

# **PROYECTO DE SOFTWARE**

**Cursada 2022**

# TEMARIO

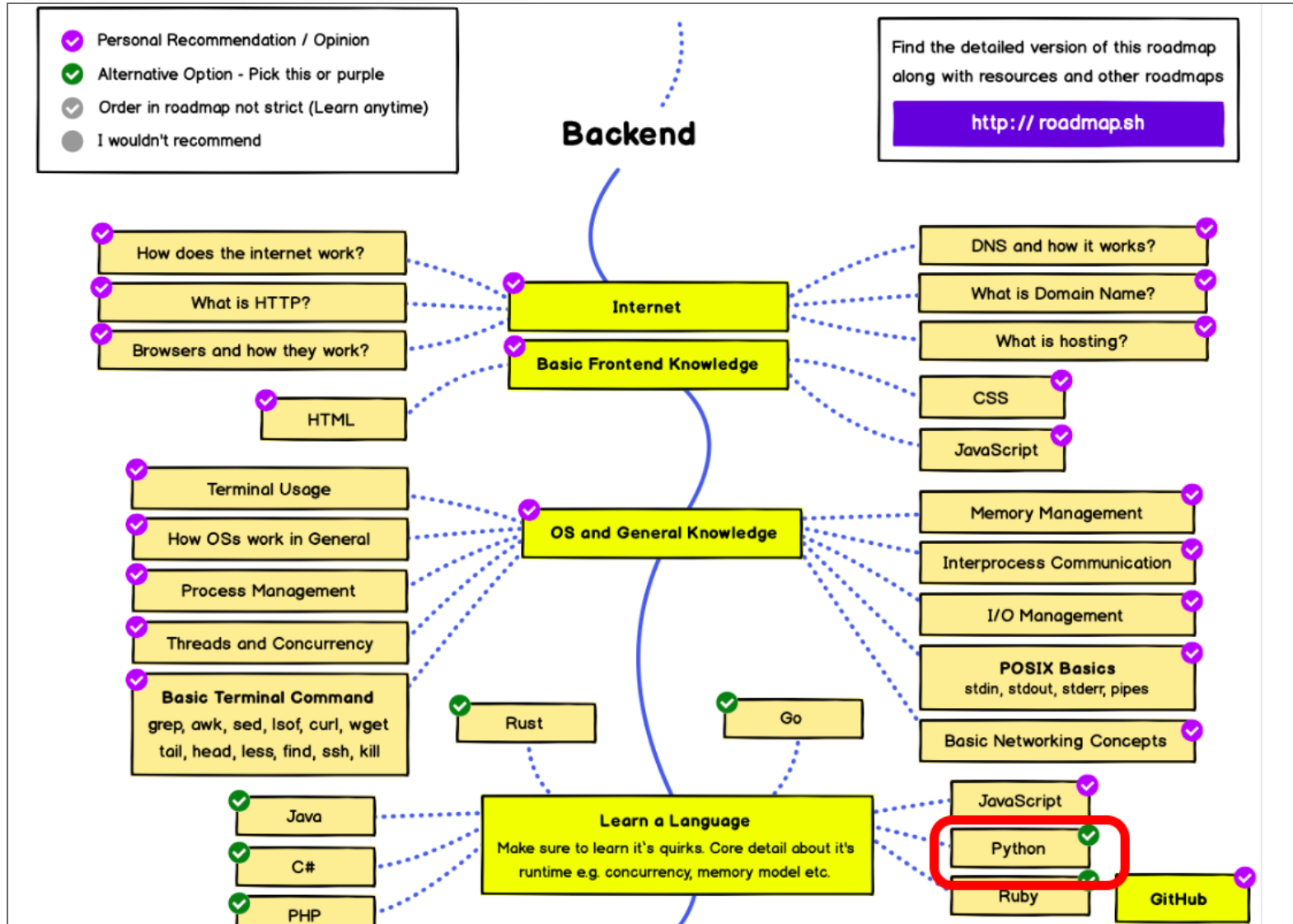
- Flask
- Templates - Introducción

# REPASO PYTHON

# REPASANDO

- ¿Por qué usamos y promovemos el uso de software libre?
- ¿Características de Python?
- ¿Desarrollo en el servidor?

