

Fundamentos de aplicaciones Web

Karla A. Fonseca Márquez

Presentación del curso

- Del 14 al 21 de agosto 2017.
- De 16:00 a 18:00 hrs.
- Curso con una duración total de 10 horas



Evaluación:

- Tareas y prácticas
- Examen final

Objetivos



- Identificar y comprender la terminología que comúnmente se emplea al trabajar en diseño y desarrollo Web.
- Brindar un panorama de alto nivel acerca de cómo funciona la Web y cuales son las tecnologías que lo habilitan.
- Conocer estas tecnologías ayuda a comprender mejor por qué algunas cosas en el desarrollo de aplicaciones Web se hacen de ciertas maneras.
- Conceptualizar los elementos que conforman una aplicación Web.

Temario



Cómo funciona la Web
Internet, WWW, Protocolos, DNS, Cliente-Servidor, HTTP, URL



Aplicaciones Web
Tipos de aplicaciones, Ventajas, Retos



Tecnologías Web
Navegadores, HTML, CSS, Javascript, Servidores, Lenguajes de programación, AJAX



Herramientas
IDE, Editores, Librerías, Frameworks, Control de versiones

¿Qué es Internet?

6



Es un término que se utiliza para referirse a una **red masiva** de millones de computadoras de todo el mundo que se usan para compartir y transmitir información.

Esto se hace a través de varios protocolos y lenguajes e involucra varios tipos de servicios como **email**, **mensajería instantánea**, **transferencia de archivos punto-a-punto**, así como a la **Web**.



<https://youtu.be/xs0j5qf4PNA>

Se requieren básicamente 3 cosas para la transferencia de datos:

01.

