

FULLSTACK CON PYTHON - FLASK

jgalindos@sena.edu.co 2024









SOFTWARE RECOMENDADO



ANACONDA

MARIADB SERVER

POSTGRES SERVER

MONGO SERVER

SQLITE SERVER

REPOSITORIO GITHUB







ENTORNO ANACONDA



JOSE FERNANDO GA

ANACONDA - PYTHON

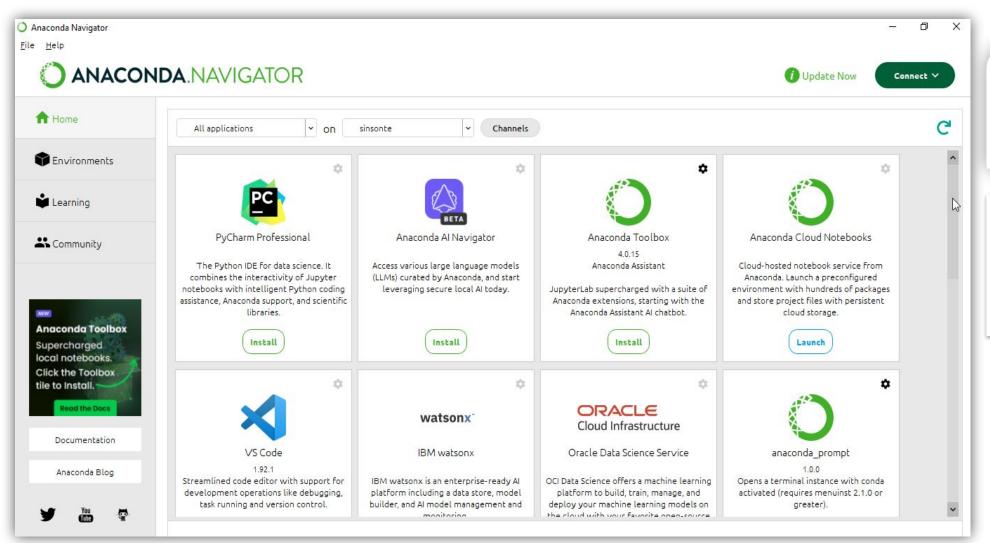


Anaconda es una distribución de software libre y de código abierto de los lenguajes Python y R,1 utilizada en ciencia de datos y aprendizaje automático (machine learning).2 Esto incluye procesamiento de grandes volúmenes de información, análisis predictivo y cómputos científicos. Está orientado a simplificar el despliegue y administración de los paquetes de software. (WIKIPEDIA)



ANACONDA - PYTHON Navigator



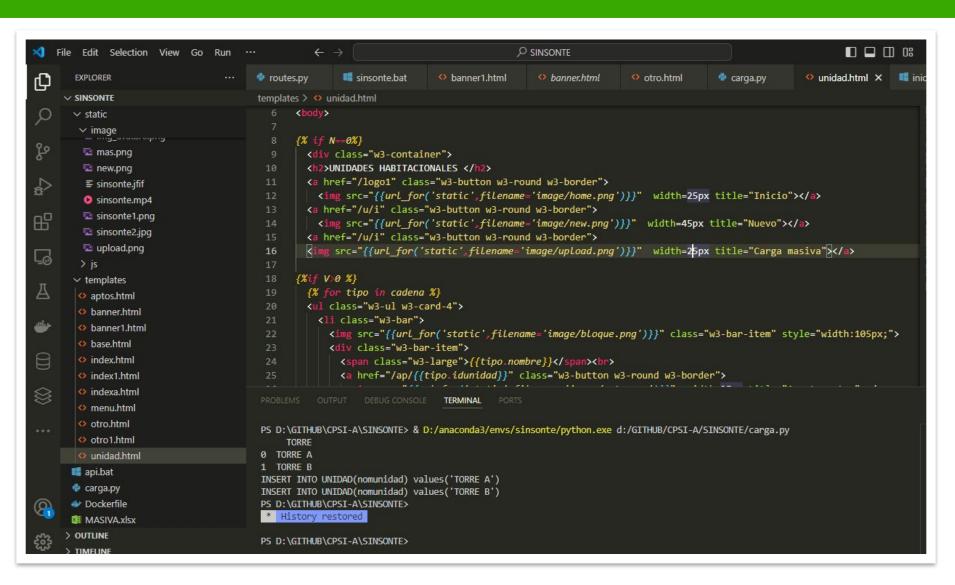






ANACONDA - PYTHON VisualCode









IOSE FERNANDO GALINDO SUARI

INFORMÁCION DE CONFIGURACION



```
Anaconda Powershell Prompt
(base) PS C:\Users\FEGASU> conda info
    active environment : base
   active env location : C:\Users\FEGASU\anaconda3
           shell level : 1
      user config file : C:\Users\FEGASU\.condarc
populated config files :
         conda version : 24.5.0
   conda-build version: 24.5.1
        python version: 3.12.4.final.0
                solver : libmamba (default)
      virtual packages : archspec=1=westmere
                           conda=24.5.0=0
                           win=0=0
      base environment : C:\Users\FEGASU\anaconda3 (writable)
     conda av data dir : C:\Users\FEGASU\anaconda3\etc\conda
 conda av metadata url : None
           channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/r/win-64
                          https://repo.anaconda.com/pkgs/r/noarch
                          https://repo.anaconda.com/pkgs/msys2/win-64
                          https://repo.anaconda.com/pkgs/msys2/noarch
         package cache : C:\Users\FEGASU\anaconda3\pkgs
                          C:\Users\FEGASU\.conda\pkgs
                          C:\Users\FEGASU\AppData\Local\conda\conda\pkgs
      envs directories : C:\Users\FEGASU\anaconda3\envs
                          C:\Users\FEGASU\.conda\envs
                          C:\Users\FEGASU\AppData\Local\conda\conda\envs
```





