

WEB Platform - Stateless/Stateful 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.

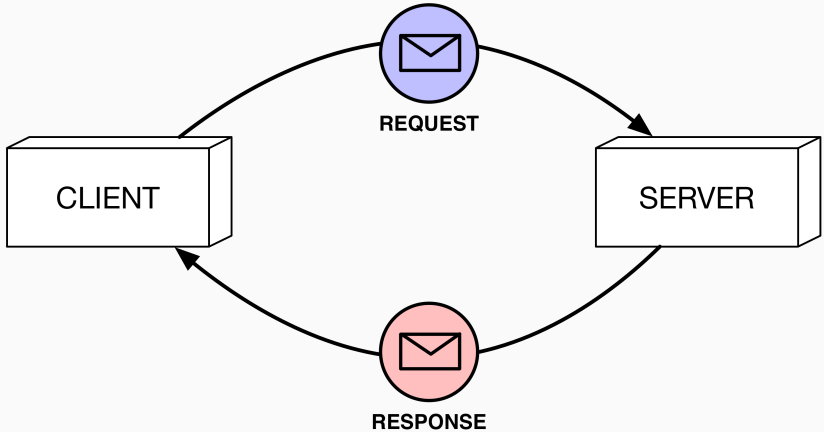
Logro de esta Sesión

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.

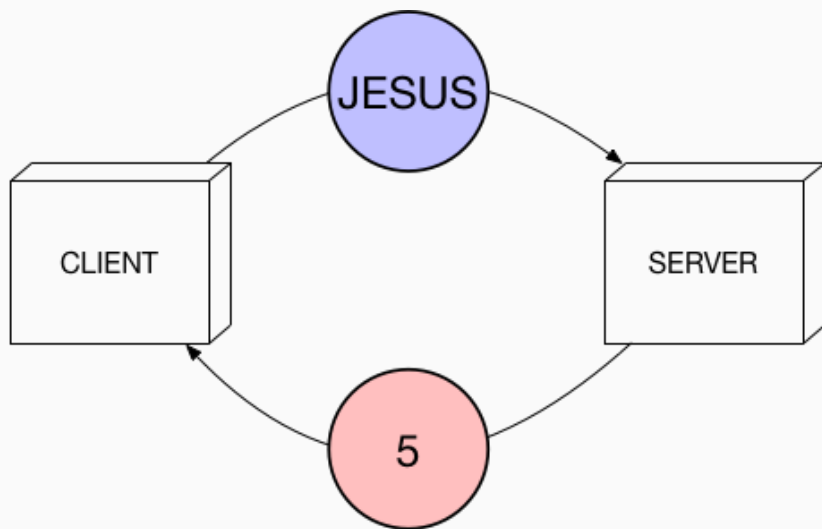
Introducción

Client-Server Communication



Group Work

- En grupos de 4 personas, junto a tus compañeros elige a una persona que sea el *servidor* y el resto del equipo serán los *clientes*.
- El servidor tiene la misión de calcular el total de caracteres de tu nombre.
- Encuentren la forma de comunicación entre el cliente y el servidor

