ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE APLICACIÓN WEB PARA EXTRACCIÓN DE DATOS CON WEB SCRAPING Y HERRAMIENTA DE ANALÍTICA DE DATOS

APLICATIVO WEB DE EXTRACCIÓN DE DATOS.

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE

CHRISTIAN GABRIEL SOLEDISPA MENDOZA

DIRECTOR: JUAN PABLO ZALDUMBIDE PROAÑO

DMQ, Agosto 2023

CERTIFICACIONES

Yo, Christian Gabriel Soledispa Mendoza declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

CHRISTIAN GABRIEL SOLEDISPA MENDOZA

christian.soledispa@epn.edu.ec

cgsoledispa@outlook.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Christian Gabriel Soledispa Mendoza, bajo mi supervisión.

ING. JUAN PABLO ZALDUMBIDE PROAÑO

Juan.zaldumbide@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CHRISTIAN GABRIEL SOLEDISPA MENDOZA

DEDICATORIA

El presente proyecto va dedicado a toda la comunidad de estudiantes de la Escuela Politécnica Nacional conformada por excelentes docentes y estudiantes que cada día se esfuerzan por alcanzar el éxito y sobre todo por el progreso de la sociedad. También a las personas a las que el siguiente proyecto para la extracción de datos con webscraping les sea de utilidad.

SOLEDISPA MENDOZA CHRISTIAN GABRIEL

AGRADECIMIENTO

Agradezco a mis padres y hermanos por ser mi principal apoyo durante todo este proceso en mi vida estudiantil, porque sin ellos no podría alcanzar todo lo que he logrado hasta ahora con su amor incondicional y valores que me supieron enseñar este triunfo más que para mí es para ellos.

A mi amiga Viviana por haber estado incondicionalmente a mi lado desde el primer semestre que estuve en la EPN, acompañarme en este largo y sacrificado proceso con su amistad.

A mi amiga Doménica por ser mi apoyo moral e incondicional este último tiempo que ha sido muy difícil, pero con su amistad y alegría me ha llenado de fuerzas para seguir adelante y alcanzar mis objetivos.

SOLEDISPA MENDOZA CHRISTIAN GABRIEL

ÍNDICE DE CONTENIDO

RESUMEN

Actualmente existen sistemas web que permiten realizar web scraping pero son difíciles de buscar o encontrar en el internet, esta funcionalidad es muy útil en muchos ámbitos tanto de estudio como laborales. Este sistema web para la extracción de datos tiene como objetivo facilitar este proceso y almacenar la información extraída de manera ordenada y estructurada en una base de datos NoSQl como lo es MongoDB.

¿Por qué es importante saber utilizarla si necesitas analizar datos de la web? De esta manera automatizada se puede obtener información de páginas que lo permitan y estructurar dicha información para que sea utilizada a conveniencia de los usuarios. Al aplicar esta herramienta la obtención de datos facilita la visualización de estos mediante un sistema de frontend.

Una herramienta muy importante para el desarrollo de este sistema web es Django ya que es enfocado al lenguaje de programación python y as vez este lenguaje es el más práctico para realizar web scraping de diferentes páginas web y estructurarlos en una base de datos que este caso es MongoDB.

PALABRAS CLAVE: web scraping, NoSQL, Django, Python, MongoDB.

ABSTRACT

Currently there are web systems that allow web scraping to be carried out but they are difficult to search or find on the internet, this functionality is very useful in many areas of both study and work. This web system for data extraction aims to facilitate this process and store the extracted information in an orderly and structured manner in a NoSQl database such as MongoDB.

Why is it important to know how to use it if you need to analyze web data? In this automated way, information can be obtained from pages that allow it and structure said information so that it can be used at the convenience of users. By applying this tool, data collection facilitates its visualization through a frontend system.

A very important tool for the development of this web system is Django since it is focused on the python programming language and at the same time this language is the most practical to carry out web scraping of different web pages and structure them in a database that in this case is MongoDB.

KEYWORDS: web scraping, NoSQL, Django, Python, MongoDB.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El sistema web permite la extracción de datos de diferentes páginas web con web scraping, los datos son almacenados en una base de datos NoSQL que en este caso es MongoDB.

El web scraping es una herramienta muy útil para recopilar todo tipo de datos que se encuentren online y al almacenarlos poder utilizarlos y manipularlos de la manera que sea conveniente y cualquier fin que tenga el usuario [1].

Se puede scrapear todo tipo de datos que se encuentren en la web, la mayoría de las páginas web disponen sus datos para que los scrapers puedan extraerlos [2]. El web scraping tiene muchos propósitos como realizar estudios de mercado, buscar potenciales clientes, extraer información valiosa como noticias y contenido que sea del interés del usuario [1].

Un buen webscraping es capaz de extraer datos de forma fiable tomando en cuenta aspectos importantes como utilizar localizadores precisos para poder especificar lo que la información que se quiere extraer exactamente. La calidad de los datos no es prudente trabajar con datos no estructurados ya que serpia desordenado y no se puede ofrecer buenos resultados en la búsqueda por eso se debe asegurar que los datos estén clasificados antes de la entrega [1].

El web scraping se relaciona con la indexación de la web, que indexa dicha información de la web mediante un robot lo que es utilizado por los motores de búsqueda. El web scraping se enfoca más en la transformación de los datos sin estructura en la web en datos estructurados que se pueden almacenar y analizar en bases de datos centrales, hojas de cálculo o en cualquier fuente de almacenamiento de datos [3] como se realiza en el siguiente proyecto de web scraping con la base de datos NoSql como MongoDB.

1.1 Objetivo general

Desarrollar un aplicativo web para la extracción de datos.

1.2 Objetivos específicos

1. OE1: Determinar los requerimientos globales.
2. OE2: Diseñar el modelo de la base de datos y la arquitectura
3. OE3: Extraer los datos de diferentes páginas web con web scraping.
4. OE4: Almacenar los datos extraídos en una base de datos de tiempo real.
5. OE5: Realizar la estructura de datos para que de esta manera puedan ser visualizados.

1.3 Alcance

El presente proyecto propone el despliegue de un sistema web para la extracción de datos con web scraping, por medio de herramientas funcionales como Java y Python para escribir la lógica del lado del servidor y tecnologías de API para permitir la comunicación entre sistemas que en la mayor parte del desarrollo se busque resultados precisos y claros.

El sistema proveerá de dos perfiles; el perfil administrador desde el cual se podrá administrar y gestionar la información extraída y realizará las conexiones con la base de datos en tiempo real y un perfil usuario específico donde los usuarios ingresan y buscan la información que desea obtener.

Este sistema permitirá al usuario acceder a información específica de ciertas páginas web como palabras más buscadas, tendencias e información precisa que se quiera buscar de ciertas publicaciones realizadas que con las herramientas que se tienen a mano es muy complicado de conseguir. El sistema de extracción se encargará únicamente de extraer, gestionar y analizar la información para almacenarlas en una base de datos en tiempo real y luego esta información pueda ser visualizada mediante el componente de frontend.

1.4 Marco Teórico

Metodología

Una metodología de desarrollo de software es un conjunto de métodos y técnicas utilizadas para diseñar y desarrollar un software informático. Por cuestión de organización es indispensable trabajar con una metodología para tener todos los factores ordenados y saber cómo utilizarlos, también sirven para que el desarrollo del proyecto tenga un control y así minimizar los márgenes de errores. Utilizar una metodología en el desarrollo hace que se gestionen de mejor manera los recursos y también ahorra tiempo [4].

Web scraping

Mediante el web scraping se extraen y almacenan todo tipo de datos de páginas web para que luego sean analizados y utilizados como sea conveniente. Todos los datos que sean extraídos se pueden almacenar en bases de datos. El web scraping es utilizado para muchas tareas como la recopilación rápida de datos, en el ámbito profesional para obtener ventajas realizando un estudio de mercado de la competencia y así comparar con los productos propios y por esta razón es posible bloquearlo como medio de seguridad mediante bots de software [5].

Sistema web

Un Sistema web es un software al que se accede por medio de un servidor web sin la necesidad de tener una aplicación de escritorio. En la actualidad el sistema web es muy utilizado ya que es muy práctico y rápido en la web. Las aplicaciones web evitan gastos para que no sea necesario el uso de programas que por lo general son muy costosos y se puede trabajar desde cualquier lugar donde se encuentre. Los sistemas web al trabajar con bases de datos pueden procesar y mostrar la información a los usuarios, estos sistemas no tienen la necesidad de ser instalados [6].

2 METODOLOGÍA

En la programación se utilizan las metodologías de desarrollo de software, el principal objetivo de estas refiriéndose a este ámbito es poder realizar un trabajo en equipo de manera organizada y con el paso del tiempo han pasado de ser únicamente para la organización a ser una base indispensable al momento de desarrollar software. Últimamente las metodologías ágiles de desarrollo software se han impuesto sobre muchas otras por lo tanto es muy importante saberlas para poder desarrollar software de calidad y eficaz [4]. El caso de estudio de este proyecto para la extracción de datos por medio de webscraping es aplicar una metodología de software ágil.

2.1 Metodología de Desarrollo

RAD (Rapid application development) es una metodología de desarrollo de software que se basa en prototipos, comentarios y pruebas rápidas en lugar de una planificación en específico. Con un desarrollo rápido de aplicaciones, los desarrolladores realizan varias iteraciones y actualizaciones del software sin la necesidad de empezar desde el inicio lo que garantiza que el principal objetivo sea la calidad del software y solvente los requisitos de los usuarios finales [5].

Al hacer que los usuarios puedan interactuar con prototipos que se encuentran en constante evolución la funcionalidad comercial del proyecto puede ser mucho mayor a la que se logra mediante el modelo cascada. Con RAD los errores que puedan ocurrir se pueden descubrir y actuar sobre estos en etapas tempranas del desarrollo [6].

RAD es la mejor manera para desarrollar prototipos de manera rápida para probar cada una de las funcionalidades del software sin tener que preocuparse por el producto final [5].

Recopilación de Requerimientos

Un requerimiento es una capacidad o una condición que un usuario necesita para logras un objetivo o resolver algún problema, son capacidades o condiciones que debe tener un sistema para solventar las necesidades específicas que tenga el usuario. Es un proceso de documentación, definición y mantenimiento de los requisitos, son un conjunto de

actividades que se encargan de identificar y transmitir el propósito que tiene un sistema o software y el contexto en que será utilizado [10].

2.2 Diseño de la arquitectura

En el diseño arquitectónico es una planificación basada en modelos, abstracciones teóricas y patrones que se toman en cuenta al momento de realizar un software con cierta complejidad como un paso previo a cualquier implementación de este. Así se obtiene una guía teórica detallada que permite armar de manera correcta dicho software [11]. A continuación, se presenta el modelo arquitectónico del que se basa el respectivo desarrollo del proyecto.

Arquitectura de Datos

En este proyecto entra Django que es un framework de código abierto escrito en Python de alto nivel con el que se fomenta el desarrollo limpio, práctico y rápido [12]. Django es una herramienta utilizada para desarrollar tanto full-stack de páginas web y aplicaciones como el desarrollo de servidores. Es el mejor framework para desarrollar aplicaciones web con Python y es uno de los más demandados por los desarrolladores que trabajan con este lenguaje en específico [13].

En la **Fig. 1** se observa en breve descripción las herramientas que forman parte del proyecto en desarrollo, y además una breve visualización del Modelo Vista Controlador (MVC).

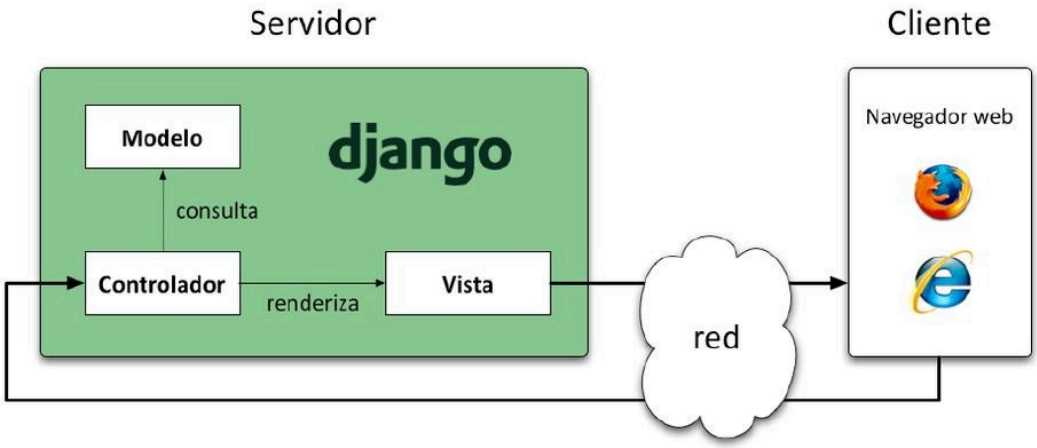

Fig. 1: Patrón arquitectónico

Patrón arquitectónico Modelo Vista Controlador

El MVC es una arquitectura de software que se utiliza para separa el código en cada una de sus funcionalidades en distintas capas las cuales se encargan de realizar una funcionalidad en específico para ofrecer diversos beneficios. Se utiliza principalmente en sistemas que requieren el uso de interfaces de usuario, aunque se puede utilizar en muchos tipos de aplicaciones. Se basa en la separación del código en tres capas distintas según la funcionalidad, Modelos, Vistas y Controladores [14].

El modelo es el backend que tiene la lógica de datos. [15]

La vista es el frontend o la interfaz gráfica de usuario. [15]

El controlador es el que controla como se muestran los datos de la aplicación. [15]

2.3 Herramientas de Desarrollo

En la **TABLA I** se observan las herramientas utilizadas debido a la estructura del proyecto y los requerimientos.

TABLA I. Herramientas fundamentales del proyecto

Herramienta	**Descripción**	**Justificación**
Django	Django es un framework de código abierto escrito en Python de alto nivel con el que se fomenta el desarrollo limpio, práctico y rápido [12].	Se utiliza este framework porque es el mejor y el más práctico según los requerimientos del proyecto y porque se utiliza Python.
Python	Python es un lenguaje de programación muy utilizado en el desarrollo de aplicaciones web, desarrollo de software, machine learning y ciencia de datos [16].	Se utiliza este lenguaje de programación ya que es el más práctico de utilizar para web scraping debido a sus librerías.
MongoDB	Es una base de datos NoSql que tiene una gran escalabilidad, flexibilidad y posee un modelo de indexación avanzado [17].	Se utiliza MongoDB porque uno de lo requerimientos del proyecto es utilizar una base de datos NoSQL y esta base de datos en tiempo real es la mejor para lo requerido.
Bs4	Es una biblioteca de Python que sirve para analizar documentos en HTML [18].	La biblioteca de beautiful soup nos permite realizar el scrapeo de las páginas web mediante scripts de pyhton.
Instaloader	Es una herramienta que permite descargar todo tipo de contenido de Instagram como fotos, videos, textos y	Instaloader se utiliza porque es una biblioteca que permite extraer información de Instagram la que se

	otros metadatos de Instagram [19].	necesita para el proyecto como tal.
Github	Es la plataforma web visual en la cual se toma a consideración las características de crear proyectos colaborarivos en repositorios y sin dificultad a la presentación, pueden ser accedidos desde lugares diferentes por medio de internet [20].	Visualiza el entorno del proyecto de una manera más gráfica y permite administrar el código por etapas a conveniencia del usuario.

3 RESULTADOS

A continuación, se muestran los resultados de los procedimientos realizados en cada uno de los pasos en el proyecto que son presentados en cada uno de los sprints aplicando la metodología RAD:

Sprint 0. Configuración del ambiente de desarrollo

Este sprint inicial hace referencia a la configuración del entorno de desarrollo, asegurándose de que todo lo necesario para empezar con este proceso y codificación esté en óptimas condiciones y con un correcto funcionamiento. Los resultados abarcados en este sprint son:

- VSCode en funcionamiento
- Proyecto en la base de datos NoSQL (MongoDB)

VSCode en funcionamiento

Se prepara el editor de código fuente creando un nuevo proyecto con el lenguaje de programación que se utiliza para el desarrollo de la aplicación como se visualiza en la **Fig 2.**

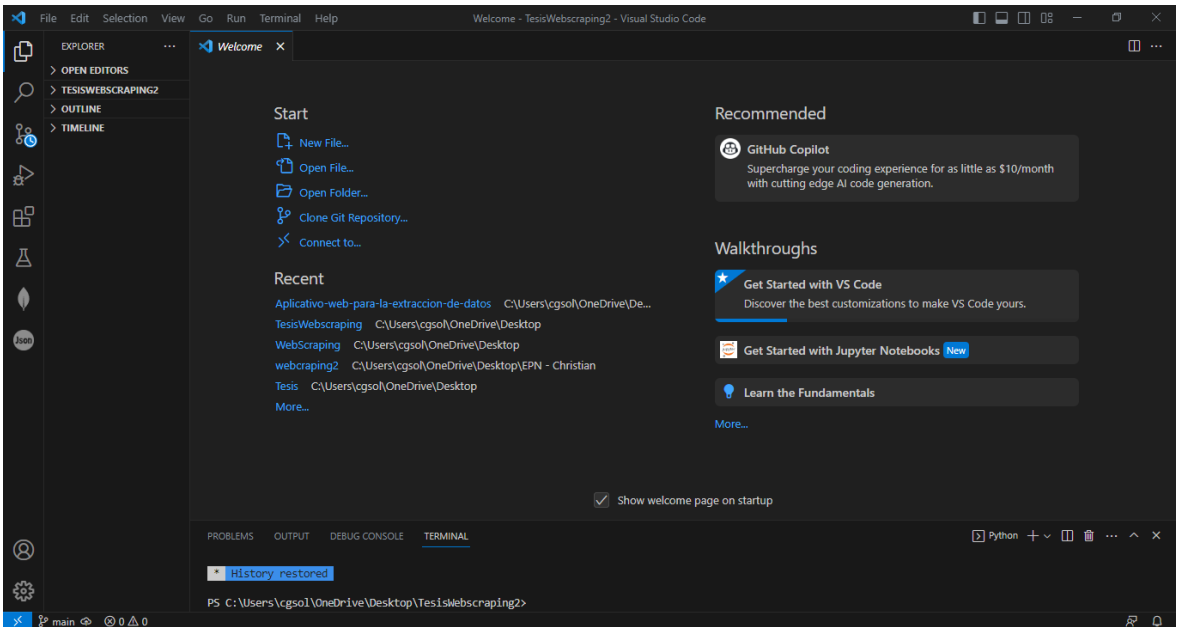

Fig 2. Ejecución del editor de código (VSCode)

Proyecto en la base de datos NoSQL (MongoDB)

Se prepara la base datos para el proyecto a desarrollar y se crea un proyecto dentro de la misma para conectar con la aplicación como se visualiza en la **Fig 3**.