CREANDO UN ENTORNO VIRTUAL CON UNA VERSION DE PYTHON



```
Anaconda Powershell Prompt
(base) PS C:\Users\FEGASU> conda create --name sinsonte python=3.8
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done
## Package Plan ##
 environment location: C:\Users\FEGASU\anaconda3\envs\sinsonte
  added / updated specs:
    - python=3.8
The following packages will be downloaded:
    package
                                             build
    ca-certificates-2024.7.2
                                        haa95532 0
                                                            128 KB
    pip-24.0
                                    py38haa95532 0
                                                            2.8 MB
    python-3.8.19
                                        h1aa4202 0
                                                           18.9 MB
    setuptools-69.5.1
                                    py38haa95532_0
                                                           1003 KB
                                        h2eaa2aa 4
                                                             10 KB
    vc-14.2
                                        h43f2093 4
    vs2015 runtime-14.29.30133
                                                            1.1 MB
    wheel-0.43.0
                                    py38haa95532 0
                                                            137 KB
                                                           24.1 MB
                                            Total:
The following NEW packages will be INSTALLED:
                     pkgs/main/win-64::ca-certificates-2024.7.2-haa95532_0
  libffi
                     pkgs/main/win-64::libffi-3.4.4-hd77b12b_1
  openssl
                     pkgs/main/win-64::openssl-3.0.14-h827c3e9 0
                     pkgs/main/win-64::pip-24.0-py38haa95532_0
  pip
 python
                     pkgs/main/win-64::python-3.8.19-h1aa4202_0
  setuptools
                     pkgs/main/win-64::setuptools-69.5.1-py38haa95532 0
                     pkgs/main/win-64::sqlite-3.45.3-h2bbff1b_0
 sqlite
                     pkgs/main/win-64::vc-14.2-h2eaa2aa_4
pkgs/main/win-64::vs2015_runtime-14.29.30133-h43f2093_4
 vs2015 runtime
  wheel
                     pkgs/main/win-64::wheel-0.43.0-py38haa95532_0
Proceed ([y]/n)? 3
```



conda create --name sinsonte Python=3.8

conda activate sinsonte

conda deactivate sinsonte

Anaconda Powershell Prompt

(base) PS C:\Users\FEGASU> conda activate sinsonte (sinsonte) PS C:\Users\FEGASU>

Anaconda Powershell Prompt

(base) PS C:\Users\FEGASU> conda env remove --name sinsonte

Remove all packages in environment C:\Users\FEGASU\anaconda3\envs\sinsonte:

Everything found within the environment (C:\Users\FEGASU\anaconda3\envs\sinsonte),
ed. Do you wish to continue?

(y/[n])? y

(base) PS C:\Users\FEGASU>



INSTALANDO PAQUETES UN ENTORNO VIRTUAL CON UNA VERSION DE PYTHON



```
Anaconda Powershell Prompt
(sinsonte) PS C:\Users\FEGASU> pip install flask
Collecting flask
 Using cached flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from flask)
 Using cached werkzeug-3.0.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
 Using cached jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
 Using cached itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
 Using cached click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
 Using cached blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting importlib-metadata>=3.6.0 (from flask)
 Downloading importlib_metadata-8.2.0-py3-none-any.whl.metadata (4.7 kB)
Collecting colorama (from click>=8.1.3->flask)
 Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting zipp>=0.5 (from importlib-metadata>=3.6.0->flask)
 Downloading zipp-3.19.2-py3-none-any.whl.metadata (3.6 kB)
```



Anaconda Powershell Prompt	
(sinsonte) PS C:\Users\FEGASU> pip list	
Package	Version
blinker	1.8.2
click	8.1.7
colorama	0.4.6
Flask	3.0.3
<pre>importlib_metadata</pre>	8.2.0
itsdangerous	2.2.0
Jinja2	3.1.4
MarkupSafe	2.1.5
pip	24.0
setuptools	69.5.1
Werkzeug	3.0.3
wheel	0.43.0
zipp	3.19.2



ADICIONANDO UN CANAL DE INSTALACION DE APAQUETES EN UN ENTORNO VIRTUAL CON UNA VERSION DE PYTHON



```
Anaconda Powershell Prompt

(sinsonte) PS C:\Users\FEGASU> conda config --add channels conda-forge
(sinsonte) PS C:\Users\FEGASU>

Anaconda Powershell Prompt

(sinsonte) PS C:\Users\FEGASU> conda config --show channels channels:
   - conda-forge
   - defaults

(sinsonte) PS C:\Users\FEGASU>
```



```
Anaconda Powershell Prompt

(sinsonte) PS C:\Users\FEGASU> conda config --get channels
--add channels 'defaults' # lowest priority
--add channels 'conda-forge' # highest priority

(sinsonte) PS C:\Users\FEGASU> __
```



```
Anaconda Powershell Prompt
```

