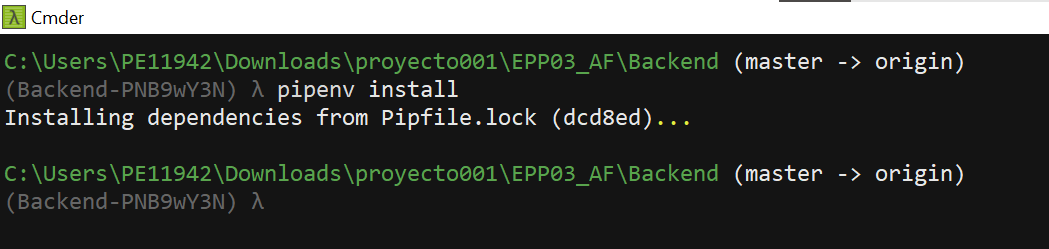
# Python – Backend:

**Se recomienda crear un entorno virtual para instalar los paquetes requeridos, en mi caso estoy usando pipenv. Para instalar pipenv se puede ejecutar “pip install pipenv” desde cualquier parte del sistema:**

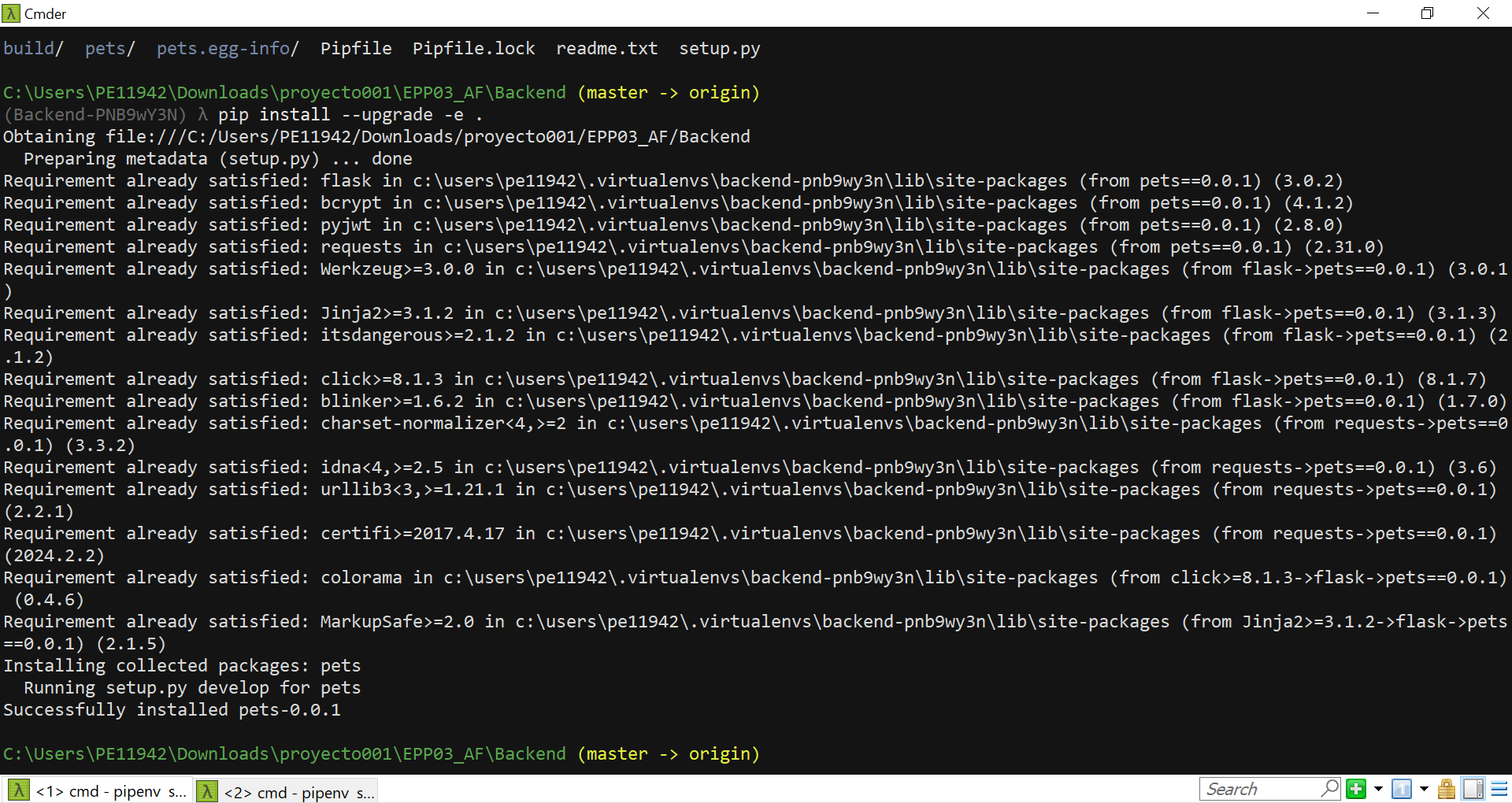
**Luego, nos debemos posicionar al nivel de la carpeta Backend para crear el entorno virtual mediante “pipenv shell”**

****

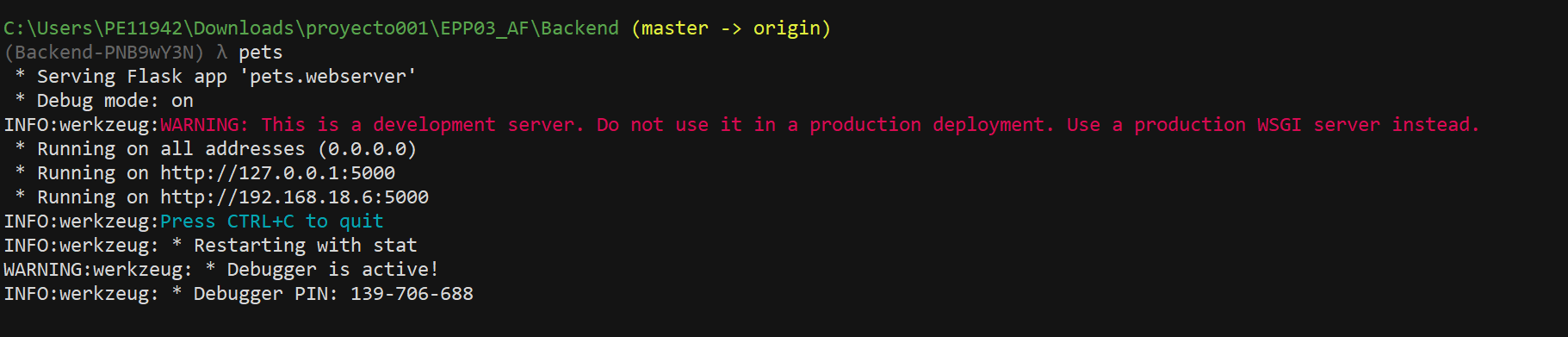
**Posteriormente, se realizará la instalación de los paquetes que forman parte de las dependencias mediante “pipenv install”**

****

**Luego instalaremos la aplicación como paquete de Python, para ello ejecutamos “pip install –upgrade -e .” posicionados en la carpeta Backend.**



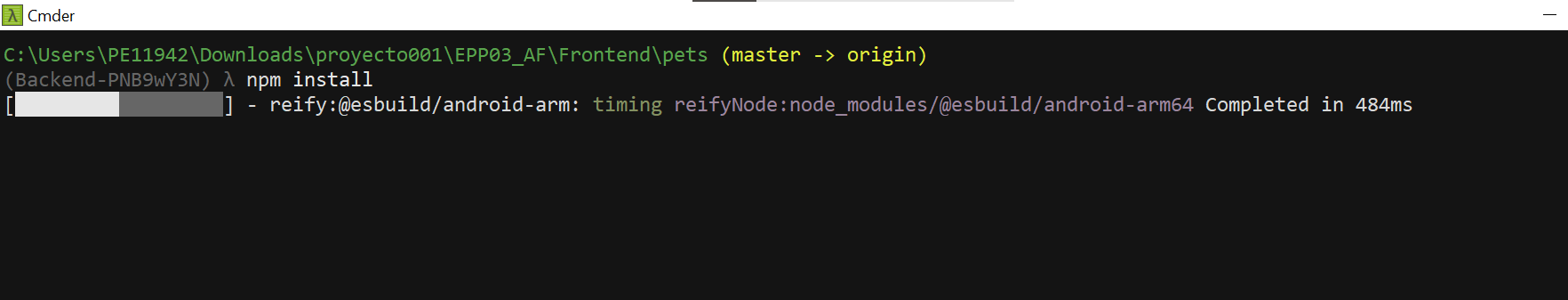
**Se procederá a iniciar la aplicación mediante la llamada al paquete “pets”**

