

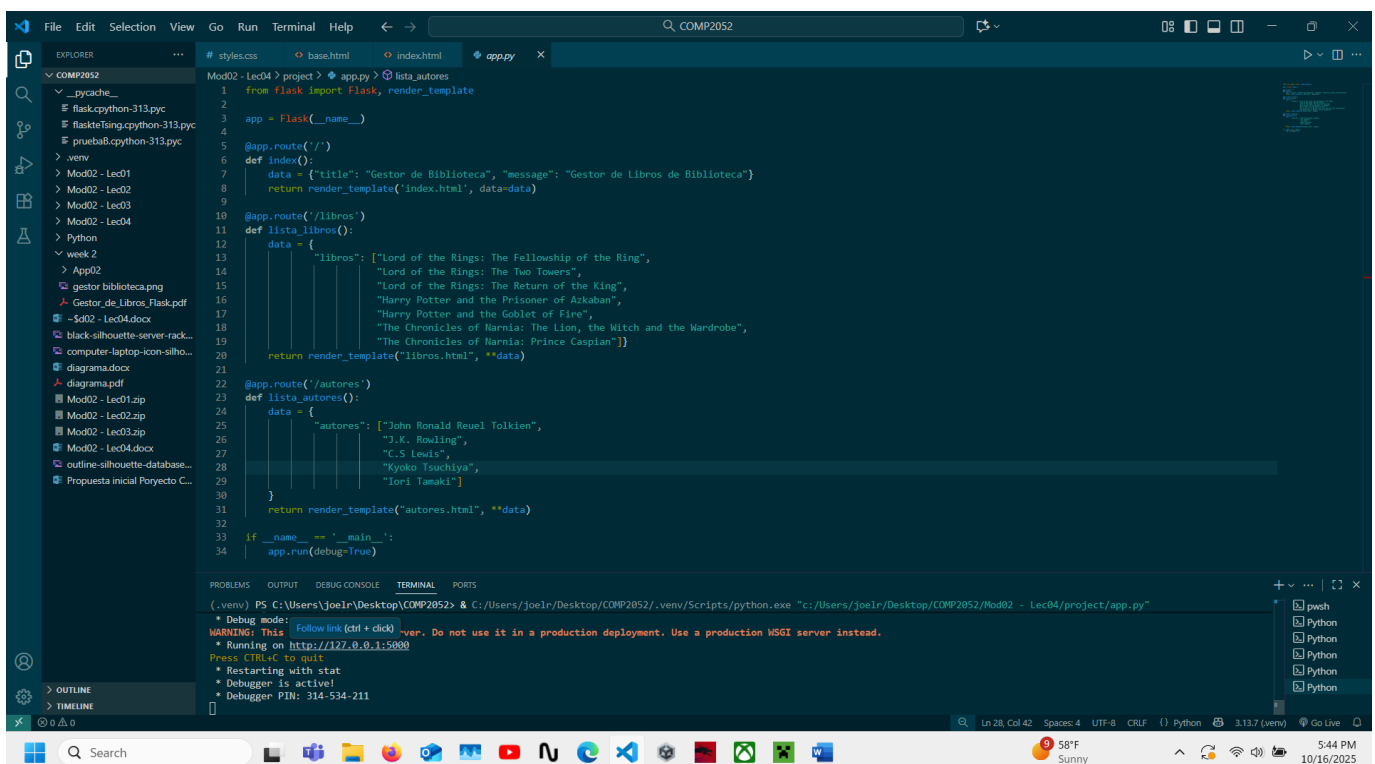
Hector J. Roman Olmo

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:

