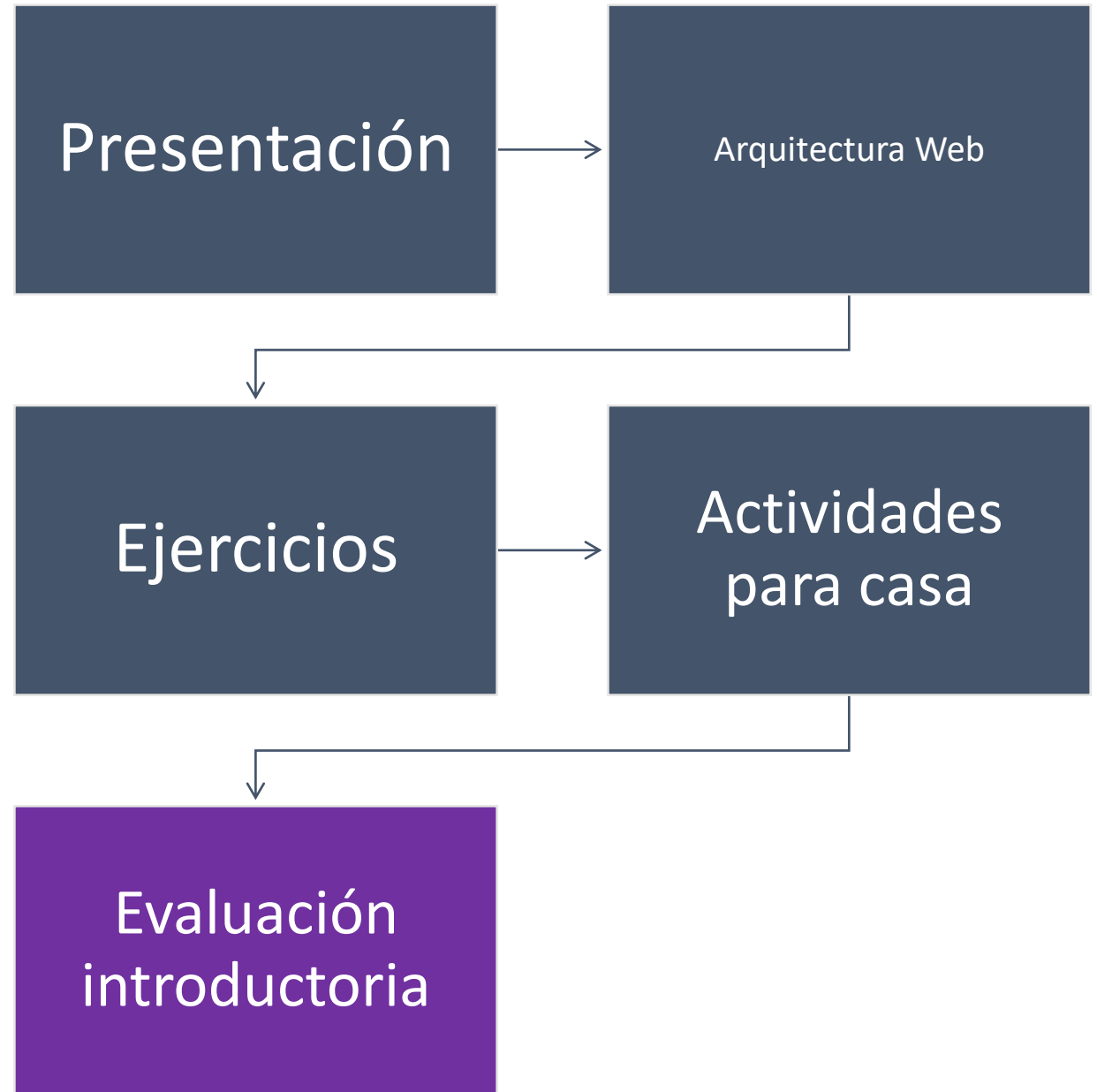




# S01 – Arquitectura Web

Ing. Jonathan Chavez  
Programación I

# Agenda de la semana

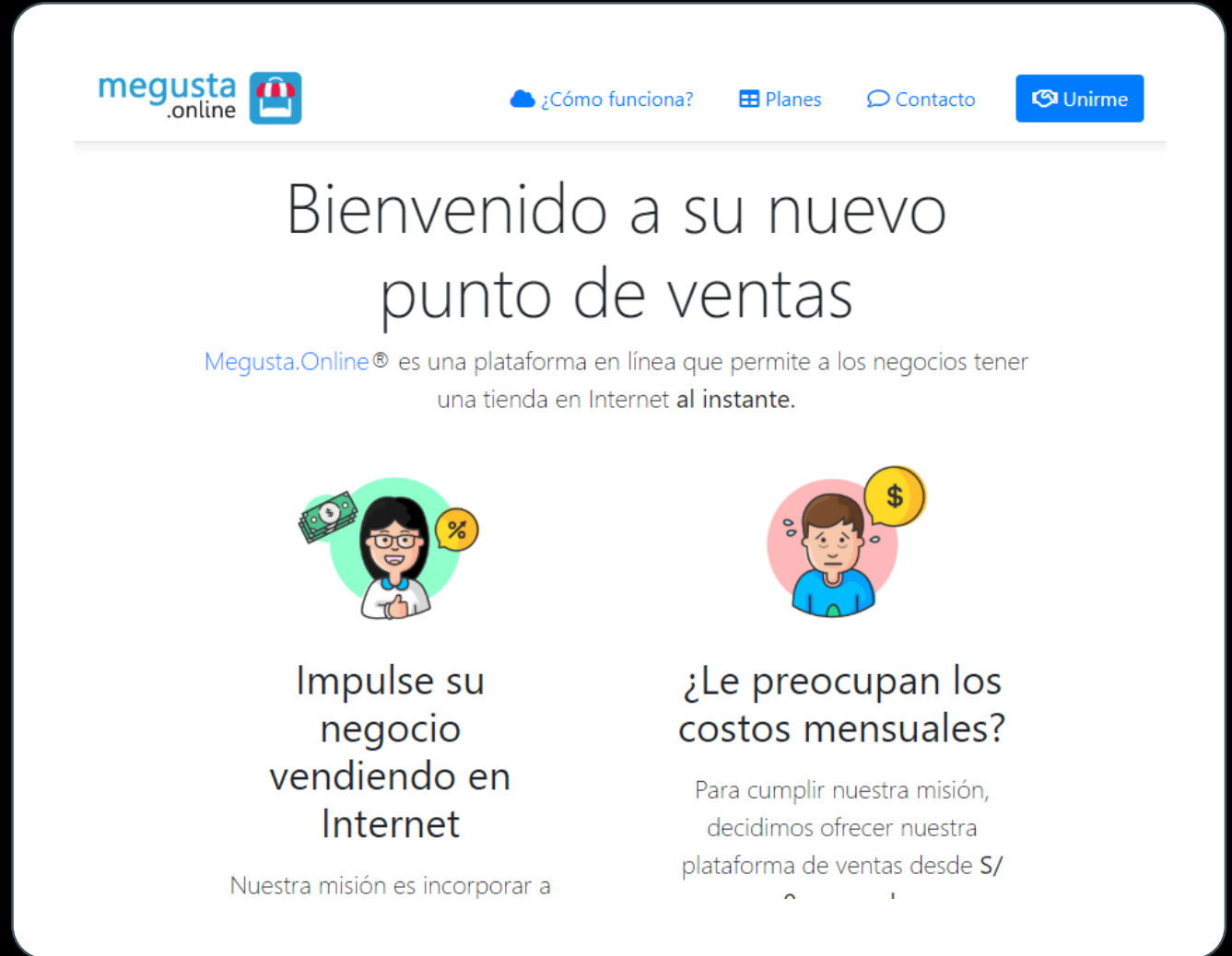


# Presentación

- Nombre
- Ciclo
- Dato interesante
- Experiencia previa en programación

# ¿De qué trata el curso?

- El curso está dirigido a que el estudiante adquiera conocimientos y técnicas necesarias para el desarrollo de aplicaciones multiplataforma orientas a Web



