

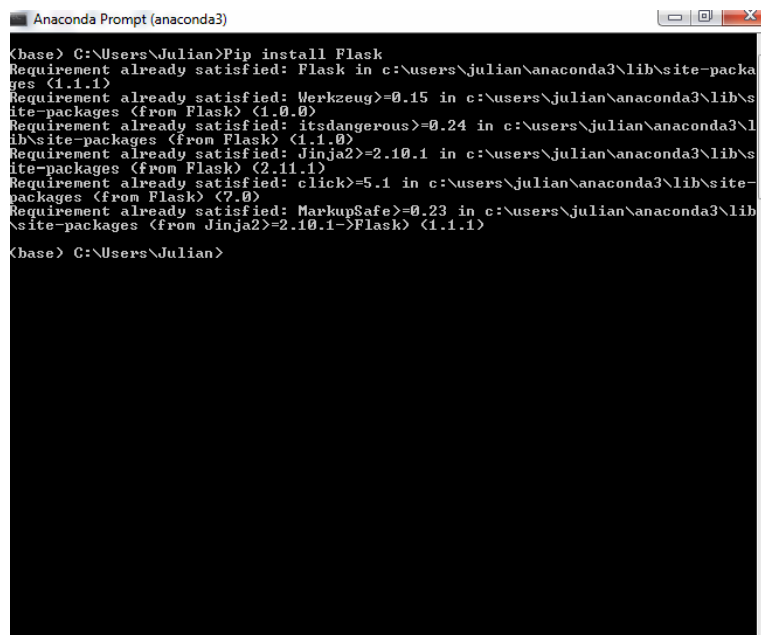
VISTAS Y FRONTEND

A) Para poder realizar la interfaz de la página, hacer las vista y el llamado al backend se usó un framework de Python de para poder crear aplicaciones.

Instrucciones para instalar el módulo Flask.

- 1) En el buscador de Windows buscar Prompt anaconda
- 2) Procederemos a escribir Pip Install Flask, esperamos unos segundos y listo se ha instalado el módulo Flask como se muestra en la imagen.

Esto en caso de usar Anaconda (si se usa otro editor de código como visual studio únicamente escribir en la terminal (Python -m pip install Flask).