(sinsonte) PS C:\Users\FEGASU> conda config --remove channels conda-forge
(sinsonte) PS C:\Users\FEGASU>





PRACTICA APLICACIÓN FLASK



RECONOCIMIENTOS PREVIOS DE PYTHON







MANUAL DE PYTHON

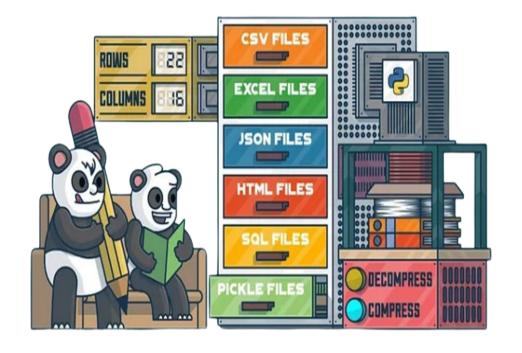
RECONOCIMIENTOS PREVIOS EN PYTHON



QUÉ ES FLASK?



Flask es un framework web ligero y flexible para Python. Es utilizado principalmente para el desarrollo de aplicaciones web y APIs (interfaces de programación de aplicaciones). Flask proporciona herramientas y bibliotecas que permiten a los desarrolladores construir aplicaciones web de manera rápida y eficiente.





CARACTERISTICAS DE FLASK



- 1. Minimalismo: Flask está diseñado para ser simple y fácil de entender, permitiendo a los desarrolladores comenzar rápidamente con proyectos web sin una curva de aprendizaje empinada.
- 2. Extensibilidad: Aunque Flask es ligero, es altamente extensible. Permite la integración con una amplia gama de extensiones y bibliotecas para añadir funcionalidades adicionales según sea necesario.

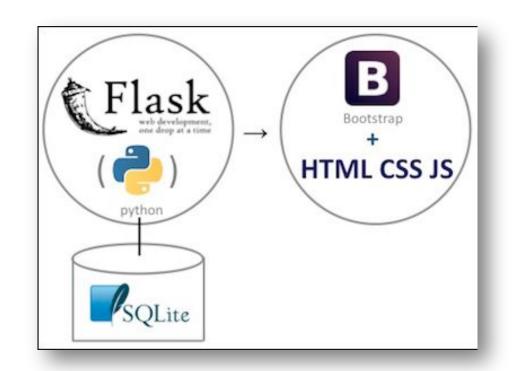




CARACTERISTICAS DE FLASK



- 3. Flexibilidad: Flask no impone una estructura de directorios o una forma específica de desarrollar aplicaciones, lo que brinda a los desarrolladores la libertad de organizar sus proyectos como deseen.
- 4. Jinja2: Flask utiliza el motor de plantillas Jinja2, lo que facilita la creación de vistas HTML dinámicas.
- 5. Servidor de desarrollo integrado: Flask incluye un servidor de desarrollo que facilita la creación y prueba de aplicaciones web localmente.



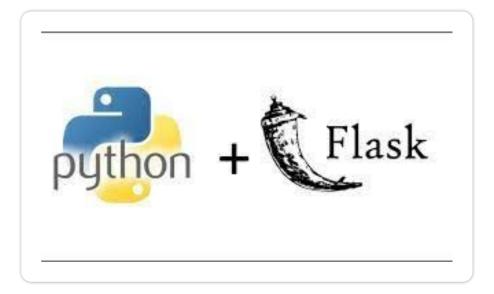


CARACTERISTICAS DE FLASK



6. Compatibilidad con WSGI: Flask es compatible con el estándar WSGI (Web Server Gateway Interface), lo que significa que puede ser desplegado en una amplia gama de servidores web.

Flask es una excelente opción para aquellos desarrolladores que prefieren un marco minimalista pero poderoso para la creación de aplicaciones web en Python.

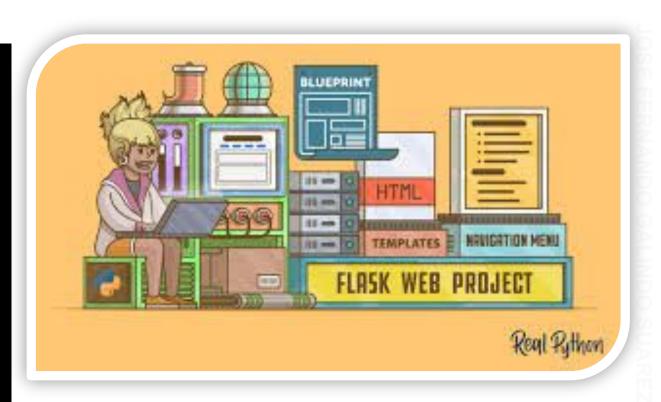




INSTALACION DEL ENTORNO VIRTUAL



INSTALANDO EL ENTORNO VIRTUAL pip install virtualenv CREANDO EL ENTORNO VIRTUAL python -m venv proyectos ACTIVANDO EL ENTORNO VIRTUAL **cd Scripts** activate cd.. DESACTIVANDO EL ENTORNO VIRTUAL **cd Scripts** deactivate







