

WEB Platform - Client-Server constraint

CS2B01 - Desarrollo Basado en Plataformas - Unidad 2

Dr. Jesus Bellido
jbellido@utec.edu.pe

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Logros

Logro de la Unidad

Al finalizar esta unidad usted estará en la capacidad de:

- Diseñar e implementar una aplicación web sencilla
- Describir las restricciones que la web pone a los desarrolladores.
- Comparar y contrastar la programación web con la programación de proposito general
- Describir las diferencias entre software como un servicio y productos de software tradicionales.
- Discutir cómo los estándares de web impactan el desarrollo de software.
- Revisar una aplicación web existente con un estándar web actual.

Logro de esta Sesión

Al finalizar esta unidad usted estará en la capacidad de:

- Diseñar e implementar una aplicación web sencilla
- Describir las restricciones que la web pone a los desarrolladores.
- Comparar y contrastar la programación web con la programación de proposito general
- Describir las diferencias entre software como un servicio y productos de software tradicionales.
- Discutir cómo los estándares de web impactan el desarrollo de software.
- Revisar una aplicación web existente con un estándar web actual.

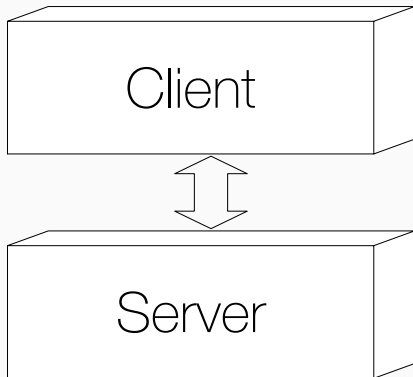
Introducción

1,250,190,025

Websites online right now

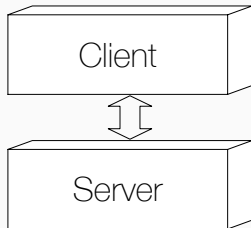
- <http://www.internetlivestats.com/total-number-of-websites/>

Client-Server constraint

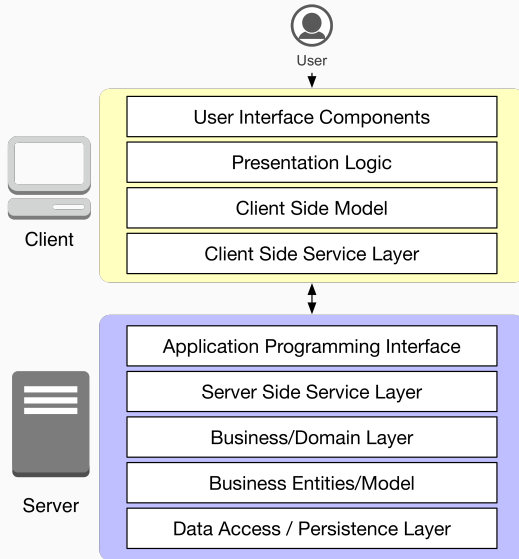


Group Work

En grupos de 4 personas defina de forma gráfica cómo se dividen las responsabilidades entre el cliente el servidor.

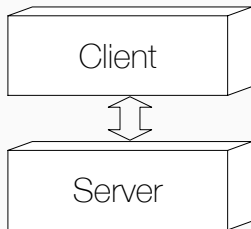


Client-Server Concerns

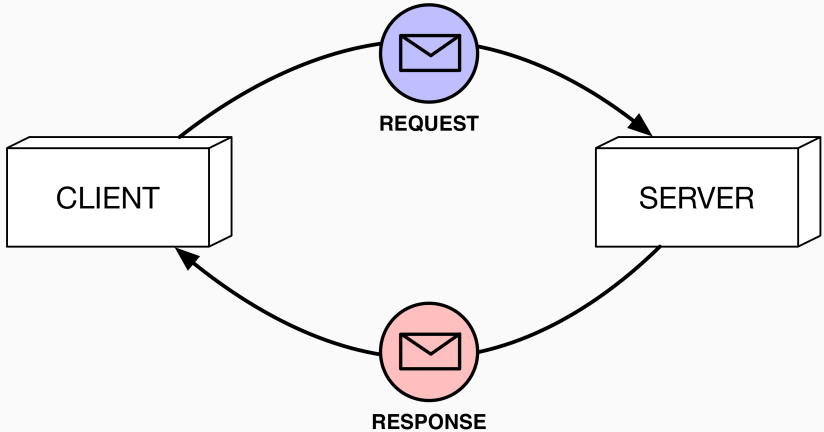


Group Work

En grupos de 4 personas defina de forma gráfica cómo ocurre la comunicación entre el cliente y el servidor.

