Arquitecturas de Sistemas WEB

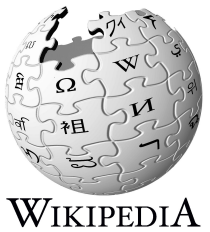
Cliente Servidor

Agenda

- Definiciones: Sistemas - Actividades
- Sistemas WEB
- Arquitecturas
- Arquitectura *Cliente - Servidor*

Sistema de información

Un **sistema de información** (SI) es un conjunto de elementos “digitales” orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una **necesidad** o un **objetivo** de una organización o individuo.



Porque desarrollamos sistemas?

Se crean sistemas para cubrir una **necesidad** o un **objetivo** de una organización o individuo.

Los sistemas se utilizan para:

- tomar decisiones,
- controlar operaciones,
- analizar problemas y facilitar actividades,
- crear nuevos productos o servicios

Actividades de un Sistema de Información

Existen cuatro actividades en un **SI** que producen la información.

Estas actividades son:

1. **Recopilación**: captura o recolecta datos.
2. **Almacenamiento**: guarda de forma estructurada la información recopilada.
3. **Procesamiento**: convierte esa entrada de datos en una forma más significativa (información).
4. **Distribución**: transfiere la información procesada a las personas o roles que la usarán.

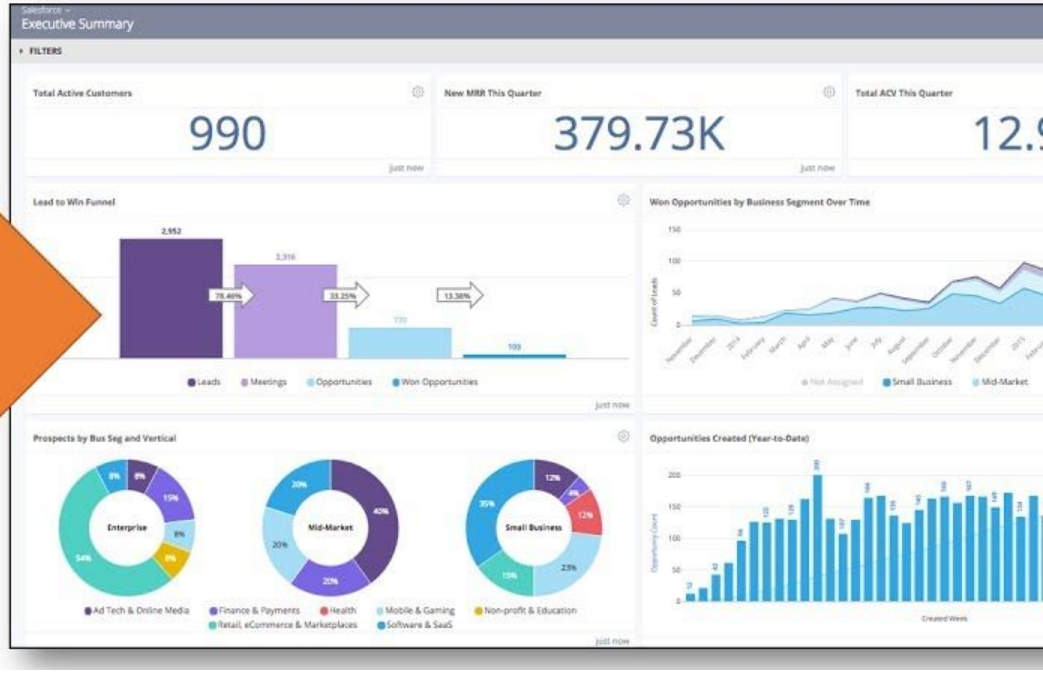
Datos vs información

DATOS != INFORMACIÓN

Data

Information

	sector	tryint
	00nil_Combined_Sector	14625
	00nil_Combined_Sector	10125
	00nil_Combined_Sector	4500
	business	1350
	consumer	3150
	00nil_Combined_Sector	5625
	business	4950
	consumer	675
	00nil_Combined_Sector	4500
	00nil_Combined_Sector	1890
	business	855
	consumer	1035
	00nil_Combined_Sector	2610
	business	1215
	consumer	1395



Sistemas WEB

Un sistema WEB es un sistema diseñado y desarrollado para que funciona a través de **Internet**

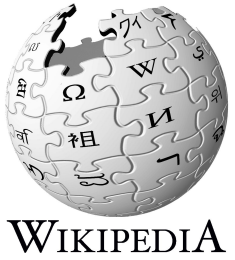
- Están basados en una arquitectura **cliente - servidor.**
- Utilizan tecnologías WEB para entregar información o servicios a otros usuarios o sistemas.