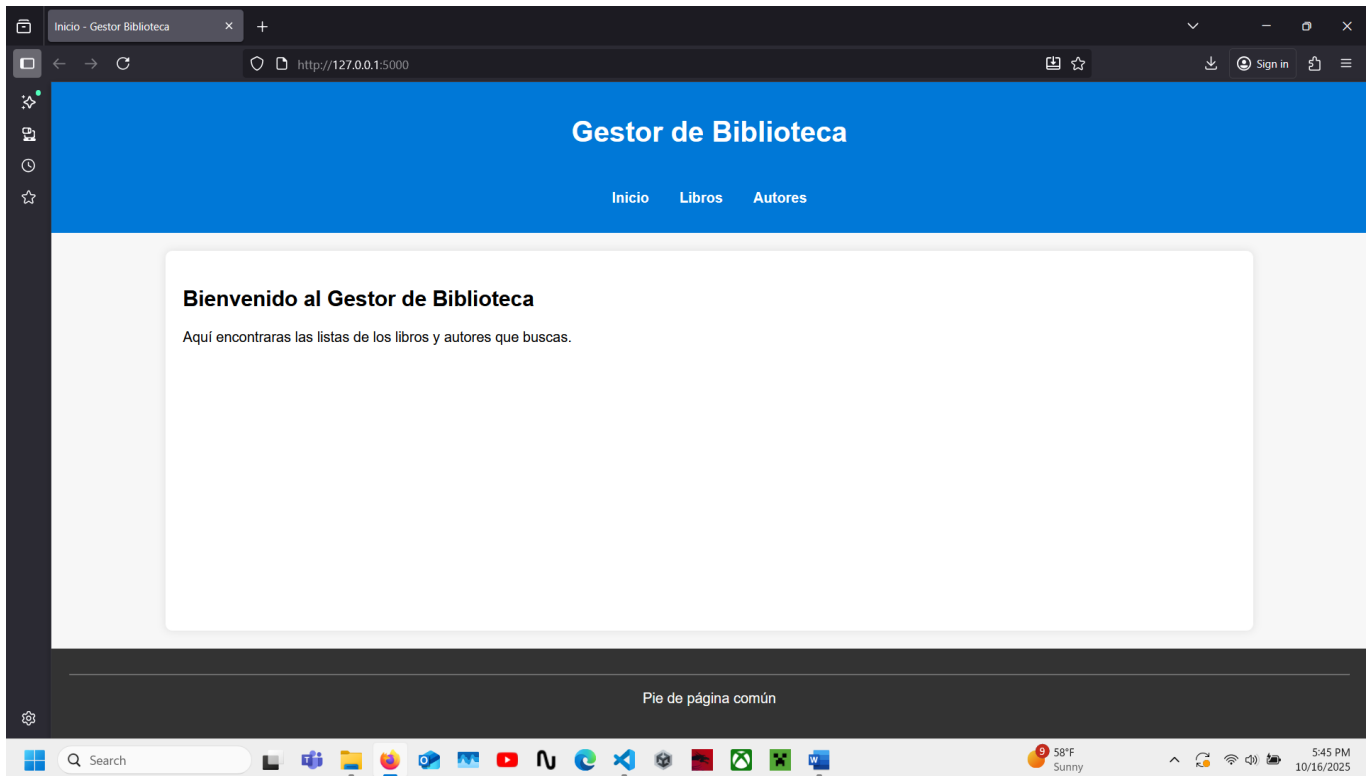


The screenshot shows a Visual Studio Code editor window with a Flask application. The Explorer pane on the left shows the project structure, including files like `styles.css`, `base.html`, `index.html`, and `app.py`. The main editor displays the `app.py` file with the following code:

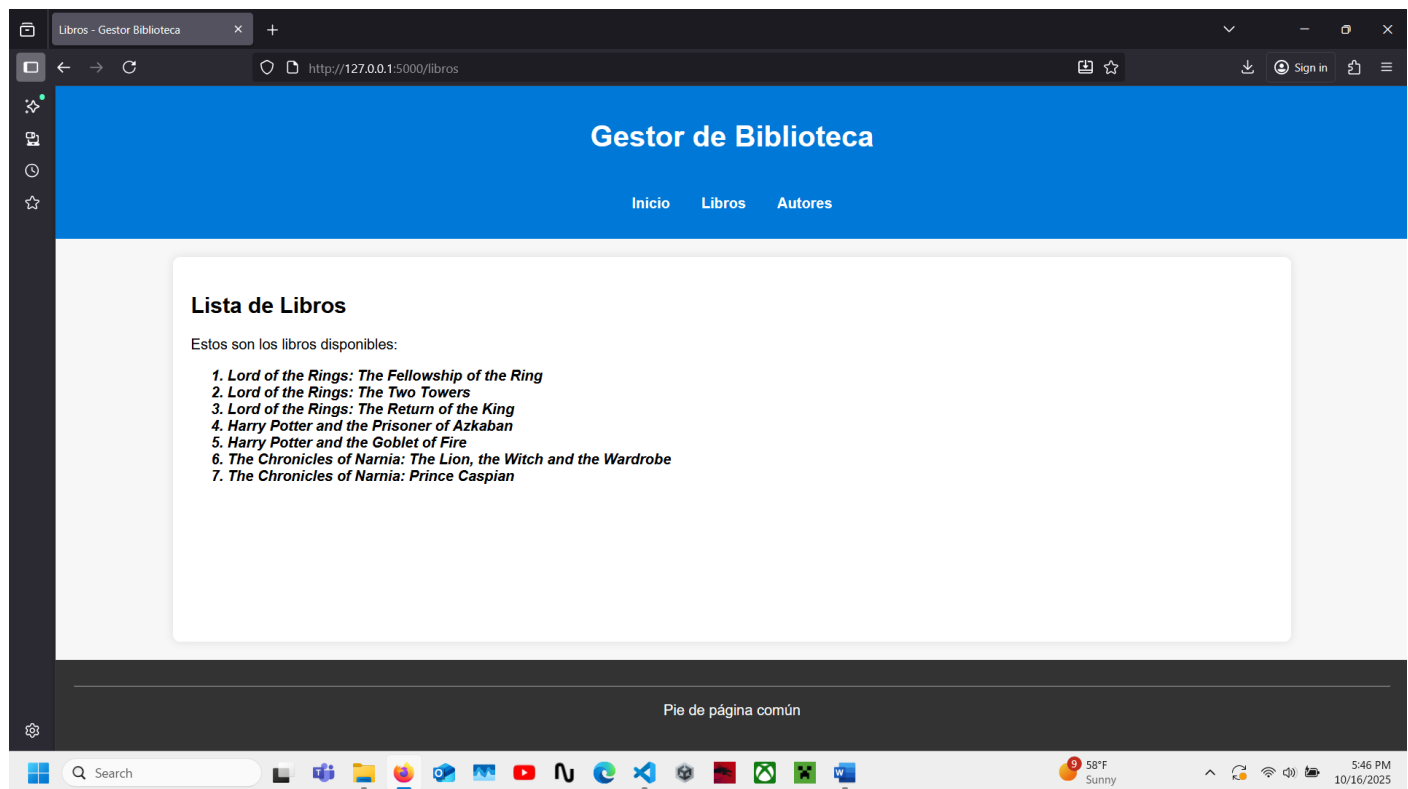
```
# styles.css
1 from flask import Flask, render_template
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def index():
7     data = {"title": "Gestor de Biblioteca", "message": "Gestor de libros de Biblioteca"}
8     return render_template("index.html", data=data)
9
10 @app.route('/libros')
11 def lista_libros():
12     data = {
13         "libros": [
14             "Lord of the Rings: The Fellowship of the Ring",
15             "Lord of the Rings: The Two Towers",
16             "Lord of the Rings: The Return of the King",
17             "Harry Potter and the Prisoner of Azkaban",
18             "Harry Potter and the Goblet of Fire",
19             "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe",
20             "The Chronicles of Narnia: Prince Caspian"]
21     }
22     return render_template("libros.html", **data)
23
24 @app.route('/autores')
25 def lista_autores():
26     data = {
27         "autores": [
28             "John Ronald Reuel Tolkien",
29             "J.K. Rowling",
30             "C.S. Lewis",
31             "Kyoko Tsuchiya",
32             "Iori Tamaki"]
33     }
34     return render_template("autores.html", **data)
35
36 if __name__ == '__main__':
37     app.run(debug=True)
```

The terminal at the bottom shows the command `python app.py` being executed, and the application is running on `http://127.0.0.1:5000`. The status bar at the bottom indicates the file is `app.py` and the Python interpreter is `3.11.7 (venv)`.