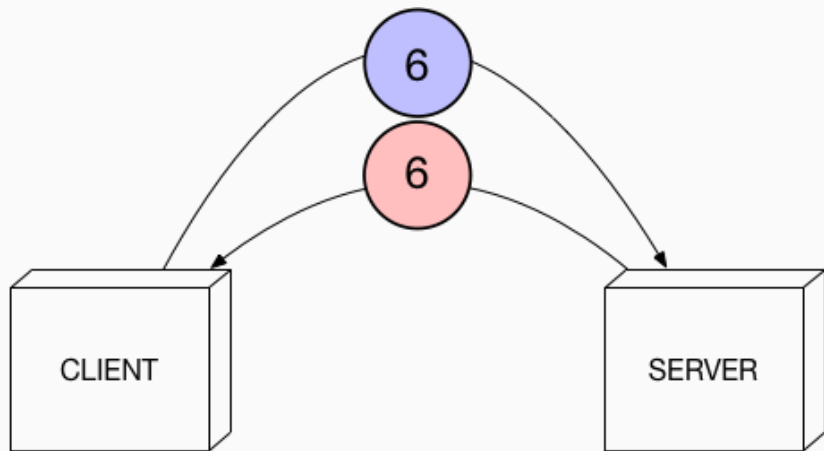


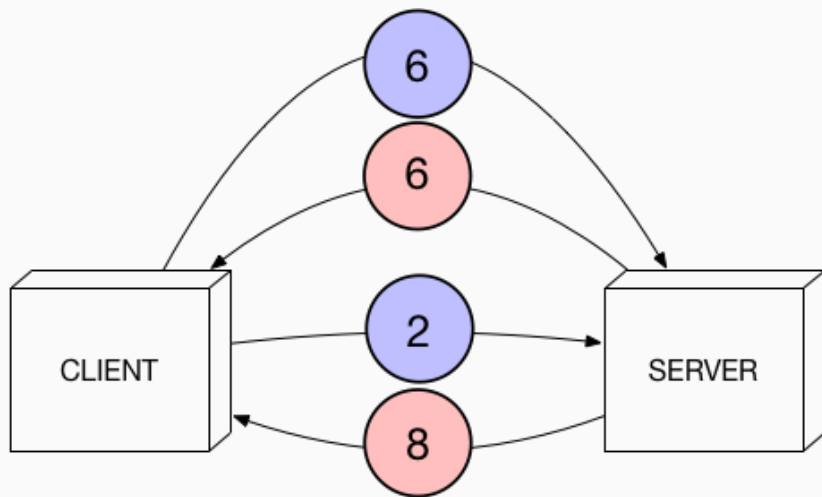
Client Server Interaction

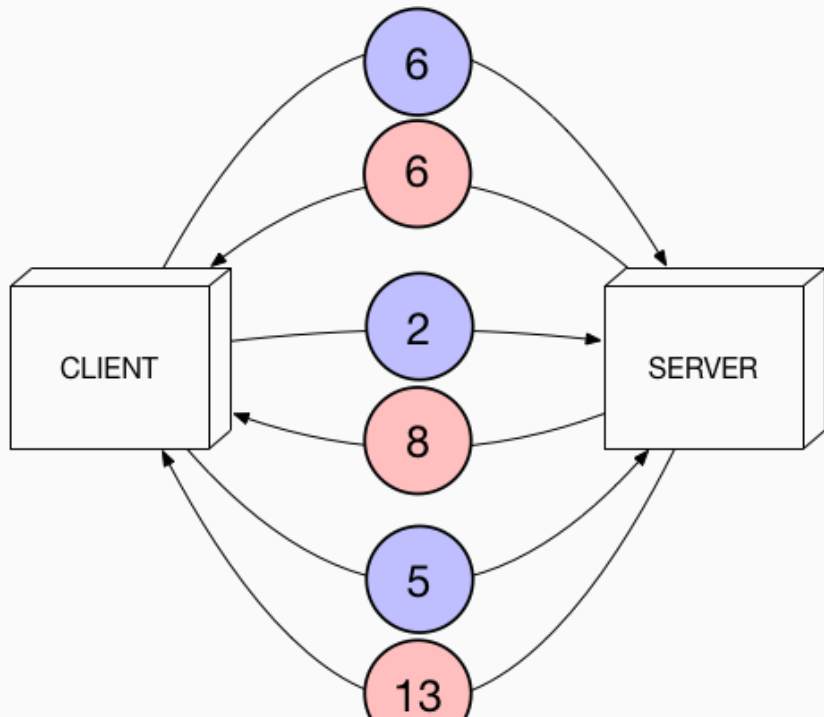
```
1 from flask import Flask, session
2
3 app = Flask(__name__)
4
5 @app.route('/cuantasletras/<nombre>')
6 def cuantas_letras(nombre):
7     return str(len(nombre))
8
9 if __name__ == '__main__':
10     app.run()
```