# PROGRAMACIÓN EN EL SERVIDOR





# NOSOTROS USAREMOS PYTHON

¿Por qué?

- <https://www.tiobe.com/tiobe-index/>
- <https://insights.stackoverflow.com/trends?tags=python>
- <https://github.info/>
- [https://madnight.github.io/github/#/pull\\_requests/](https://madnight.github.io/github/#/pull_requests/)

# PYTHON

- Fue desarrollado por Guido Van Rossum a finales de los años 80.
- El nombre proviene del grupo de cómicos ingleses Monty Python.
- Sitio Oficial: <https://python.org/>
- Python Argentina: <https://python.org.ar/>



# CARACTERÍSTICAS GENERALES

- Es un lenguaje de alto nivel, fácil de aprender. Muy expresivo y legible.
- Es interpretado.
- Multiplataforma y multiparadigma.
- Posee una sintáxis elegante y tipado dinámico y fuerte.
- Tiene un manejo eficiente de estructuras de datos de alto nivel.

# SINTAXIS BÁSICA

- **NO** hay declaración de variables: las variables se crean **dinámicamente** cuando se les asigna un valor.
- Tiene **tipado dinámico**: las variables pueden cambiar de tipo a lo largo del programa.
- Bloques, indentación.

# FRAMEWORK

- Esquema de trabajo que nos permite:
  - Evitar la escritura de código de manera repetitiva
  - Buenas prácticas
  - Estructura del código relativamente homogéneo.

# FLASK

- Instalamos un nuevo paquete al entorno, Flask.

```
poetry env use
/home/diego/anaconda3/envs/proyecto2022/bin/python3.10
poetry add flask
```

```
diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/Proyecto2020$ poetry add flask
The currently activated Python version 3.8.10 is not supported by the project (3.10.4).
Trying to find and use a compatible version.

Poetry was unable to find a compatible version. If you have one, you can explicitly use it via the "env use" command.
diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/Proyecto2020$ poetry env use /home/diego/anaconda3/envs/proyecto2022/bin/python3.10
Creating virtualenv proyecto2022-elplLz6u-py3.10 in /home/diego/.cache/pypoetry/virtualenvs
Using virtualenv: /home/diego/.cache/pypoetry/virtualenvs/proyecto2022-elplLz6u-py3.10
diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/Proyecto2020$ poetry add flask
Using version ^2.2.2 for Flask

Updating dependencies
Resolving dependencies... (0.2s)

Writing lock file

Package operations: 6 installs, 0 updates, 0 removals

  • Installing markupsafe (2.1.1)
  • Installing click (8.1.3)
  • Installing itsdangerous (2.1.2)
  • Installing jinja2 (3.1.2)
  • Installing werkzeug (2.2.2)
  • Installing flask (2.2.2)
```



# FLASK

- Hola Mundo

```
# app.py

from flask import Flask
app = Flask(__name__)

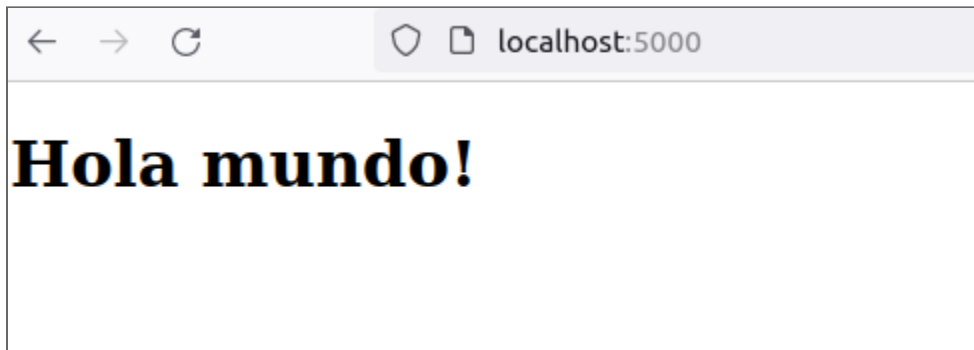
@app.route('/')
def hello_world():
    return 'Hello, World!'
```