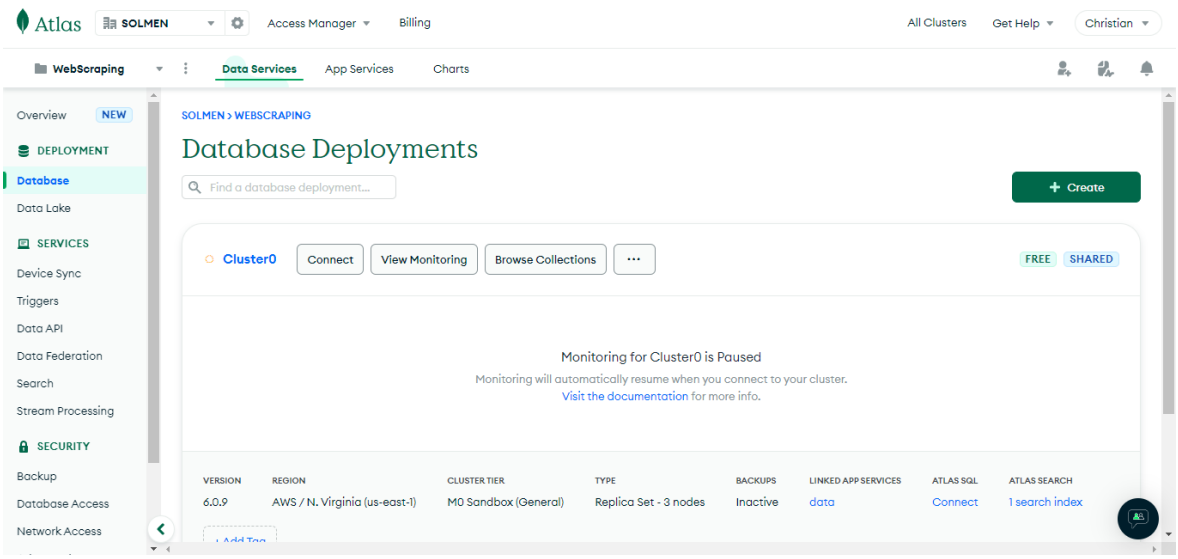

Fig 3. Base de datos en MongoDB

Sprint 1. Elección del lenguaje de programación

El sprint 1 se enfoca en el inicio de la codificación con el lenguaje de programación JavaScript en un inicio y el desarrollo del script para realizar web scraping. Los resultados abarcados en este sprint son:

- Uso de la herramienta playwright.
- Desarrollo del script con JavaScript.
- Error TimeOut.

Uso de la herramienta playwright

Se utiliza la herramienta playwright que permite realizar pruebas end to end en aplicaciones web de manera rápida y práctica [21] para realizar el webscraping mediante Google Chrome como se muestra en la **Fig 4.**

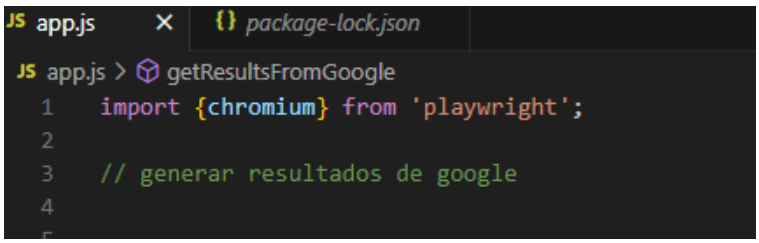

Fig 4. Se importa chromium desde playwright

Desarrollo del script con JavaScript

Se utiliza el lenguaje de programación JavaScript para codificar el script con la herramienta playwright como se muestra en la **Fig 5.**

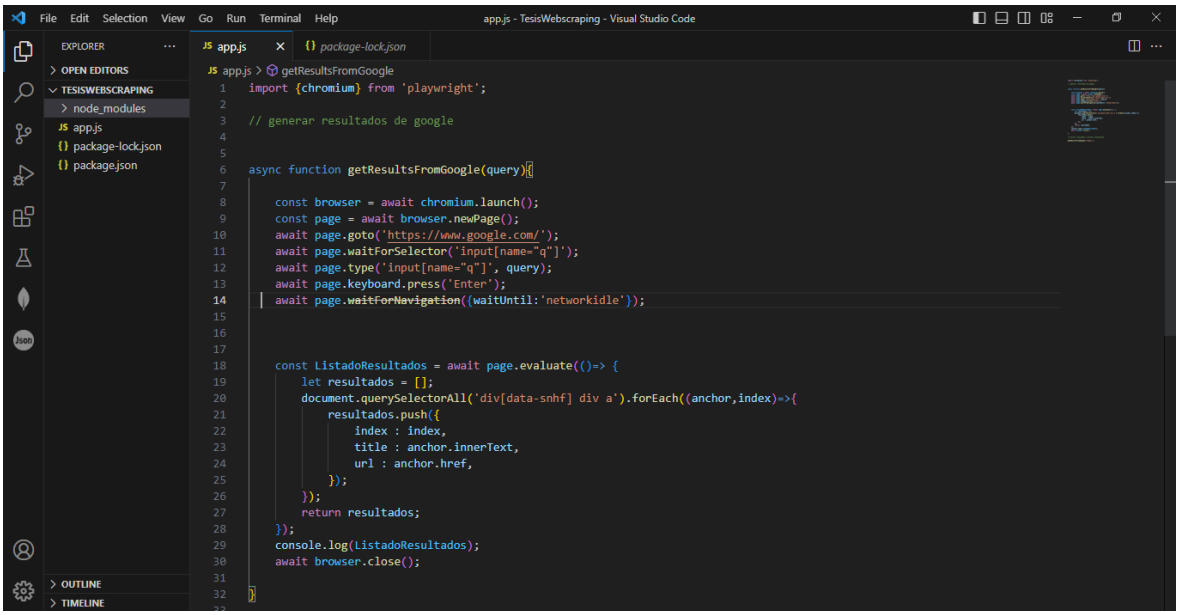

Fig 5. Script desarrollado con JavaScript

Error TimeOut

Se ejecuta el script de JavaScript con la herramienta playwright y se presenta el error TimeOut 30000ms exceeded que se visualiza en la **Fig 6.**

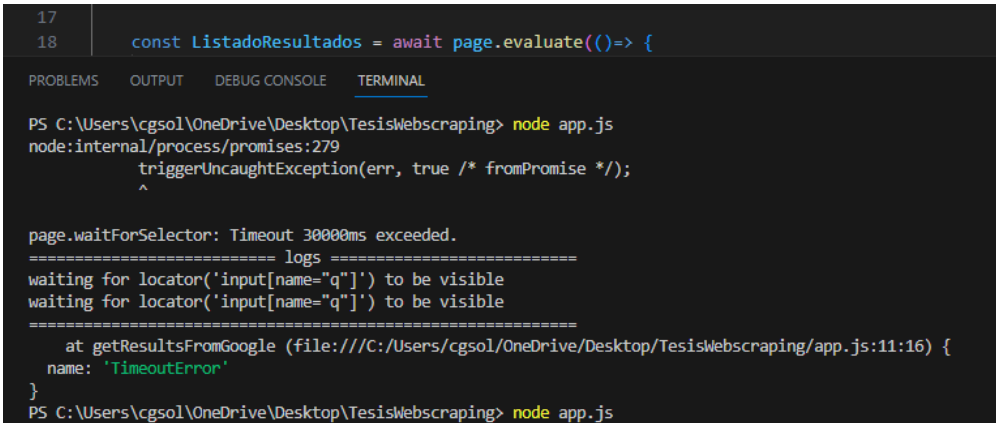

Fig 6. Error TimeOut 30000ms exceeded.

Sprint 2. Cambio del lenguaje de programación

El sprint 2 se enfoca al cambio de lenguaje de programación debido a la practicidad del nuevo lenguaje que es Python, con más variedad de librerías y documentación que permiten que desarrollo del sistema web sea de una manera más rápida, práctica y eficiente. Los resultados abarcados en este sprint son:

- Creación del nuevo proyecto con Python.

Creación del nuevo proyecto con Python.

Se crea el nuevo proyecto con el lenguaje de programación Python para evitar errores suscitados y que puedan suceder a futuro, como se visualiza en la **Fig 7**.

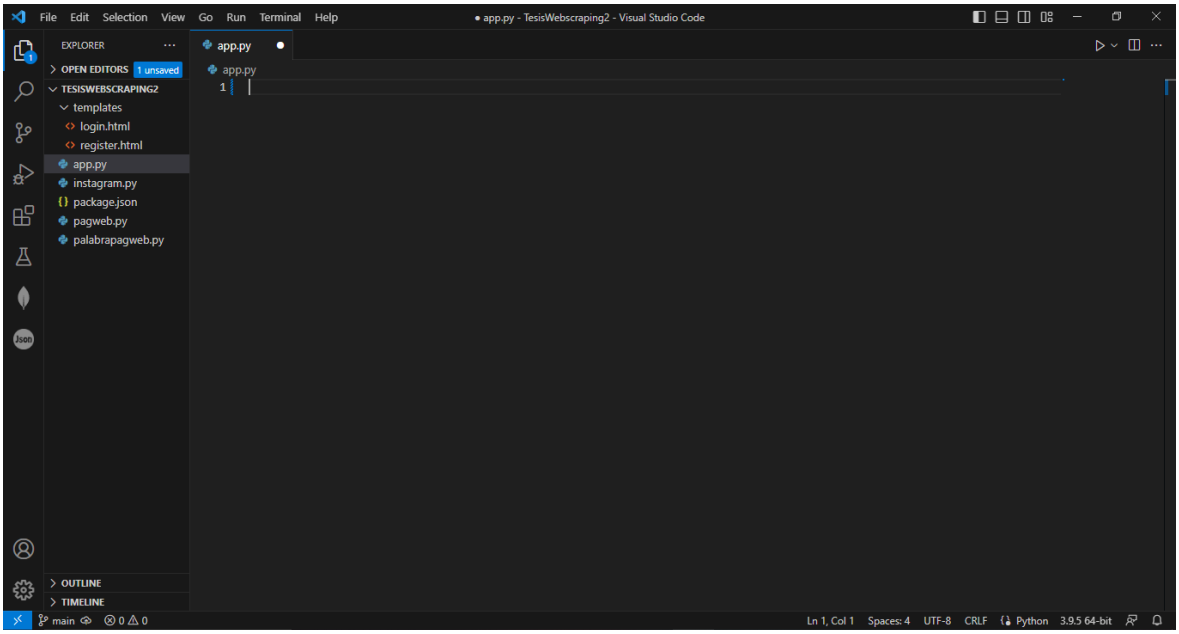

Fig 7. Nuevo proyecto en python

Sprint 3. Codificación de Scripts en pyhton para web scraping

El sprint 3 trata sobre la codificación de los scripts que van a realizar web scraping con el lenguaje de programación Python. Los resultados abarcados en este sprint son:

- Script para extraer toda la información de una página web.
- Script para realizar conteo de palabras en una página web.
- Script para extraer información de perfiles de Instagram.
- Script autenticación de usuario con flask.

Script para extraer toda la información de una página web.

Se desarrolla un script que al ejecutar se debe ingresar el link de una página web y extraer toda la información de esta, utilizando las librerías beautifulsoup4 y requests como se visualiza en las siguientes Fig 8 , Fig 9 y Fig 10.

```
PS C:\Users\cgsol\OneDrive\Desktop\TesisWebscraping2> pip install requests
Requirement already satisfied: requests in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (2.25.1)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (from requests) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (from requests) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (from requests) (1.26.6)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (from requests) (2021.5.30)
PS C:\Users\cgsol\OneDrive\Desktop\TesisWebscraping2>
```

Fig 8. Instalación de Requests

```
PS C:\Users\cgsol\OneDrive\Desktop\TesisWebscraping2> pip install beautifulsoup4
Requirement already satisfied: beautifulsoup4 in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (4.9.3)
Requirement already satisfied: soupsieve>1.2 in c:\users\cgsol\appdata\local\programs\python\python39\lib\site-packages (from beautifulsoup4) (2.2.1)
PS C:\Users\cgsol\OneDrive\Desktop\TesisWebscraping2>
```

Fig 9. Instalación de beautifoulsoup4

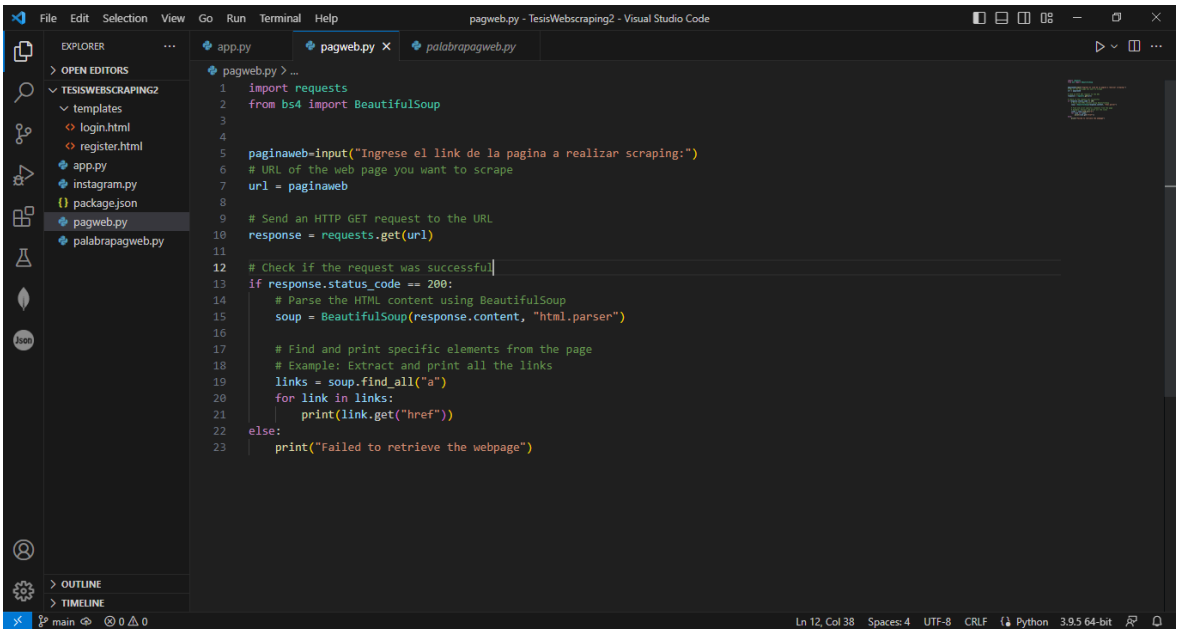

Fig 10. Codificación del script para extraer información de una página web

Se ejecuta el script y realiza el scraping ingresando el enlace de la página web de la que se desea extraer la información como se muestra en la Fig 11.

