Auxiliar 5

Factous de Flask 2

START

Templates

Flask usa plantillas para enviar el HTML.

Usa un motor de plantillas llamado **Jinja 2** que tiene su propio lenguaje para hacer el HTML de forma inteligente.

Podemos usar variables, ciclos, condiciones, funciones, todo dentro del HTML.

```
<ol>
{% for usuario in usuarios %}
 <li>{{ usuario.nombre }}</li>
{% endfor %}
</ol>
```

Templates

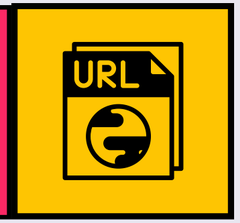
Los *Layouts* evitan la repetición de código y mantienen consistencia del diseño de la página.

Usualmente se hace un archivo *base* donde luego se inyectan secciones de HTML

Tags importantes: extends, block

```
{% extends 'base.html' %}
{% block contenido %}
<h1>HOLAAA ESTA ES MI PAGINA WEB!!
AAAAAAAA</h1>
{% endblock %}
```

Rutas

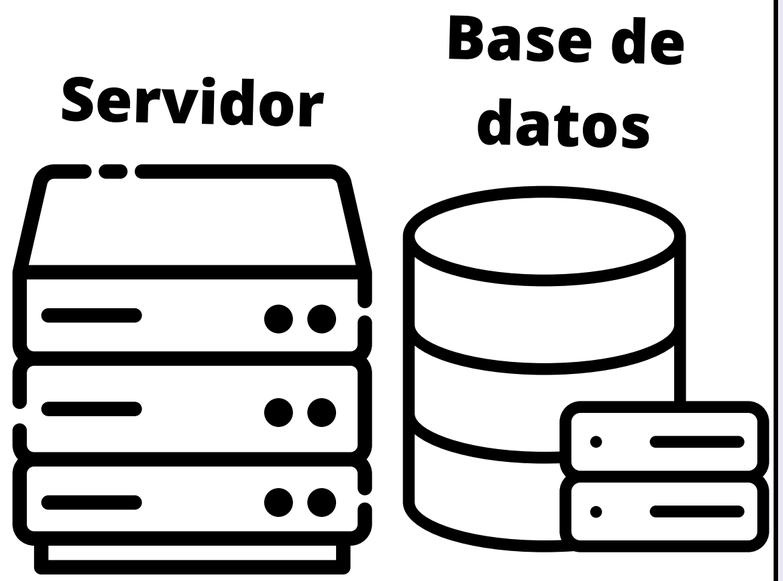


Como accedemos a nuestro sitio. (/login, /index.html)

Hay que registrarlas. Cada una tiene que devolver una respuesta. (200, 404, etc)

En el cuerpo de la respuesta podemos enviar información como el documento HTML o un objeto JSON.

```
@app.route("/index/", method=(GET,POST))
def index():
  return "Hola"
```


Servidor
Base de datos

Request (Petición)

- Método (GET)
- Cabeceras
- URL (/inicio)
- Cuerpo

Servidor

Base de datos

Request (Petición)

Método (GET)

URL (/inicio)

Cabeceras

Cuerpo

Servidor

Base de datos

```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    return render_template(mi_temp)
```

Request (Petición)

- Método (GET)
- URL (/inicio)
- Cabeceras
- Cuerpo

Servidor

Base de datos

Response (Respuesta)

- Status (200)
- Cabeceras

```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    return render_template(mi_temp)
```

Request (Petición)

- Método (GET)
- URL (/inicio)
- Cabeceras
- Cuerpo

Servidor

Base de datos

```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    usuario = db.get_usuario(1)
    return render_template(mi_temp, usuario)
```

Usuario 1 (nombre:"Vicente", apellido:"Valdes")

Request (Petición)

Método (GET)

Cabeceras

URL (/inicio)

Cuerpo

Servidor

Base de datos

Response (Respuesta)

Status (200)

Cabeceras


```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    usuario = db.get_usuario(1)
    return render_template(mi_temp, usuario)
```

Request (Petición)

Método (POST)

URL (/inicio)

Cabeceras

Cuerpo (Con formulario)

Servidor

Base de datos

```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    return render_template(mi_temp)
  elif request.method == "POST":
    input = request.form['mi-input']
    validar(input)
    db.save(input)
    return redirect(url_for('index'))
```

Request (Petición)

Método (POST)

URL (/inicio)

Cabeceras

Cuerpo (Con formulario)

Response (Respuesta)

Status (302)

Cabeceras

Método (GET)

URL (/inicio)

Servidor

Base de datos

```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    return render_template(mi_temp)
  elif request.method == "POST":
    input = request.form['mi-input']
    validar(input)
    db.save(input)
    return redirect(url_for('index'))
```

Request (Petición)

Método (GET)

URL (/inicio)

Cabeceras

Cuerpo

Servidor

Base de datos

Response (Respuesta)

Status (200)

Cabeceras

```
@app.route("/inicio/", method=(GET,POST))
def index():
  if request.method == "GET":
    return render_template(mi_temp)
```

Archivos

No se guardan en la base de datos convencional (MySQL)

Se guardan en el sistema de archivos del servidor o un servidor dedicado a archivos

Pueden ser muy maliciosos así que TIENEN que ser validados

Llegan en la request a través de `request.files`

Validación de Archivos

- Restringir tamaño del archivo
- Restringir cantidad de archivos
- Restringir el tipo del archivo
- Cuidado con el nombre del archivo

MIME type: especifican tipos de datos, como por ejemplo texto, imagen, audio, etc. que los archivos contienen. (Ej: image/gif)