# FLASK

- Miramos nuestro código en el browser:

```
poetry run flask run
```

```
diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/ejemplos-proyecto-2022/admin$ poetry run flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
```



# FLASK

- Tips

```
poetry shell
```

```
flask run
```

```
diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/ejemplos-proyecto-2022/admin$ poetry shell
Spawning shell within /home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDJr-py3.10
. /home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDJr-py3.10/bin/activate
diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/ejemplos-proyecto-2022/admin$ . /home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDJr-py3.10/bin/activate
(admin-py3.10) diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/ejemplos-proyecto-2022/admin$ flask run
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
█
```



# FLASK

- Modificamos nuestro código:

```
# app.py

from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    mensaje = 'Hello, World!'
    return mensaje + 2
```

## Internal Server Error

The server encountered an internal error and was unable to complete your request. Either the server is overloaded or there is an error in the application.

# FLASK

- Activamos el entorno de desarrollo:

```
flask --debug run
```

```
(admin-py3.10) diego@diego-Lenovo-V330-15IKB:~/Documentos/Git/ejemplos-proyecto-2022/admin$ flask --debug run
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 585-477-601
```

# FLASK

- Ahora podemos ver un poco más:

## TypeError

`TypeError: can only concatenate str (not "int") to str`

### Traceback (most recent call last)

```
File "/home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDjr-py3.10/lib/python3.10/site-packages/flask/app.py", line 2548, in __call__
    return self.wsgi_app(environ, start_response)
File "/home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDjr-py3.10/lib/python3.10/site-packages/flask/app.py", line 2528, in wsgi_app
    response = self.handle_exception(e)
File "/home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDjr-py3.10/lib/python3.10/site-packages/flask/app.py", line 2525, in wsgi_app
    response = self.full_dispatch_request()
File "/home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDjr-py3.10/lib/python3.10/site-packages/flask/app.py", line 1822, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDjr-py3.10/lib/python3.10/site-packages/flask/app.py", line 1820, in full_dispatch_request
    rv = self.dispatch_request()
File "/home/diego/.cache/pypoetry/virtualenvs/admin-uPRSrDjr-py3.10/lib/python3.10/site-packages/flask/app.py", line 1796, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**view_args)
File "/home/diego/Documentos/Git/ejemplos-proyecto-2022/admin/app.py", line 7, in hello_world
    return "<p>Hello, World!</p>" + 2
```

`TypeError: can only concatenate str (not "int") to str`

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it.

# FLASK

- Antes de continuar vamos a crear algunas carpetas dentro de admin
  - ■ public
  - ■ src



# FLASK

- Vamos a agregar dentro de src dos carpetas
- ■ core
- ■ web



# FLASK

- Pasamos la creación de la app flask al módulo web.

```
# src/web/__init__.py

from flask import Flask

def create_app():
    app = Flask(__name__)

    # Define home
    @app.route("/")
    def hello_world():
        return "<p>Hello, World!</p>"

    return app
```

# FLASK

- Modificamos app.py

```
# app.py

from src.web import create_app

app = create_app()

def main():
    app.run()

if __name__ == "__main__":
    main()
```