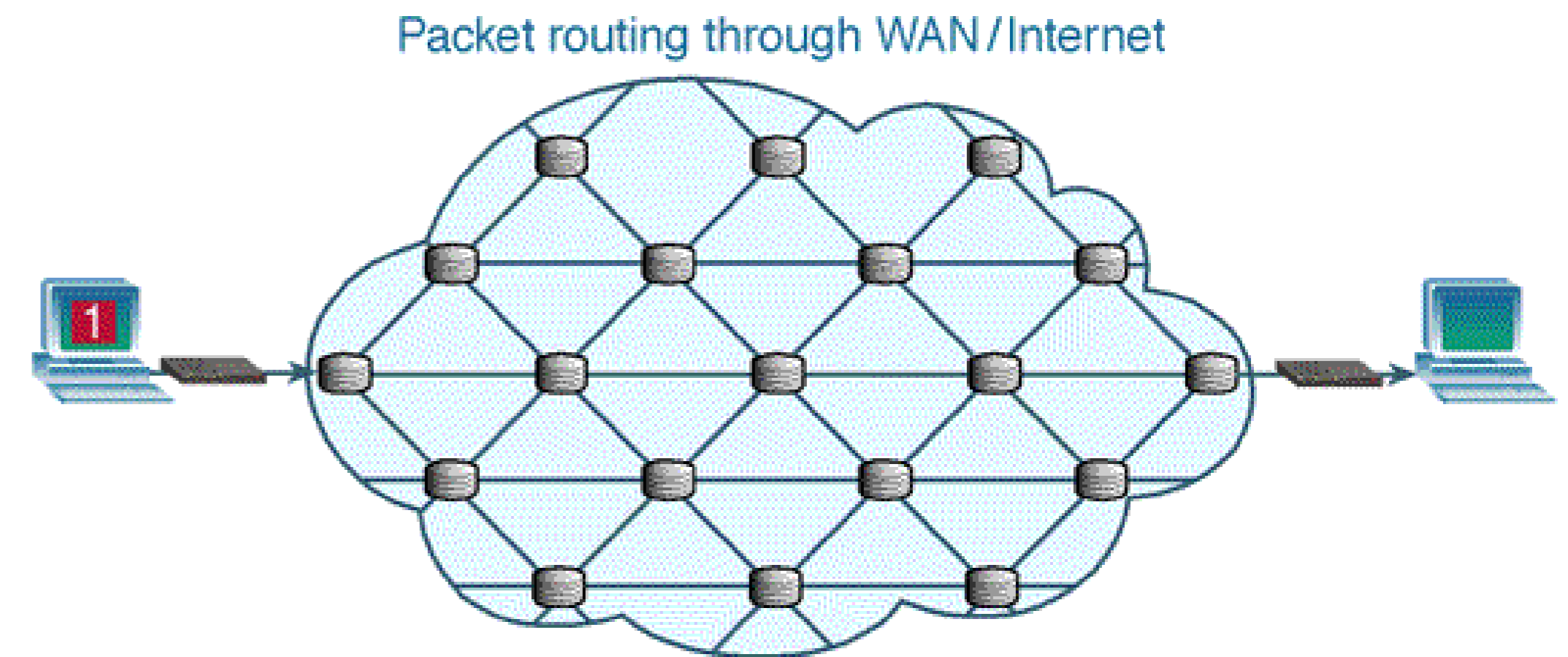
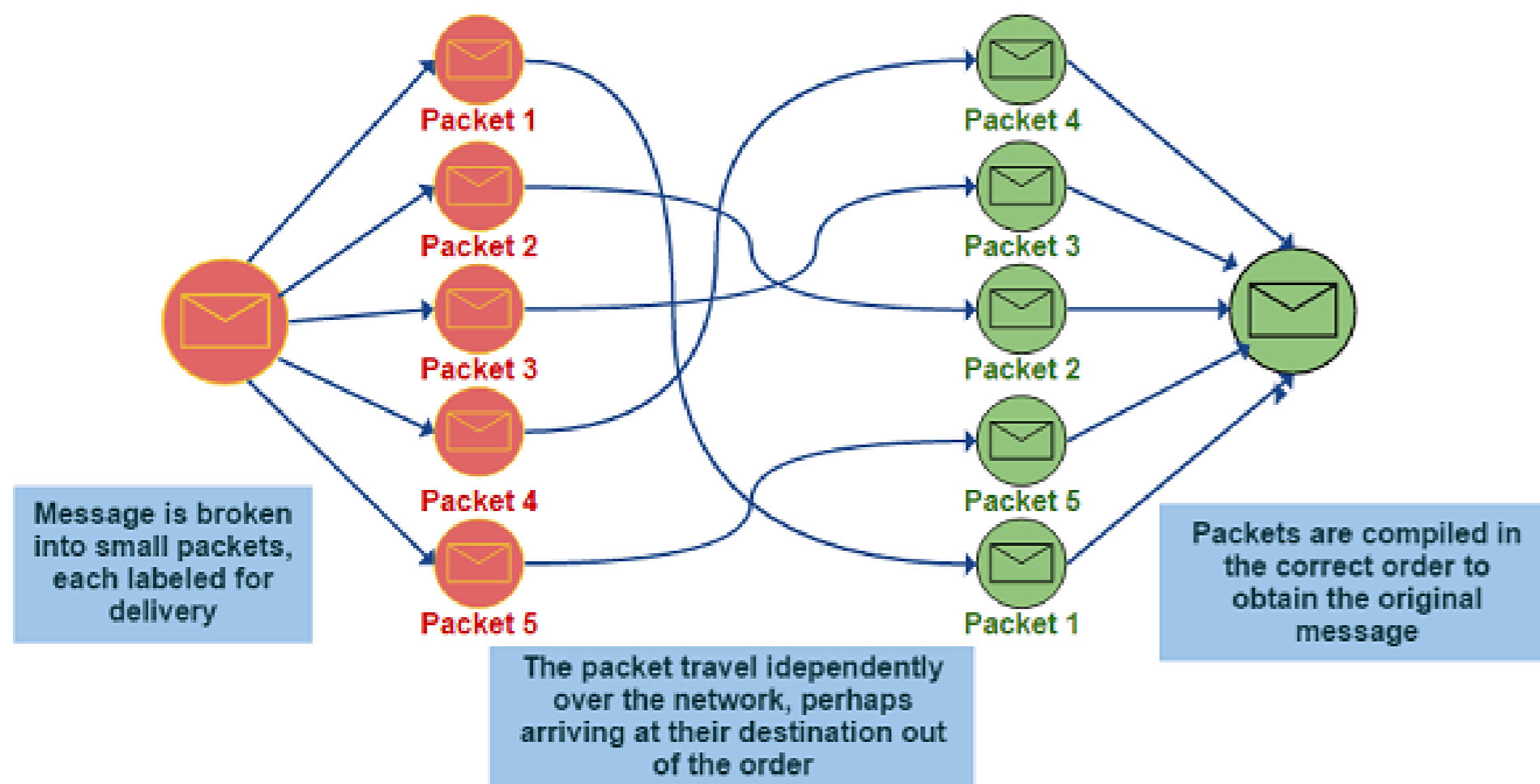
Reglas para el empaquetamiento de datos que se envían y reciben.