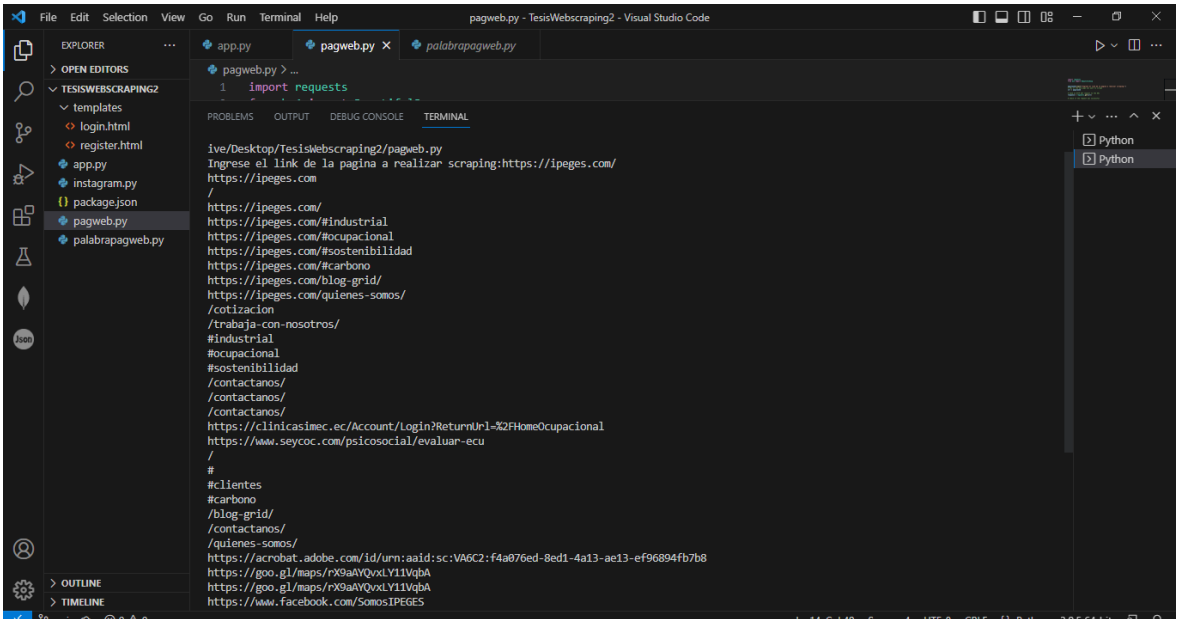

Fig 11. Visualización de la información extraída

Script para realizar conteo de palabras en una página web

Se desarrolla un script que al ejecutar se debe ingresar el enlace de una página web y una palabra que se requiera buscar para realizar un conteo de esta palabra y obtener la información de cuántas veces se la encontró en la página web para que requerimos la instalación de librerías como requests y re lo que se muestra en Fig 12.

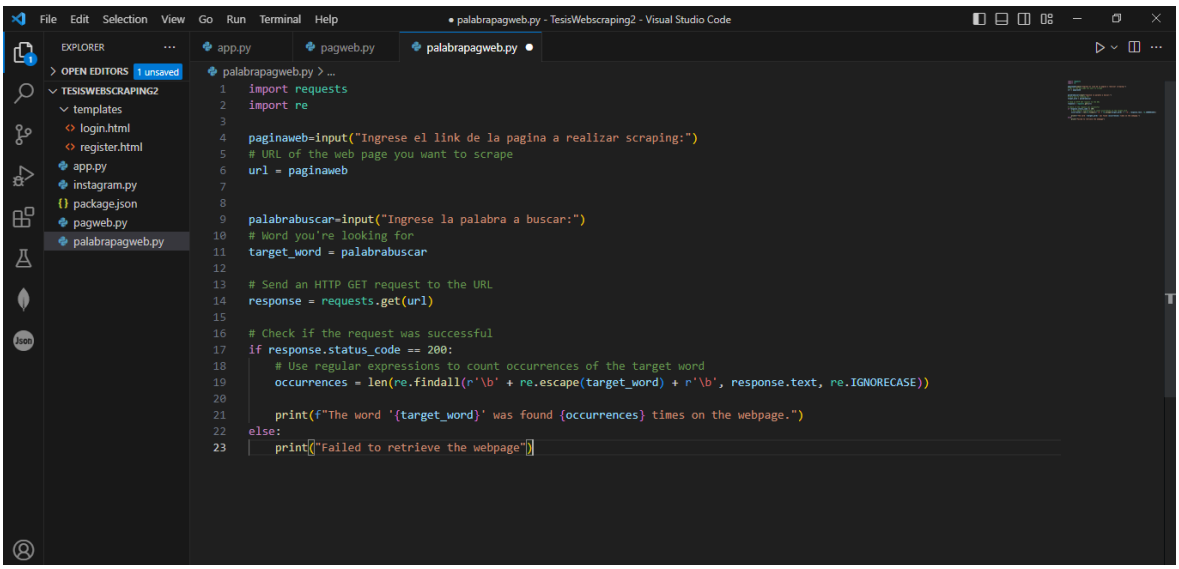

Fig 12. Script para conteo de palabras en una página web.

Se ejecuta el script y al ingresar el enlace de la página web y la palabra de la cual se desea extraer este tipo de información para que realice es scraping y se visualicen los resultados como se muestra en la Fig 13.

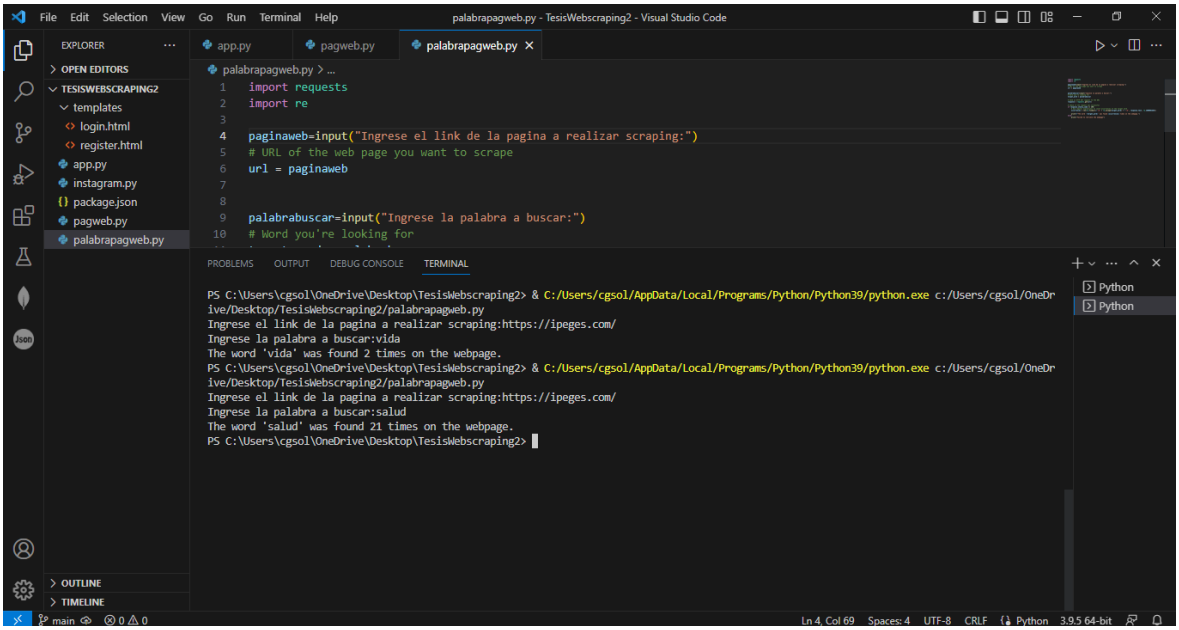

Fig 13. Ejecución y visualización de los datos extraídos del conteo de palabras

Script para extraer información de perfiles de Instagram

Se desarrolla un script utilizando las librerías instaloader y pandas para la extracción de datos de perfiles de la red social Instagram y visualice la información como se muestra en Fig 14, Fig 15 y Fig 16.

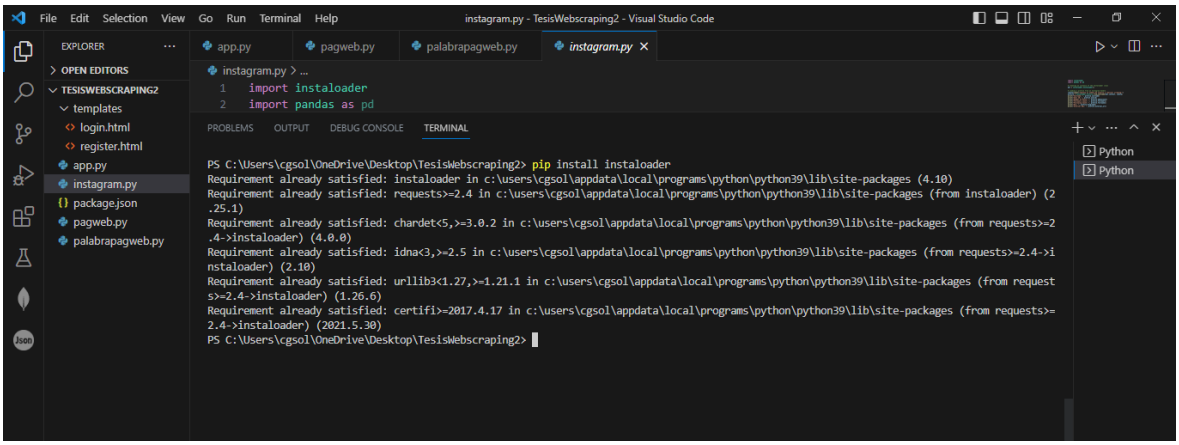

Fig 14. Instalación de Instaloader

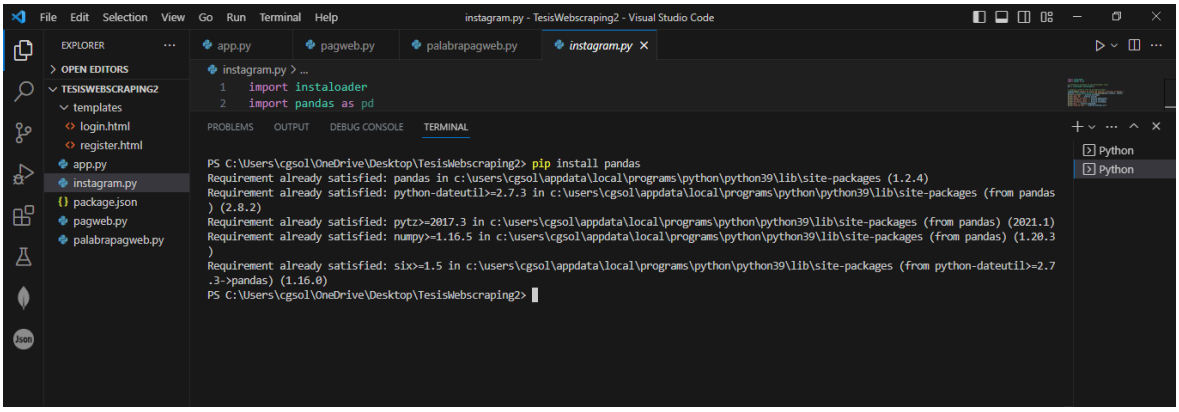

Fig 15. Instalación de pandas

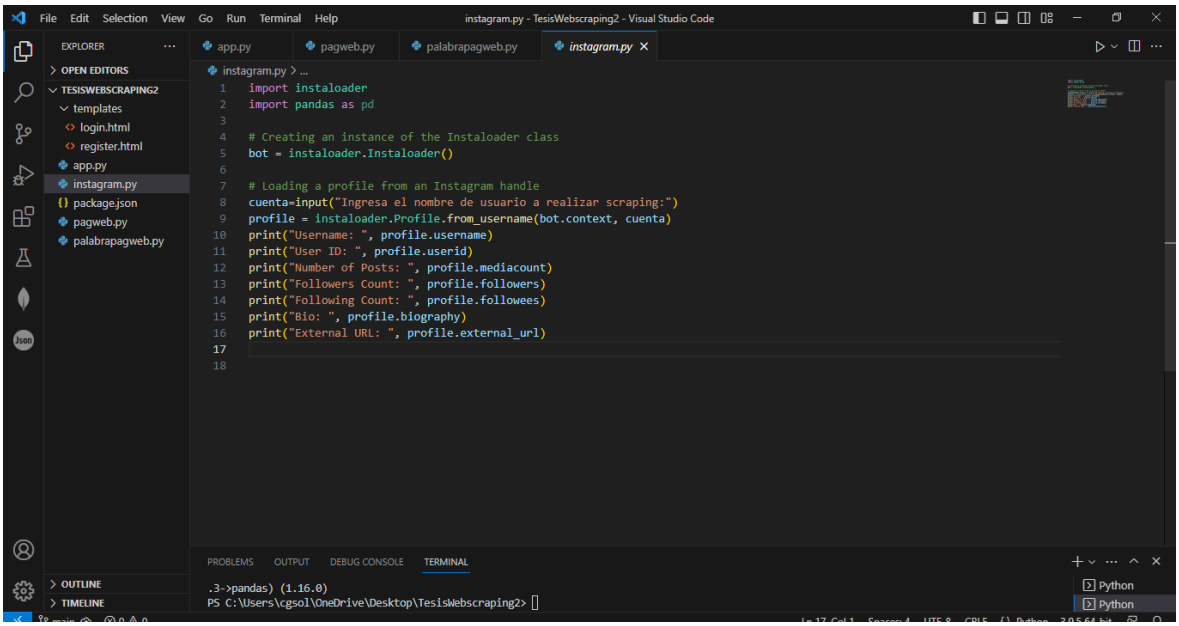

Fig 16. Script para la extracción de datos de Instagram

Se ejecuta el script, se ingresa el usuario de la red social del que se desea extraer la información del perfil para luego visualizarla como se muestra en la Fig 17.

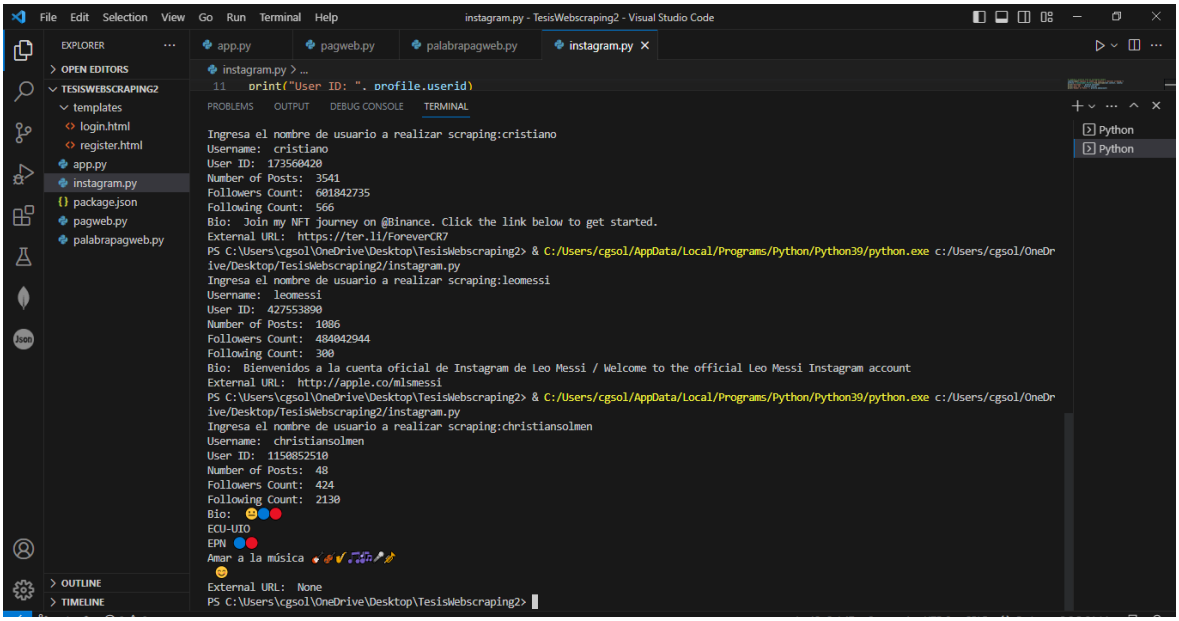

Fig 17. Ejecución y visualización de la información de los perfiles

Script autenticación de usuario con flask.

Se codifica un script para la autenticación de usuario ya conectado a la base de datos NoSQL (MongoDB), realizando la conexión con la base pero generando un error en el script como se muestra en Fig 18, Fig 19 y Fig 20.

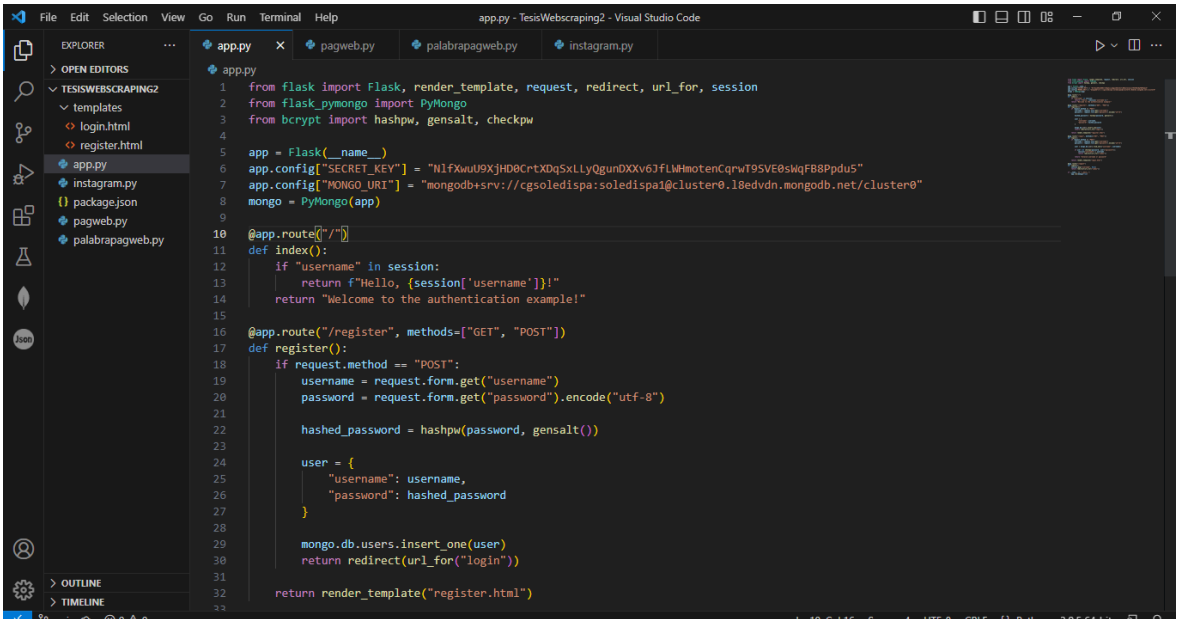

Fig 18. Primera parte del Script de autenticación con flask.