Group Work

- En grupos de 4 personas, junto a tus compañeros elige a una persona que sea el *servidor* y el resto del equipo serán los *clientes*.
- El servidor tiene la misión de ir sumando lo que el cliente le va pidiendo progresivamente.
- Encuentren la forma de comunicación entre el cliente y el servidor



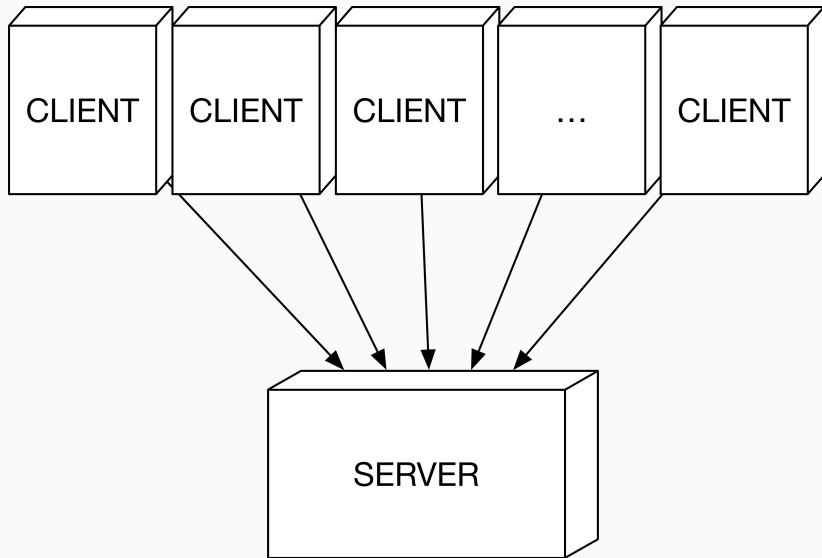




Client Server Interaction

```
1 from flask import Flask, session
2
3 app = Flask(__name__)
4 app.secret_key = 'You Will Never Guess'
5
6 @app.route('/sumar/<numero>')
7 def sumar(numero):
8     if 'suma' not in session:
9         session['suma'] = "0"
10
11     session['suma'] = int(session['suma']) + int(
12         numero)
13     return str(session['suma'])
14
15 if __name__ == '__main__':
16     app.run()
```

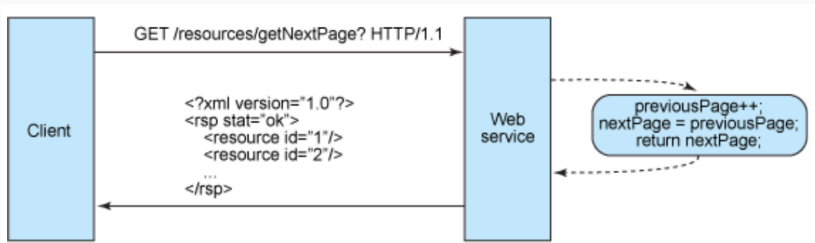
How many clients can servers handle at a time?



Estado de Sesión

Estado de Sesión en el Servidor (stateful)

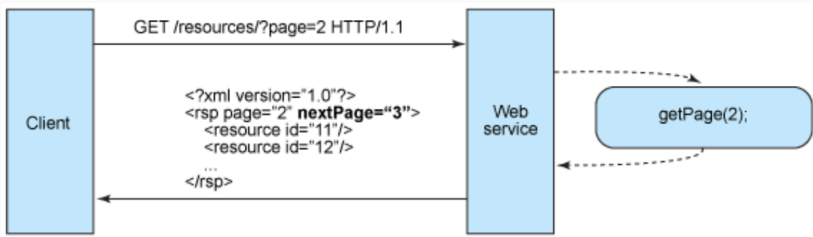
Figure 1: Patrón Estado de Sesión en el Servidor



- Mantiene en el **servidor** el estado de la interacción entre dos componentes.

