

# Desarrollo de aplicaciones web con Python y Django

De cero a ciento y pico en poco tiempo

Colmux 2011

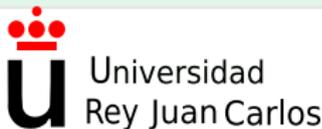
Jesús M. González Barahona

jgb@gsyc.es

<http://identi.ca/jgbarah> <http://twitter.com/jgbarah>

GSyC/LibreSoft, Universidad Rey Juan Carlos

Colmenar Viejo, 17 de marzo de 2011



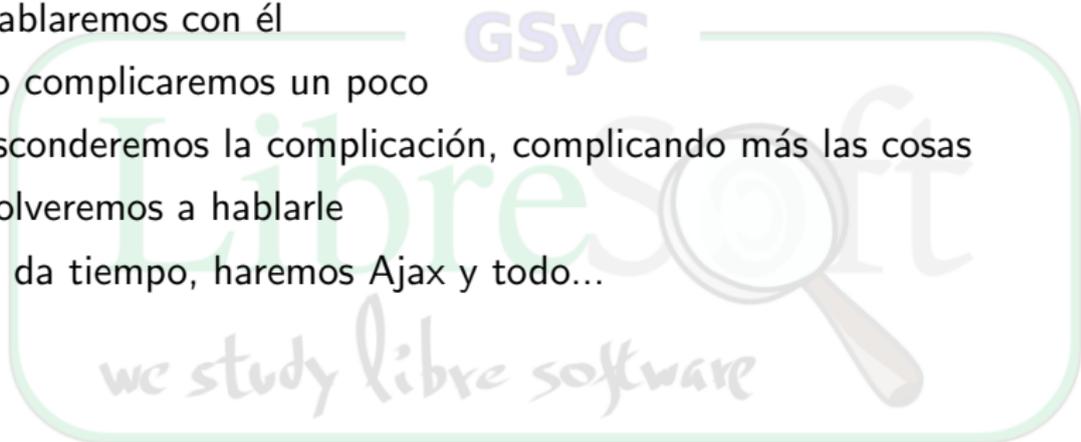
GSyC

©2011 Jesús M. González Barahona.  
Algunos derechos reservados.  
Este artículo se distribuye bajo la licencia  
“Reconocimiento-CompartirIgual 3.0 España” de Creative  
Commons,  
disponible en

[http:  
//creativecommons.org/licenses/by-sa/3.0/es/deed.es](http://creativecommons.org/licenses/by-sa/3.0/es/deed.es)

# ¿En qué vamos a malgastar la próxima hora?

- Haremos un servidor web sencillito
- Hablaremos con él
- Lo complicaremos un poco
- Escondremos la complicación, complicando más las cosas
- Volveremos a hablarle
- Si da tiempo, haremos Ajax y todo...



# Un servidor web muuuuuuy sencillito

- `servidor-http-simple.py`

```
import socket
mySocket = socket.socket(socket.AF_INET,
                          socket.SOCK_STREAM)
mySocket.bind(('localhost', 1234))
mySocket.listen(5)
while True:
    (recvSocket, address) = mySocket.accept()
    print recvSocket.recv(1024)
    recvSocket.send("HTTP/1.1 200 OK\r\n\r\n" +
                    "<html><body><h1>Hello!</h1></body></html>" +
                    "\r\n")
    recvSocket.close()
```

- Para ejecutarlo:

```
python servidor-http-simple.py
```

# ¿Cómo hablamos con él?

- Pues desde un navegador... (probamos con Firefox)
- O desde un plugin específico (probamos Web Developer de Firefox)
- Pero también podemos ver qué nos dice
- O hablar HTTP “a pelo” (desde un telnet)  
`telnet localhost 1234`
- O pedirle al amigo curl que hable por nosotros  
`curl http://localhost:1234/`
- O al amigo wget  
`wget http://localhost:1234/`
- Y podemos espiar cualquier conversación con wireshark...

- servidor-http-random.py

```
import socket
import random
mySocket = socket.socket(socket.AF_INET,
                          socket.SOCK_STREAM)
mySocket.bind(('localhost', 1234))
mySocket.listen(5)
random.seed()
while True:
    (recvSocket, address) = mySocket.accept()
    print recvSocket.recv(1024)
    nextPage = str (random.randint (0,10000))
    nextUrl = "http://localhost:"+str(myPort)+"/"+nextPage
    htmlBody = "<p>Next page: <a href=" + \
        + nextUrl + '>' + nextPage + "</a></p>"
    recvSocket.send("HTTP/1.1 200 OK \r\n\r\n" +
        "<html><body>" + htmlBody + "</body></html>" +
        "\r\n")
    recvSocket.close()
```

# Y ahora con Django 1.2.3

```
cd myproject  
python manage.py runserver 1234
```

- Una aplicación realmente simple: myproject  
http://localhost:1234/  
http://localhost:1234/hello  
http://localhost:1234/bye/Maria  
http://localhost:1234/number/3
- Un gestor de contenidos bien simple: cms\_users

# Y el no va más: ¡un poco de Ajax!

- Una página que genera frases:  
`sentences_generator.html`
  - JavaScript encastrado en el HTML
  - Se ejecuta en el navegador al cargar la página
- Una aplicacioncilla Ajax (bien simple):  
`words_provider`  
`http://localhost:1234/apps/sentences_generator.html`
  - Sevidor Django que sirve:
    - Una página HTML con JavaScript encastrado
    - Un canal con partes de frases
  - Se ejecuta el JavaScript en el navegador
  - Pide partes de frases al servidor
  - Muestra el resultado

- Python: <http://python.org>
- Django: <http://djangoproject.com>
- Para aprender Python (por ejemplo):  
Dive into Python  
<http://diveintopython.org>
- Para aprender Django (por ejemplo):  
The Django Book  
<http://djangobook.com/>
- “Tecnologías y Servicios para Aplicaciones en Internet”,  
asignatura con materiales libremente accesibles,  
su temario incluye el contenido de esta presentación

<http://docencia.etsit.urjc.es/moodle/course/view.php?id=3>