Crear ambiente virtual PRUEBA

Anaconda Prompt - conda create --name sinsonte Python=3.8 - conda create --name PRUEBA Python=3.8

```
(base) C:\Users\Administrador>md d:\PRUEBA

(base) C:\Users\Administrador>cd d:\PRUEBA

(base) C:\Users\Administrador>d:

(base) d:\PRUEBA>conda create --name PRUEBA Python=3.8

WARNING: A directory already exists at the target location 'd:\anaconda3\envs\PRUEBA'
but it is not a conda environment.
Continue creating environment (y/[n])? y
```

Activar ambiente virtual PRUEBA

Anaconda Prompt - conda create -- name sinsonte Python=3.8 (base) d:\PRUEBA>conda activate PRUEBA (PRUEBA) d:\PRUEBA>

Crear archivo "requi.txt

```
Anaconda Prompt - conda create --name sinsonte Python=3.8

(base) d:\PRUEBA>copy con requi.txt
flask
flask_cors
request
^Z
1 archivo(s) copiado(s).
```

Anaconda Prompt - conda create --name sinsonte Python=3.8 - pip install flask

```
(PRUEBA) d:\PRUEBA>pip install flask
Collecting flask

(PRUEBA) d:\PRUEBA>pip install flask_cors
Collecting flask cors

(PRUEBA) d:\PRUEBA>pip install requests

Collecting requests
```



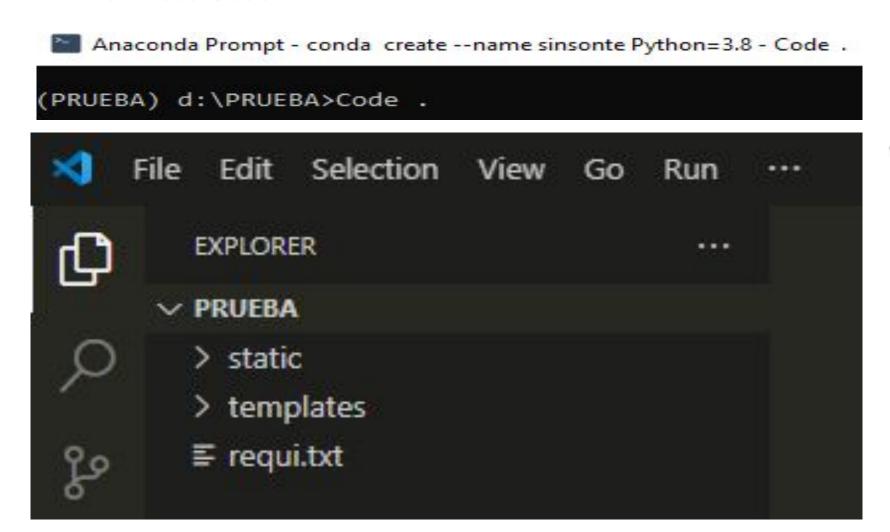
Construir esta estructura de archivos



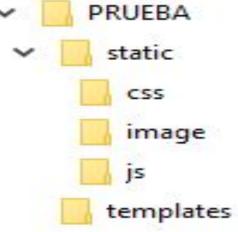




Activar VisualCode



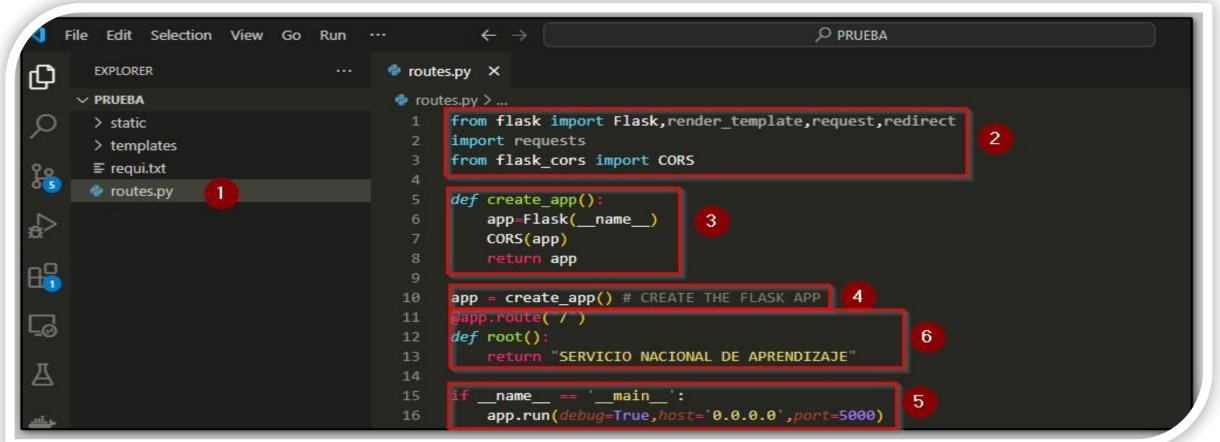








Crear el archivo "routes.py"