Arquitectura Cliente - Servidor

Arquitectura de un Sistema

La **Arquitectura** del **Software** es el diseño de más alto nivel de la estructura de un sistema.

Una **arquitectura** consiste en un conjunto de **patrones** y **abstracciones** coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del **software.**



Diseño de una Arquitectura

- La **arquitectura** le da la estructura a la aplicación
- Permite analizar y diseñar sin programar los principales problemas que podemos tener en nuestra aplicación
    - No es lo mismo la arquitectura de **WhatsApp** que la de **Dropbox**
- Debe asistir a los servicios/funcionalidad que debe cumplir un sistema (requerimientos funcionales) teniendo en cuenta cuestiones que hacen a la operación (requerimientos no funcionales)

“Otras” arquitecturas

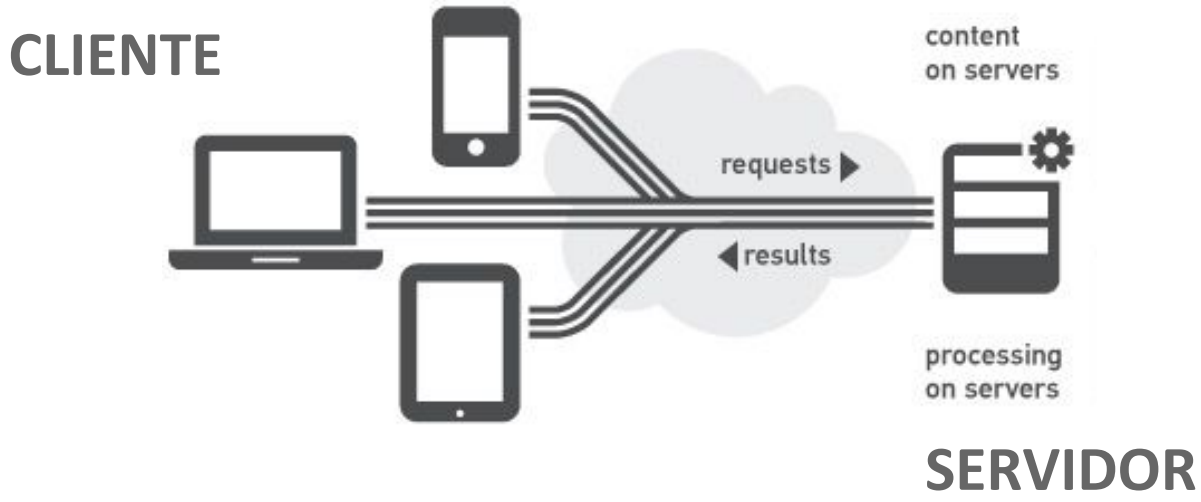
- Arquitectura de la información
  - Todos los conceptos que definen al contexto para el usuario

- Arquitectura física
  - Equipamiento físico donde corren los sistemas (conectividad, servidores, PCs, routers..)

Arquitectura “Cliente - Servidor”

Es la **arquitectura preponderante** en la WEB

En este tipo de interacción, el usuario **(cliente)** realiza **peticiones (http request)** a un programa remoto **(servidor)**, quien le devolverá a cambio una **respuesta (http response)**.



Qué hace un Cliente?

- Es quien inicia las solicitudes **(HTTP REQUEST)**
- Espera y recibe las respuestas del servidor.
- Interactúa, en general, mediante una interfaz gráfica con el usuario (UI/VISTA)

¿Se les ocurren ejemplos?

Qué hace un Servidor?

- Esperan a que lleguen las solicitudes de los clientes (papel pasivo en la comunicación)
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente. **(HTTP RESPONSE)**
- Por lo general, acepta las conexiones de un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado)

¿Se les ocurren ejemplos?