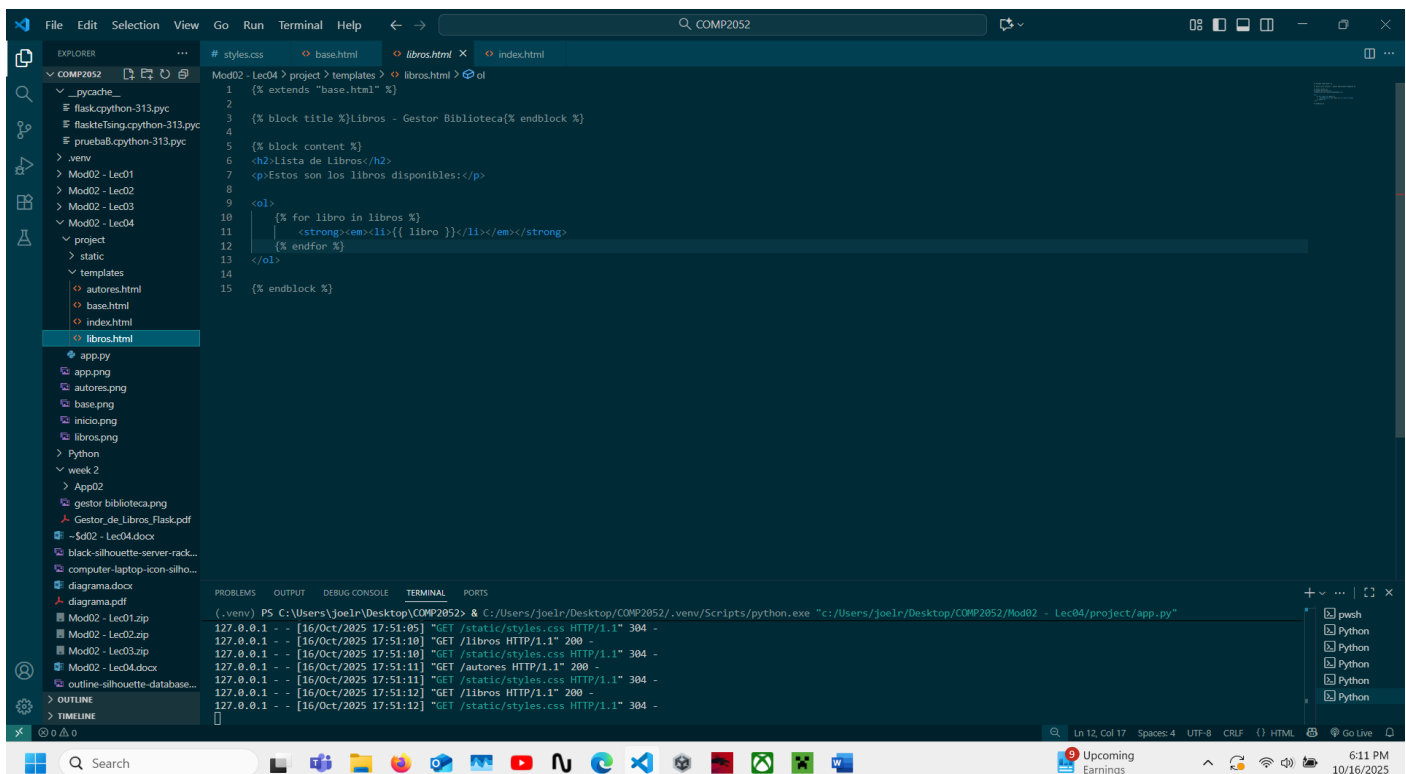
Esta es la captura del momento en que se corrió el código. En la terminal muestra que esta funcionando. También podemos observar 3 rutas: (`/`, `/libros`, `/autores`) con sus respectivas listas.