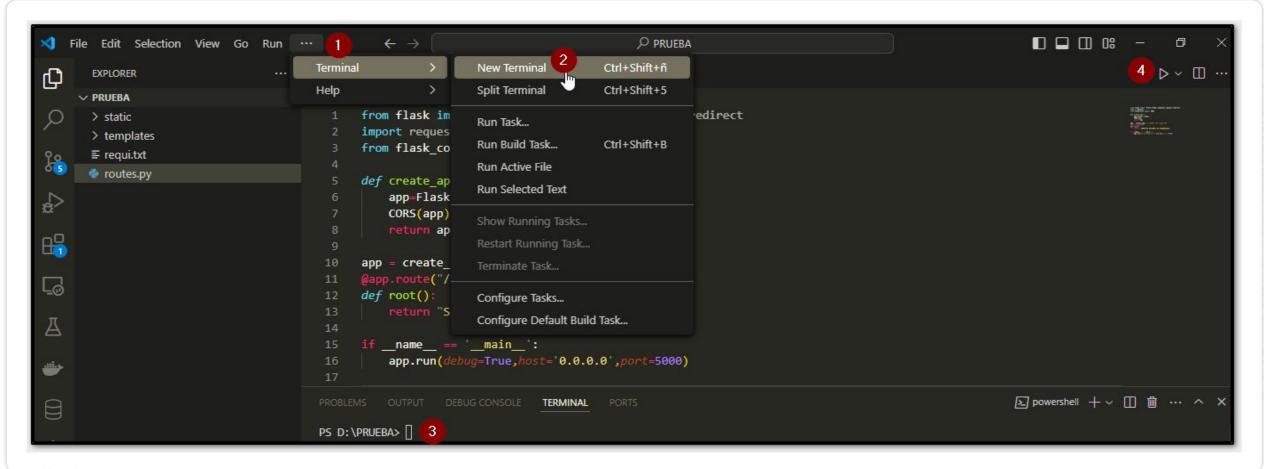








Crear una terminal

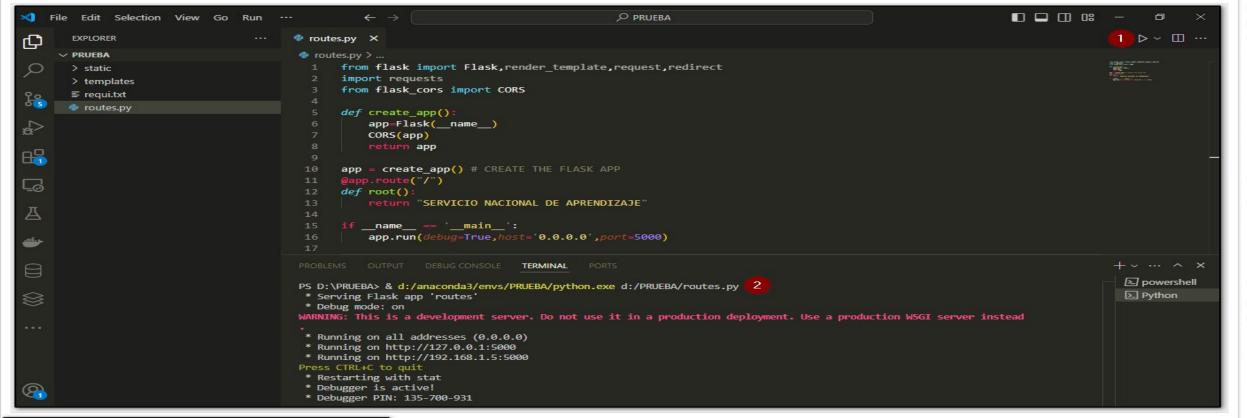








Ejecutar la aplicación







pythonanywhere by ANACONDA.