# Unidades del curso

Introducción a  
arquitectura de  
aplicaciones Web



Semana 1, 2, 3, **4**

Implementación  
de aplicaciones  
multiplataforma



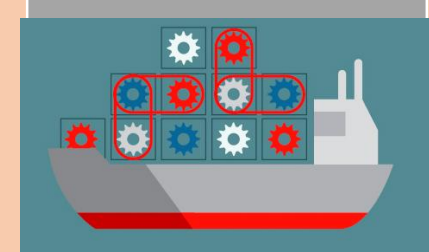
Semana 5, 6, 7, **8**

Patrones de  
diseño



Semana 9, 10, 11, **12**

Tendencias en  
arquitectura para  
la Web



Semana 13, 14, **15, 16**

# Evaluaciones

- Promedio de Laboratorios =  $(L1 + L2 + L3 + L4) / 4$
- Promedio de Prácticas =  $(P1 + P2 + P3 + P4 - MN) / 3$
- Promedio Evaluaciones =  $(\text{Prom. Labs.} + \text{Prom. Prács.} + \text{Proyecto}) / 3$
- **Promedio Final** =  $(2 * \text{Prom. Eval.} + \text{Ex. Parcial} + \text{Ex. Final}) / 4$

# Sugerencias para aprovechar el curso



Puntualidad y asistencia



Participación y  
**bitácora de  
aprendizaje**



Integridad

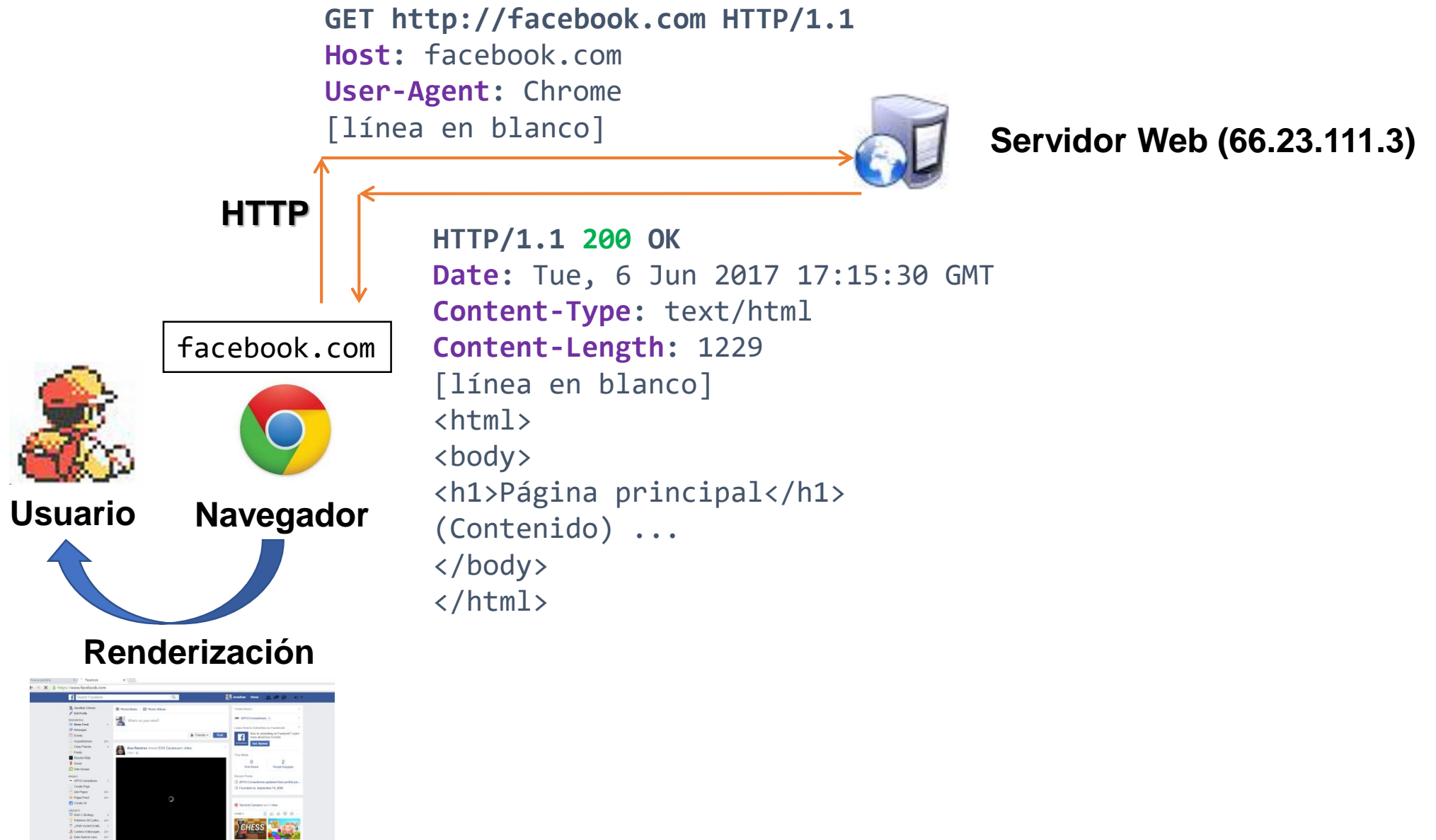


Dudas



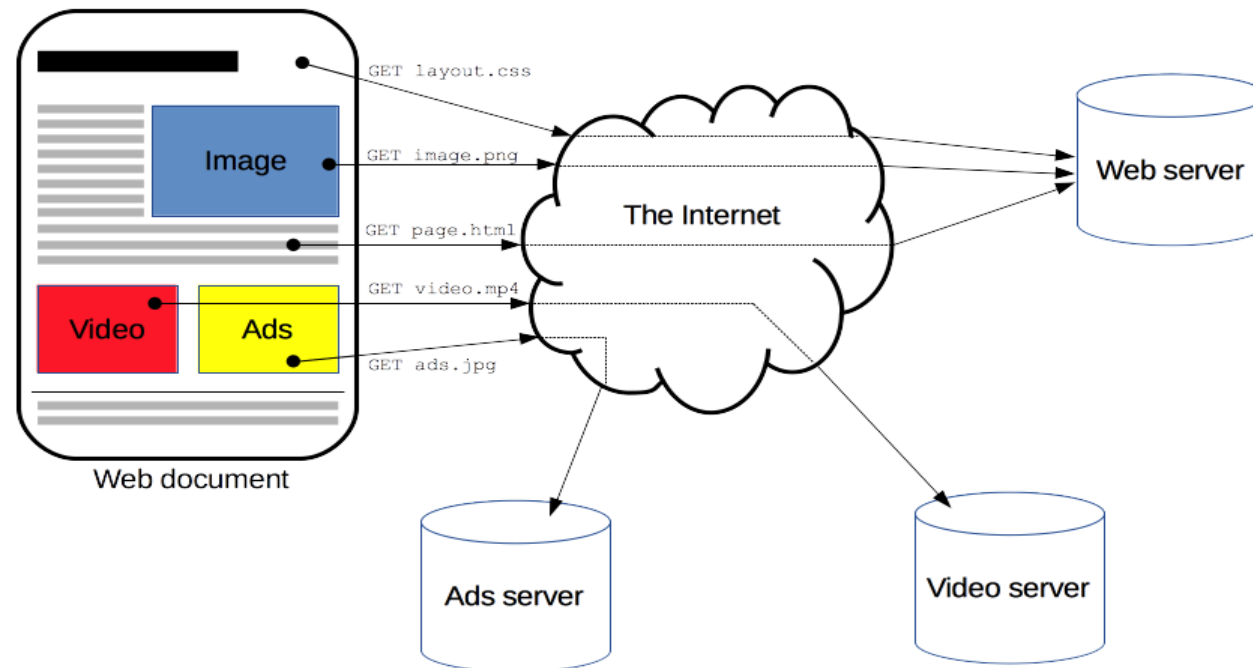


# ¿Cómo navegamos en la Web?



# Protocolo HTTP

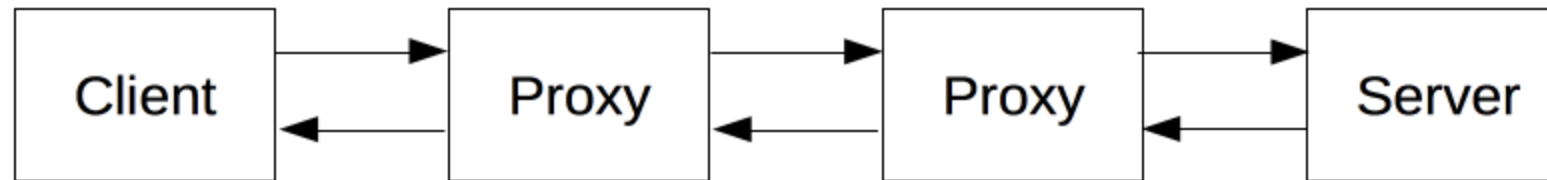
- HTTP es un protocolo que permite la obtención de \_\_\_\_\_ (como documentos html, css, png, ico, etc.) a través de la red.