Arquitectura “Cliente-Servidor” WEB

Una típica arquitectura WEB cuenta con:

- Un **navegador**: hace de cliente y realiza peticiones al servidor
- Un **servidor**: administra y responde las peticiones de clientes o de otros servidores
- Una **API**: es la forma se comunican a través de internet
- El **protocolo** HTTP: es el **protocolo** de **comunicación** entre Cliente y Servidor (basado en TCP/IP)

Cliente-Servidor WEB- Introducción

- El concepto es a nivel **software** (programas).
  - Un software es el **cliente**
  - Otro software es el **servidor WEB**
- A nivel **hardware**, una **misma** máquina puede ser cliente y servidor en simultáneo para diferentes programas (o incluso un mismo programa)
- La capacidad de proceso está repartida
- Se separan las responsabilidades

Qué hace un servidor WEB? (I)

Al principio, la forma de trabajar con los servidores web era de la siguiente forma:

1. el navegador hace una petición al servidor mediante http.
2. el servidor transforma la URL en una ruta el sistema de archivos
3. devuelve el archivo al navegador. (para una misma URL el servidor siempre va a devolver el mismo recurso HTML, CSS, JavaScript, imágenes...)

Qué hace un servidor WEB? (II)

Hoy en día, la mayoría de los servidores web permiten que en cada petición se ejecute un programa que genera **dinámicamente** el recurso que se envía al usuario (server-side scripting).

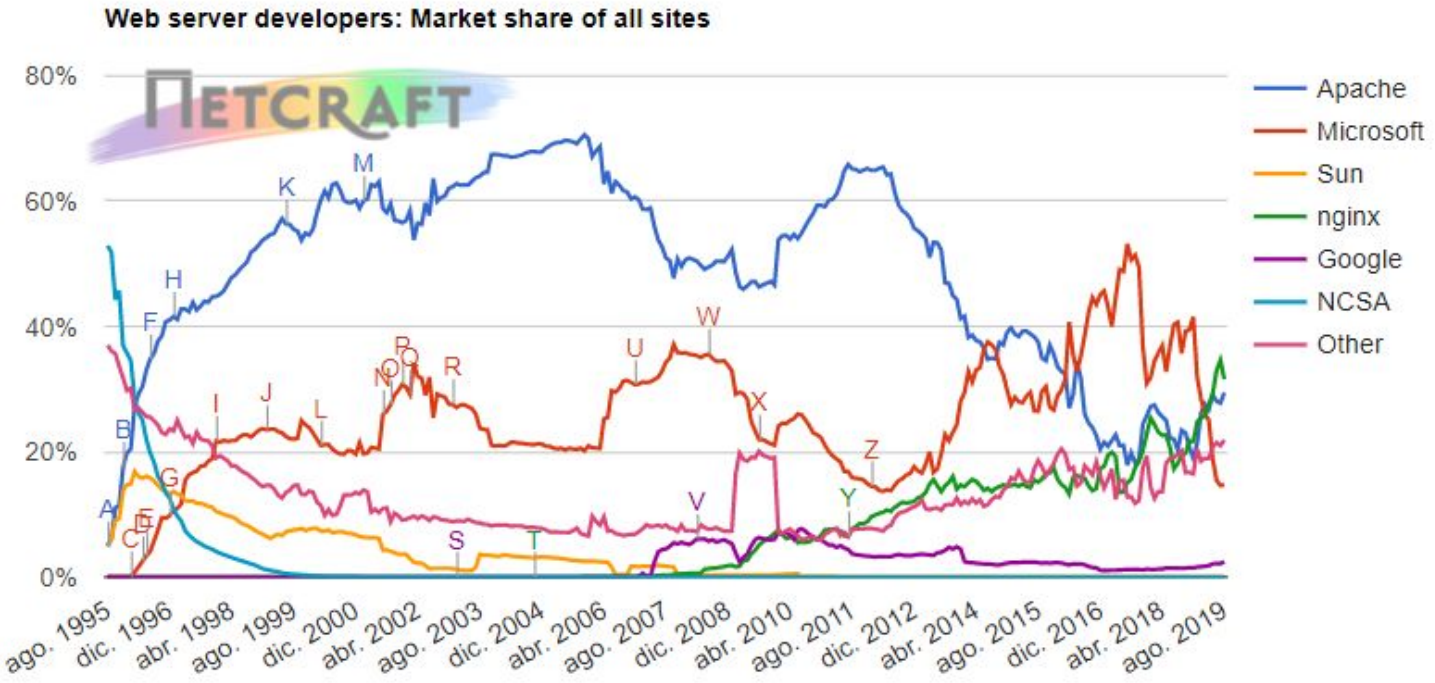
- el contenido dinámico se genere con la información de una base de datos.
- procesan información que les llega del mismo (autenticación, formulario, upload archivos)

Esta funcionalidad permite el desarrollo de completas **aplicaciones web**.