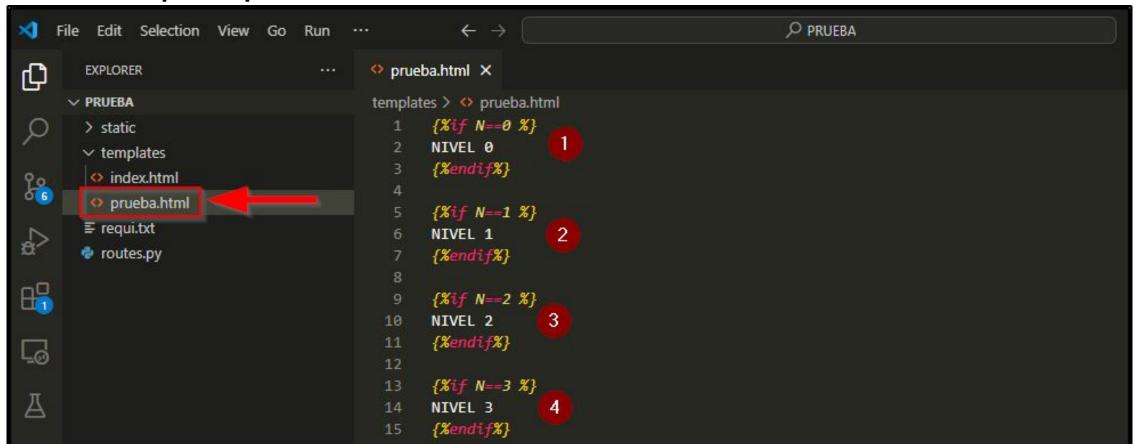


JINJA2 Y MULTINIVEL





Crear el template "prueba.html

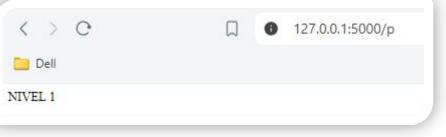








Crear un ENDPOINT "/p" en "routes.py", llamando el template "prueba.html" y pasando el valor M



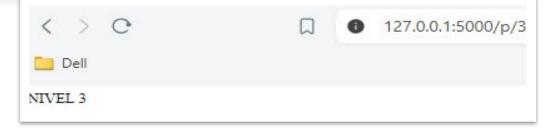






Se modifica el ENDPOINT "/p" en "routes.py", llamando el template "prueba.html" y pasando el valor M









Se crea el ENDPOINT "/p2" en "routes.py", llamando el template "prueba.html" y pasando el valor M y la lista cadena

```
@app.route("/p2") 1

def IrNivelp2():
    M=int(2) 2

3 cadena=("Juan", "Maria", "Carlos") 4 5 6
    return render_template("prueba.html", N=M, cadena=cadena)
```

```
O Dell

NIVEL 2
Juan
Maria
Carlos
```







Se modifica el ENDPOINT "/p2" en "routes.py", llamando el template "prueba.html" y pasando el valor M y la lista cadena

Se modifica "prueba.html"

```
{%if N==2 %}
NIVEL 2<br>
{% for a in cadena %}
{{a}}<br>
{% endfor %}
{%endif%}
```

```
Carlos
```





