

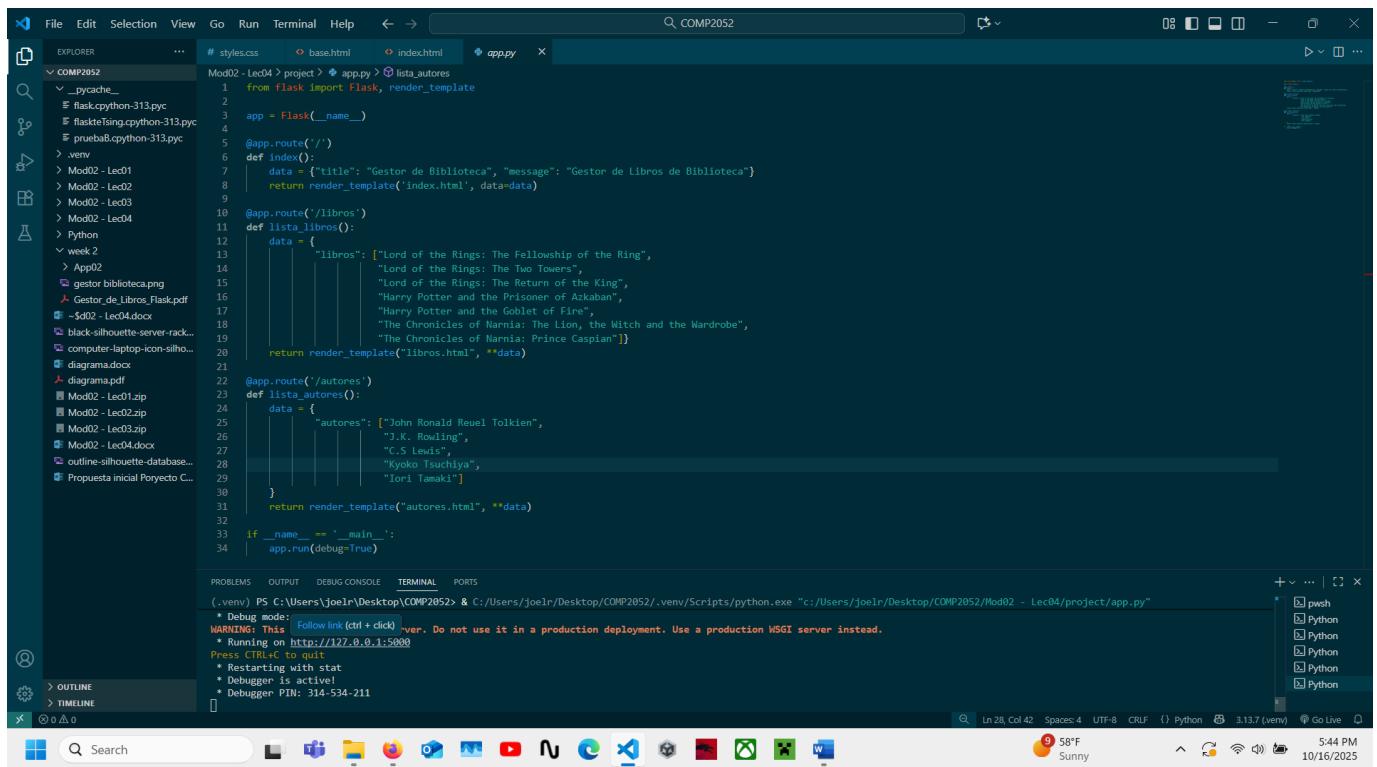
Hector J. Roman Olmo

R00482942

Modulo 02 – Lección 04: Plantillas (Templates) en Back-End con Flask

El separar el front-end del back-end me facilita actualizar la parte visual sin tener que modificar la lógica o viceversa. En la escalabilidad me permite adaptar el front-end del proyecto a diferentes plataformas como móvil, web o escritorio mientras que el back-end sigue funcionando igual.

