



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



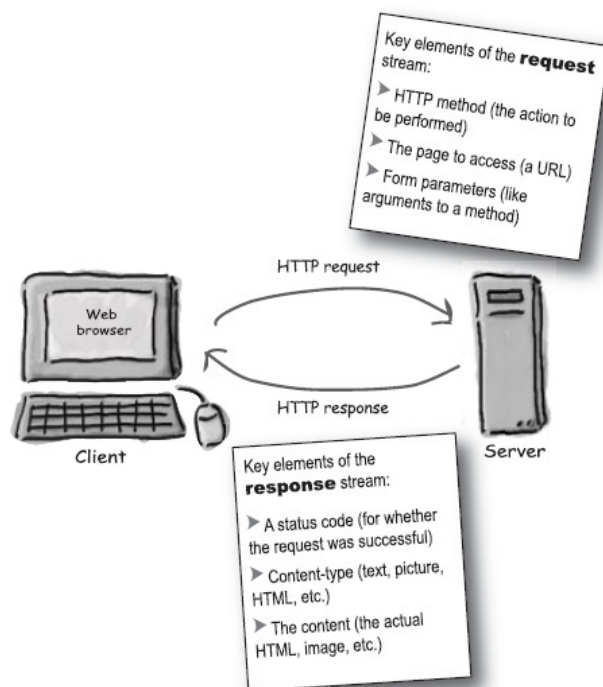
Aplicaciones web en Python

Fernando Berzal, berzal@acm.org

Aplicaciones web



Funcionamiento de una aplicación web



Aplicación web con Flask



```
from flask import Flask
```

```
app = Flask(__name__)
```

```
@app.route('/')
```

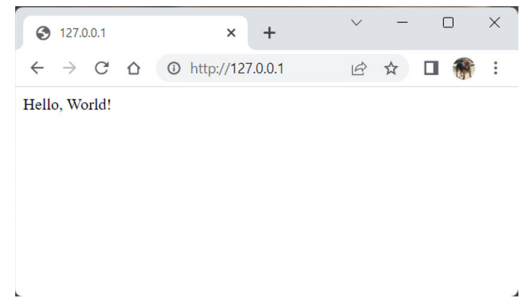
```
def hello_world():
```

```
    return 'Hello, World!'
```

```
if __name__ == '__main__':
```

```
    # Ejecución local de la aplicación
```

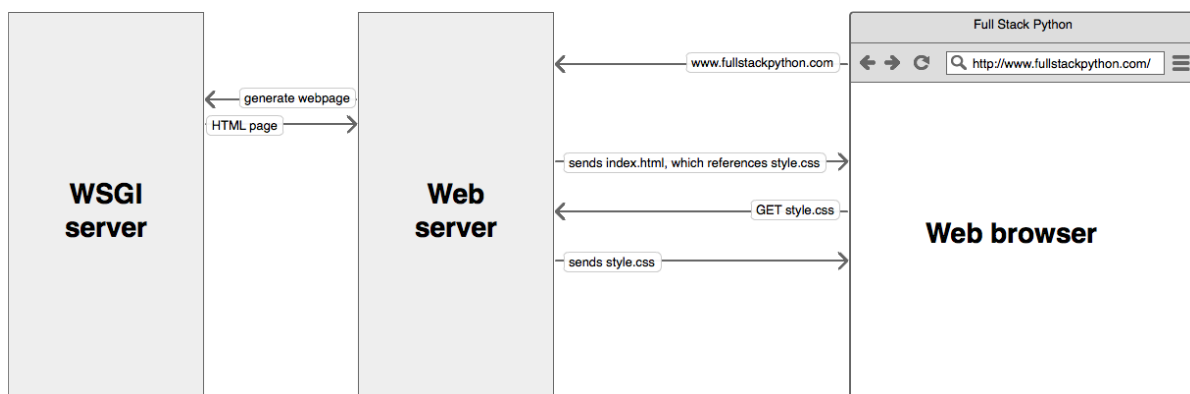
```
    app.run(host='0.0.0.0', port=80, debug=True)
```



NOTA: Así probamos nuestra aplicación localmente con el servidor HTTP interno de Flask. En producción nunca lo haríamos así, utilizaríamos un servidor WSGI como gunicorn o waitress...