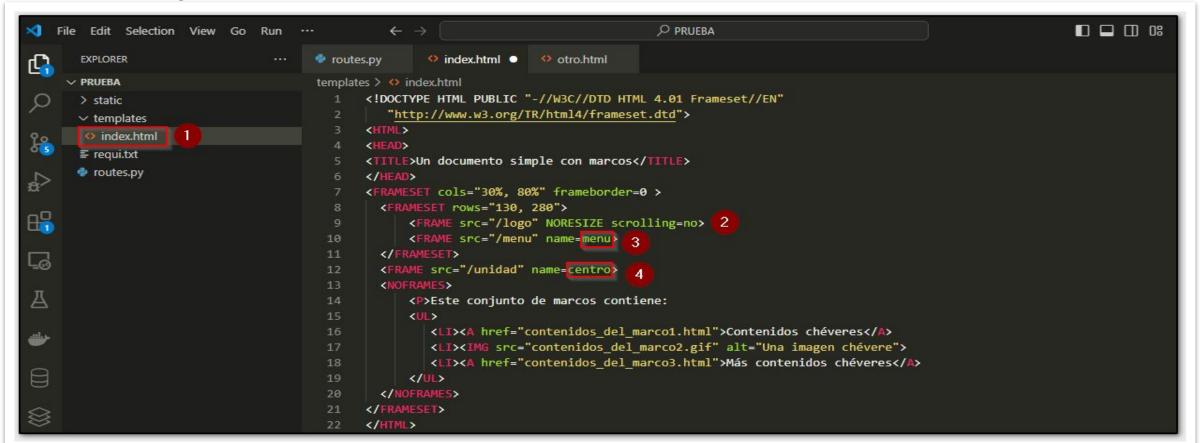


DISEÑO WEB TEMPLATES





Crear el template "index.html









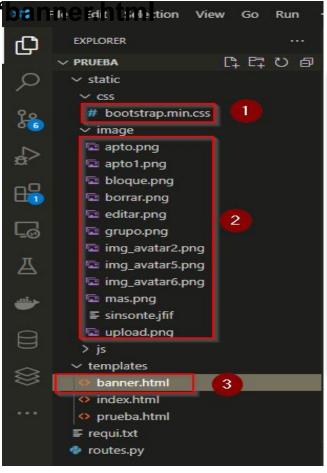








Descargue, descomprima y coloque en static

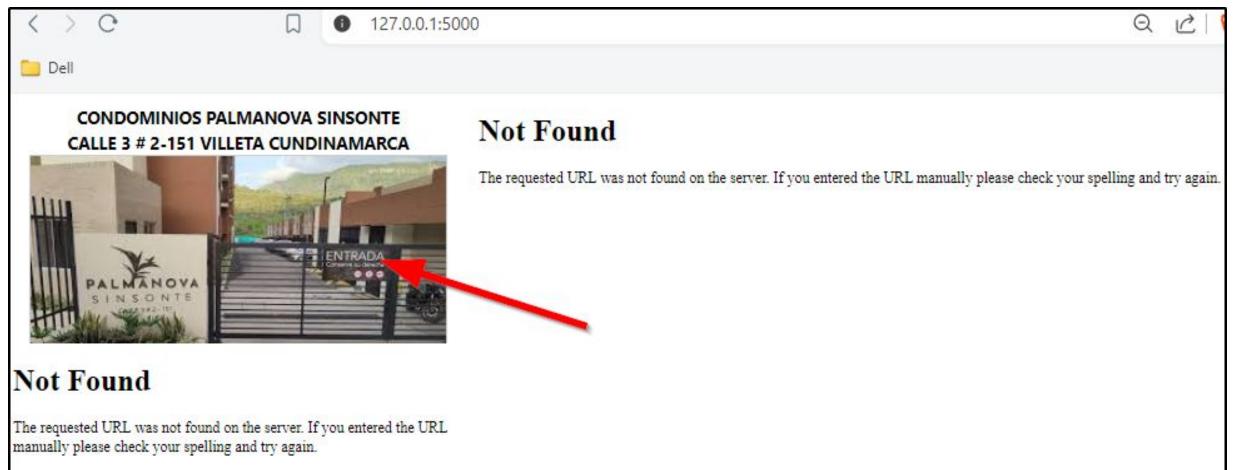


Crear el template "banner.html

Crear el ENDPOINT "/logo"

```
@app.route("/logo")
def inilogocio():
    return render_template("banner.html")
```



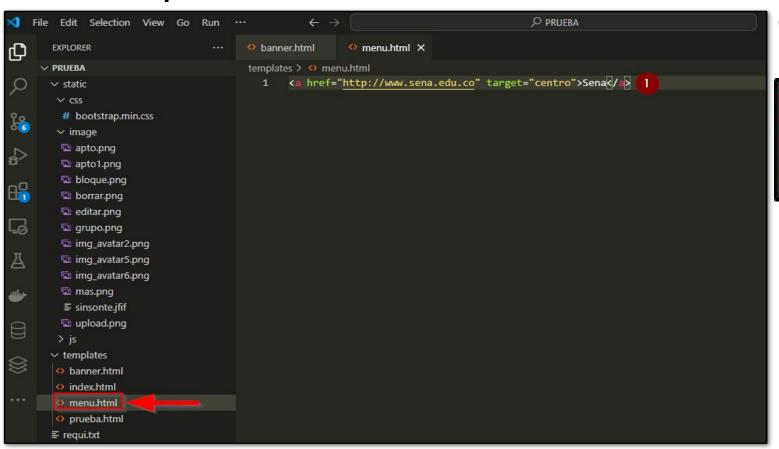








Crear el template "menú.html"



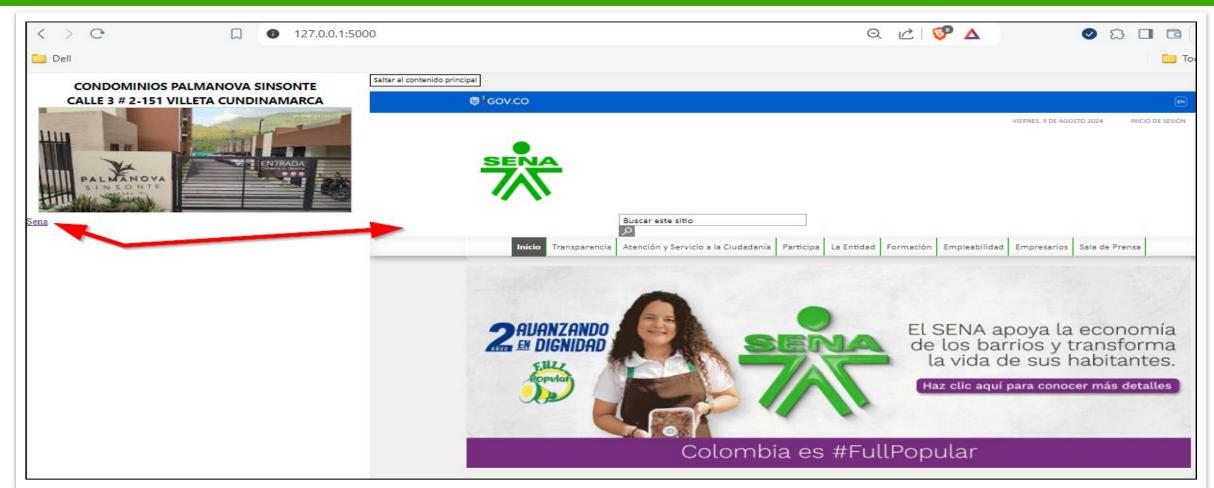
Crear el ENDPOINT "/menú", en "routes.py"

```
@app.route("/menu")
def menup():
    return render_template("menu.html")
```















Modificar el template "menú.html"