****

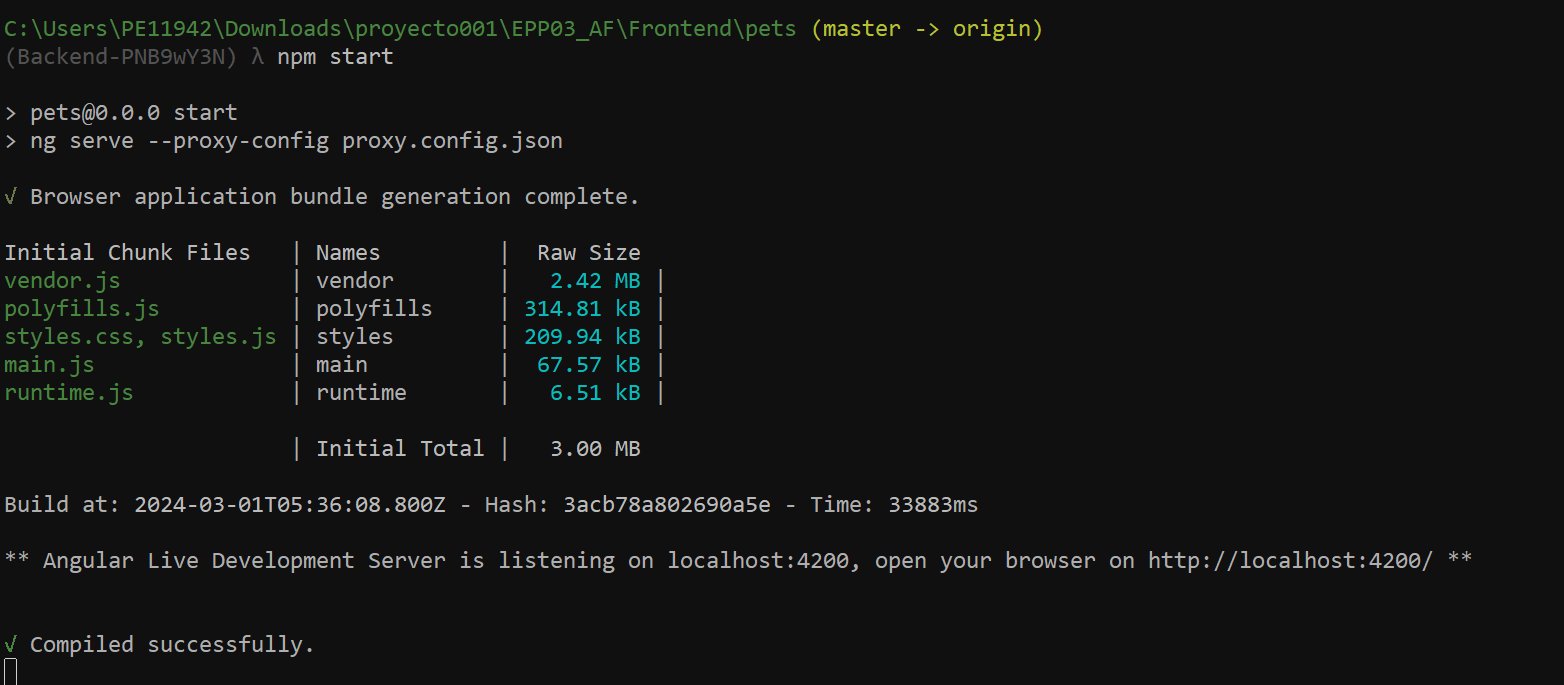
# Angular – Front End:

Para la parte de Front End, las acciones se realizarán desde la carpeta ‘..\Frontend\pets\’ a menos que se indique lo contrario en algún paso especifico.

Primero, se realizará la instalación de las dependencias. Para ello se ejecutará “npm install”



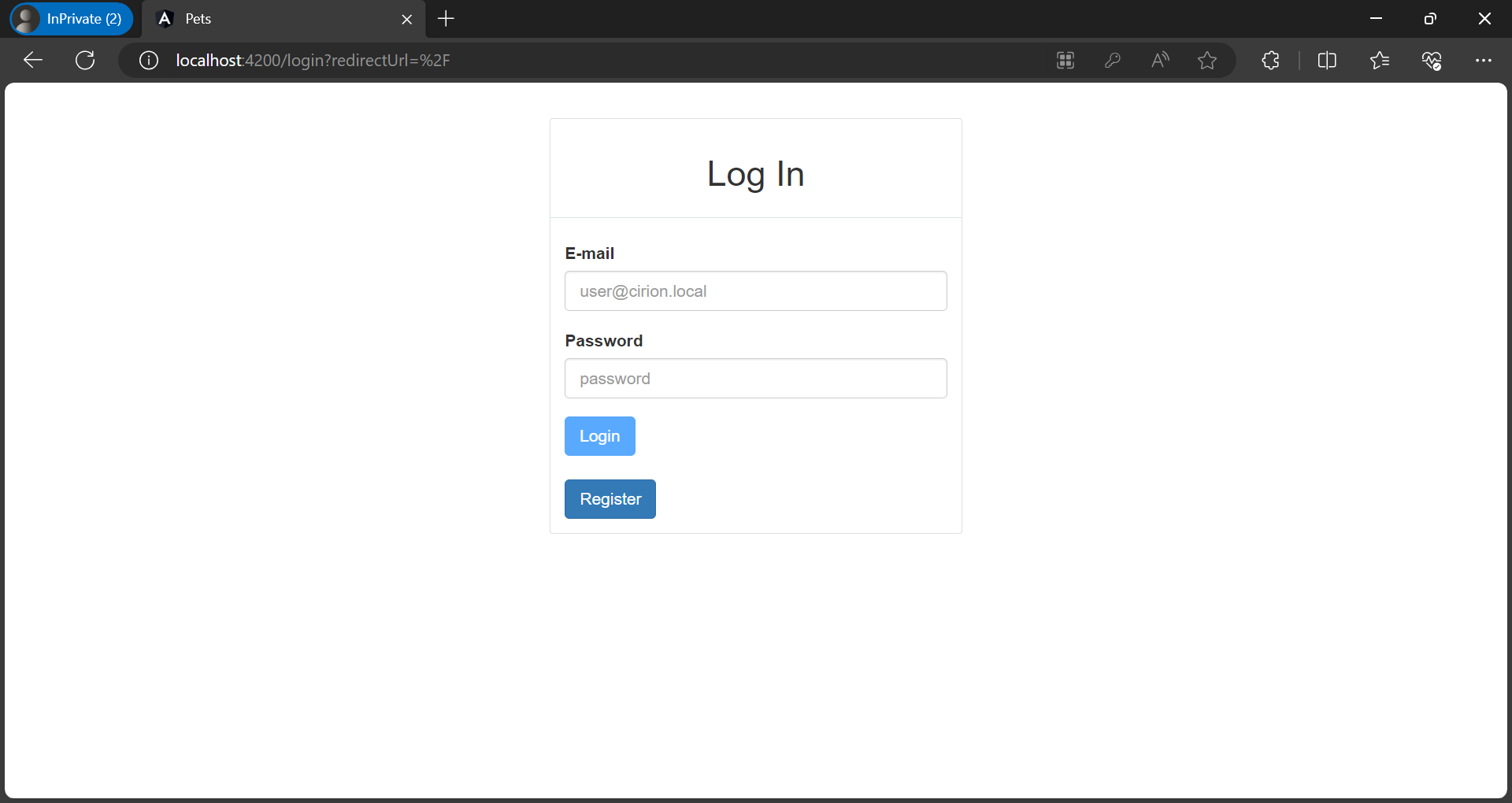
Luego de que la instalación finalice, se deberá ejecutar “npm start” para iniciar node y poder consultarlos mediante el navegador web:



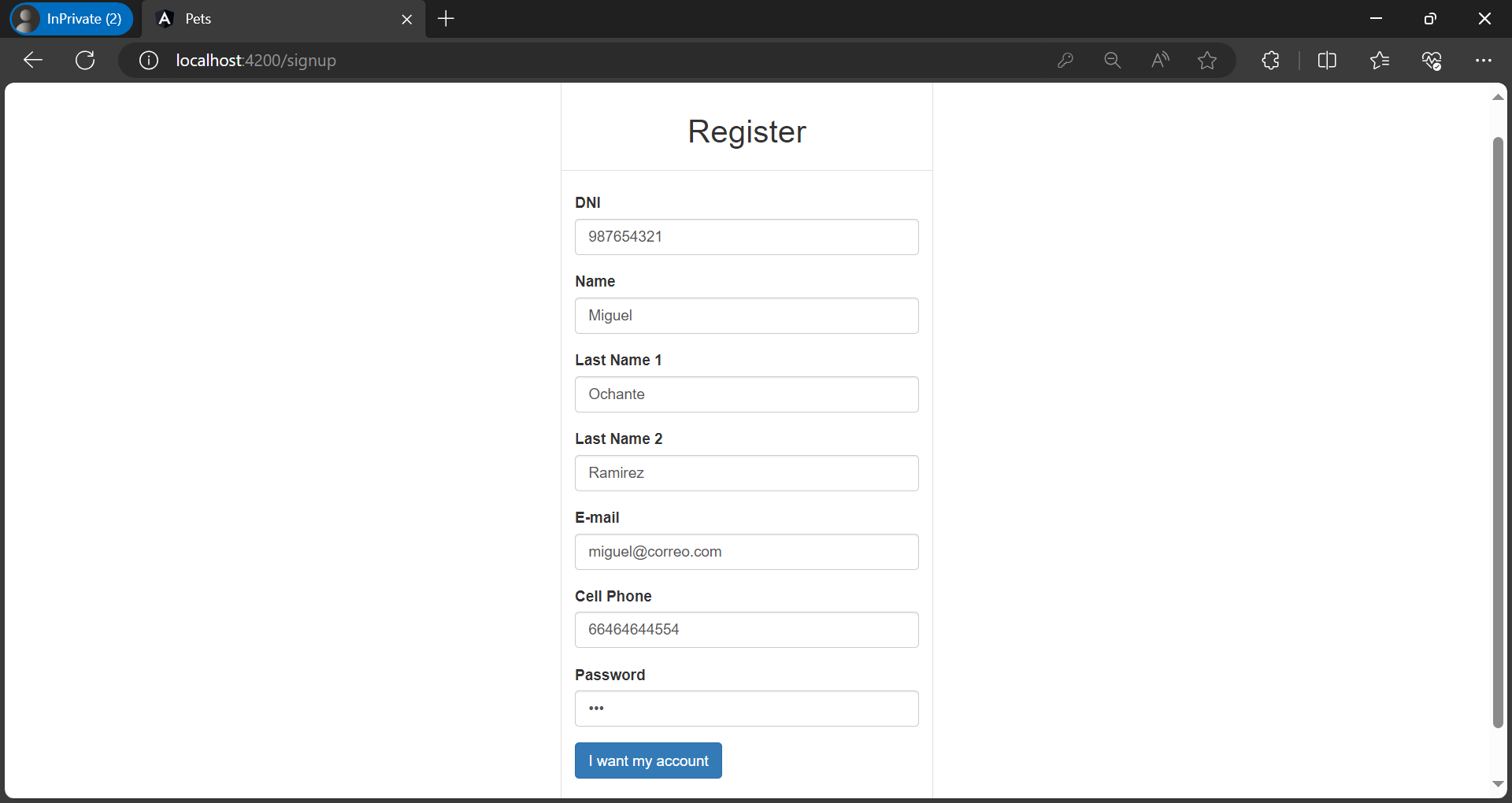
# Validación:

Luego de haber iniciado tanto la parte Front End como la Back End, se puede consultar el sitio web mediante la url brindada en el último paso: <http://localhost:4200/>

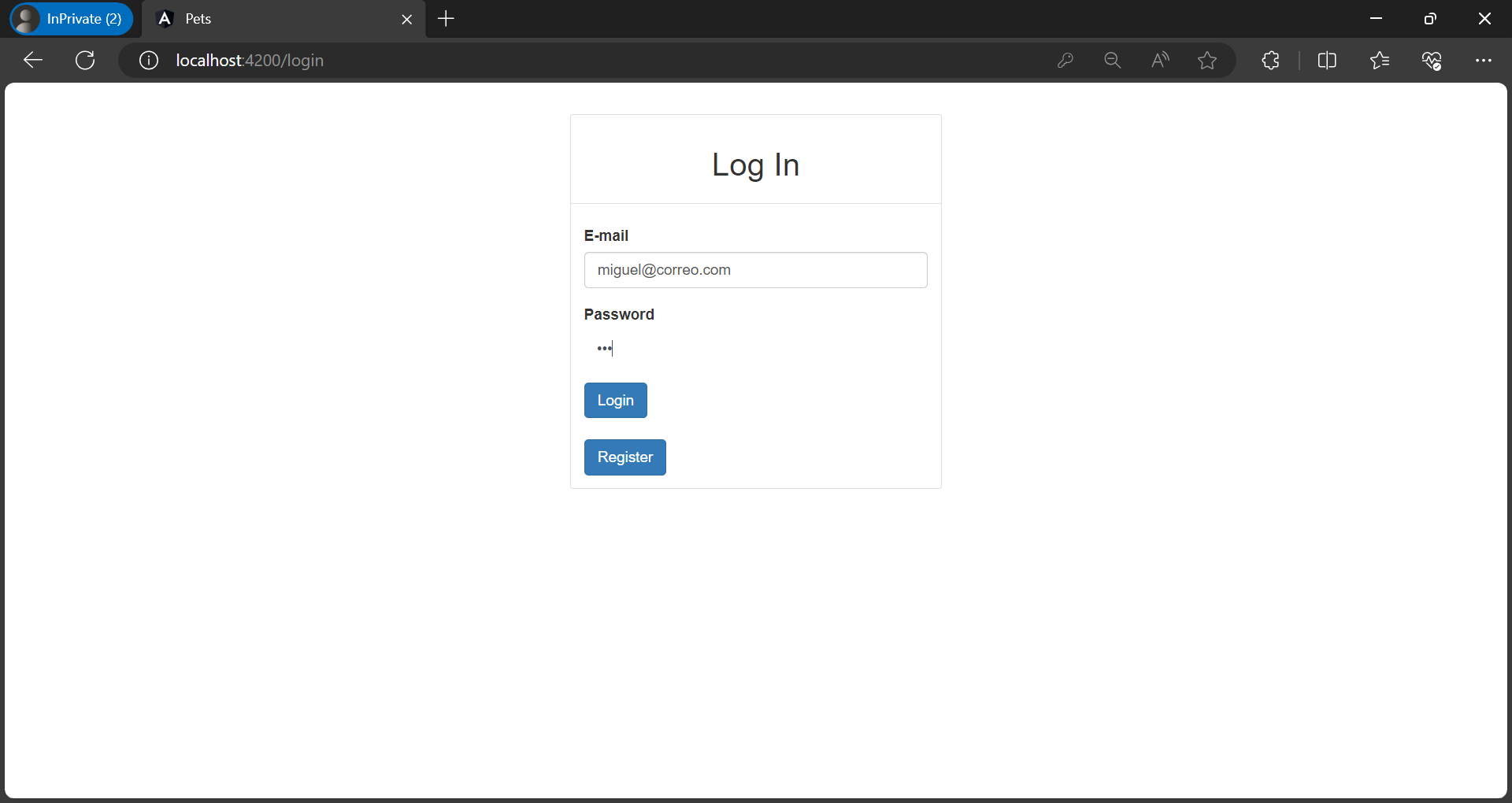
Esta URL nos redirija al login



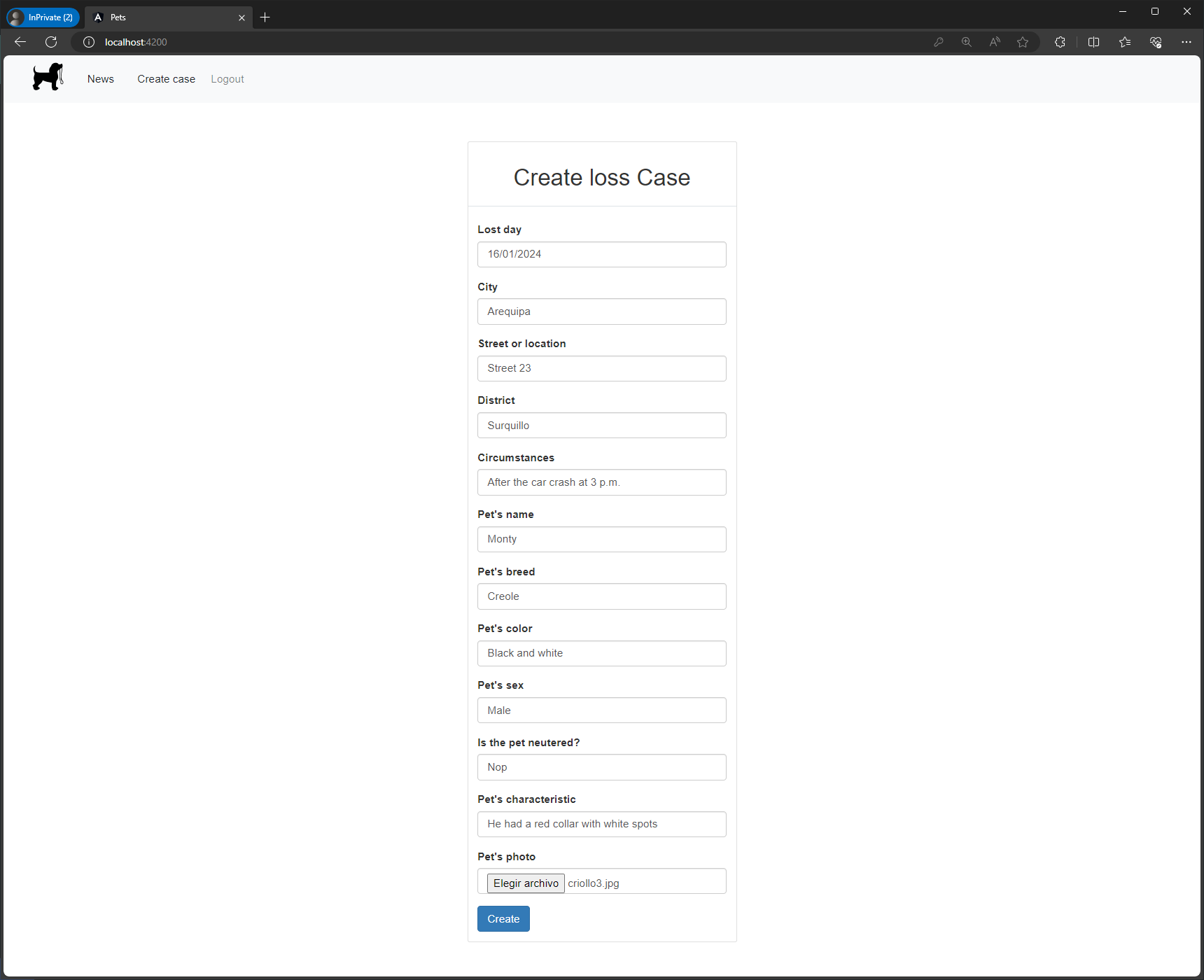
Para poder acceder debemos de crearnos una cuenta, para ello seleccionamos el botón “Register”, en la pagina de Sign Up, llenamos los datos requeridos para crear una cuenta:



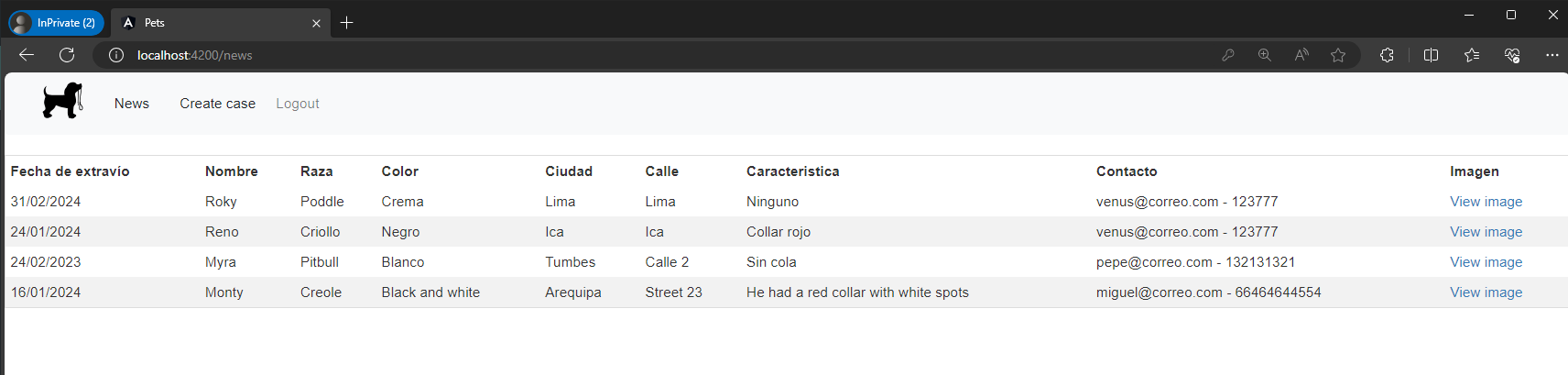
Luego de seleccionar “I want my account” nos redirija al login, donde ingresaremos con los datos antes ingresados:



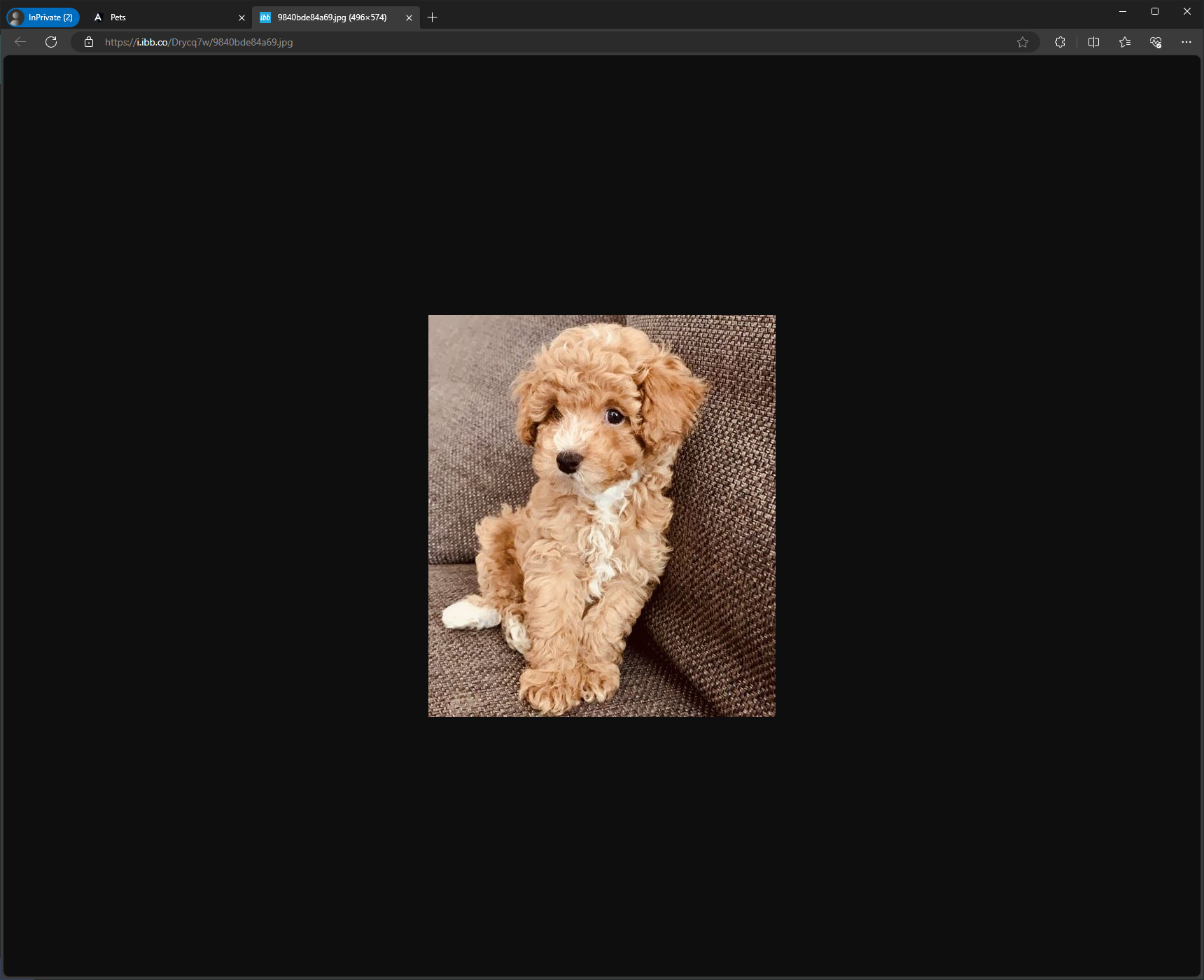
Como sección principal se muestra la posibilidad de crear un caso directamente, para ellos se requiere llenar algunos campos sobre los detalles de la mascota y datos generales:



Luego de llenar los datos, seleccionamos “create”, esto nos redirigirá a la sección “News” donde podremos ver los distintos reportes de los usuarios (*el tiempo es un recurso escaso cuando de personalizaciones se trata 😐*).



Aquí podremos visualizar algunos datos de los reportes, así como la foto de la mascota, para ello podemos pulsar en el hipervínculo de la columna “Imagen”



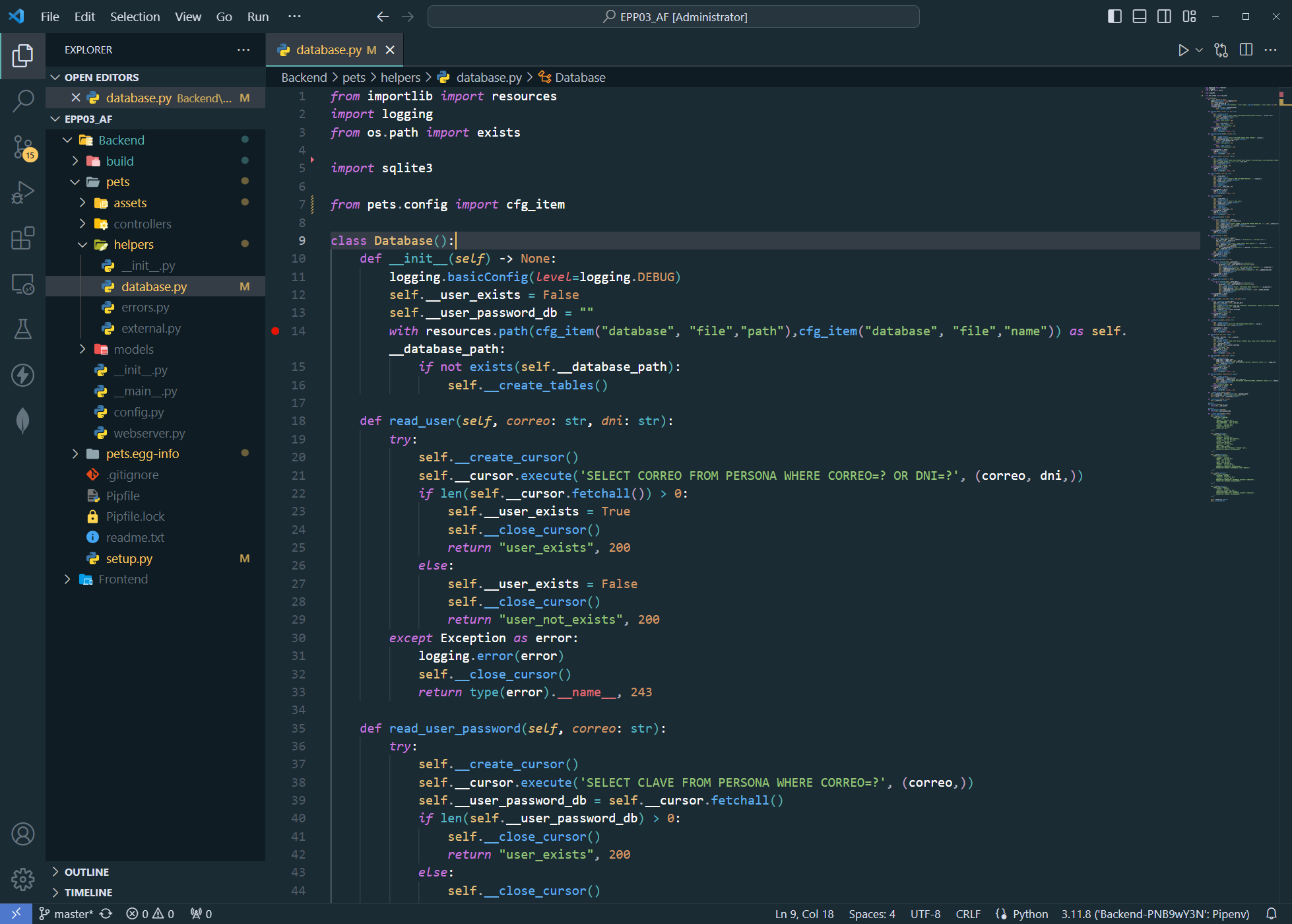
Se abrirá una nueva ventana del navegador con la imagen de la mascota.

La parte de base de datos y aplicación (Flask) soporta múltiples fotos/rasgos por mascota. Sin embargo, no llegue a agregar esta funcionalidad a la parte de FrontEnd. Esto se podrá ver en el diagrama ER y MER que está disponible en la última sección de este documento.