Estado de Sesión en el Cliente (stateless)

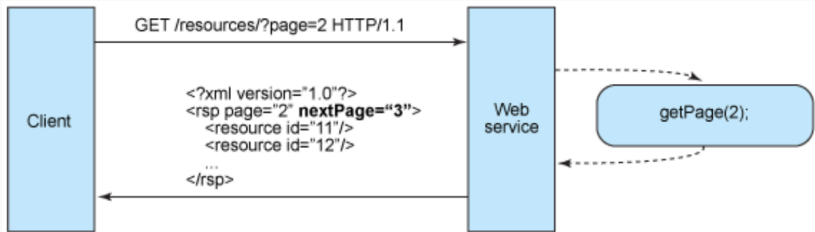
Figure 2: Patrón Estado de Sesión en el Cliente



- Mantiene en el **cliente** el estado de la interacción entre dos componentes.

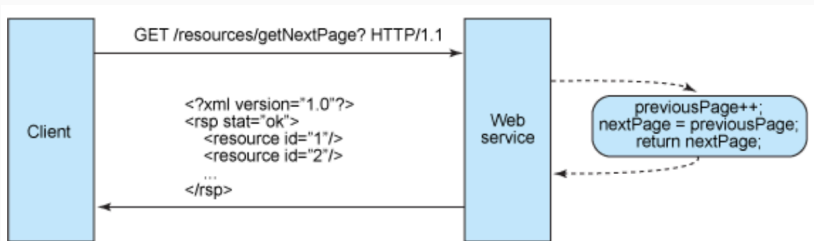
Ventajas

Stateless : Ventajas



- **ESCALABILIDAD:** No ocupa memoria del servidor para almacenar variables de sesión.

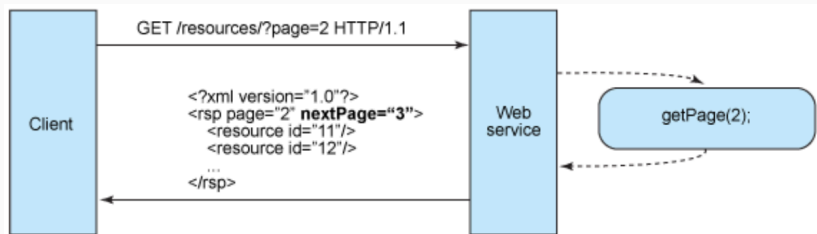
Stateful : Ventajas



- **LATENCIA:** Reduce el tamaño del mensaje.

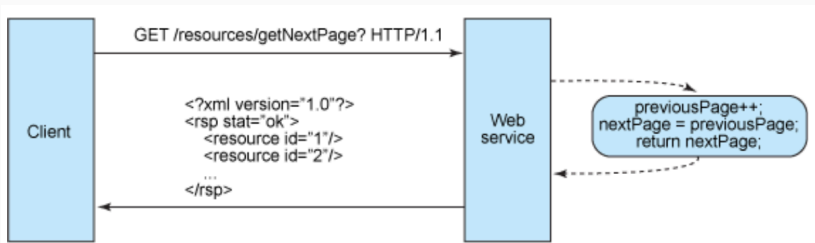
Desventajas

Stateless : Desventajas



- LATENCIA Aumenta el tamaño del mensaje.

Stateful : Desventajas



- **ESCALABILIDAD** Usa la memoria del servidor para almacenar variables de sesión.

Implementación

Stateful

```
1 from flask import Flask, session
2
3 app = Flask(__name__)
4 app.secret_key = 'You Will Never Guess'
5
6 @app.route('/setuser/<user>')
7 def setuser(user):
8     session['user'] = user
9     return 'User value set to: ' + session['user']
10
11 @app.route('/getuser')
12 def getuser():
13     return 'User value was previously set to: ' +
14         session['user']
15
16 if __name__ == '__main__':
17     app.run()
```

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/say_hello/<user>')
6 def hello(user):
7     return 'Hello ' + user
8
9 if __name__ == '__main__':
10     app.run()
```

1. Logros
2. Introducción
3. Estado de Sesión
4. Ventajas
5. Desventajas
6. Implementación

Preguntas