Capturas de pantalla de prueba:



The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure named "COMP2052". Inside, there are several files and folders:
 - __pycache_
 - flask.pyhon-313.pyc
 - flaskTsing.pyhon-313.pyc
 - pruebaB.pyhon-313.pyc
 - venv
 - Mod02 - Lec01
 - Mod02 - Lec02
 - Mod02 - Lec03
 - Mod02 - Lec04
 - Python
 - week 2
 - App02
 - gestor biblioteca.png
 - Gestor_de_Libros_Flask.pdf
 - Mod02 - Lec04.docx
 - black-silhouette-server-rack...
 - computer-laptop-icon-silho...
 - diagrama.docx
 - diagrama.pdf
 - Mod02 - Lec01.zip
 - Mod02 - Lec02.zip
 - Mod02 - Lec03.zip
 - Mod02 - Lec04.zip
 - outline-silhouette-database...
 - Propuesta inicial Poryecto C...
- Code Editor:** The active file is "app.py" with the following content:

```
from flask import Flask, render_template
app = Flask(__name__)

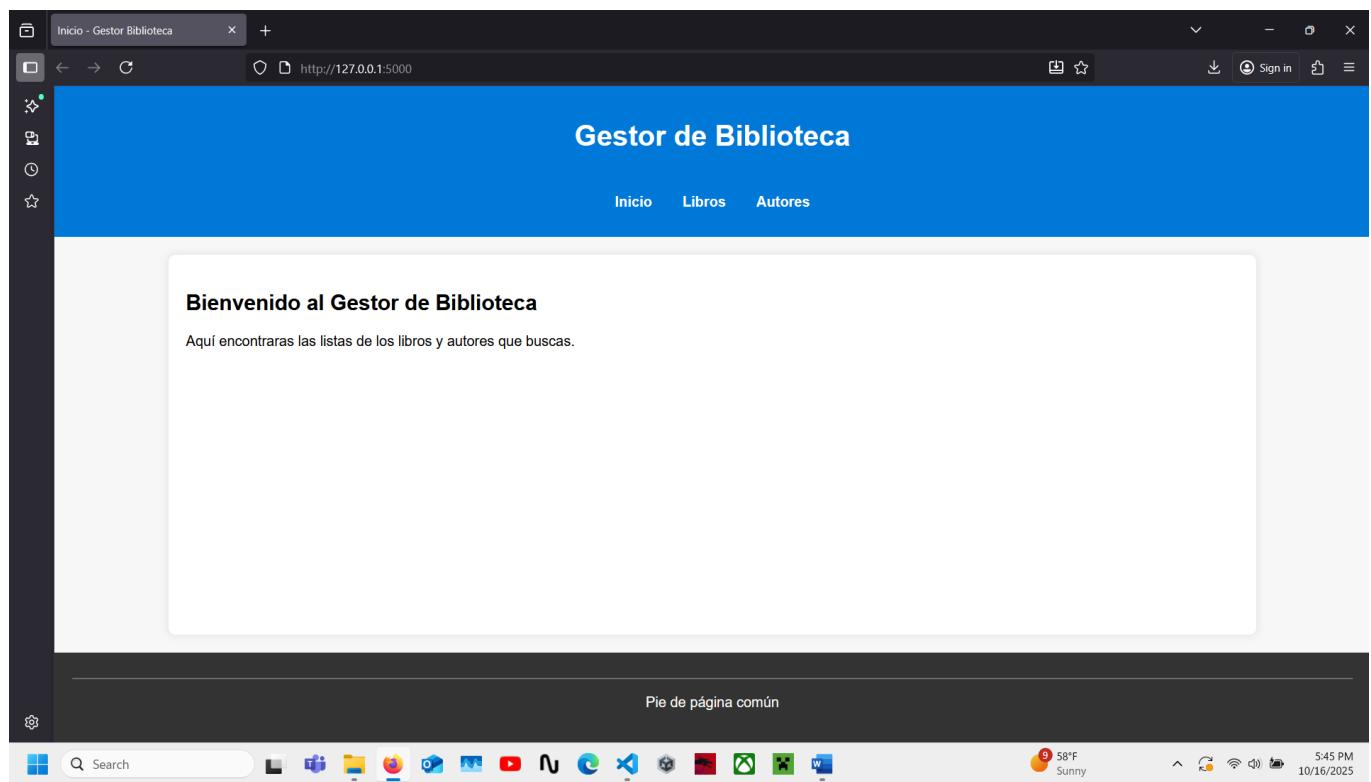
@app.route('/')
def index():
    data = {"title": "Gestor de Biblioteca", "message": "Gestor de Libros de Biblioteca"}
    return render_template('index.html', data=data)

@app.route('/libros')
def lista_libros():
    data = {
        "libros": [
            "Lord of the Rings: The Fellowship of the Ring",