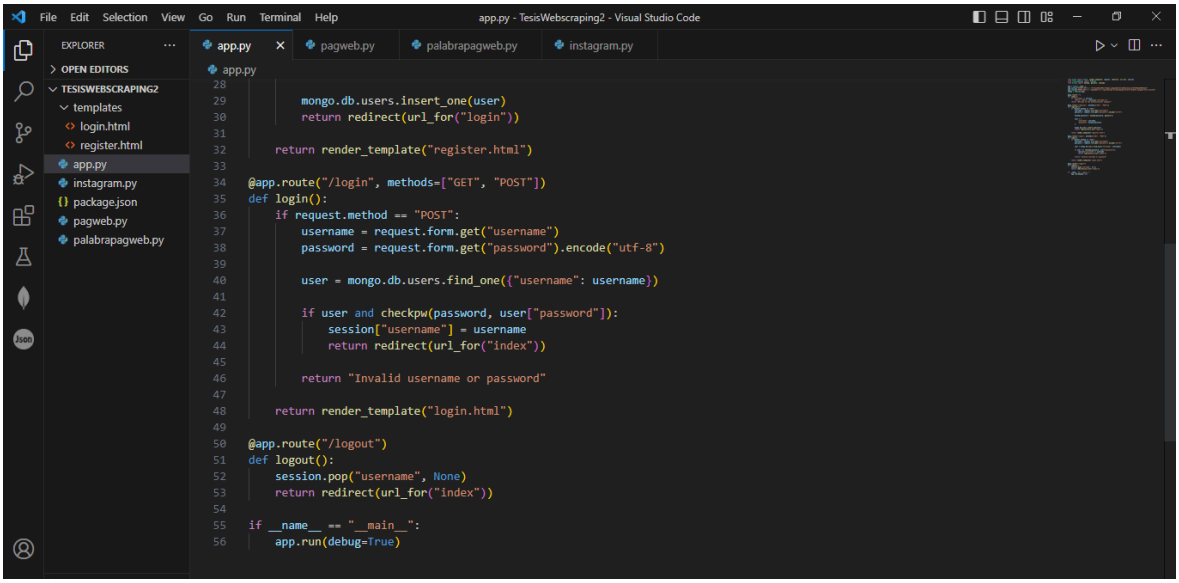

Fig 19. Segunda parte del Script de autenticación con flask.

Se ejecuta el script y se realiza la conexión con la base de datos pero el editor de código muestra un error como se visualiza en Fig 20 y Fig 21.

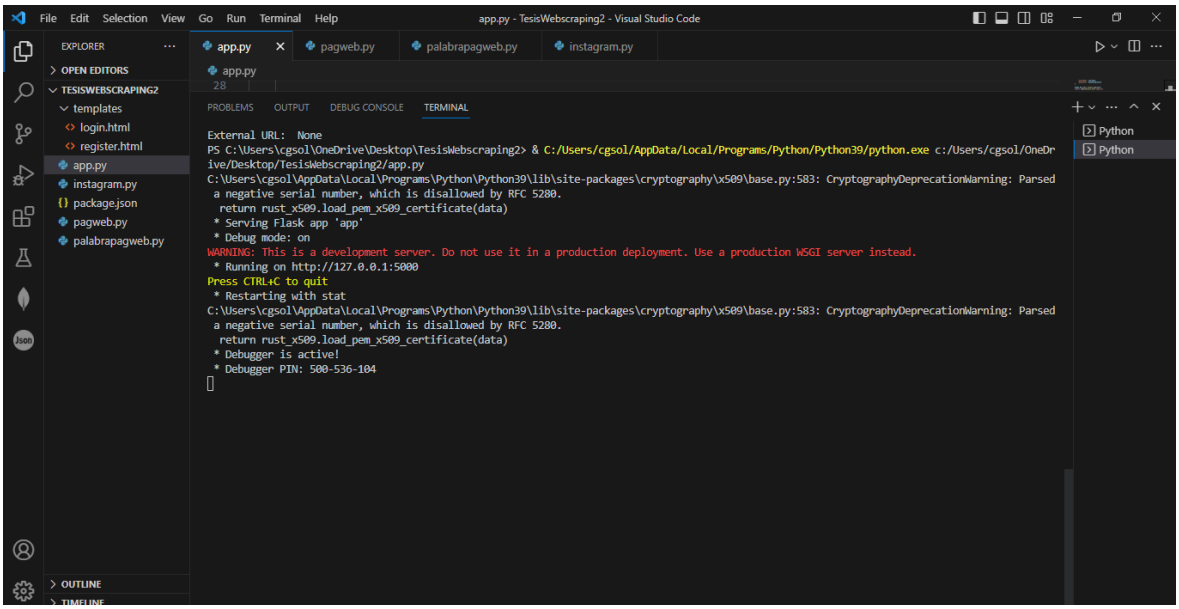

Fig 20. Error en la ejecución del script de autenticación con flask

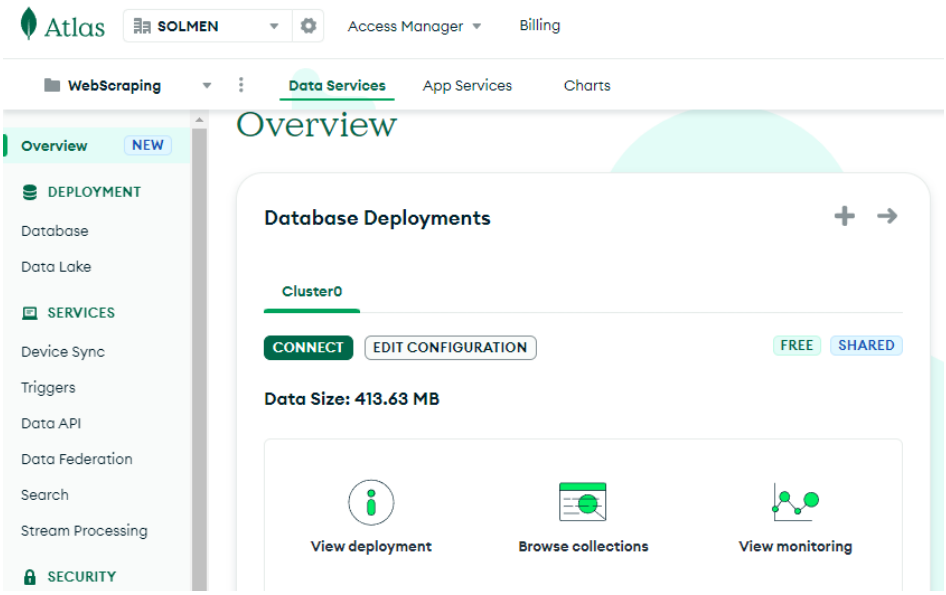

Fig 21. Conexión exitosa con MongoDB

Sprint 4. Implementación de registro de usuarios

El sprint 4 se enfoca en la implementación del registro de usuario para lo cual se utiliza Django ya que con este framework el desarrollo de la autenticación de usuarios se vuelve más práctica al tener facilidad de crear una interfaz amigable que se ajusta a lo requerimientos del sistema de web scraping. Los resultados abarcados en este sprint son:

- Registro de un usuario

Registro de un usuario

El script realiza el proceso del registro de un usuario mediante el host en formato JSON para que los mismos puedan ser almacenados de una manera ordenada y estructurada como se muestra en la Fig 22.

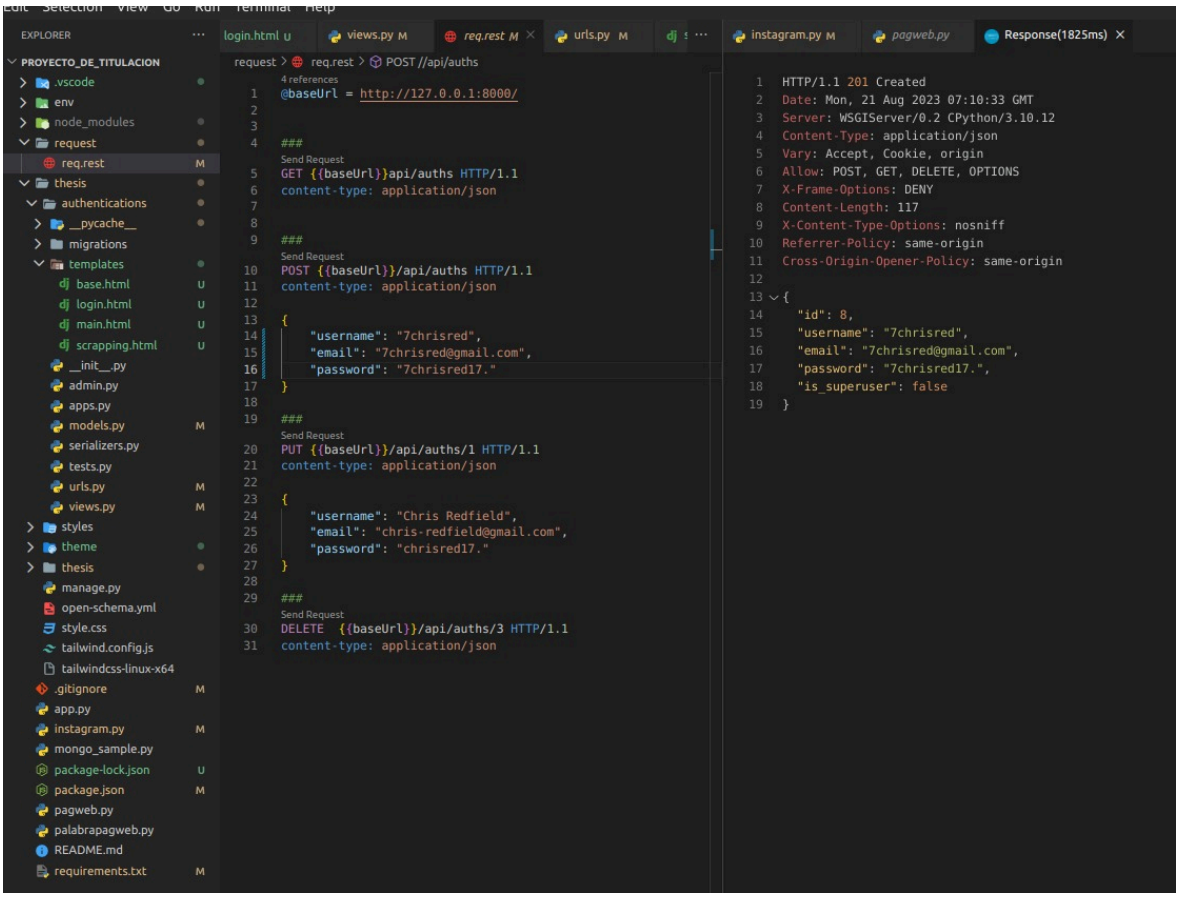

Fig 22. Script del registro de usuarios

Sprint 5. Implementación de Lista de usuarios

El sprint 5 se enfoca en la creación de la lista de usuarios que hayan sido registrados mediante el host y con la estructura definida. Los resultados abarcados en este sprint son:

- Lista de usuarios.

Lista de usuarios

Con el script se genera una lista de todos los usuarios que son registrados, esta lista tiene la estructura de username, email y password como se visualiza en la Fig 23.

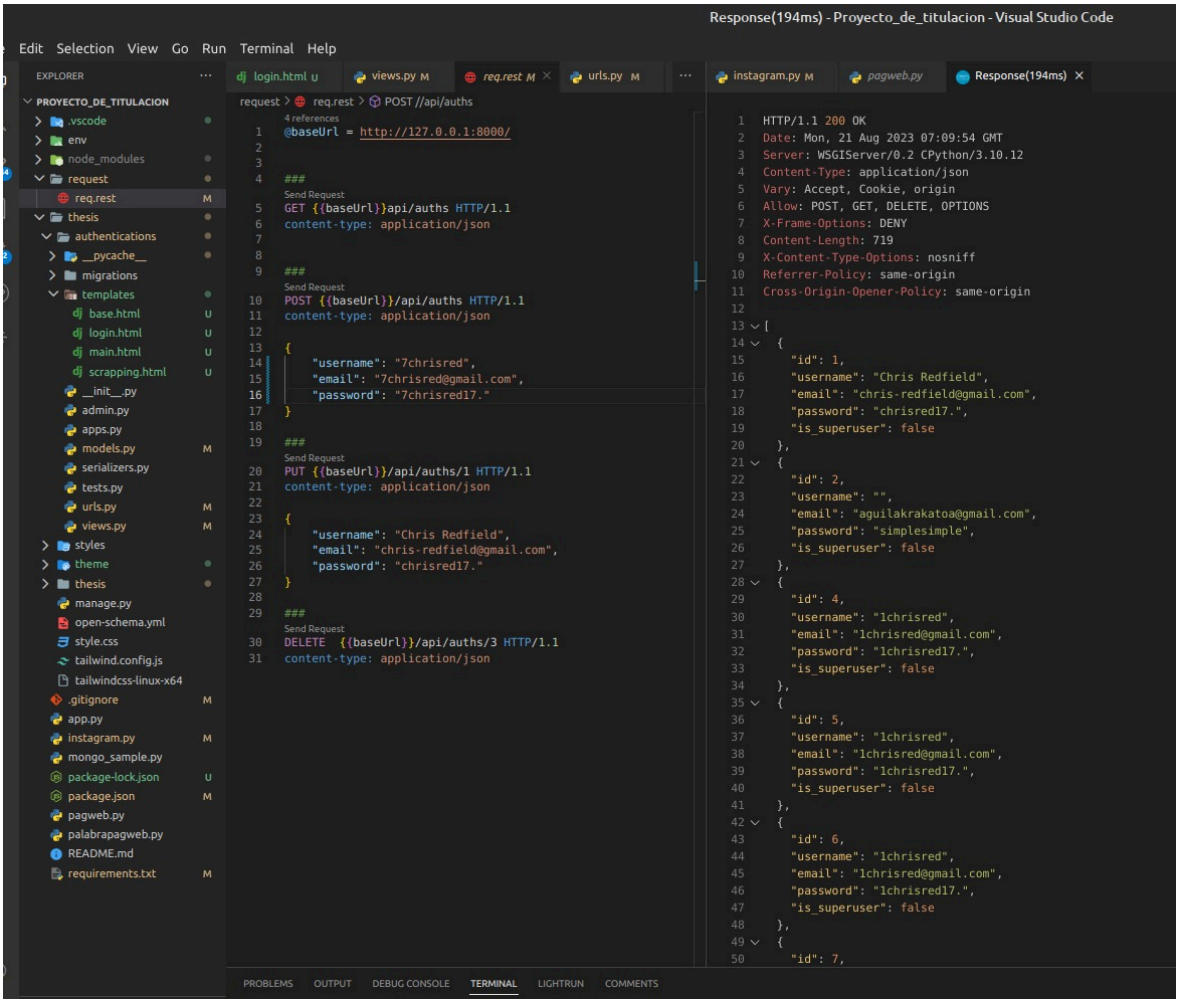

Fig 23. Lista de usuarios

Sprint 6. Implementación de Actualización de usuarios

El sprint 6 se enfoca en la implementación de la función para actualizar un usuario registrado con nueva información en las listas de usuarios registrados sin tener problemas en la autenticación. Los resultados abarcados en este sprint son:

- Actualización de usuarios.

Actualización de usuarios.

Toda la información de un usuario puede ser actualizada mediante el script con Django, esto permite realizar la autenticación luego de la actualización sin tener problemas como se muestra en la Fig 24.