# FLASK

- Vamos recibir argumentos

```
# src/web/__init__.py
from flask import Flask

def create_app():
    app = Flask(__name__)

    # Define home
    @app.route("/")
    def hello_world():
        return "<p>Hello, World!</p>"

    @app.route("/personas")
    def personas():
        return "personas"

    @app.route("/personas/<string:nombre>")
    def persona(nombre):
        nombre = nombre.capitalize()
        return f"Hola {nombre}"

    return app
```



# FLASK

- Armamos un html:

```
# src/web/__init__.py
from flask import Flask

def create_app():
    app = Flask(__name__)

    # Define home
    @app.route("/")
    def hello_world():
        mensaje = '<!DOCTYPE html>'
        mensaje += '<html lang="es"> '
        mensaje += '<head>'
        mensaje += '</head>'
        mensaje += '<body>'
        mensaje += 'Hola mundo!'
        mensaje += '</body>'
        mensaje += '</html>'
        return mensaje
```

# FLASK

- y otro...:

```
# src/web/__init__.py
from flask import Flask

def create_app():
    app = Flask(__name__)

    .....

@app.route("/personas")
def personas():
    mensaje = '<!DOCTYPE html>'
    mensaje += '<html lang="es"> '
    mensaje += '<head>'
    mensaje += '</head>'
    mensaje += '<body>'
    mensaje += 'Hola personas!'
    mensaje += '</body>'
    mensaje += '</html>'
    return mensaje
```

# FLASK

- y otro...:

```
# src/web/__init__.py
from flask import Flask

def create_app():
    app = Flask(__name__)

    .....

@app.route("/personas/<string:nombre>")
def persona(nombre):
    nombre = nombre.capitalize()
    mensaje = '<!DOCTYPE html>'
    mensaje += '<html lang="es"> '
    mensaje += '<head>'
    mensaje += '</head>'
    mensaje += '<body>'
    mensaje += f"Hola {nombre}"
    mensaje += '</body>'
    mensaje += '</html>'
    return mensaje
```

**FLASK**

**¿Problemas? ¿Inconvenientes?**

# FLASK

- Una solución

```
from flask import Flask, render_template

def create_app(static_folder="static"):
    app = Flask(__name__)

    # Define home
    @app.route("/")
    def hello_world():
        return render_template('index.html')

    .....

    return app
```

# FLASK

- index.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Ejemplo Proyecto</title>
  </head>
  <body>
    <h1>Hola mundo!</h1>
  </body>
</html>
```

# FLASK

- Con variables

```
.....  
  
@app.route("/personas")  
def personas():  
    contenido = "Hola personas!"  
    return render_template('index.html',  
contenido=contenido)  
  
@app.route("/personas/<string:nombre>")  
def persona(nombre):  
    nombre = nombre.capitalize()  
    contenido = f"Hola {nombre}"  
    return render_template('index.html',  
contenido=contenido)  
  
.....
```

# FLASK

- index.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Ejemplo Proyecto</title>
  </head>
  <body>
    <h1>{{contenido}}</h1>
  </body>
</html>
```



# FLASK

- Condicionales

```
....  
  
@app.route("/")  
def hello_world():  
    contenido = "mundo"  
    return render_template('index.html',  
contenido=contenido)  
  
@app.route("/personas")  
def personas():  
    contenido = "personas"  
    return render_template('index.html',  
contenido=contenido)  
  
@app.route("/personas/<string:nombre>")  
def persona(nombre):  
    nombre = nombre.capitalize()  
    contenido = nombre  
    return render_template('index.html',  
contenido=contenido)  
  
....
```



# FLASK

- index.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Ejemplo Proyecto</title>
  </head>
  <body>
    {% if contenido == "Argentina" %}
      <h1>La Ciudad Autónoma de Buenos Aires es la
capital de {{contenido}} <h1>
    {% else %}
      <h1>Hola {{contenido}} !!<h1>
    {% endif %}
  </body>
</html>
```

# FLASK

- Definimos una lista de issues