02.

Red interconectada (alámbrica o inalámbrica) sobre la que se enviarán los datos.

03.

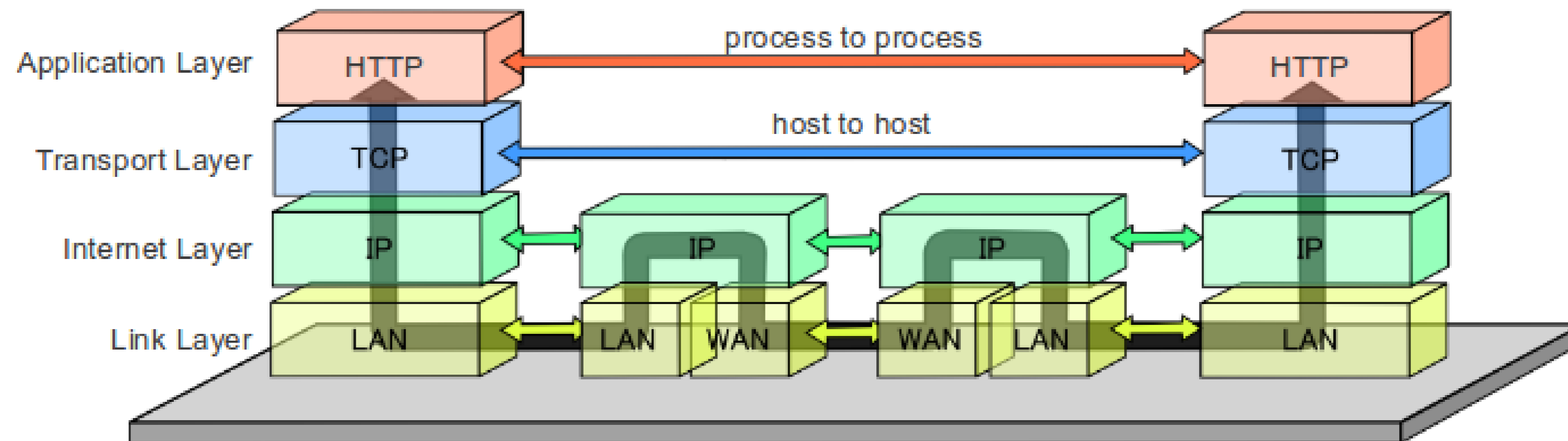
Manera de encontrar la ruta para hacer llegar los datos desde la computadora que envía hasta la que los recibe.



Familia de protocolos de Internet

- Grupo de protocolos **en los que se basa Internet** y que permiten la transmisión de datos.
- Son más de **100** los protocolos que conforman la suite, entre los que destacan:
 - **TCP** (*Transmission Control Protocol*), Protocolo de Control de Transmisión,
 - **IP** (*Internet Protocol*), Protocolo de Internet.
 - **ARP** (*Address Resolution Protocol*), Protocolo de Resolución de Direcciones,
 - **FTP** (*File Transfer Protocol*), Protocolo de Transferencia de Archivos,
 - **HTTP** (*HyperText Transfer Protocol*), Protocolo de Transferencia de HiperTexto,
 - **POP** (*Post Office Protocol*), Protocolo de Oficina Postal, para correo electrónico,
 - **SMTP** (*Simple Mail Transfer Protocol*), Protocolo de Transferencia Simple de Correo, para correo electrónico,
 - **Telnet** (*Teletype Network*), para acceder a equipos remotos.