- Estos últimos son obtenidos a solicitud de un agente (usualmente un navegador Web).
- Una sola página Web puede ser producto de múltiples solicitudes como se muestra en la siguiente figura.



# Protocolo HTTP

- En este protocolo, los clientes y los servidores se comunican a través de un intercambio de mensajes.
- Los mensajes enviados por los clientes (navegadores Web) son llamados                     , mientras que las respuestas enviadas por el servidor son llamados:                     .
- Fue diseñado en 1990s como un protocolo extensible y ha ido evolucionando en el tiempo. Actualmente no solo podemos obtener recursos, sino también, colocar contenido en los servidores a través de los formularios HTML.

# Componentes de sistemas basados en HTTP

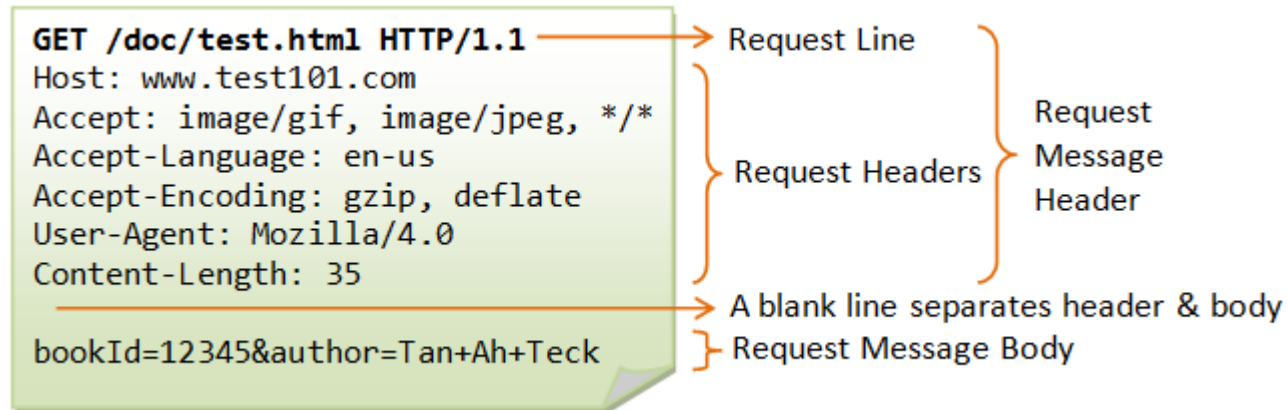


- HTTP es un protocolo cliente-servidor en donde los requests son enviados por una sola entidad llamada **user-agent**. El user-agent es el software que actúa en representación del usuario, siendo la mayoría de veces un navegador Web.
- Al otro extremo, tenemos a los servidores quienes sirven los documentos solicitados por el cliente. Un servidor es representado virtualmente por una máquina, aunque en realidad pueda ser una colección de servidores que compartan la carga.
- Gracias a la versión HTTP/1.1, se introdujo la cabecera           , que permite a diferentes sitios Web compartir la misma dirección IP.
- Cada solicitud independiente es iniciada por el cliente y dirigida hacia el servidor que proporcionará una respuesta. Entre esta solicitud y respuesta, podrían haber numerosas entidades llamadas también: proxies. Estos proxies son transparentes para el protocolo pero pueden servir para las siguientes funciones: caching, filtering, load balancing, logging, etc.

# Principales características de HTTP

## HTTP es simple

- HTTP fue diseñado para ser simple y entendido por humanos. Los mensajes transmitidos en el protocolo son de fácil lectura, proporcionando mecanismos de testeo y depuración.