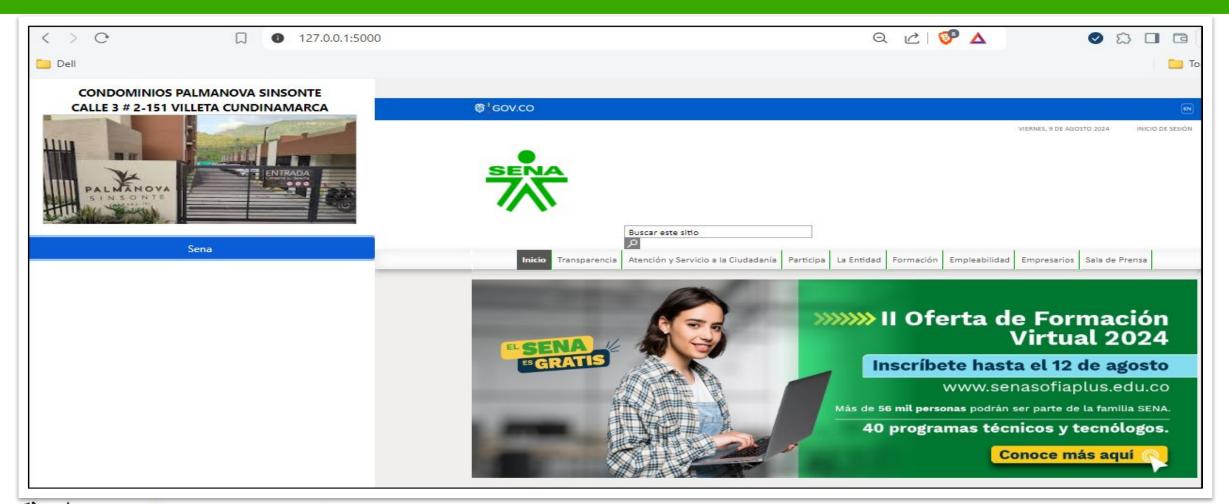
```
menu.html X
templates > 0 menu.html
       <link href="{{url for('static', filename='css/bootstrap.min.css')}}" rel="stylesheet">
       <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"</pre>
       integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>
       (br)
       <a class="btn btn-primary" href="http://www.sena.edu.co" target="centro" style="width:100%;">SENA</a>
```





APLICACIÓN FLASK











MODULARIZANDO CON BLUEPRINT



BLUEPRINT PARA FLASK



En Flask, "Blueprint" (o "Blueprints" en plural) es una característica que permite organizar y estructurar una aplicación web en módulos o componentes más pequeños y reutilizables. Un Blueprint en Flask funciona de manera similar a una aplicación Flask completa, pero se puede registrar en una aplicación principal en lugar de ser independiente.





CARACTERISTICAS DE BLUEPRINT



- •Organización modular: Con Blueprint, puedes dividir tu aplicación en módulos más pequeños y separar funcionalidades específicas en diferentes archivos. Esto facilita la organización y la gestión de aplicaciones web más grandes.
- •Reutilización de código: Puedes definir rutas, controladores (views), y otros aspectos de una aplicación dentro de un Blueprint y luego registrar ese Blueprint en múltiples aplicaciones Flask. Esto facilita la reutilización del código en diferentes partes de tu proyecto o en proyectos diferentes.





CARACTERISTICAS DE BLUEPRINT



Independencia y flexibilidad: Aunque un Blueprint se define como una unidad modular, sigue siendo independiente y puede tener su propia configuración, plantillas, archivos estáticos, etc. Esto proporciona flexibilidad en cómo estructuras y despliegas diferentes partes de tu aplicación.





DEFINIR UN BLUEPRINT



```
CREAR UN DIRECTORIO LLAMADO "PERROS"
routes.py
from flask import Blueprint, render template
  Crear un Blueprint
perros = Blueprint('perros', __name_
url prefix='/perros')
 Definir rutas y vistas dentro del
Blueprint
@perros.route('/1')
def login():
    return 'Manada de perros'
```





REGISTRAR EL BLUEPRINT



```
app.py
```

```
from flask import Flask
from flask cors import CORS
from PERROS.routes import perros
 importar el Blueprint
 Crear la aplicación Flask
app = Flask( name
 Registrar el Blueprint en la aplicación
app.register_blueprint(perros)
              main
    name == '
app.run(debug=True, port=5000, host='0.0.0.0')
```





EJECUTAR EL SERVIDOR WEB



EJECUTAR EL SERVIDOR WEB

(env) D:\GITHUB\CPSI\Bluep>python app.py

- * Serving Flask app 'app'
- * Debug mode: on

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

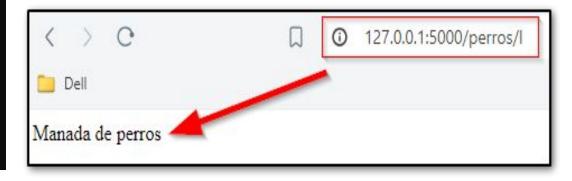
server instead.

- * Running on all addresses (0.0.0.0)
- * Running on http://127.0.0.1:5000
- * Running on http://192.168.1.88:5000

Press CTRL+C to quit

- * Restarting with stat
- * Debugger is active!
- * Debugger PIN: 104-357-436







DUDAS Y PREGUNTAS





MATERIALES DE FORMACIÓN







CUENTAME CÓMO TE FUE?





CREDITOS

Por: José Fernando Galindo Suarez jgalindos@sena.edu.co







GRACIAS

Línea de atención al ciudadano: 01 8000 910270 Línea de atención al empresario: 01 8000 910682



www.sena.edu.co