Aplicación web con Flask



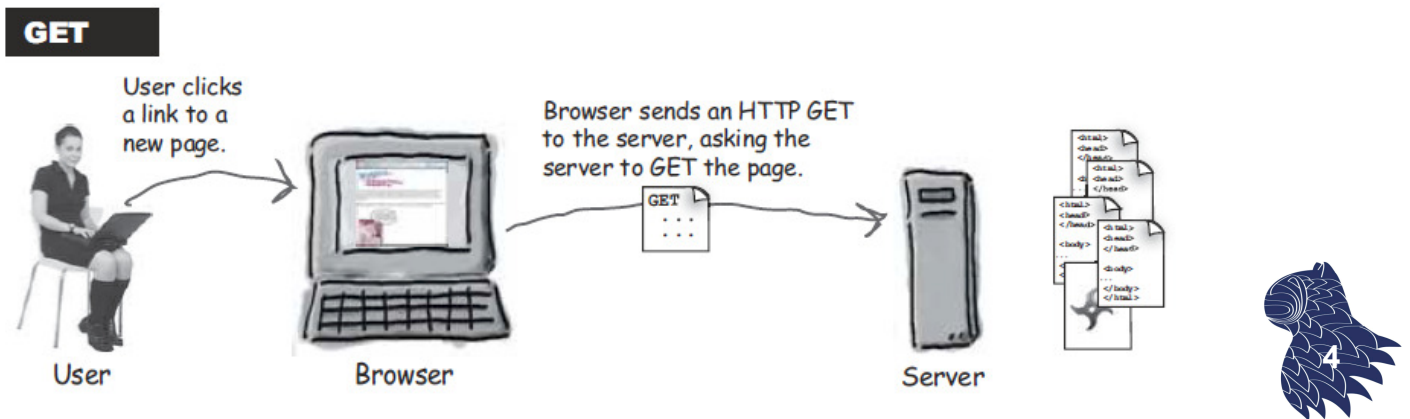
Solicitudes HTTP



Solicitud HTTP GET

```
@app.route("/page", methods=["GET"])  
def view_page ():
```

...



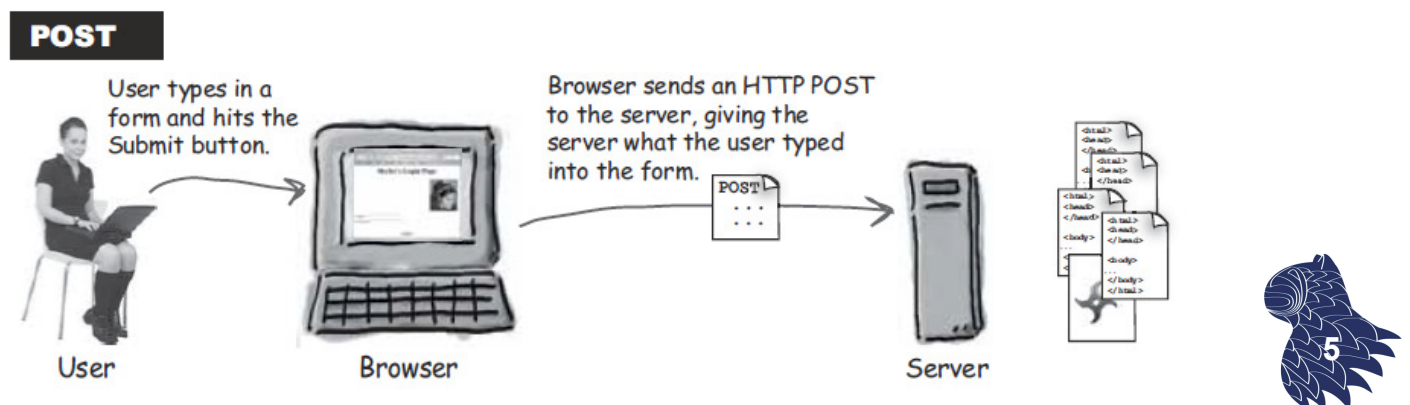
Solicitudes HTTP



Solicitud HTTP POST

```
@app.route("/form", methods=["POST"])  
def process_form ():
```

...



Solicitudes HTTP



Parámetros de una solicitud HTTP

p.ej. HTTP GET

<https://isi.appspot.com/hello?clave1=valor1&clave2=valor2>

Desde la aplicación web, se reciben como cadenas...

```
from flask import request
request.args.get('clave1', default='valor_por_defecto_1')
request.args.get('clave2', default='valor_por_defecto_2')
```

NOTA:

Igual para los parámetros de las solicitudes HTTP POST.



Solicitudes HTTP



Redirecciones

```
from flask import redirect, url_for
```

En el cliente [HTTP redirect]...

```
@app.route('/')
def hello():
    return redirect(url_for('target'), code=302)
```

En el servidor [request dispatch]...

```
@app.route('/')
def hello():
    another_endpoint ()
```



Datos en una aplicación web



Ámbito de los datos en una aplicación web

- Aplicación: Variables globales (no recomendadas) y bases de datos compartidas entre usuarios.
- Sesión: Variables de sesión (`session`)

```
@app.route('/')
def index():
    if 'username' in session:
        username = session['username']
        return 'Logged in as ' + username
    return "You are not logged in"
```