            "Lord of the Rings: The Two Towers",
            "Lord of the Rings: The Return of the King",
            "Harry Potter and the Prisoner of Azkaban",
            "Harry Potter and the Goblet of Fire",
            "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe",
            "The Chronicles of Narnia: Prince Caspian"
        ]
    }
    return render_template('libros.html', **data)

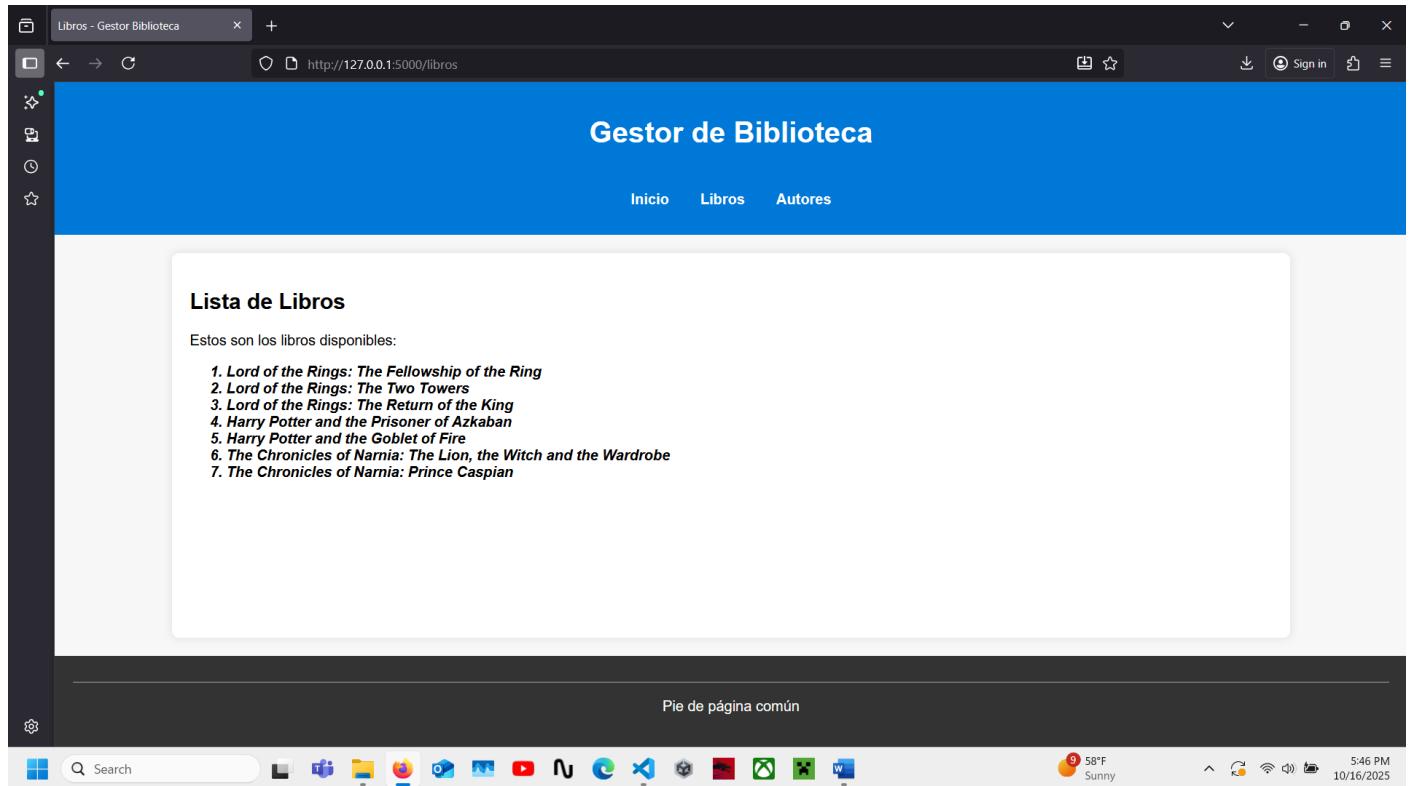
@app.route('/autores')
def lista_autores():
    data = {
        "autores": [
            "John Ronald Reuel Tolkien",
            "J.K. Rowling",
            "C.S. Lewis",
            "Kyoko Tsukiyama",
            "Iori Tamaki"
        ]
    }
    return render_template('autores.html', **data)

if __name__ == '__main__':
    app.run(debug=True)
```
- Terminal:** Shows the command being run: "PS C:\Users\joelr\Desktop\COMP2052> & C:/Users/joelr/Desktop/COMP2052/.venv/Scripts/python.exe "c:/Users/joelr/Desktop/COMP2052/Mod02 - Lec04/project/app.py"
- Status Bar:** Shows system information including battery level (9%), temperature (58°F), and date/time (10/16/2025).