```
issues = [  
    {  
        "id": 1,  
        "user": "José",  
        "title": "Mi computadora no funciona.",  
        "description": "Mi departamento me compró una  
nueva computadora y necesito configurarla con todos mis  
emails y documentos de mi vieja computadora.",  
        "status": "new",  
    },  
    {  
        "id": 2,  
        "user": "María",  
        "title": "No puedo obtener mis emails.",  
        "description": "Estoy tratando de acceder a mi  
correo desde casa, pero no puedo obtenerlos. Estoy  
tratando con Outlook en mi casa pero en la oficina tengo  
Thunderbird.",  
        "status": "in_progress",  
    },  
    {  
        "id": 3,  
        "user": "Rubén",  
        "title": "No puedo imprimir",  
        "description": "Cada vez que trato de imprimir mi  
presentación el programa se cierra. Esto sólo me pasa con
```

```
PowerPoint en Word puedo imprimir. Ya me aseguré que la  
impresora está prendida. Tengo una HP LaserJet 5.",  
    "status": "done",  
  },  
]
```

# FLASK

- Iteraciones

```
def create_app(static_folder="static"):  
    ....  
    @app.route("/issues/")  
    def issues_index():  
        return render_template("issues/index.html",  
issues=issues)  
    return app
```

# FLASK

- index.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Ejemplo Proyecto</title>
  </head>
  <body>
    <h1>Consultas</h1>

    <table class="index-table box">
      <thead>
        <tr>
          <th>#</th><th>Autor</th><th>Título</th>
        <th>Descripción</th><th>Estado</th>
        </tr>
      </thead>
      <tbody>
        {% for issue in issues %}
        <tr>
          <th>{{ issue.id }}</th><th>{{ issue.user }}</th>
          <th>{{ issue.title }}</th><th>{{ issue.description }}</th>
          <th>{{ issue.status }}</th>
        </tr>
        {% endfor %}
      </tbody>
    </table>
```

```
</body>  
</html>
```



# FLASK

- Hermoso listado

## Consultas

#	Autor	Título	Descripción	Estado
1	José	Mi computadora no funciona.	Mi departamente me compró una nueva computadora y necesito configurarla con todos mis emails y documentos de mi vieja computadora.	new
2	María	No puedo obtener mis emails.	Estoy tratando de acceder a mi correo desde casa, pero no puedo obtenerlos. Estoy tratando con Outlook en mi casa pero en la oficina tengo Thunderbird.	in_progress
3	Rubén	No puedo imprimir	Cada vez que trato de imprimir mi presentación el programa se cierra. Esto sólo me pasa con PowerPoint en Word puedo imprimir. Ya me aseguré que la impresora está prendida. Tengo una HP LaserJet 5.	done

# FLASK

- Un poco de estilo, en los templates index.html ponemos lo siguiente:

```
<head>
  <title>Ejemplo Proyecto</title>
  <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}" />
</head>
```

# FLASK

- Un poco de estilo

## Consultas

#	Autor	Título	Descripción	Estado
1	José	Mi computadora no funciona.	Mi departamento me compró una nueva computadora y necesito configurarla con todos mis emails y documentos de mi vieja computadora.	new
2	María	No puedo obtener mis emails.	Estoy tratando de acceder a mi correo desde casa, pero no puedo obtenerlos. Estoy tratando con Outlook en mi casa pero en la oficina tengo Thunderbird.	in_progress
3	Rubén	No puedo imprimir	Cada vez que trato de imprimir mi presentación el programa se cierra. Esto sólo me pasa con PowerPoint en Word puedo imprimir. Ya me aseguré que la impresora está prendida. Tengo una HP LaserJet 5.	done

**FLASK**

**¿Problemas? ¿Inconvenientes?**

# FLASK

- Url

```
....  
@app.route("/issues/")  
def issues_index():  
    return render_template("issues/index.html",  
issues=issues)  
....
```