# Principales características de HTTP

## **HTTP es extensible**

- En su versión 1.0. HTTP introduce el concepto de "cabeceras" (headers) lo que permite al protocolo ser extensible y experimental.
- Nuevas funcionalidades pueden ser introducidas por un simple acuerdo entre cliente y servidor.

# Principales características de HTTP

## **HTTP es stateless, pero no sessionless**

- HTTP es stateless. No hay ningún vínculo entre 2 requests.
- A pesar de esta limitante, la extensibilidad de las cabeceras permite al protocolo implementar mecanismos de sesión a través de "HTTP cookies".

# El flujo HTTP

- Cuando un cliente desea comunicarse con un servidor, suceden los siguientes pasos:

- 1. El cliente abre una conexión TCP.
- 2. El cliente envía un mensaje HTTP (request)
- 3. El servidor responde

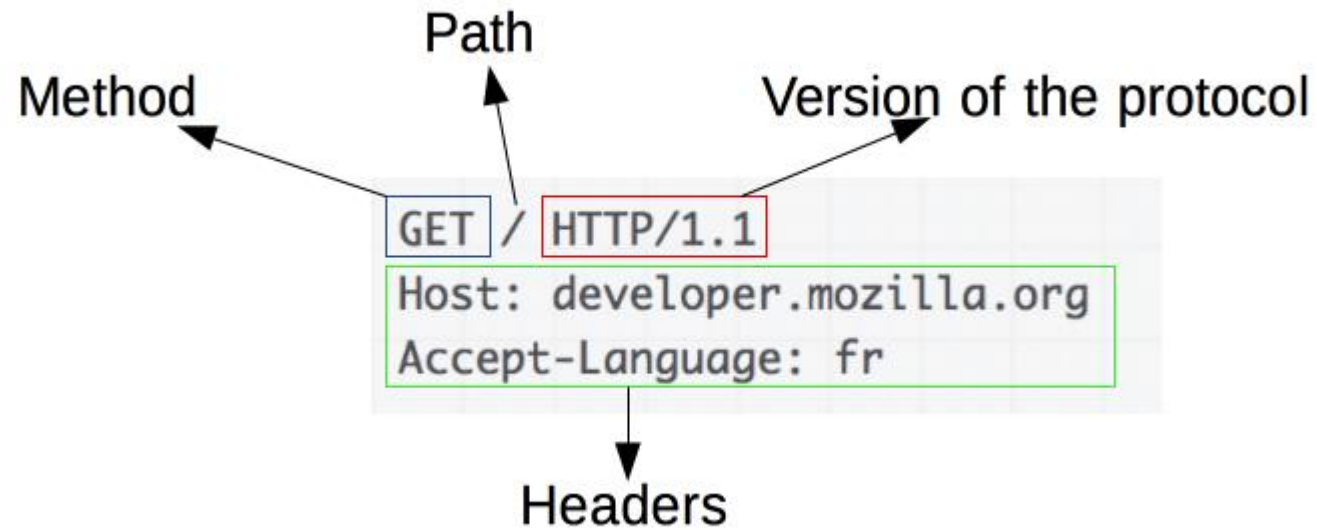
```
1 GET / HTTP/1.1
2 Host: developer.mozilla.org
```

```
1 HTTP/1.1 200 OK
2 Date: Sat, 09 Oct 2010 14:28:02 GMT
3 Server: Apache
4 Last-Modified: Tue, 01 Dec 2009 20:18:22 GMT
5 ETag: "51142bc1-7449-479b075b2891b"
6 Accept-Ranges: bytes
7 Content-Length: 29769
8 Content-Type: text/html
9
10 <!DOCTYPE html... (here comes the 29769 bytes of the requested
```