Esta es la captura del momento en que se corrió el código. En la terminal muestra que esta funcionando. Tambien podemos observar 3 rutas: (/ , /libros, /autores) con sus respectivas listas.



Esta es la página inicial {index.html ruta = ('/'). Esta página hereda el “header” con el título y los 3 botones/links que permiten saltar a otras páginas de la página “base.html”. Las otras dos (“Libros” y “Autores”) también heredan el “header” de la página “base” como muestran las próximas dos capturas.



Esta captura muestra la página “libros.html” con una lista de libros.

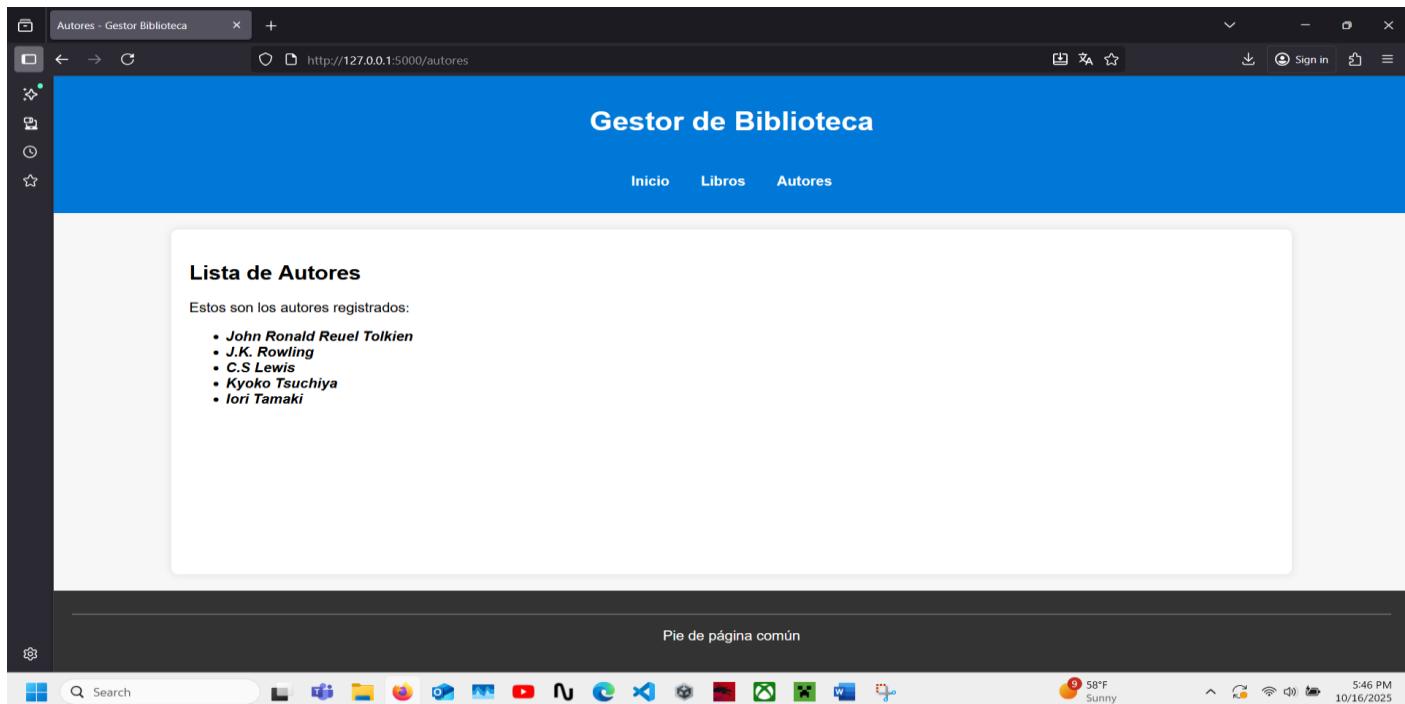
```
# styles.css
# base.html
# libros.html
# index.html

Mod02 - Lec04 > project > templates > libros.html > ol
1  {% extends "base.html" %} 
2  {% block title %}Libros - Gestor Biblioteca{% endblock %} 
3  {% block content %} 
4  <h2>Lista de Libros</h2> 
5  <p>Estos son los libros disponibles:</p> 
6  <ol> 
7  |   {% for libro in libros %} 
8  |       <strong><em>{{ libro }}</em></strong> 
9  |   {% endfor %} 
10 | </ol> 
11 | {% endblock %} 

libros.html
app.py
app.png
autores.png
base.png
inicio.png
libros.png
week 2
App02
gestor_biblioteca.png
Gestor_de_Libros_Flask.pdf
Mod02 - Lec04.docx
black-silhouette-server-rack...
computer-laptop-icon-silhouette...
diagrama.docx
diagrama.pdf
Mod02 - Lec01.zip
Mod02 - Lec02.zip
Mod02 - Lec03.zip
Mod02 - Lec04.docx
outline-silhouette-database...
OUTLINE
TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(.venv) PS C:\Users\joelr\Desktop\COMP2052> & C:/Users/joelr/Desktop/COMP2052/.venv/Scripts/python.exe "c:/Users/joelr/Desktop/COMP2052/Mod02 - Lec04/project/app.py"
127.0.0.1 - - [16/Oct/2025 17:51:05] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [16/Oct/2025 17:51:10] "GET /libros HTTP/1.1" 200 -
127.0.0.1 - - [16/Oct/2025 17:51:10] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [16/Oct/2025 17:51:10] "GET /static/base.html HTTP/1.1" 304 -
127.0.0.1 - - [16/Oct/2025 17:51:11] "GET /static/styles.css HTTP/1.1" 304 -
127.0.0.1 - - [16/Oct/2025 17:51:12] "GET /libros HTTP/1.1" 200 -
127.0.0.1 - - [16/Oct/2025 17:51:12] "GET /static/styles.css HTTP/1.1" 304 -
```

La captura anterior muestra el código de la página “libros.html” la cual hereda la mayoría de sus elementos de la página “base.html”. Aquí se utiliza una lista ordenada la cual recibe la lista que está en la app (“app.py”) dentro de la función “lista_libros()” de la ruta “/libros”



Esta captura muestra la página “autores.html” con una lista de autores.

```

<h2>Lista de Autores</h2>
<p>Estos son los autores registrados:</p>
<ul>
    {% for autor in autores %}
        <li><strong><em>{{ autor }}</em></strong></li>
    {% endfor %}
</ul>

```

Esta captura muestra el código de la página “autores.html” la cual hereda la mayoría de sus elementos de la página “base.html”. Aquí se utiliza una lista desordenada la cual recibe la lista que está en la app (“app.py”) dentro de la función “lista_autores()” de la ruta “/autores”.

Enlace github:

<https://github.com/hjroman/COMP2052>