# FLASK

- index.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Ejemplo Proyecto</title>
    <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}" />
  </head>
  <body>
    {% if contenido == "Argentina" %}
      <h1>La Ciudad Autónoma de Buenos Aires es la
capital de {{contenido}} </h1>
    {% else %}
      <h1>Hola {{contenido}} !!</h1>
    {% endif %}
    <a href="{{url_for('issues_index')}}"> Issues </a>
  </body>
</html>
```

# FLASK

- Url

Hola mundo !!

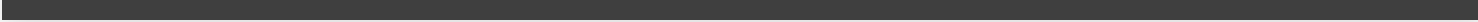
[Issues](#)

# FLASK

- Herencia
- layout.html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    {% block head %}
      <link rel="stylesheet" href="{{ url_for('static',
filename='style.css') }}" />
      <title>{% block title %}{% endblock %}</title>
    {% endblock %}
  </head>
  <body>
    <div id="navbar">
      {% block navbar %}
      {% endblock %}
    </div>
    <div id="main">
      {% block content %}
      {% endblock %}
    </div>
    <div id="footer">
      {% block footer %}
      {% endblock %}
    </div>
  </body>
</html>
```





# FLASK

- Herencia
- index.html

```
{% extends "layout.html" %}
{% block title %}Ejemplo Proyecto{% endblock %}
{% block head %}
    {{ super() }}
{% endblock %}
{% block content %}
    {% if contenido == "Argentina" %}
        <h1>La Ciudad Autónoma de Buenos Aires es la
capital de {{contenido}} <h1>
    {% else %}
        <h1>Hola {{contenido}} !!<h1>
    {% endif %}
    <a href="{{url_for('issues_index')}}"> Issues </a>
{% endblock %}
```

# FLASK

- Herencia
- issues/index.html

```
{% extends "layout.html" %}
{% block title %}Consultas{% endblock %}
{% block head %}
    {{ super() }}
{% endblock %}
{% block content %}
    <h1>Consultas</h1>

    <table class="index-table box">
        <thead>
            <tr>
                <th>#</th>
                <th>Autor</th>
                <th>Título</th>
                <th>Descripción</th>
                <th>Estado</th>
            </tr>
        </thead>
        <tbody>
            {% for issue in issues %}
            <tr>
                <th>{{ issue.id }}</th>
                <th>{{ issue.user }}</th>
                <th>{{ issue.title }}</th>
```

```
    <th>{{ issue.description }}</th>
    <th>{{ issue.status }}</th>
  </tr>
  {% endfor %}
</tbody>
</table>
{% endblock %}
```

# FLASK

- Formularios
- issues/index.html

```
{% extends "layout.html" %}
{% block title %}Consultas{% endblock %}
{% block head %}
    {{ super() }}
{% endblock %}
{% block content %}
    .....

</table>
<form action="{{ url_for('issues_add')}}" method="post">
    <input type="text" name="id" placeholder="Ingrese el
id">
    <input type="text" name="user" placeholder="Ingrese el
autor">
    <input type="text" name="title" placeholder="Ingrese
el título">
    <input type="text" name="description"
placeholder="Ingrese la descripción">
    <input type="text" name="status" placeholder="Ingrese
el estado">
    <button>Enviar</button>
</form>
<a href="{{url_for('hello_world')}}"> Volver al inicio
</a>
```

```
{% endblock %}
```

# FLASK

- Formularios

```
from flask import Flask, render_template, request

def create_app(static_folder="static"):
    app = Flask(__name__, static_folder=static_folder)
    ....

@app.route("/issues/add", methods=["POST"])
def issues_add():
    issue = {"id": request.form.get("id"),
            "user": request.form.get("user"),
            "title": request.form.get("title"),
            "description": request.form.get("description"),
            "status": request.form.get("status"),
            }
    issues.append(issue)
    return render_template("issues/index.html",
issues=issues)

....
```