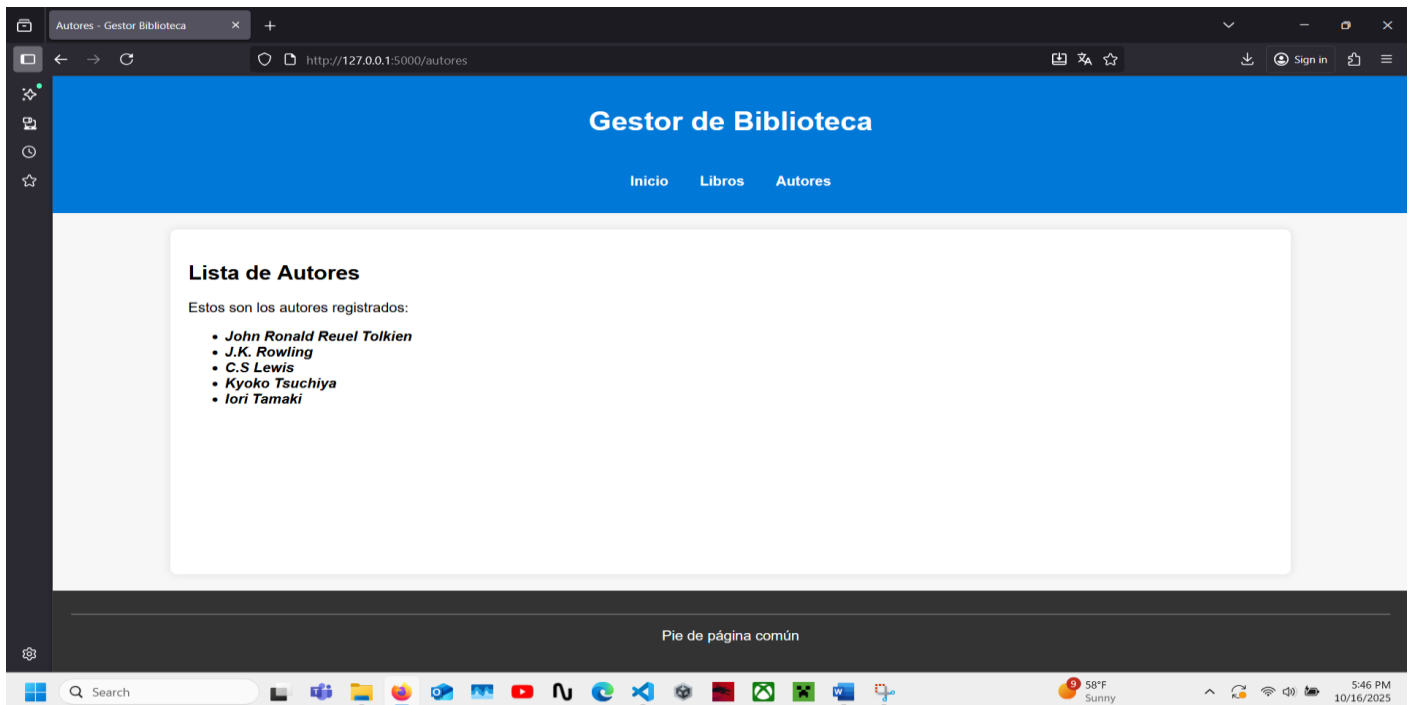
Esta es la página inicial {index.html ruta = ('/'}}. Esta página hereda el “header” con el título y los 3 botones/links que permiten saltar a otras páginas de la página “base.html”. Las otras dos (“Libros” y “Autores”) también heredan el “header” de la página “base” como muestran las próximas dos capturas.



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



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.

The screenshot shows a Visual Studio Code editor with the file 'autores.html' open. The code is a Jinja2 template that extends 'base.html'. It defines a block title 'Autores - Gestor Biblioteca', a block content with an unordered list of authors, and a block endblock. The list is generated by a loop over 'autores'. The terminal window at the bottom shows the command prompt and several HTTP requests and responses, including GET requests for static files and the /autores endpoint.

```
1 {% extends "base.html" %}
2
3 {% block title %}Autores - Gestor Biblioteca{% endblock %}
4
5 {% block content %}
6 <h2>Lista de Autores</h2>
7 <p>Estos son los autores registrados:</p>
8
9 <ul>
10     {% for autor in autores %}
11     <li><strong><em><i>{{ autor }}</i></em></strong>
12     {% endfor %}
13 </ul>
14 {% endblock %}
```

Terminal output:

```
(.venv) PS C:\Users\joeln\Desktop\COMP2052> & C:/Users/joeln/Desktop/COMP2052/.venv/Scripts/python.exe "c:/Users/joeln/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 17:51:11] "GET /autores HTTP/1.1" 200 -
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 -
```

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>