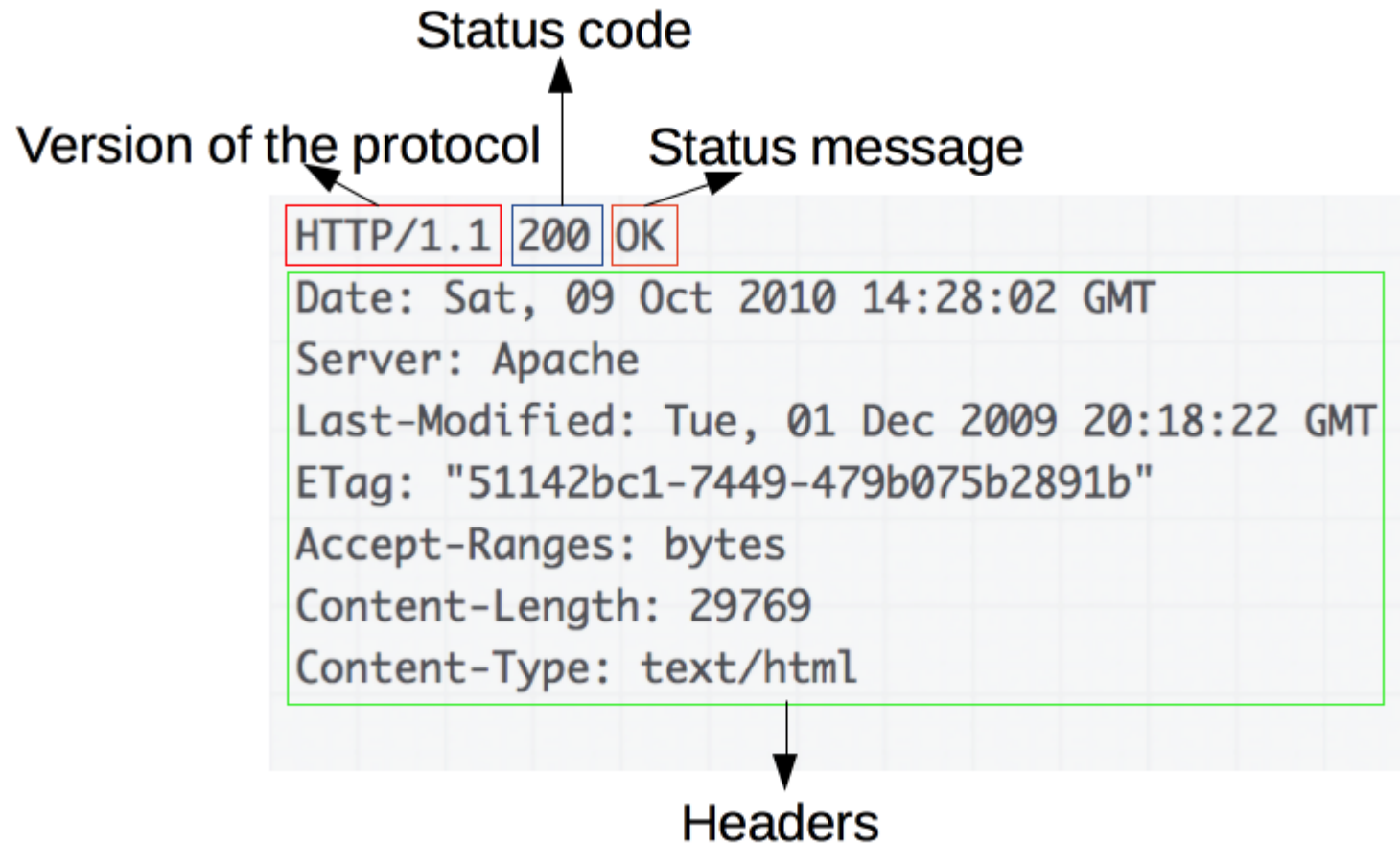
- 4. El cliente cierra o reusa la conexión TCP.



# Analizando las solicitudes



# Analizando las respuestas



# Ejercicio de investigación en clase

- Investigar 4 verbos adicionales a GET y anotar su uso en sus cuadernos.

# Repaso

1. ¿Qué significa HTTP?
2. ¿Cuáles son los componentes de un sistema basado en HTTP?
3. ¿Qué es un user-agent?
4. ¿Qué cabecera permite a diversos sitios Web compartir una misma IP?
5. ¿Cuáles son las 3 principales características del protocolo?
6. ¿Cuáles son los 2 verbos más usados en el protocolo HTTP?

# Actividad para casa (para la bitácora)

1. Investigar sobre la versión 2.0 del protocolo HTTP y comprender las principales ventajas sobre la versión 1.1. Elaborar un resumen ejecutivo en sus cuadernos.

# Evaluación introductoria

# Fuentes

- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>