A screenshot of the Anaconda Prompt terminal window. The title bar reads "Anaconda Prompt (anaconda3)". The terminal shows the command "C:\Users\Julian>Pip install Flask" being executed. Below the command, several lines of output confirm the installation, stating that various dependencies (Werkzeug, Jinja2, MarkupSafe, etc.) are already satisfied. The prompt then returns to "C:\Users\Julian>".

```

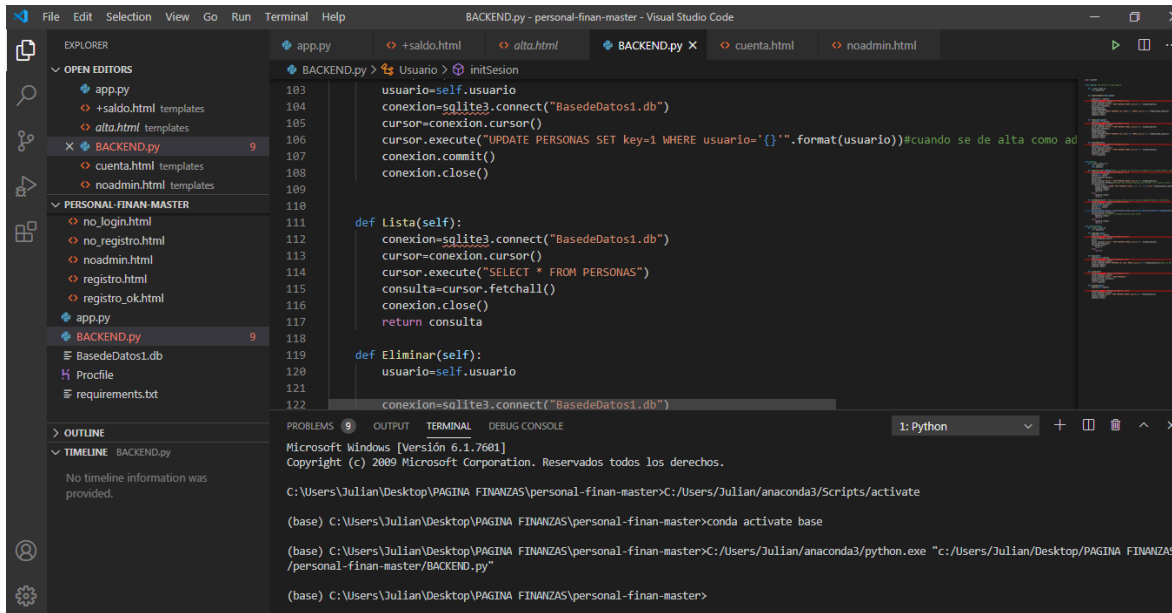
C:\Users\Julian>Pip install Flask
Requirement already satisfied: Flask in c:\users\julian\anaconda3\lib\site-packa
ges (1.1.1)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\julian\anaconda3\lib\s
ite-packages (from Flask) (1.0.0)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\julian\anaconda3\l
ib\site-packages (from Flask) (1.1.0)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\julian\anaconda3\lib\s
ite-packages (from Flask) (2.11.1)
Requirement already satisfied: click>=5.1 in c:\users\julian\anaconda3\lib\site-
packages (from Flask) (7.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\julian\anaconda3\lib
\site-packages (from Jinja2>=2.10.1->Flask) (1.1.1)
C:\Users\Julian>
```

En el archivo app.py se diseñó toda la interfaz para poder crear la página dependiendo de las necesidades del backend cada de sus clases se mandó llamar como una librería.

Por cada función del frontend para que no se perdiera la información que el usuario introdujera se hicieron paginas puentes por cada función, vinculándolos con lo diferentes templates creados en HTML, están son invocadas en el archivo app.py por medio de los métodos GET y POST que sirven respectivamente para enviar información visible y no visible.

B) PARA QUE SE MUESTRE LA RUTA DE LA PÁGINA GENERADA LO QUE SE DEBE DE HACER ES:

1. Verificar primero que la parte del Backend no haya ningún error y que se ejecute de manera correcta el programa.



The screenshot shows the Visual Studio Code interface with the 'BACKEND.py' file open. The file contains Python code for a backend application, including database connections and user management functions. The terminal at the bottom shows the command prompt output, indicating that the application is running successfully.

```
103 usuario=self.usuario
104 conexion=sqlite3.connect("BasedeDatos1.db")
105 cursor=conexion.cursor()
106 cursor.execute("UPDATE PERSONAS SET key=1 WHERE usuario={}""".format(usuario))#cuando se de alta como ad
107 conexion.commit()
108 conexion.close()
109
110
111 def Lista(self):
112     conexion=sqlite3.connect("BasedeDatos1.db")
113     cursor=conexion.cursor()
114     cursor.execute("SELECT * FROM PERSONAS")
115     consulta=cursor.fetchall()
116     conexion.close()
117     return consulta
118
119 def Eliminar(self):
120     usuario=self.usuario
121
122     conexion=sqlite3.connect("BasedeDatos1.db")
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: Python

Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

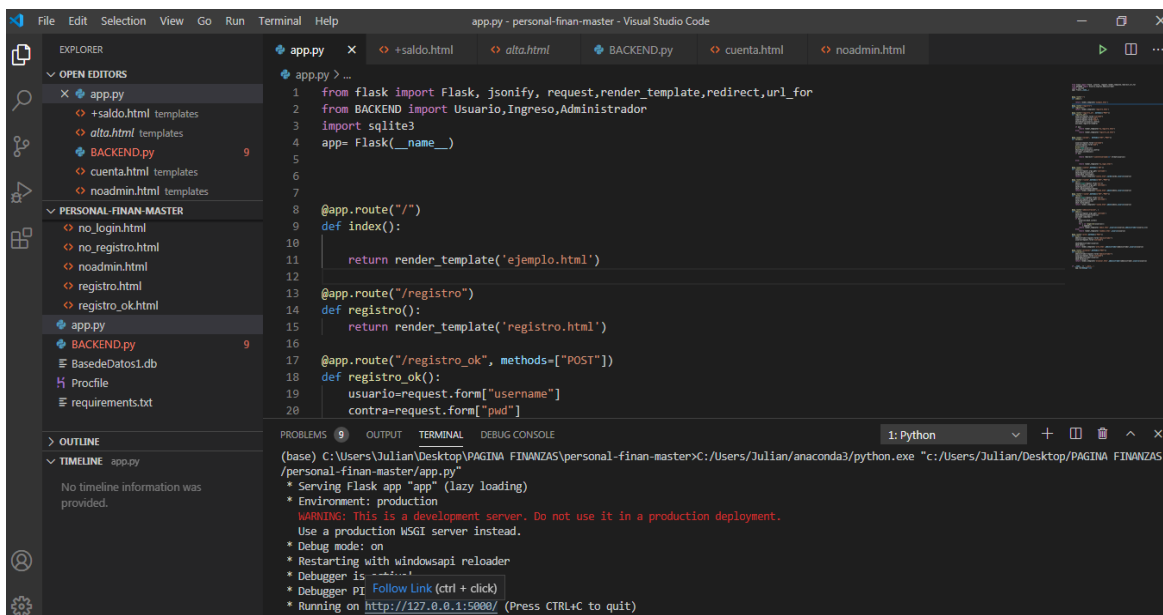
C:\Users\Julian\Desktop\PAGINA FINANZAS\personal-finan-master>C:/Users/Julian/anaconda3/Scripts/activate

(base) C:\Users\Julian\Desktop\PAGINA FINANZAS\personal-finan-master>conda activate base

(base) C:\Users\Julian\Desktop\PAGINA FINANZAS\personal-finan-master>C:/Users/Julian/anaconda3/python.exe "c:/Users/Julian/Desktop/PAGINA FINANZAS/personal-finan-master/BACKEND.py"

(base) C:\Users\Julian\Desktop\PAGINA FINANZAS\personal-finan-master>

2. Una vez ya teniendo instalado el modulo correr el archivo app.py y en la terminal se generara una dirección de enrutamiento presionar ctrl+click para poder ir a la ruta en el navegador.



The screenshot shows the Visual Studio Code interface with the 'app.py' file open. The file contains Python code for a Flask application, including imports, app initialization, and route definitions. The terminal at the bottom shows the command prompt output, indicating that the application is running successfully.

```
1 from flask import Flask, jsonify, request,render_template,redirect,url_for
2 from BACKEND import Usuario,Ingreso,Administrador
3 import sqlite3
4 app= Flask(__name__)
5
6
7
8 @app.route("/")
9 def index():
10     return render_template('ejemplo.html')
11
12
13 @app.route("/registro")
14 def registro():
15     return render_template('registro.html')
16
17 @app.route("/registro_ok", methods=["POST"])
18 def registro_ok():
19     usuario=request.form["username"]
20     contra=request.form["pwd"]
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: Python

(base) C:\Users\Julian\Desktop\PAGINA FINANZAS\personal-finan-master>C:/Users/Julian/anaconda3/python.exe "c:/Users/Julian/Desktop/PAGINA FINANZAS/personal-finan-master/app.py"

* Serving Flask app "app" (lazy loading)

* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

* Debug mode: on

* Restarting with windowsapi reloader

* Debugger is on

* Debugger pt Follow Link (ctrl + click)

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)