Plantillas Jinja



Para no incluir HTML en el código en Python...

```
<html>
  <head><title>Hello, World!</title></head>
  <body>
    Hello, {{username}}!
  </body>
</html>
```

```
@app.route('/')
def index():
    if 'username' in session:
        return render_template("templates/home.html",
                               username = session['username'] )
    return redirect(url_for('login'))
```



Plantillas Jinja



Modularización y estructuras de control en las plantillas:

```
{% extends "base.html" %}
{% block content %}

<h2>{% block title %} {{title}} {% endblock %}</h2>

<table>
  <tr><th>Artículo</th><th>Descripción</th></tr>
  {% for item in collection %}
  <tr>
    <td>{{item["name"]}}</td>
    <td>{{item["description"]}}</td>
  </tr>
  {% endfor %}
</table>

{% endblock %}
```



Plantillas Jinja



Plantilla base:

```
<!doctype html>
<html lang="en">
  <head>
    <title>{% block title %} {% endblock %}</title>
  </head>
  <body>
    <!-- Encabezado común -->
    ...

    <div>{% block content %} {% endblock %}</div>

    <!-- Pie de página -->
    ...
  </body>
</html>
```



Ejemplo



Clases en Python



app.py

```
class Book:
    def __init__(self, isbn, title, authors, publisher, date):
        self.isbn = isbn
        self.title = title
        self.authors = authors
        self.publisher = publisher
        self.date = date
        self.image = f'http://images.amazon.com/images/P/{isbn}.01._SCLZZZZZZZ_.jpg'
        self.offers = []
```

```
class BookOffer:
    def __init__(self, provider, url, price):
        self.provider = provider
        self.url = url
        self.price = price
```



Ejemplo

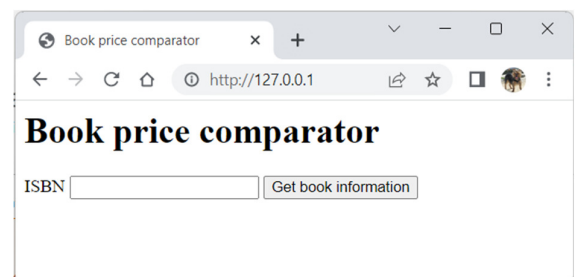


Formulario HTML



form.html

```
<html>
<head>
  <title>Book price comparator</title>
</head>
<body>
  <h1>Book price comparator</h1>
  <form action="/book">
    <label>ISBN
    <input type="text" name="isbn">
    </label>
    <input type="submit" value="Get book information">
  </form>
</body>
</html>
```



Ejemplo



Aplicación Flask

```
from flask import Flask, request, render_template

app = Flask(__name__, template_folder='./templates')

@app.route('/')
def search_form():
    return render_template("form.html")

@app.route('/book')
def display_book():
    isbn = request.args.get("isbn", default="1731265387")
    book = search_by_isbn(isbn)
    return render_template("book.html", book=book)
```



app.py



Ejemplo



Visualización de datos: Plantilla Jinja

```
<html>
<head>
  <title>Book price comparator</title>
</head>
<body>

  <h2>{{book.title}}</a></h2>
  <h3>
    {% for author in book.authors %}
      {{author}} <br/>
    {% endfor %}
  </h3>
```

...



book.html



Ejemplo



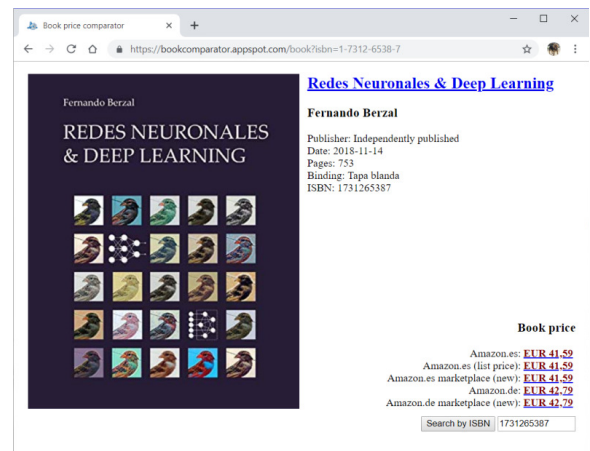
Visualización de datos: Plantilla Jinja

```
...
Publisher: {{book.publisher}} <br/>
Date: {{book.date}} <br/>
ISBN: {{book.isbn}} <br/>

<h3>Book price</h3>

{% for offer in book.offers %}
  {{offer.provider}}:
  <a href="{{offer.url}}"><b>{{offer.price}}</b></a>
<br/>
{% endfor %}

</body>
</html>
```



Frameworks



- Flask
- Django
- CherryPy
- Masonite
- Hug
- TurboGears
- Bottle
- CubicWeb
- Falcon
- Web2Py
- MorePath
- Giotto
- Pycnic
- Pylons
- Dash
- Pyramid

django



WEB2PY