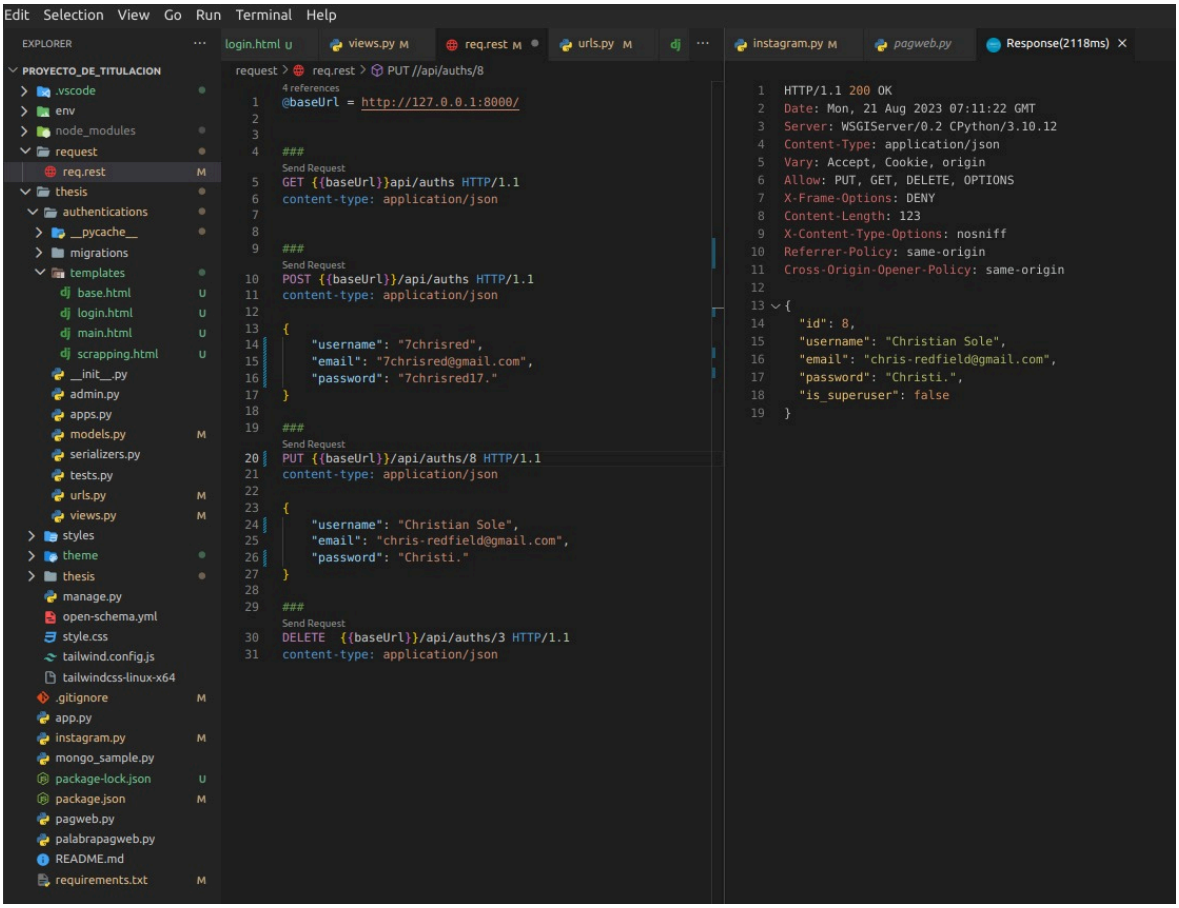

Fig 24. Actualización de usuarios

Sprint 7. Implementación de Borrar usuarios

En el sprint 7 se desarrolla la función que permite eliminar un usuario registrado y que no conste en la base, así cuando se realiza el proceso de autenticación aparecerá como usuario inexistente. Los resultados que abarca este sprint son:

- Borrar usuarios.

Borrar usuarios

En el script se crea una función que elimina el usuario que previamente debe estar registrado y al autenticar muestra el mensaje de que el usuario no existe como se muestra en la Fig 25.

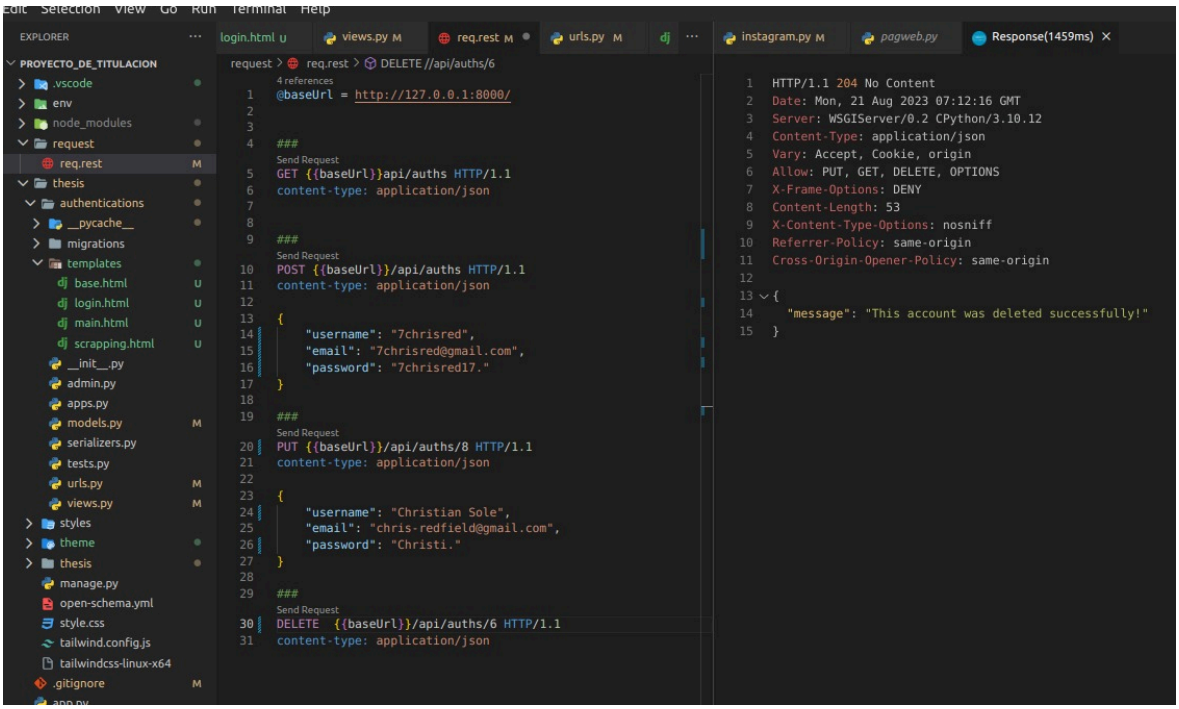

Fig 25. Borrar usuarios

Sprint 8. Implementación Login para el usuario en la web

El sprint 8 se enfoca en el desarrollo de la interfaz en la web con el uso de Django para el login del usuario. Los resultados que abarca este sprint son:

- Login para el usuario en la web

Login para el usuario en la web

En el script se desarrolla el código en html donde el usuario hace login mediante la interfaz proporcionada igual por Django como se muestra en la Fig 26.

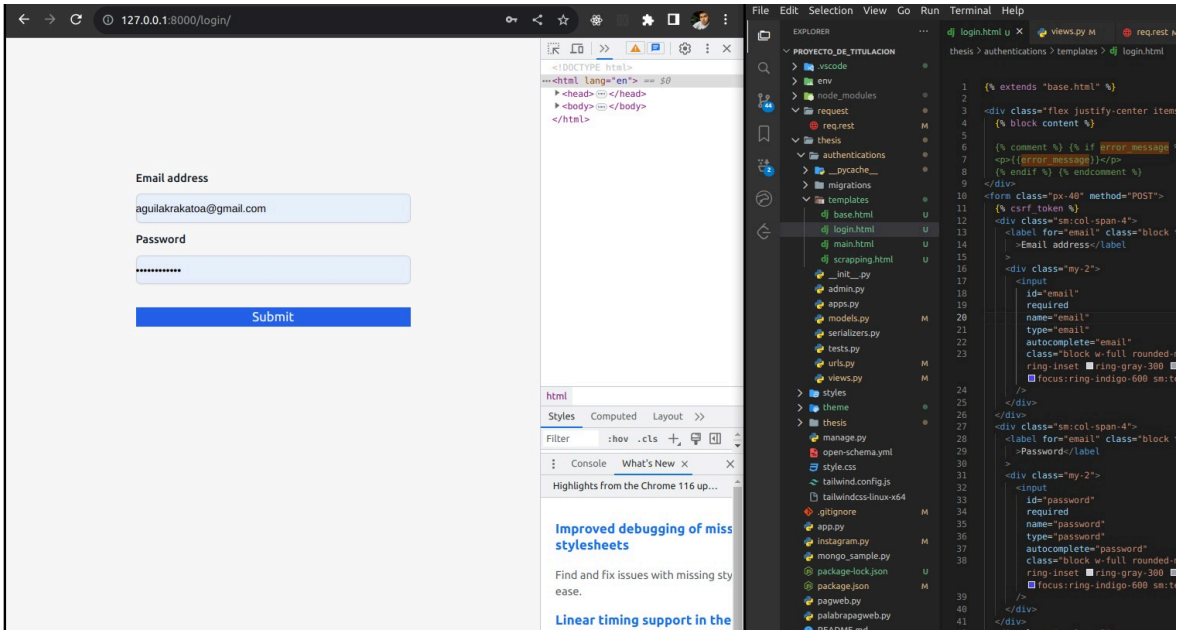

Fig 26. Login para el usuario

Sprint 9. Implementación de scraping en la interfaz web

El sprint 9 se enfoca en el desarrollo de la interfaz web mediante Django y html donde el usuario ingresa y busca tanto usuario de Instagram como página web a scrapear. Los resultados que abarca este sprint son:

- Scraping en la web.

Scraping en la web

En el script se desarrolla un código en html con Django que presenta al usuario una interfaz donde puede ingresar la información para lo que se quiere realizar scraping como se muestra en la Fig 27.

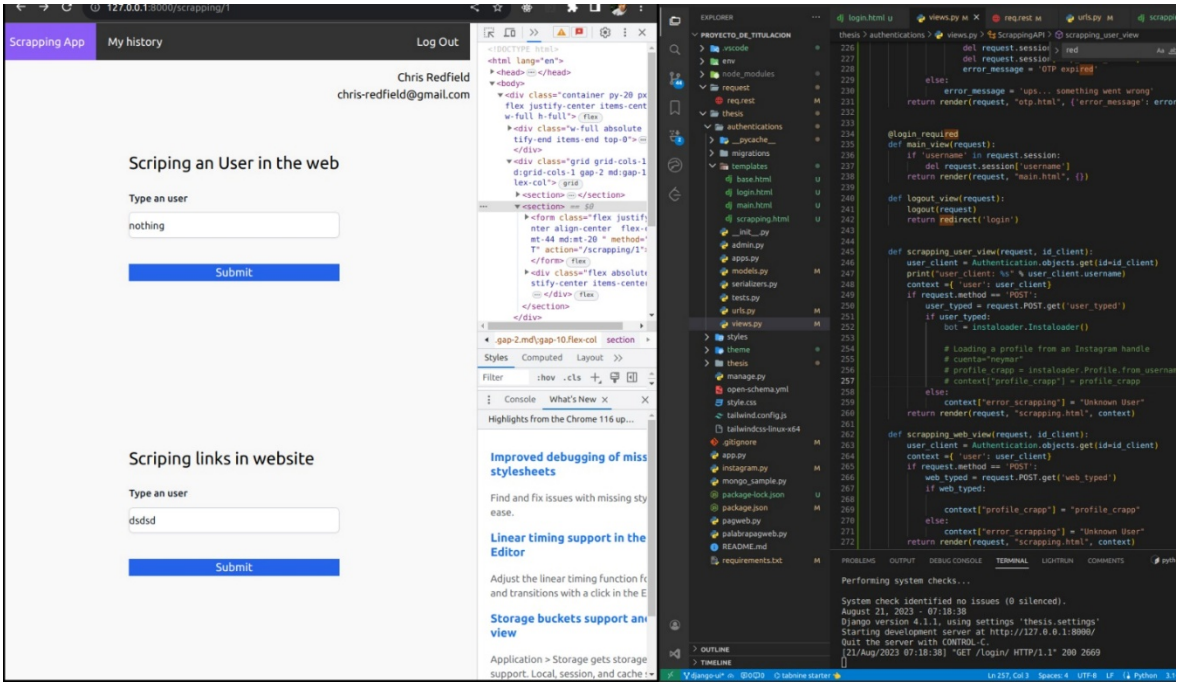

Fig 27. Scraping en la web

Sprint 10. Almacenamiento de usuarios registrados en la base de datos

El sprint 10 se enfoca en almacenar los datos de los usuarios registrados, en la base de datos MongoDB. Los resultados que abarca este sprint son:

- Almacenamiento de usuarios en la base de datos.

Almacenamiento de usuarios en la base de datos

Todos los datos de los usuarios registrados son almacenados en la base de datos NoSQL como se muestra en la Fig 28.

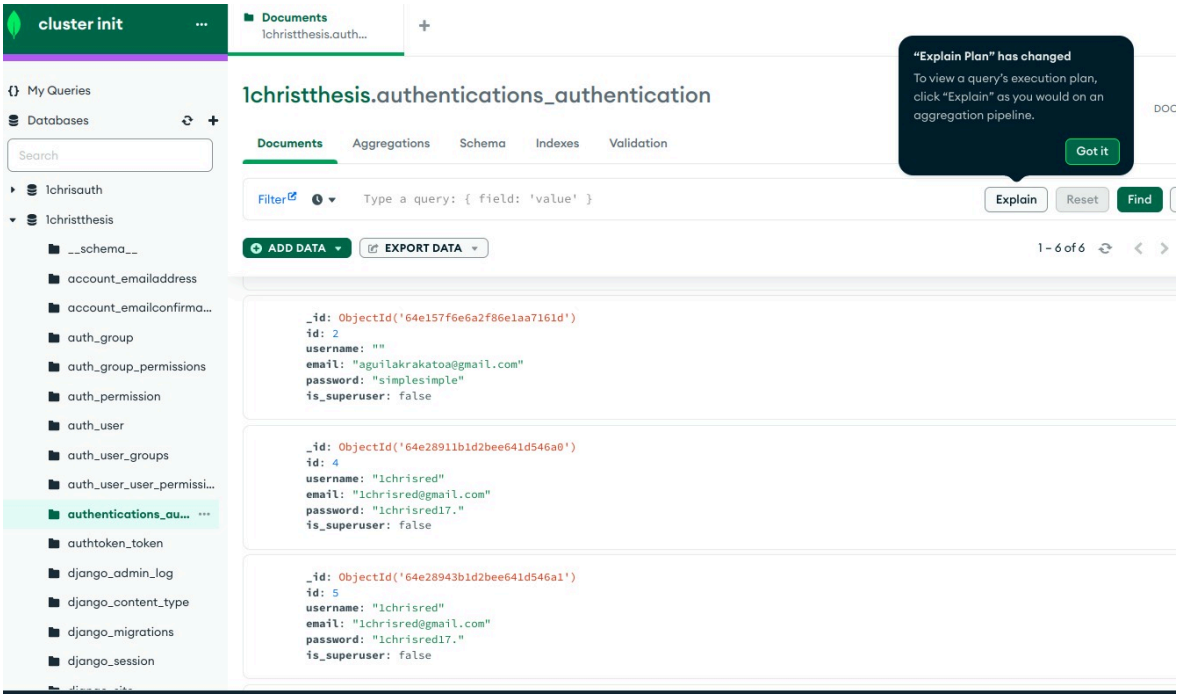

Fig 28. Información en la base de datos

Sprint 11. Scraping anchors en la web

El sprint 11 se enfoca en mostrar los resultados del scraping realizado en la interfaz, con una consulta de la API el servidor arroja los resultados de un sitio web. Los resultados sobre este sprint son:

- Scraping anchors en el sitio web.
- Consulta de la API.

Scraping anchors en el sitio web

Los resultados de realizar scraping se visualizan en la interfaz luego de realizar el login como se muestra en la Fig 28.

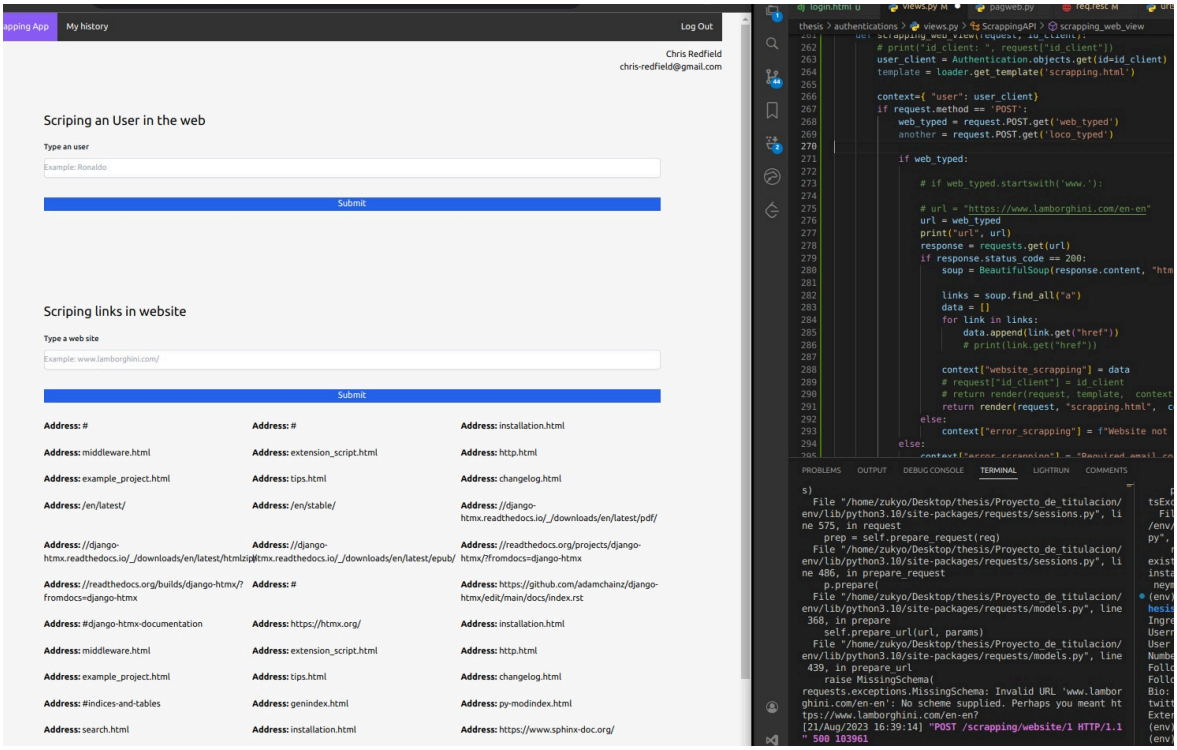

Fig 28. Scraping anchors en la web

Consulta de la API

La consulta de la API al servidor arroja los resultados de anchors de un sitio web como se muestra en la Fig 29.