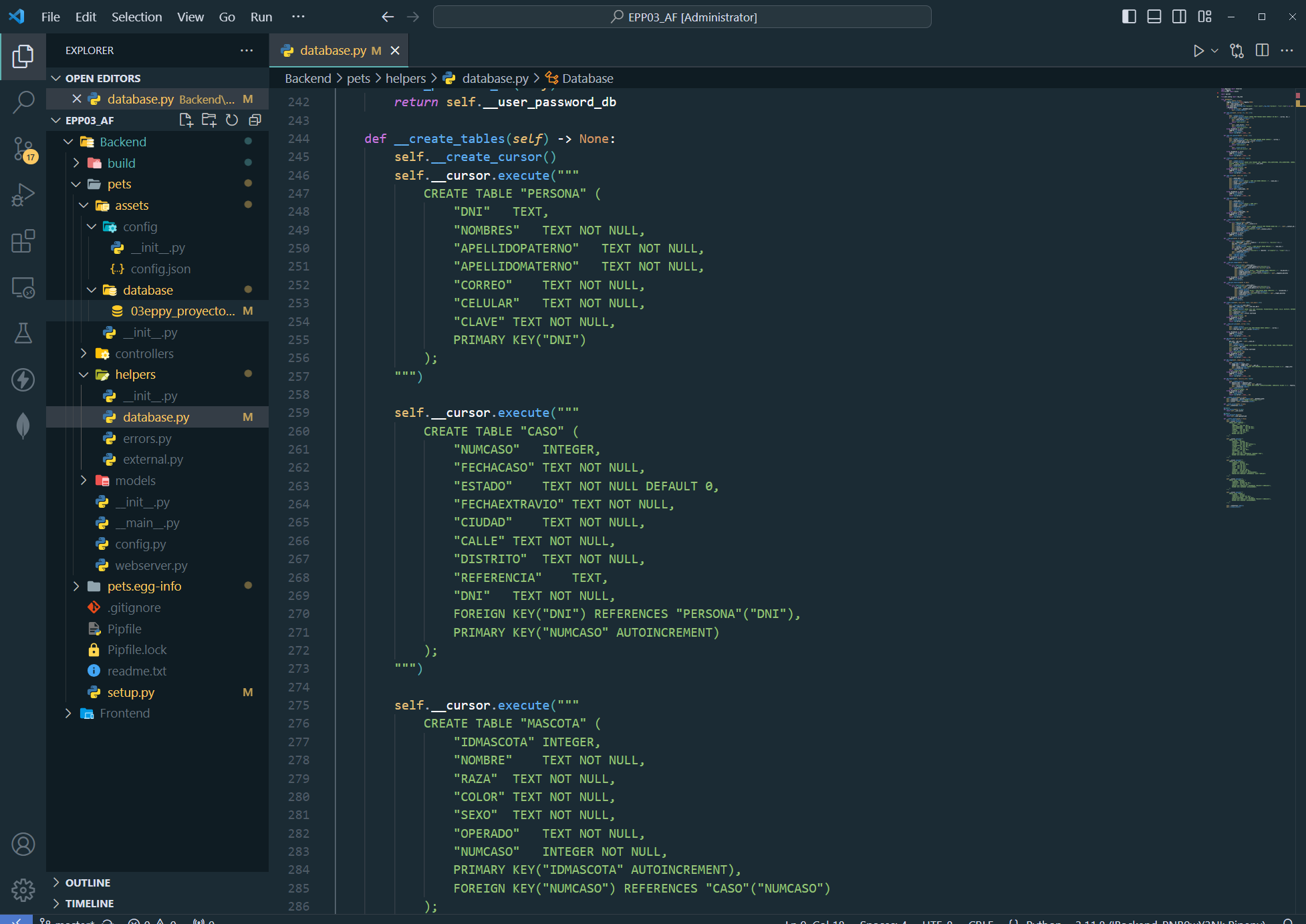
Las imágenes al ser cargadas por Angular, se codifican en Base64 y se envían a Flask para que pueda llamar a la API del servicio ImgBB (<https://es.imgbb.com/>), el cual retorna una URL que se guarda en la tabla “Imágenes” de SQLite.

# Código relevante para el curso:

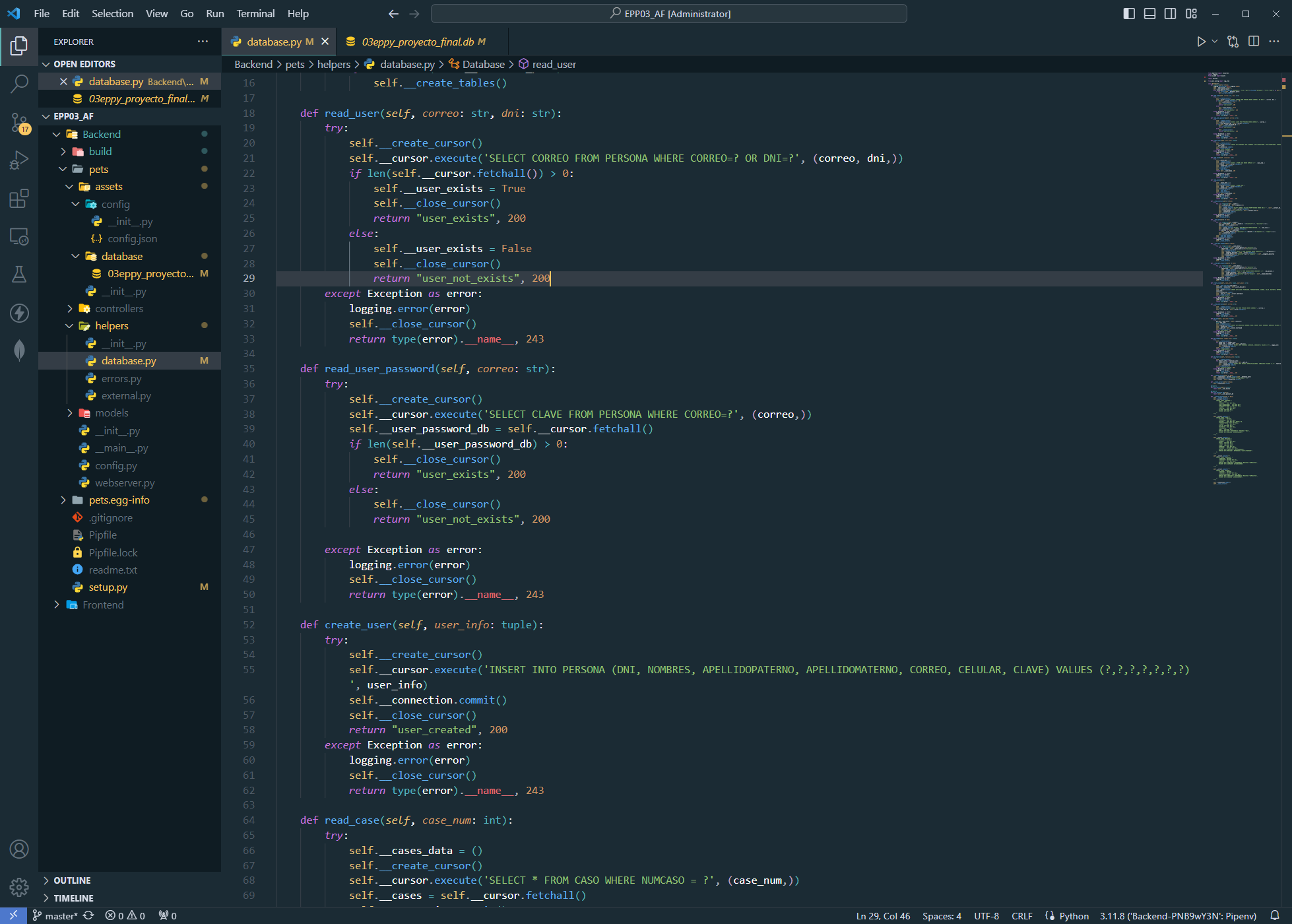
El archivo database.py que se encuentra en la ruta “./Backend/pets/helpers” contiene una clase que se encarga de manejar todas las interacciones con la base datos de SQLite.



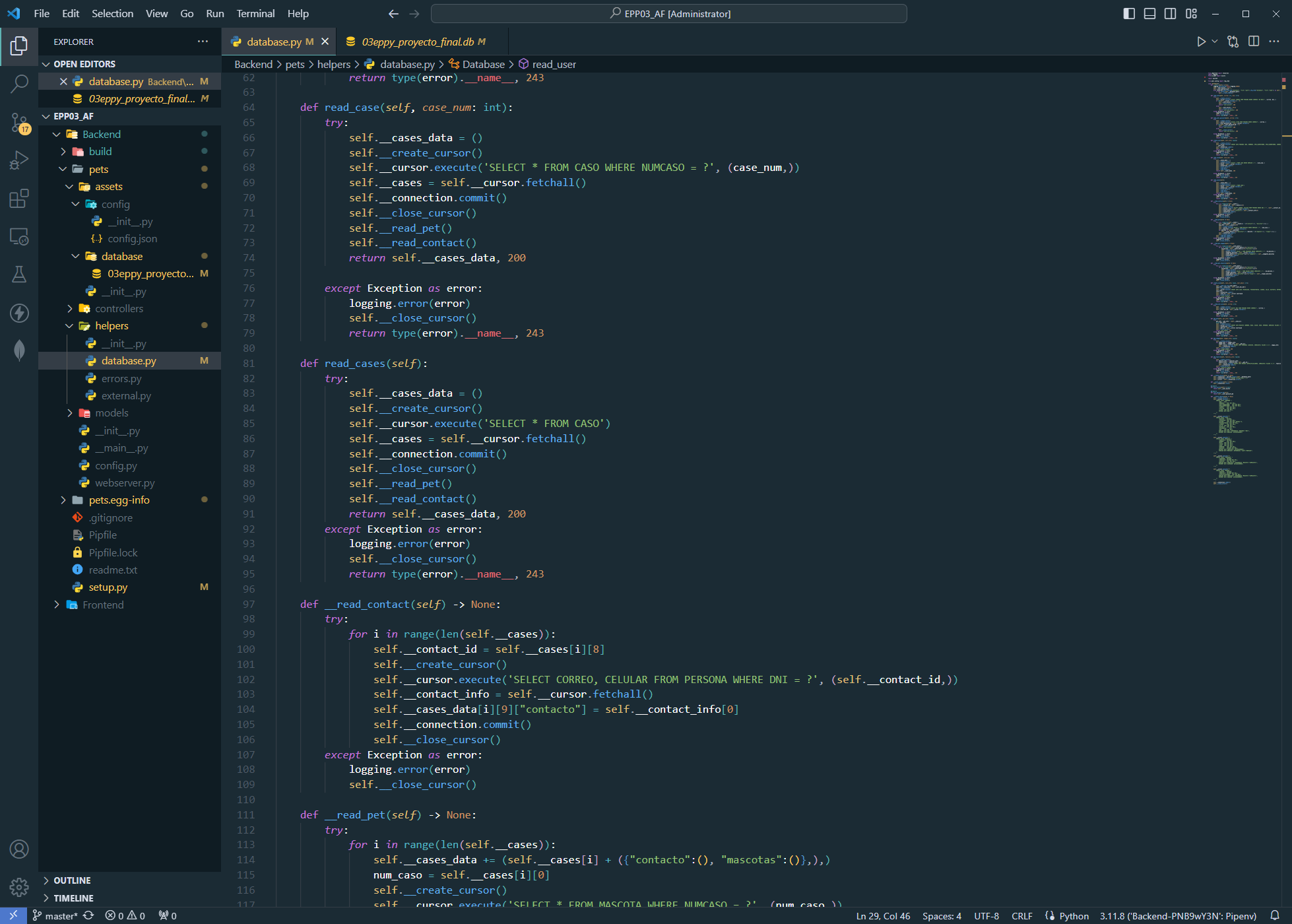
Al instanciar la clase, el método constructor verificará la ruta donde se encuentra el archivo de base datos llamado “03eppy\_proyecto\_final.db”, si este archivo no está disponible en la ruta “./Backend/pets/assets/database”, entonces creará una nueva base de datos con el esquema definido, el detalle del esquema se encuentra en la última sección del archivo database.py.