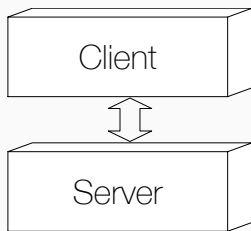


Client-Server Communication



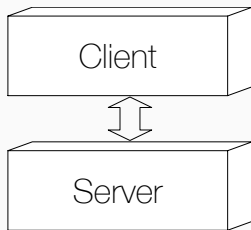
Ventajas

Figure 1: *Cliente-Servidor*



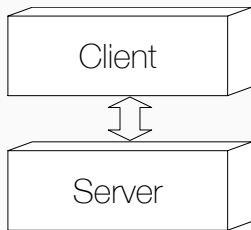
- **MODULARIDAD:** Separación de Temas

Figure 1: *Cliente-Servidor*



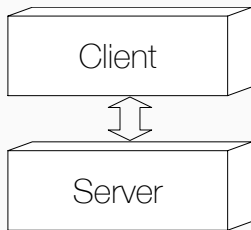
- **MODULARIDAD:** Separación de Temas
- **COMUNICACIÓN SINCRÓNICA:** request - wait - answer

Figure 1: *Cliente-Servidor*



- **MODULARIDAD:** Separación de Temas
- **COMUNICACIÓN SINCRÓNICA:** request - wait - answer
- **IMPLANTACIÓN FLEXIBLE:** capas distribuídas, heterogéneas, etc.

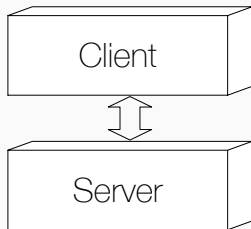
Figure 1: *Cliente-Servidor*



- **MODULARIDAD:** Separación de Temas
- **COMUNICACIÓN SINCRÓNICA:** request - wait - answer
- **IMPLANTACIÓN FLEXIBLE:** capas distribuidas, heterogéneas, etc.
- **ESCALABLE:** Gran número de clientes, soluciones concurrentes, requiere poco tiempo de proceso.

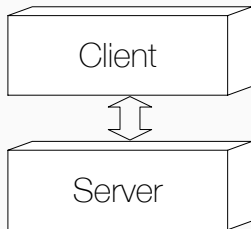
Desventajas

Figure 2: *Cliente-Servidor*



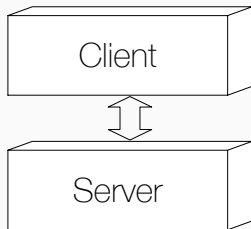
- MAYOR COMPLEJIDAD

Figure 2: *Cliente-Servidor*



- MAYOR COMPLEJIDAD
- CONGESTIÓN DE LA RED

Figure 2: *Cliente-Servidor*



- MAYOR COMPLEJIDAD
- CONGESTIÓN DE LA RED
- MENOS ROBUSTO

Implementación

```
1 import http.server
2 import socketserver
3
4 PORT = 8000
5
6 Handler = http.server.SimpleHTTPRequestHandler
7
8 with socketserver.TCPServer(("", PORT), Handler) as
   httpd:
9     print("serving at port", PORT)
10    httpd.serve_forever()
```

```
1 import http.client
2
3 conn = http.client.HTTPSConnection("localhost",
4                                     8080)
5 conn.set_tunnel("www.python.org")
6 conn.request("HEAD", "/index.html")
```


Implementación

```
1 from flask import Flask
2
3 app = Flask(__name__)
4 @app.route("/")
5 def main():
6     return "Holaaa!"
7
8 if __name__ == "__main__":
9     app.run()
```

```
1 from flask import Flask, render_template
2
3 app = Flask(__name__)
4 @app.route("/")
5 def main():
6     return render_template('login.html')
7
8 if __name__ == "__main__":
9     app.run()
```

1. Logros
2. Introducción
3. Client-Server constraint
4. Ventajas
5. Desventajas
6. Implementación
7. Implementación

Preguntas