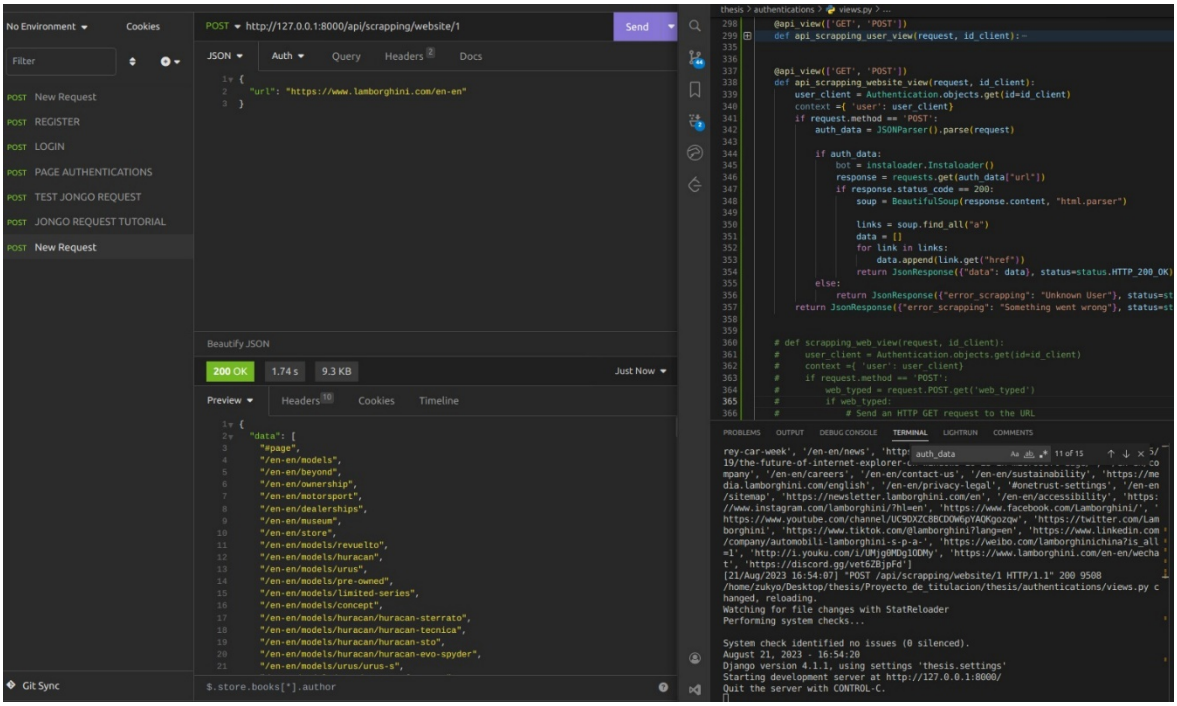

Fig 29. Consulta de la API

Sprint 12. Rutas de Django y Modelos de usuario

El sprint 12 se enfoca en la creación de las rutas con Django y la ceración de los modelos de usuarios. Los resultados que abarcan este sprint son:

- Rutas de Django
- Modelos de usuario

Rutas Django

El script muestra las rutas creadas para los datos tanto de autenticación como los datos extraídos de las páginas web como se muestra en las Fig 30 y Fig 31.

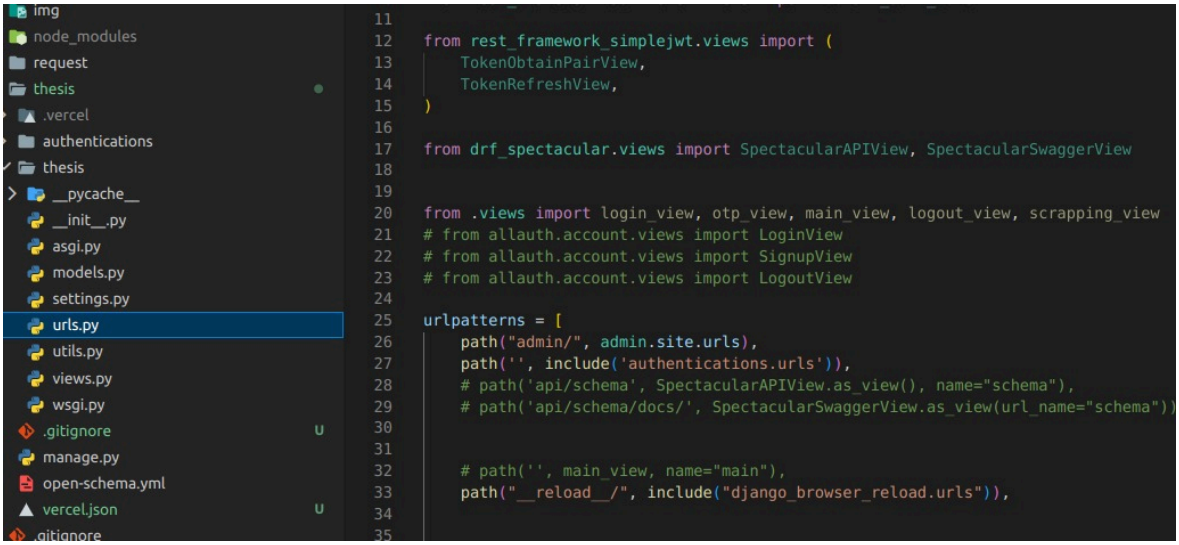

Fig 30. Definición de las rutas

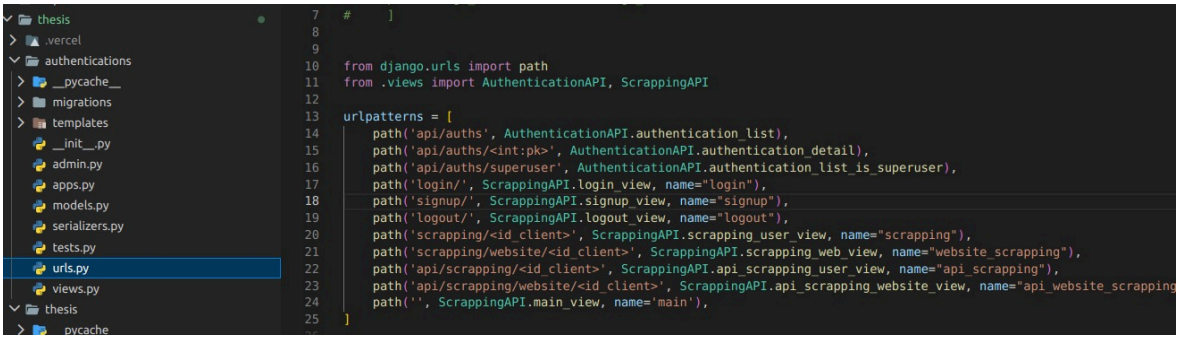

Fig 31. Rutas Django

Modelos de usuario

Se estructuran los modelos de usuarios mediante djongo y tener la estructura deseada como se muestra en la Fig 32.

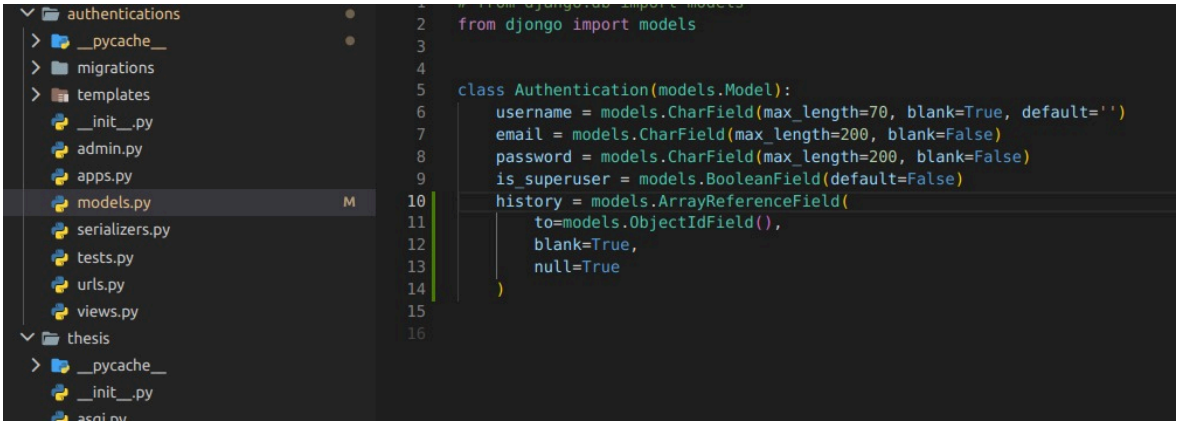

Fig 32. Modelos de usuarios

Sprint 13. Implementación de vistas

El sprint 13 se enfoca en la implementación de vistas de frontend tanto para el login y singup como para el scraping. Los resultados de este sprint son los siguientes:

- Vistas frontend de login y singup
- Vistas frrontend del scraping

Vistas Frontend de login y singup

El script contiene las vistas en la interfaz login y singup como se visualiza en la Fig 33.

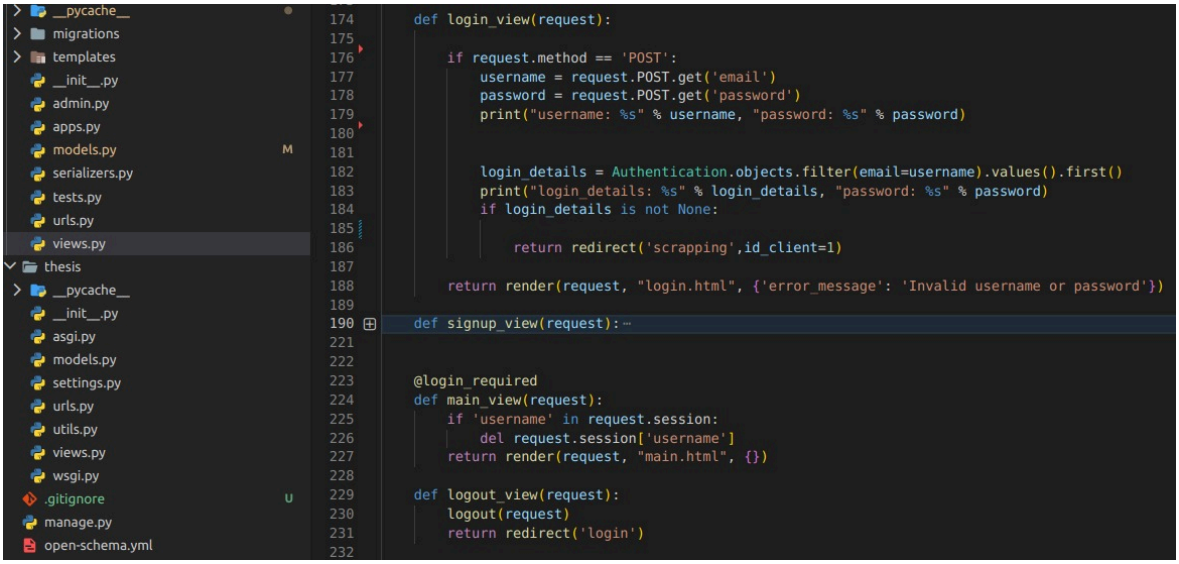

Fig 33. Vistas Login y Singup

Vistas Frontend del scraping

El script contiene las vistas en de la interfaz para el web scraping como se visualiza en la Fig 34.

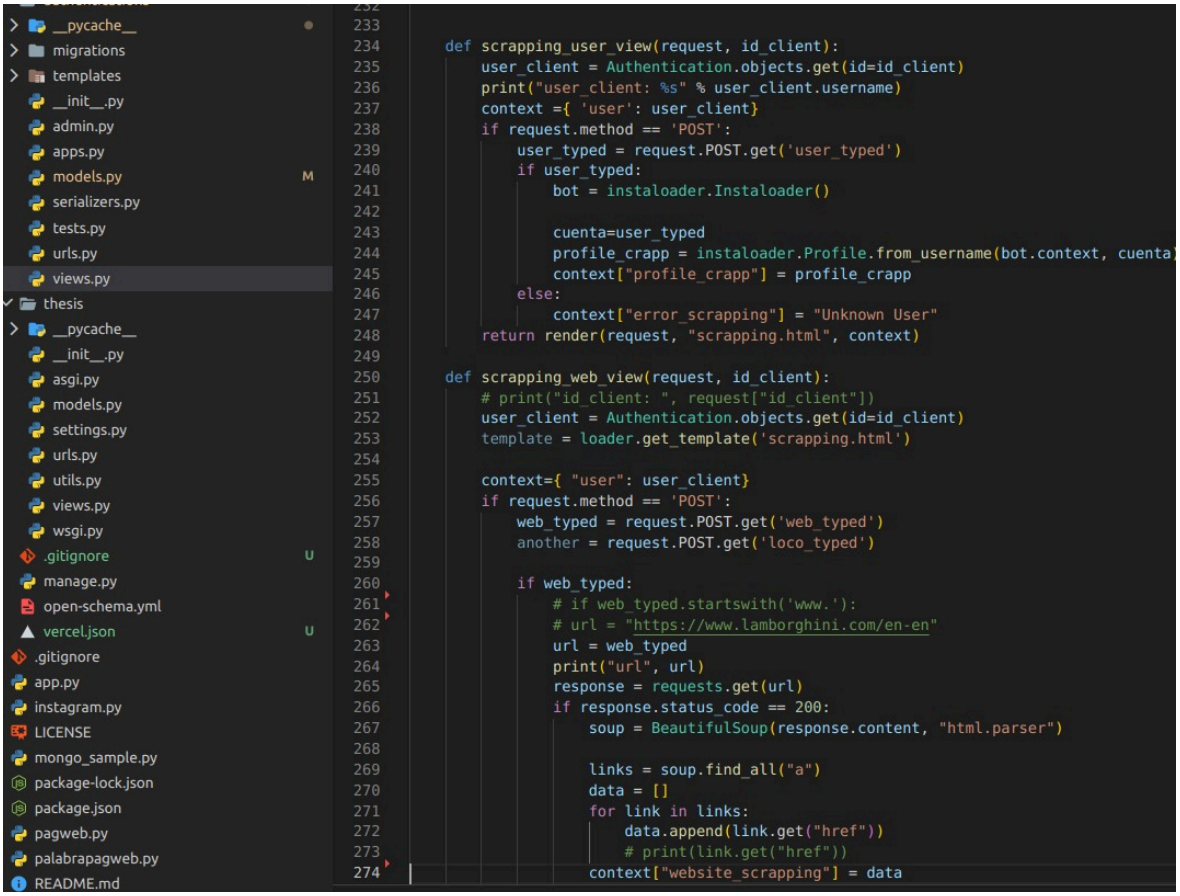

Fig 34. Vistas scraping

Sprint 14. Implementación de API Rest de autenticación y API Rest de listas de usuarios

En el sprint 14 se desarrolla el consuma de las API Rest tanto de autenticación como de listas de usuarios. Los resultados de este sprint son:

- API Rest de autenticación
- API Rest de Listas de usuarios

API Rest de autenticación

Se codifica la API Rest de autenticación para que arroje los resultados de los usuarios registrados y se realice la autenticación correctamente como se visualiza en la Fig 35.

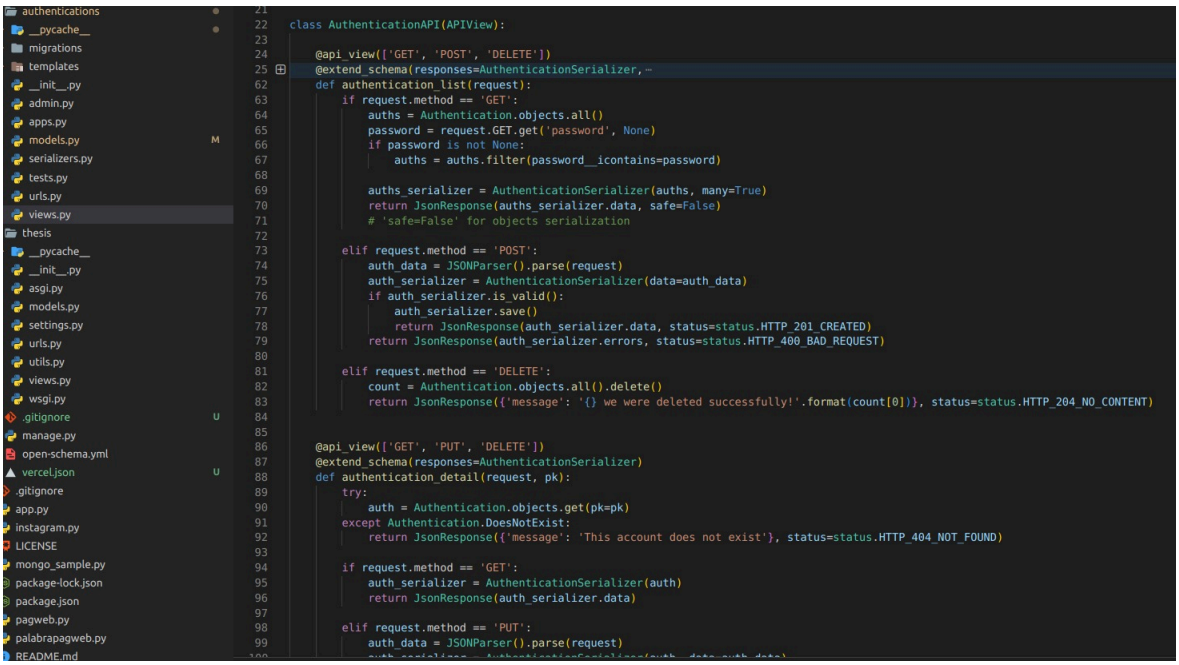

Fig 35. API Rest de authentication

API Rest de Listas de usuarios

Se codifica la API Rest de Listas de usuarios para que esta arroje los resultados de las listas correctamente y de igual manera no tener problemas con la autenticación de los usuarios como se muestra en la Fig 36.