Qué es un servidor WEB? (III)

- Lanzado en 1995 y desarrollado por la Apache Software Fundation, hoy en día es el servidor más popular
- Es un servidor web multiplataforma y con una licencia de Software Libre (Apache License)
  https://github.com/apache/httpd
- Esta implementado en C
- Su nombre completo es Apache HTTP Server Project.

Qué es un servidor WEB? (IV)

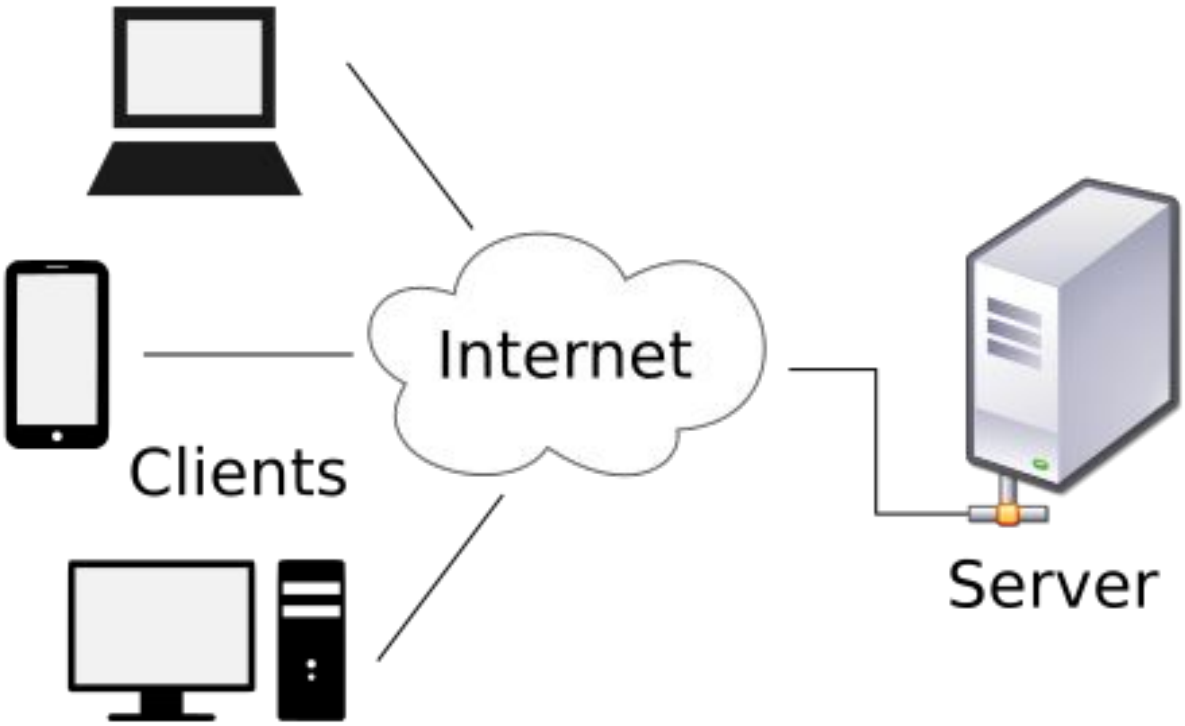


Developer	July 2019	Percent	August 2019	Percent	Change
nginx	482,877,275	34.59%	401,454,029	31.56%	-3.03
Apache	387,366,826	27.75%	374,277,243	29.43%	1.68
Microsoft	203,673,344	14.59%	187,109,423	14.71%	0.12
Google	29,385,065	2.11%	30,969,259	2.43%	0.33