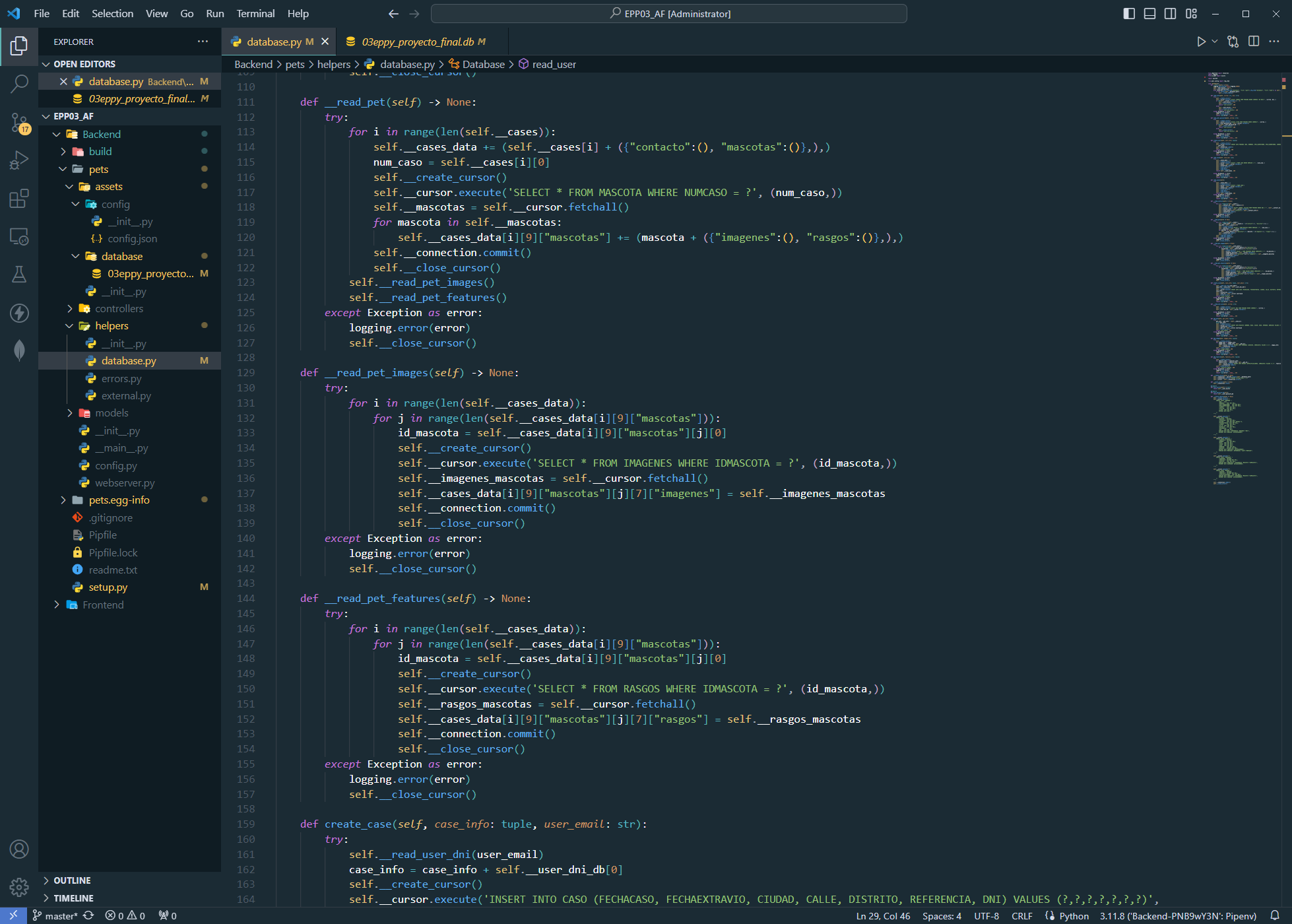


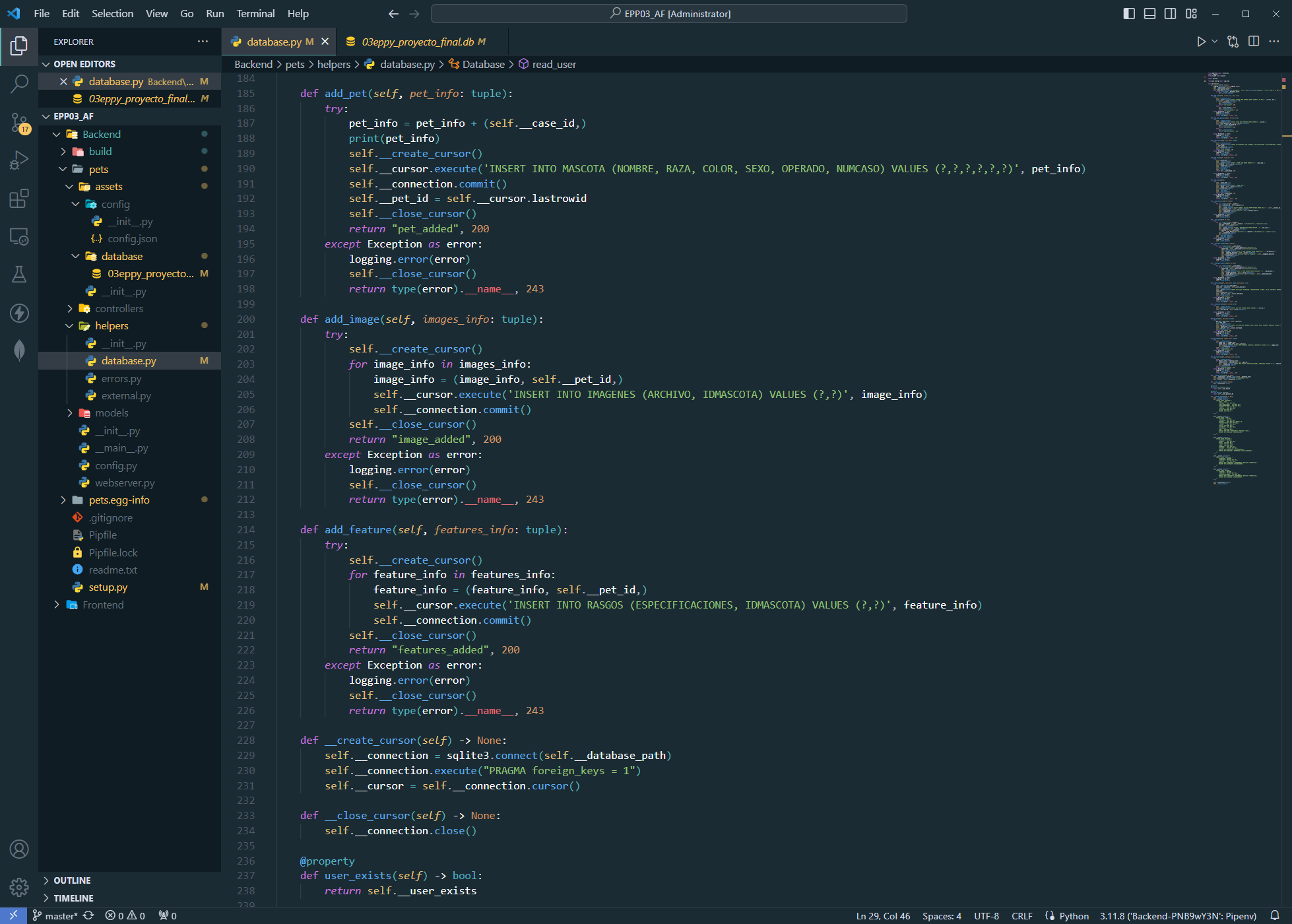
Este archivo, database.py, muestra como interactúa la aplicación con la BD, a continuación, podemos observar lo relacionado a la validación y registro de usuarios:



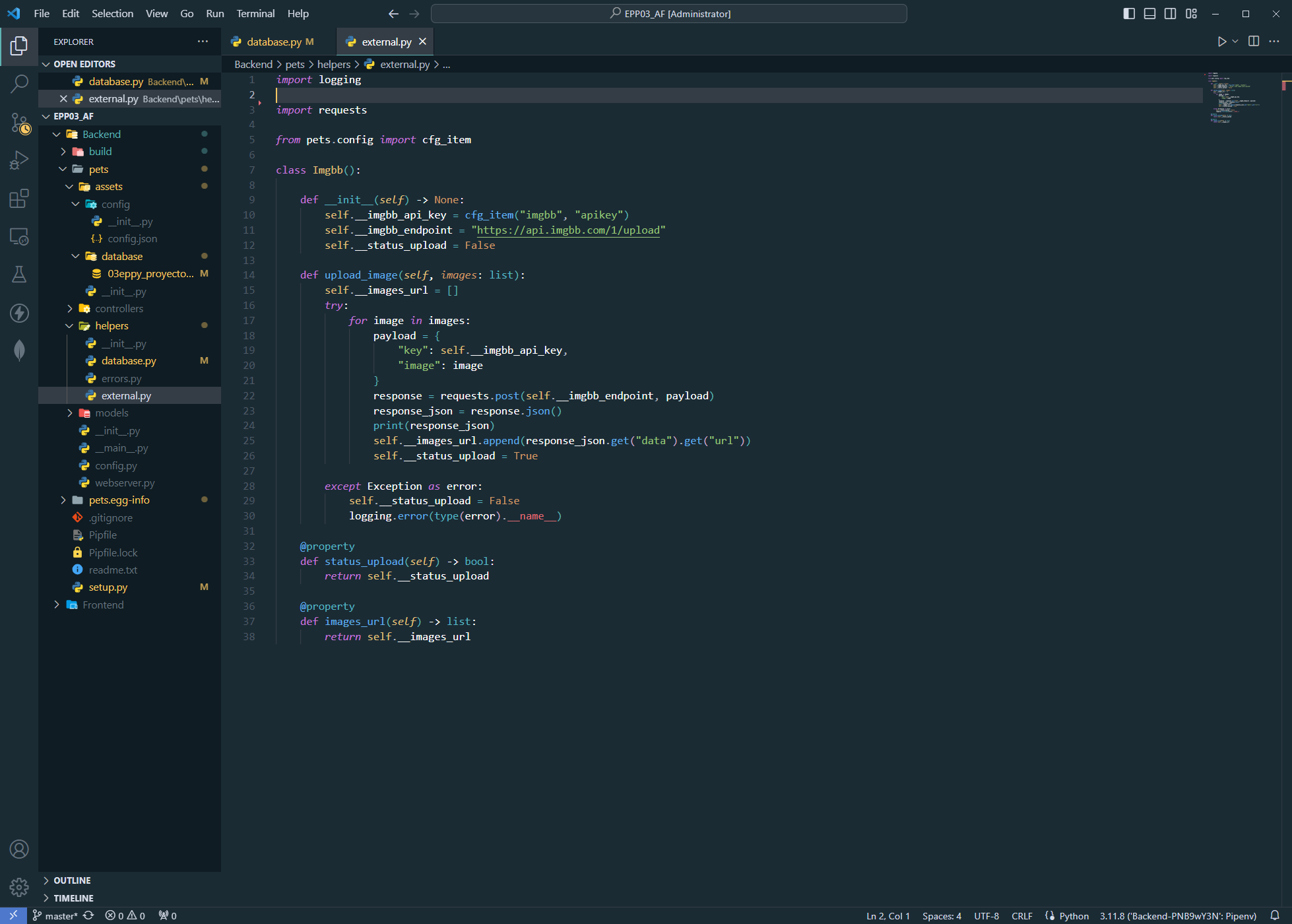
De igual forma, la interacción para los datos de los casos de extravío de las mascotas:





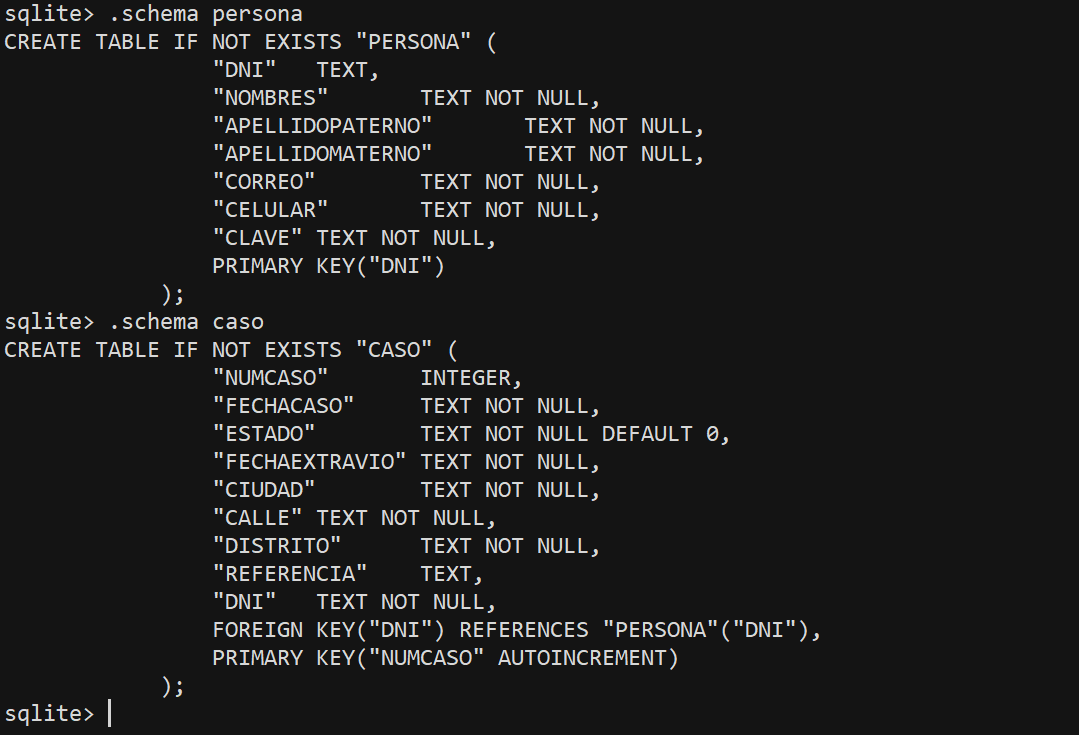


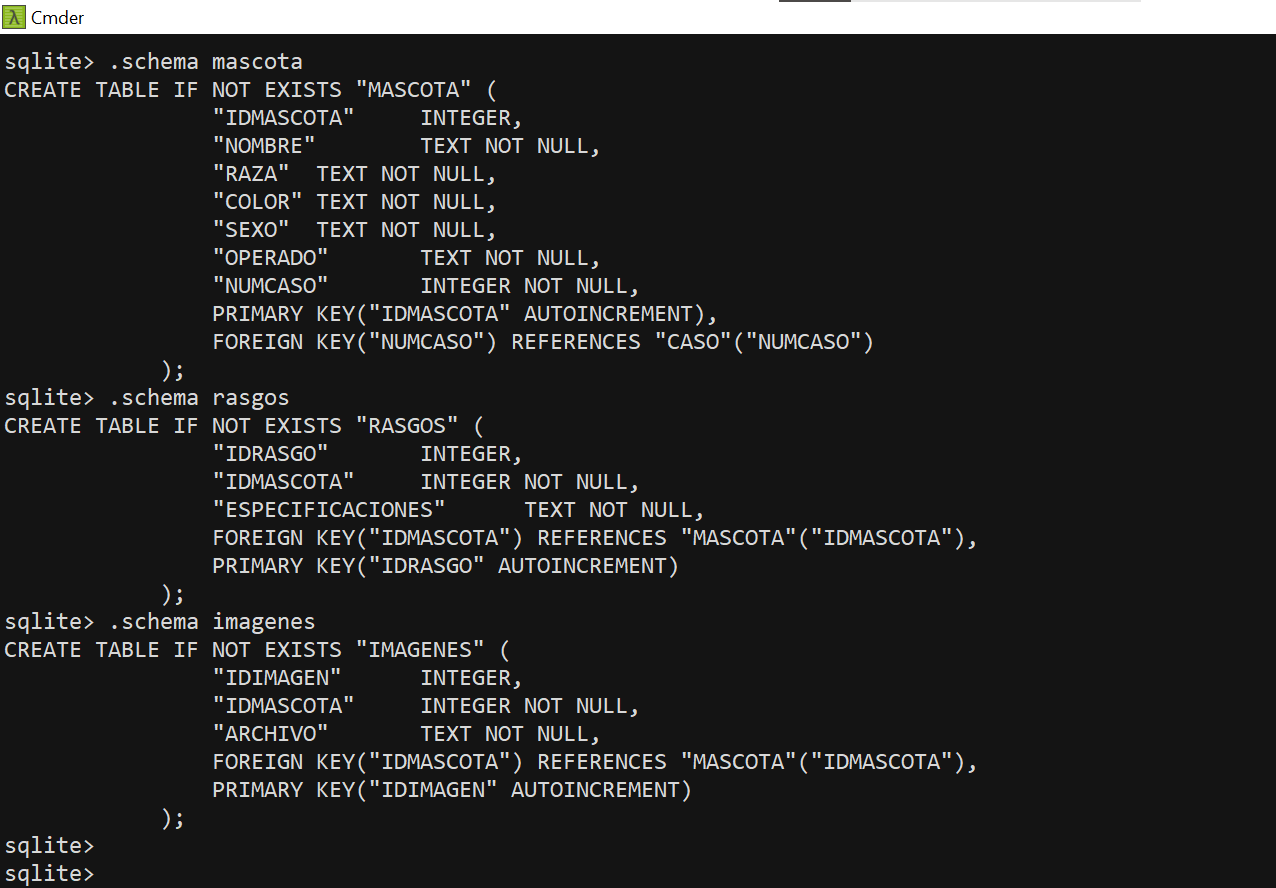
De igual forma, se muestra la clase que al ser llamada permitirá almacenar la imagen en el servicio externo ImgBB y recibir la URL que posteriormente se aloja en la BD de SQLite, como se puede visualizar aquí, esto se encuentra configurado para guardar múltiples imágenes, al igual que los métodos de la clase definidos en database.py:



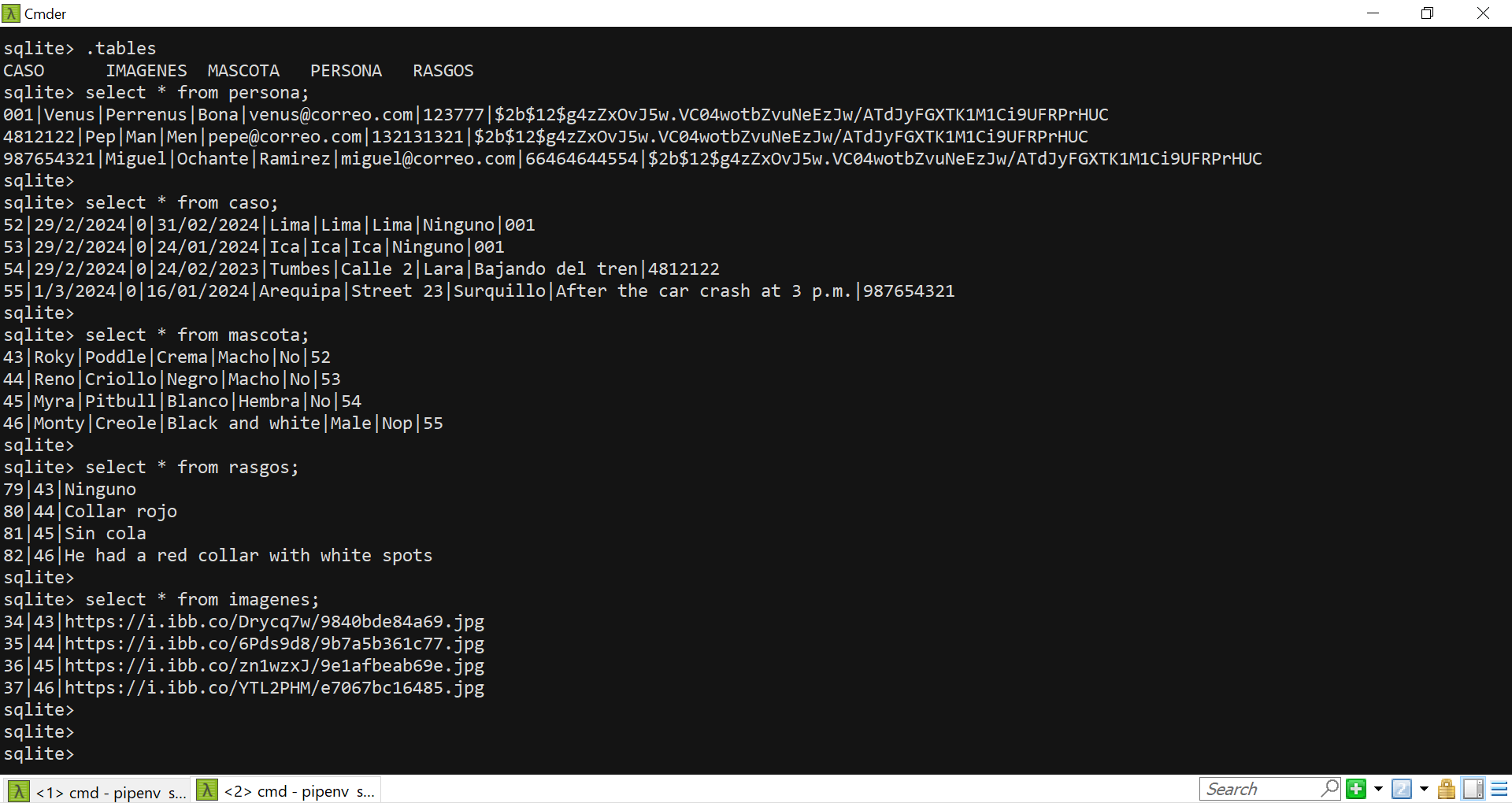
# Consulta de datos directamente en SQLite:

Consultando los esquemas:



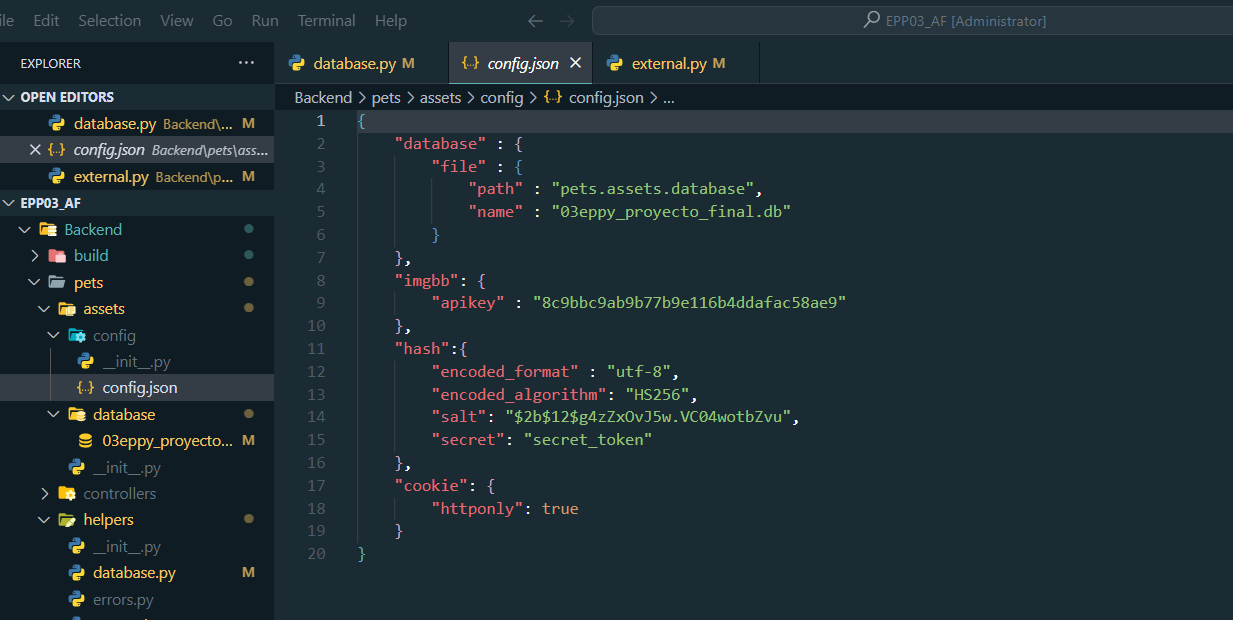


Leyendo los datos existentes



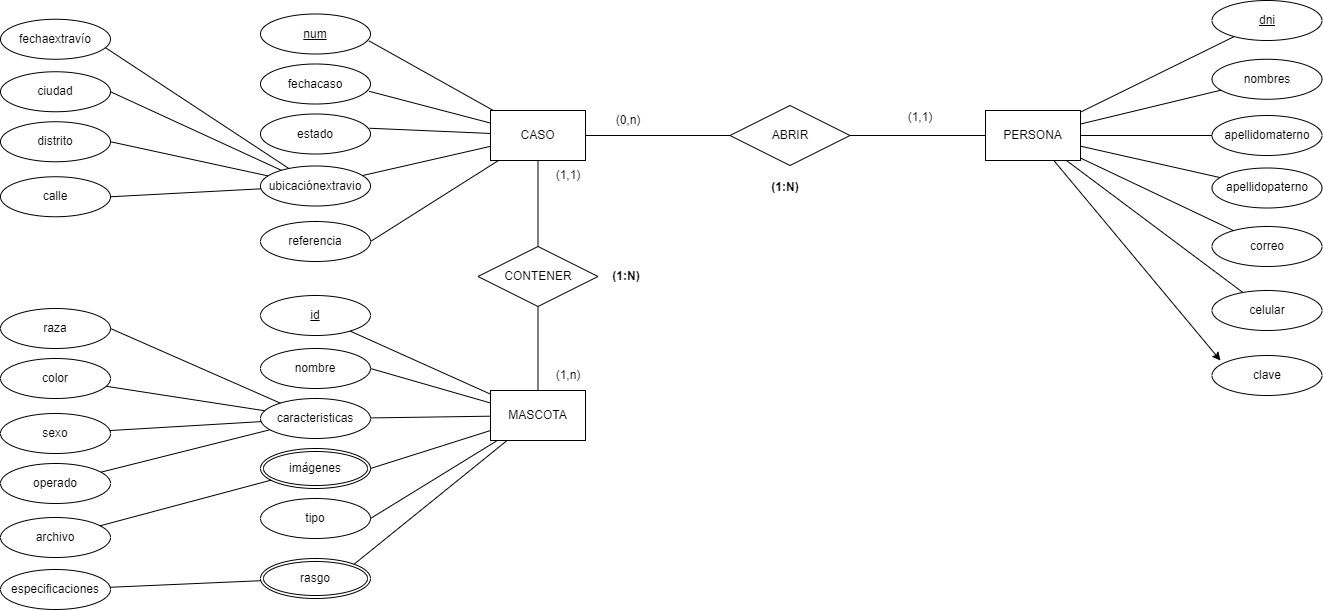
# Parámetros de configuración:

Los parámetros de configuración para iniciar la aplicación se encuentran dentro del archivo config.json que está alojado en la ruta ./Backend/pets/assets/config/.



# Parte opcional requerida:

## Modelo Entidad Relación



## Modelo relacional