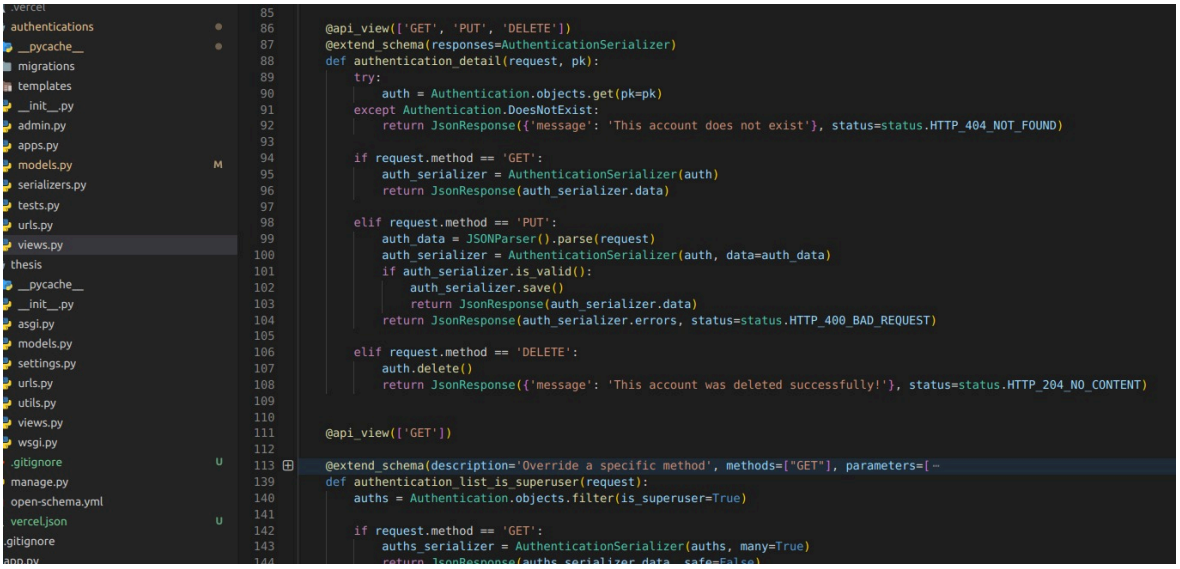

Fig 36. API Rest de Listas de usuarios

Sprint 15. Implementación total del web scraping

En el sprint 15 se implementa todo el sistema para realizar web scraping como la extracción de los perfiles de Instagram, extracción de la información de páginas web, la vista en el fronted y un historial. Los resultados que abarca este sprint son:

- Scraping de perfiles de Instagram.
- Scraping de páginas web.
- Historial de scraping.

Scraping de perfiles de Instagram

Se desarrolla la interfaz mediante Django donde el usuario ingresa el nombre del perfil y se extrae la información de este y la visualiza dentro de la misma interfaz como se muestra en la Fig 37.

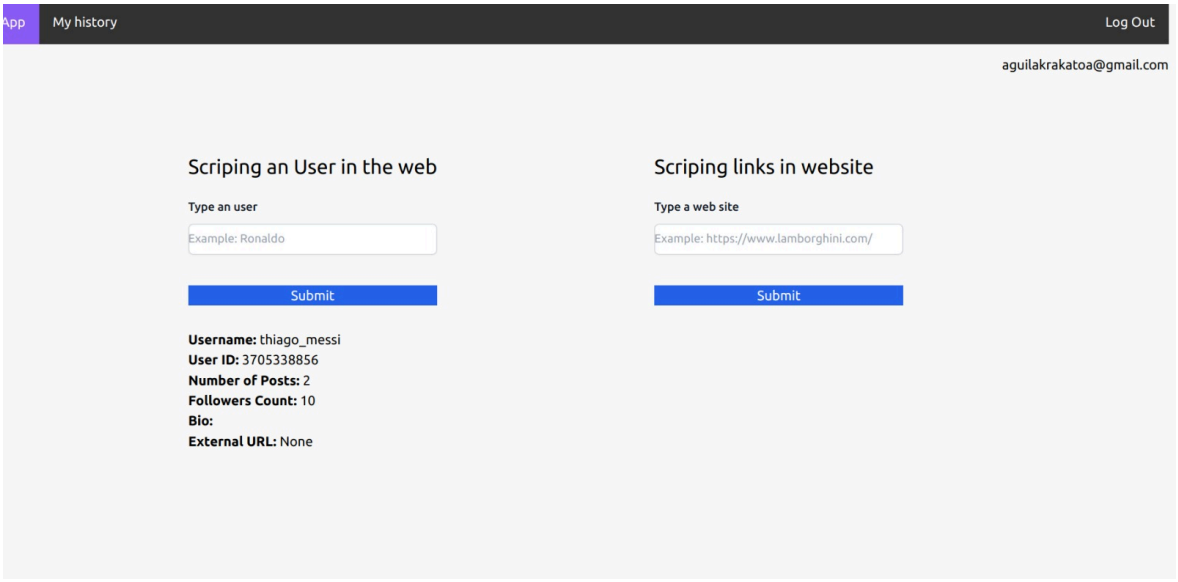

Fig 37. Scraping perfil de usuario

Scraping de páginas web

Se desarrolla la interfaz mediante Django donde el usuario ingresa el enlace de la página web, se extrae toda la información y se visualiza dentro de la interfaz como se muestra en la Fig 38.

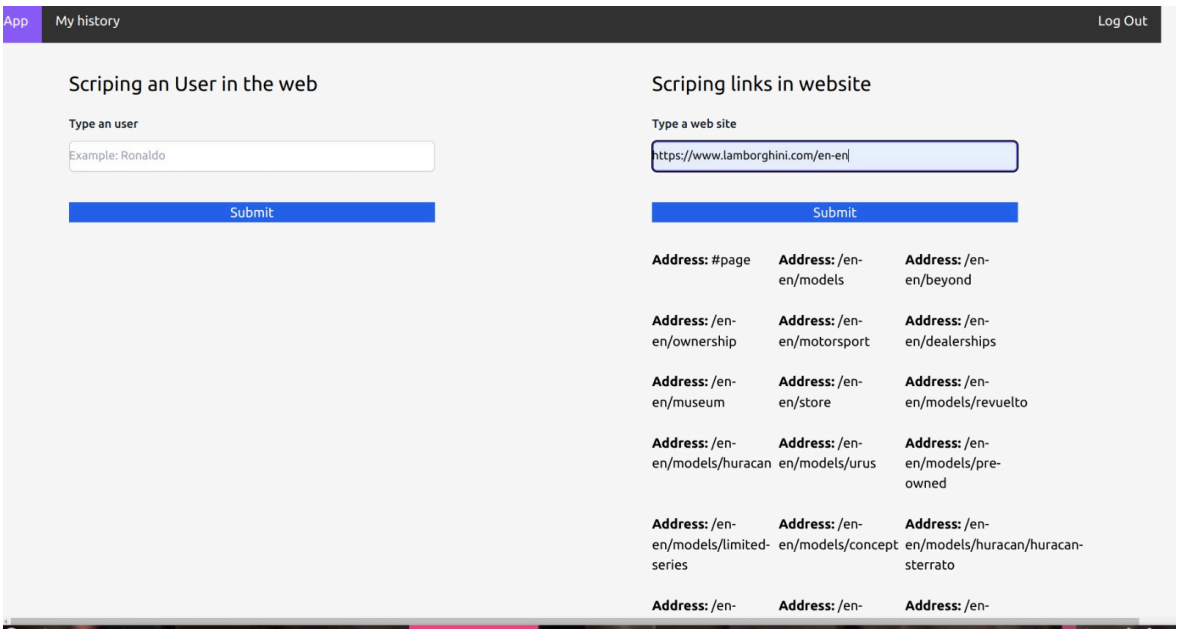

Fig 39. Scriping página web

Historial de web scraping

Se desarrolla la vista del scraping mediante el código y se lo visualiza mediante una pequeña interfaz como se muestra en las Fig 39 y Fig 40.

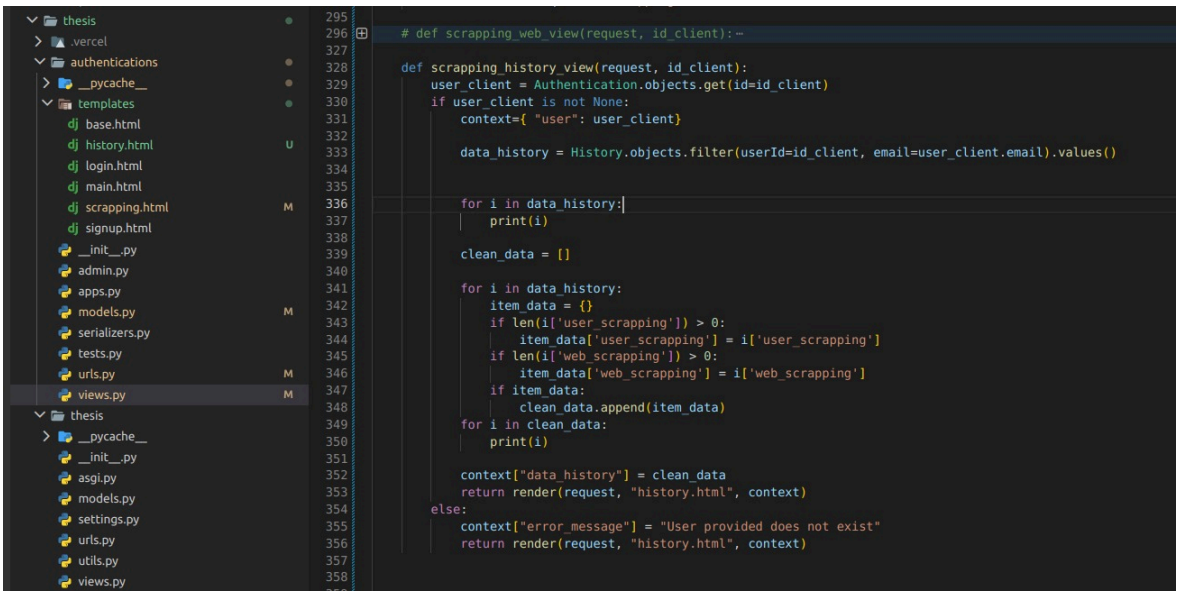

Fig 39. Vista de frontend del historial

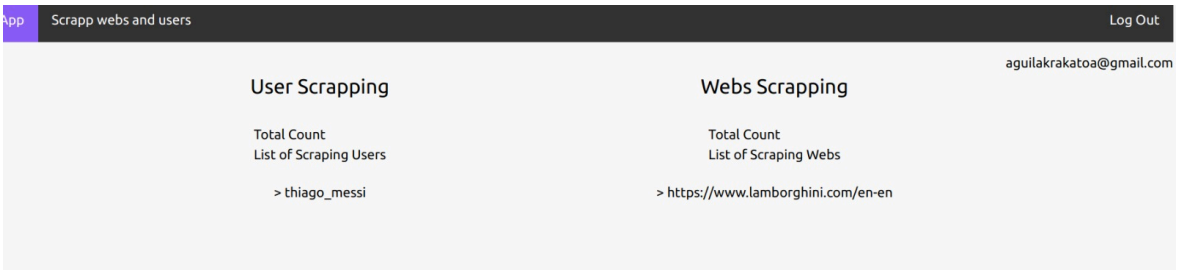

Fig 40. Historial del scraping

Sprint 16. Almacenamiento de los datos del historial en la base de datos

El sprint 16 se enfoca en almacenar los datos extraídos del historial al hacer web scraping en la base de datos MongoDB. Los resultados que abarca este sprint son:

- Base de datos del Historial.

Base de datos del historial

Los datos obtenidos al realizar tanto el web scraping de Instagram como de las páginas web se almacenan en la base de datos de MongoDB como se visualiza en la Fig 41.

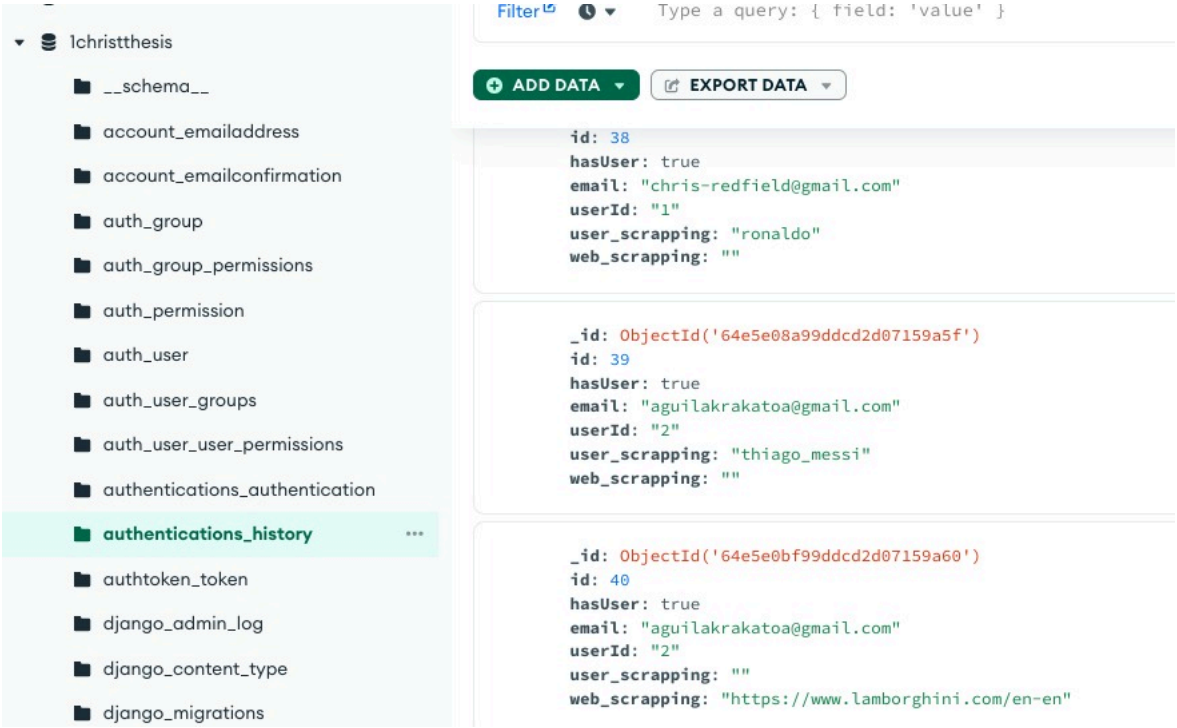

Fig 41. Base de datos del historial de web scraping

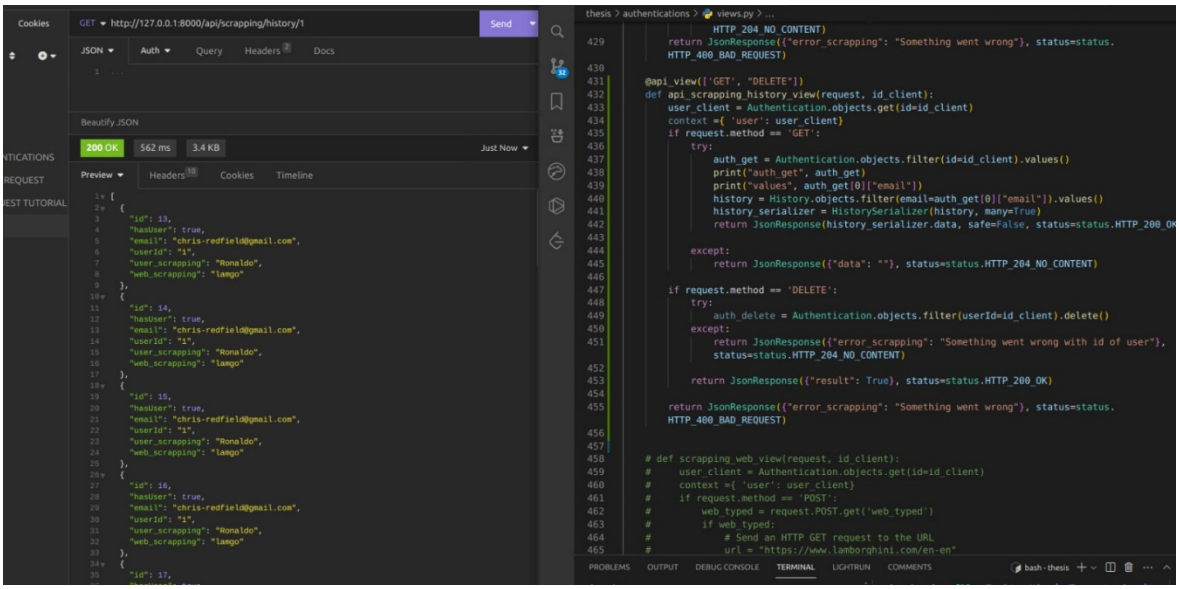

Fig 42. API Rest del historial

Sprint 17. Eliminación de la API Rest y base de datos

El sprint 17 se enfoca en probar la eliminación de los componentes tanto la API Rest y verificamos la base de datos para probar su funcionamiento. Los resultados de este sprint son:

- Eliminación de la API Rest.