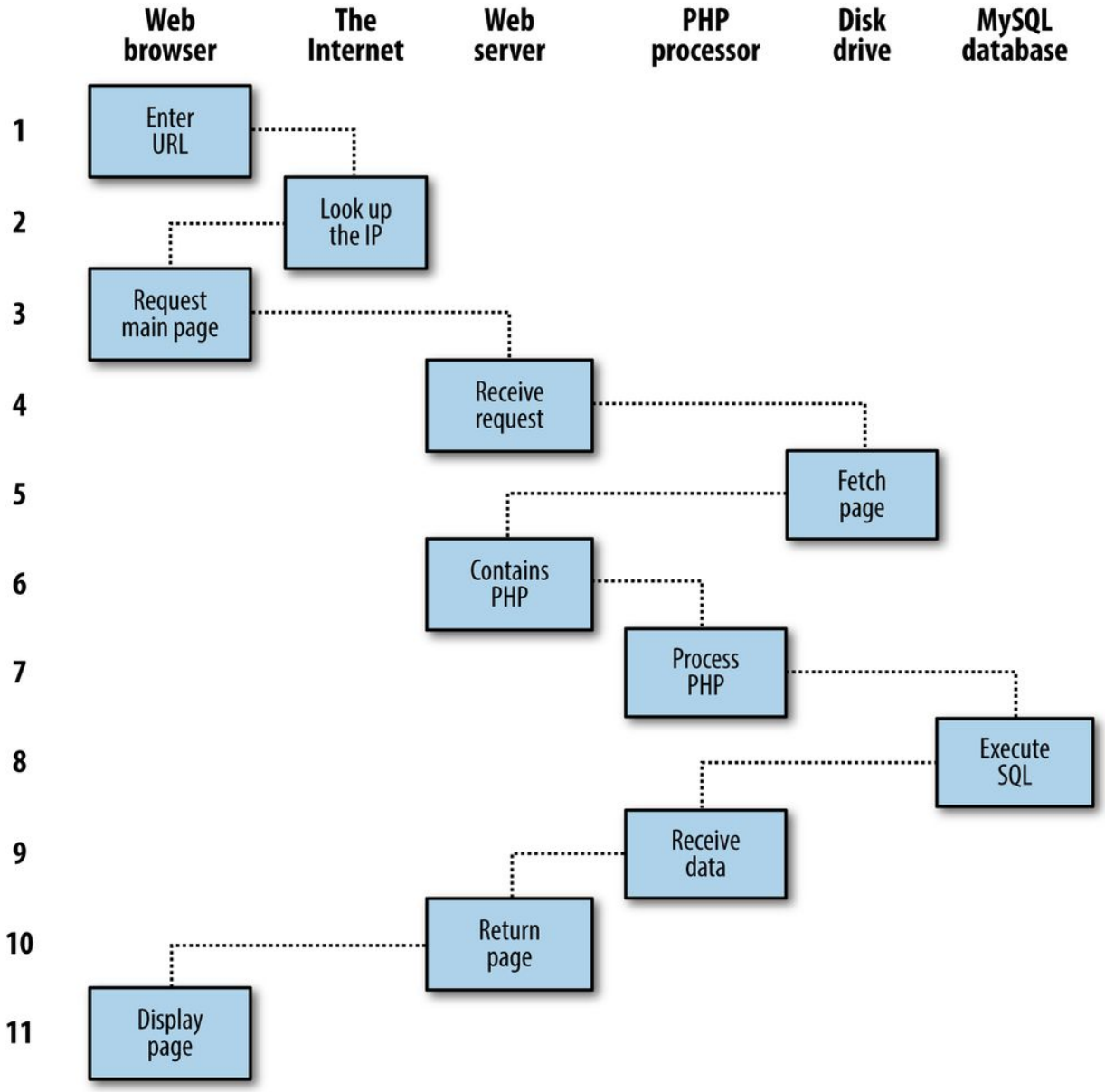
Como se genera una WEB page? (I)

Queremos consultar una página web:

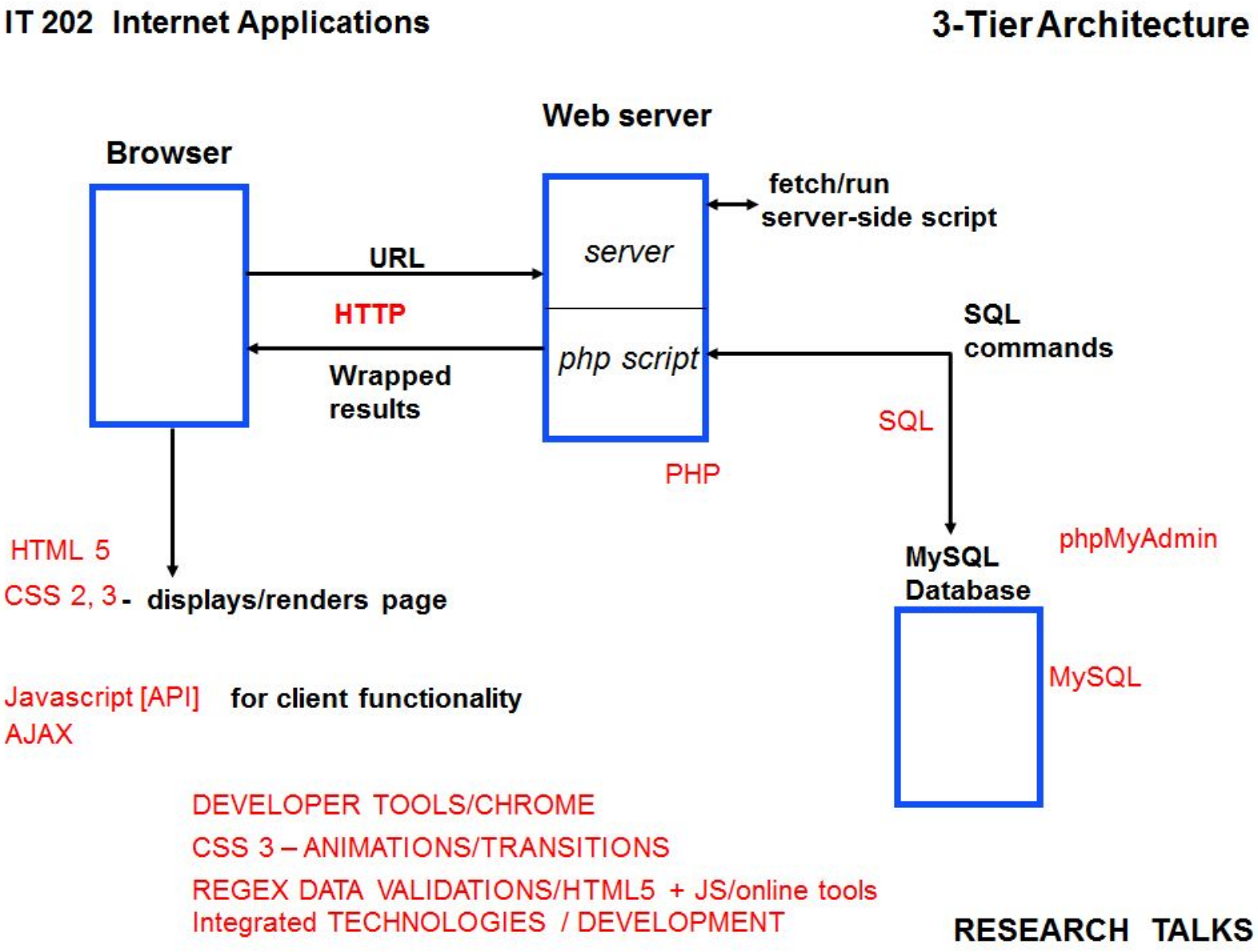
- Nuestro **navegador** (Chrome) es el **Cliente**
- El **Web Server** (Apache**) que tiene la página es el **Servidor**



Como se genera una WEB page? (II)



Como se genera una WEB page? (III)



Como se genera una WEB page? (III)

Esta discusión actualmente se ve en los sitios WEB. Existen dos formas de hacerlo

- El servidor envía el HTML completo del sitio
- El servidor envía datos (JSON, fragmentos HTML) y el cliente los procesa y muestra en la página

Existen casos **híbridos**, donde se baja el HTML completo inicialmente (procesado en el servidor) pero luego se actualiza mediante AJAX