Eliminación de la API Rest

Se procede a realizar la eliminación de la API Rest y luego se verifica el funcionamiento de la base de datos como se muestra en las Fig 43, Fig 44 y Fig 45.

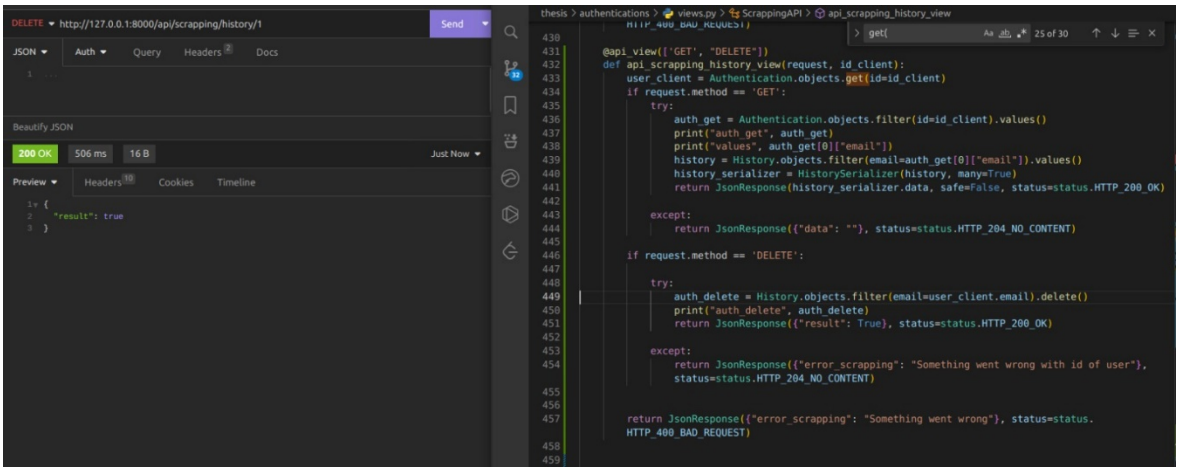

Fig 43. Eliminando la API Rest

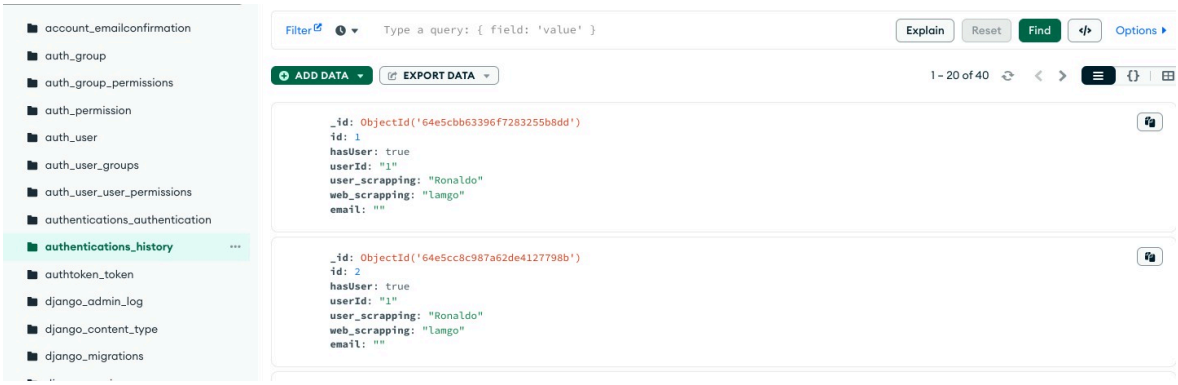

Fig 44. Base de datos antes de eliminar

Fig 45. Base de datos después de eliminar

4 CONCLUSIONES

- Se ha encontrado algunas limitaciones al utilizar la biblioteca instaloader ya que al realizar muchas peticiones la biblioteca hacer redireccionamientos, una manera de no saturarse es limitándose al margen de cuarenta peticiones.
- El funcionamiento del proyecto de acuerdo con los requerimientos establecidos ha sido exitoso, cumpliendo con las funciones principales tanto como el registro y login de los usuarios y le herramienta para realizar el web scraping en las diferentes páginas web.
- La metodología utilizada que en este caso fue RAD (Rapid application developer) fue indispensable para el desarrollo ya que al realizar las pruebas de manera constante según lo que se fue desarrollando permitió que el resultado final sea un software de calidad que solvente todos los requerimientos de una manera práctica y muy rápida.
- La utilización de un framework como Django fue acertada ya que permitió que la app funcione de la mejor manera ya que todos los scripts se realizaron con el lenguaje de programación Python.
- El lenguaje de programación Python apoyó mucho al desarrollo rápido del sistema web ya que con este lenguaje es más fácil realizar webscraping debido a su practicidad y librerías que permiten agilitar mucho más este proceso de extracción de datos.

5 RECOMENDACIONES

- Antes de empezar con el desarrollo es importante revisar las limitaciones que pueden tener algunas librerías y comprobar si los requerimientos necesitan de muchas solicitudes de usuario
- Es recomendable utilizar la metodología RAD para aplicaciones que no requieran enfocarse tanto en la planificación y más en la calidad del software ya que con las pruebas rápidas constantes que se van realizando conforme se van desarrollando cada una de las funcionalidades permite que se ahorre tiempo y los errores se corrijan apenas van apareciendo.
- Es importante saber que el hacer web scraping en algunas ocasiones puede resultar ilegal por las políticas de seguridad que manejan muchos creadores de contenido que se encuentra en la web, por eso es muy importante saber qué tipo de datos se pueden extraer y si estos mismo tengan alguna seguridad anti scrapeo.
- Cuando se realiza web scraping es importante manipular los datos de manera ordenada y estructurada porque de lo contrario la información obtenida no será de mucha ayuda complicando el entendimiento de esta.

6 REFERENCIAS BIBLIOGRÁFICAS

[1] K. T. l. d. reservados, «¿Qué Es el Web Scraping? Cómo Extraer Legalmente el Contenido de la Web,» [En línea]. Available: https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/. [Último acceso: 18 Agosto 2023].

[2] K. I. t. l. d. reservados, «¿Qué Es el Web Scraping? Cómo Extraer Legalmente el Contenido de la Web,» [En línea]. Available: https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/.

[3] M. Martí, «¿Qué es el Web scraping? Introducción y herramientas,» [En línea]. Available: https://sitelabs.es/web-scraping-introduccion-y-herramientas/. [Último acceso: 18 Agosto 2023].

[4] S. (. a. L. C. PLANETA DEAGOSTINI FORMACIÓ I UNIVERSITATS ANDORRA, «Metodologías de desarrollo de software,» [En línea]. Available: https://www.universitatcarlemany.com/actualidad/blog/metodologias-de-desarrollo-de-software/. [Último acceso: 18 Agosto 2023].

[5] I. Cloud, «¿Qué es el web scraping?,» [En línea]. Available: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-web-scraping/. [Último acceso: 18 Agosto 2023].

[6] T. l. d. r. Crea System, «¿Qué Es Un Sistema Web?,» [En línea]. Available: https://www.creasystem.net/posts/que-es-un-sistema-web. [Último acceso: 18 Agosto 2023].

[7] T. l. d. reservados, «Santander Becas,» [En línea]. Available: https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html. [Último acceso: 18 Agosto 2023].

[8] T. l. d. r. Kissflow, «What is Rapid Application Development (RAD)?,» [En línea]. Available: https://kissflow.com/application-development/rad/rapid-application-development/#:~:text=17.08.2023-,What%20is%20Rapid%20Application%20Development%20(RAD)%3F,a%20prototype%2C%20rather%20than%20planning.. [Último acceso: 18 Agosto 2023].

[9] K. Beck, «Extreme Programming Explained,» Addison Wesley, 2000, pp. 3-7.

[10] V. Solutions, «Qué es la recopilación de requisitos: definición y herramientas,» [En línea]. Available: https://visuresolutions.com/es/blog/requirements-gathering/. [Último acceso: 19 Agosto 2023].

[11] O. S. t. l. d. reservados, «Arquitectura de software,» [En línea]. Available: https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/. [Último acceso: 19 Agosto 2023].

[1 D. S. Foundation, «Django,» [En línea]. Available: https://www.djangoproject.com/. [Último
2] acceso: 19 Agosto 2023].

[1 T. N. T. School., «¿Qué es Django y para qué se utiliza?,» [En línea]. Available:
3] https://www.tokioschool.com/noticias/que-es-django/. [Último acceso: 18 Agosto 2023].

[1 C. S.L., «Qué es MVC,» [En línea]. Available: https://desarrolloweb.com/articulos/que-es-
4] mvc.html. [Último acceso: 18 Agosto 2023].

[1 R. D. Hernandez, «El patrón modelo-vista-controlador: Arquitectura y frameworks
5] explicados,» [En línea]. Available: https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/. [Último acceso: 18 Agosto 2023].

[1 I. o. s. f. T. l. d. r. Amazon Web Services, «¿Qué es Python?,» [En línea]. Available:
6] https://aws.amazon.com/es/what-is/python/. [Último acceso: 18 Agosto 2023].

[1 MongoDB, «¿Qué es MongoDB?,» [En línea]. Available: https://www.mongodb.com/es/what-
7] is-mongodb. [Último acceso: 19 Agosto 2023].

[1 L. Richardson., «Beautiful Soup,» [En línea]. Available:
8] https://www.crummy.com/software/BeautifulSoup/. [Último acceso: 19 Agosto 2023].

[1 A. Caballero, «Descargar Imágenes y Videos desde un Perfil Público de Instagram utilizando
9] Instaloader,» [En línea]. Available:
Descargar_Imagenes_y_Videos_desde_un_Perfil_Publico_de_Instagram_utilizando_Instaloader. [Último acceso: 15 Agosto 2023].

[2 Y. Fernández, «Qué es Github y qué es lo que le ofrece a los desarrolladores,» xataka, 30
0] Octubre 2019. [En línea]. Available: https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores. [Último acceso: 24 Mayo 2022].

[2 Platzi, «Curso de Automatización de Test con Playwright,» [En línea]. Available:
1] https://platzi.com/cursos/playwright/#:~:text=Playwright%20es%20una%20herramienta%20para,cualquier%20lenguaje%20de%20programaci%C3%B3n%20moderno.. [Último acceso: 20 Agosto 2023].

[2 «Create API Documentation with Postman,» API PLATFORM, [En línea]. Available:
2] https://learning.postman.com/docs/getting-started/introduction/. [Último acceso: 24 Mayo 2022].

7 ANEXOS

ANEXO I. Certificado turnitin

ANEXO I. Manual técnico

ANEXO II. Manual de usuario

ANEXO III. Manual de instalación

ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 28 de agosto de 2023

De mi consideración:

Yo, Juan Pablo Zaldumbide Proaño, en calidad de Director del Trabajo de Integración Curricular titulado APLICATIVO WEB DE EXTRACCIÓN DE DATOS asociado al DESARROLLO DE APLICACIÓN WEB PARA EXTRACCIÓN DE DATOS CON WEB SCRAPING Y HERRAMIENTA DE ANALÍTICA DE DATOS elaborado por el estudiante SOLEDISPA MENDOZA CHRISTIAN GABRIEL de la carrera en TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE , certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 10%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

Firmado electrónicamente por:
JUAN PABLO ZALDUMBIDE PROANO

Juan Pablo Zaldumbide Proaño
Docente Ocasional a Tiempo Completo
ESFOT

ANEXO II Manual Técnico

A continuación, se describen con mayor entendimiento los componentes faltantes en el informe técnico del proyecto

1. Recopilación de requerimientos

TABLA II. Recopilación de requerimientos

RECOPILACION DE REQUERMIENTOS		
TIPO DE SISTEMA	**ID - RR**	**ENUNCIADO DEL ITEM**
Web para la extracción de datos con web scraping	**RR001**	Como usuario necesito iniciar sesión.
	RR002	Como usuario necesito acceder al sistema web.
	RR003	Como usuario necesito actualizar mi perfil en la aplicación.
	RR004	Como usuario necesito poder registrarme si no tengo cuenta registrada todavía.
	RR005	Como usuario necesito poder ingresar la página web de la que quiero información.
	RR006	Como usuario necesito ver información sobre lo solicitado.
	RR007	Como usuario necesito poder acceder al a base de datos donde se almacena la información.

2. HISTORIAS DE USUARIO

En las tablas consiguientes de la **TABLA I** hasta la Error! Reference source not found. se muestran la lista de historias de usuario completas del proyecto.

TABLA III Historia de Usuario HU001

HISTORIA DE USUARIO	
Identificador (ID): HU001	**Usuario:** Administrador, cliente.

Nombre Historia: Iniciar sesión	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Alta
Iteración Asignada: 3	
Responsable: Christian Soledispa	
Descripción: El administrador y cliente deberán permitir la funcionalidad de inicio de sesión en el *sistema web*, para que los usuarios puedan iniciar sesión en el sistema.	
Observación: El inicio de sesión deberá contener los siguientes campos: email, contraseña.	

TABLA IV Historia de Usuario HU002

HISTORIA DE USUARIO	
Identificador (ID): HU002	**Usuario:** Administrador.
Nombre Historia: Mostrar usuarios	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Alta
Iteración Asignada: 2	
Responsable: Christian Soledispa	
Descripción: El administrador deberá tener la funcionalidad de visualizar a los potenciales usuarios finales desde el *backend*.	
Observación: Los potenciales usuarios deben mostrarse por sus campos: nombre, email.	

TABLA V Historia de Usuario HU003

HISTORIA DE USUARIO	
Identificador (ID): HU003	**Usuario:** Cliente

Nombre Historia: Editar perfil	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Iteración Asignada: 3	
Responsable: Christian Soledispa	
Descripción: El usuario como cliente debe tener la opción de actualizar su perfil.	
Observación: Las actualizaciones del perfil del usuario deben ser realizadas en cualquier momento y sin restricción por los mismos dueños del perfil.	

TABLA VI Historia de Usuario HU004

HISTORIA DE USUARIO	
Identificador (ID): HU004	**Usuario:** Administrador
Nombre Historia: Eliminar usuarios	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Alta
Iteración Asignada: 2	
Responsable: Christian Soledispa	
Descripción: El administrador deberá tener la funcionalidad de eliminar a los usuarios registrados en el *backend*, dependiendo de cual sea el caso y la razón de ser eliminado.	
Observación: El administrador debe contar con el privilegio de poder eliminar a los usuarios mal intencionados que quieran dañar el sistema.	

TABLA II Historia de Usuario HU005

HISTORIA DE USUARIO	
Identificador (ID): HU005	**Usuario:** Cliente
Nombre Historia: Elegir la página web	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Alta

Iteración Asignada: 2
Responsable: Christian Soledispa
Descripción: El Cliente podrá elegir antes de ingresar la información a buscar, en cual de ellas quiere realizar la búsqueda.
Observación: Las búsquedas se podrán realizar en las redes sociales más utilizadas como Facebook, Twitter, Tik Tok, Instagram.

TABLA III Historia de Usuario HU006

HISTORIA DE USUARIO	
Identificador (ID): HU006	**Usuario:** Administrador.
Nombre Historia: Visualizar la información extraída.	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Media
Iteración Asignada: 2	
Responsable: Christian Soledispa	
Descripción: El administrador podrá visualizar por medio del backend la información solicitada por el cliente.	
Observación: La información se mostrará en una interfaz sencilla ya que la misma será proporcionada y mostrada mediante el frontend.	

TABLA IX Historia de Usuario HU007

HISTORIA DE USUARIO	
Identificador (ID): HU007	**Usuario:** Administrador y cliente
Nombre Historia: Manipular los datos	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Media
Iteración Asignada: 2	

Responsable: Christian Soledispa
Descripción: El administrador podrá manipular y realizar una estructura de los datos extraídos para que sean proporcionados al cliente como tal.
Observación: Los datos serán almacenados en una base de datos NoSQL.

TABLA X Historia de Usuario HU008

HISTORIA DE USUARIO	
Identificador (ID): HU008	**Usuario:** Administrador y Cliente
Nombre Historia: Acceder a la base de datos	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Alta
Iteración Asignada: 2	
Responsable: Christian Soledispa	
Descripción: Tanto el Administrador como el cliente podrán acceder a la base de datos para estructurar los datos y puedan ser visualizados.	
Observación: Los datos podrán ser seleccionados para que únicamente sean visualizados los datos que se necesiten.	

ANEXO III Manual de usuario

El siguiente encale contiene el manual de usuario del sistema, como video en la plataforma youtube:

https://youtu.be/csEMXSEwu-w

ANEXO IV Manual de instalación

En el repositorio de GitHub se presenta el manual de cómo instalar y utilizar el sistema, describe el desarrollo completo del proyecto: https://github.com/ChristianSoledispa/Proyecto_de_titulacion.git