Data Flow of the Internet Protocol Suite

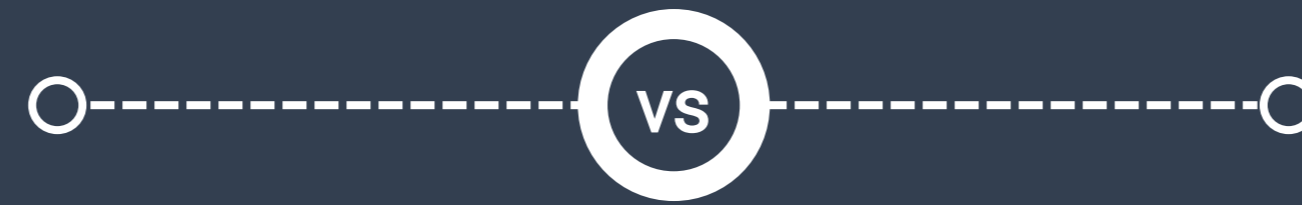


Modelo TCP/IP vs Modelo OSI

Modelo TCP/IP

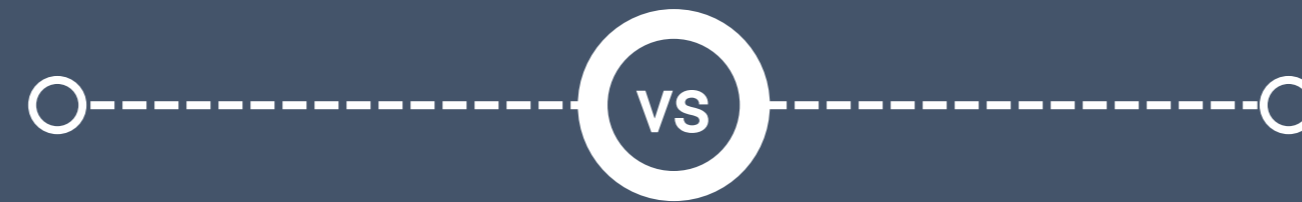
Modelo OSI

Implementación del modelo OSI



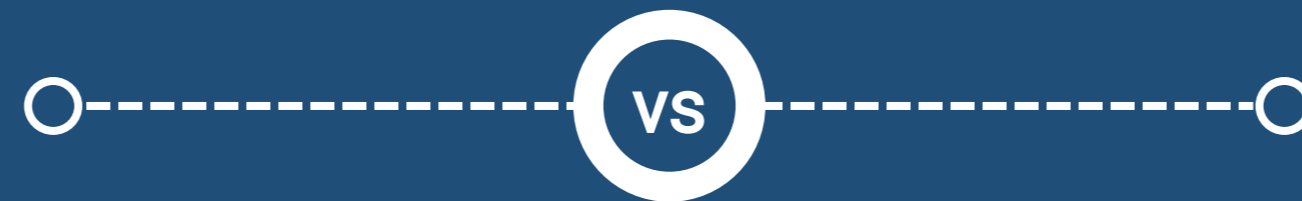
Modelo de referencia

Modelo sobre el cual se desarrolló Internet



Modelo teórico

4 capas



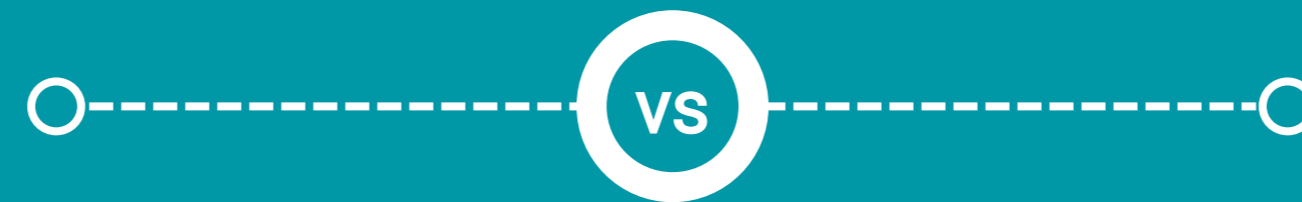
7 capas

Los protocolos se desarrollaron antes que el modelo



El modelo se desarrolló antes que los protocolos

No orientado a la conexión únicamente



Orientado y no orientado a la conexión

Dependiente de los protocolos

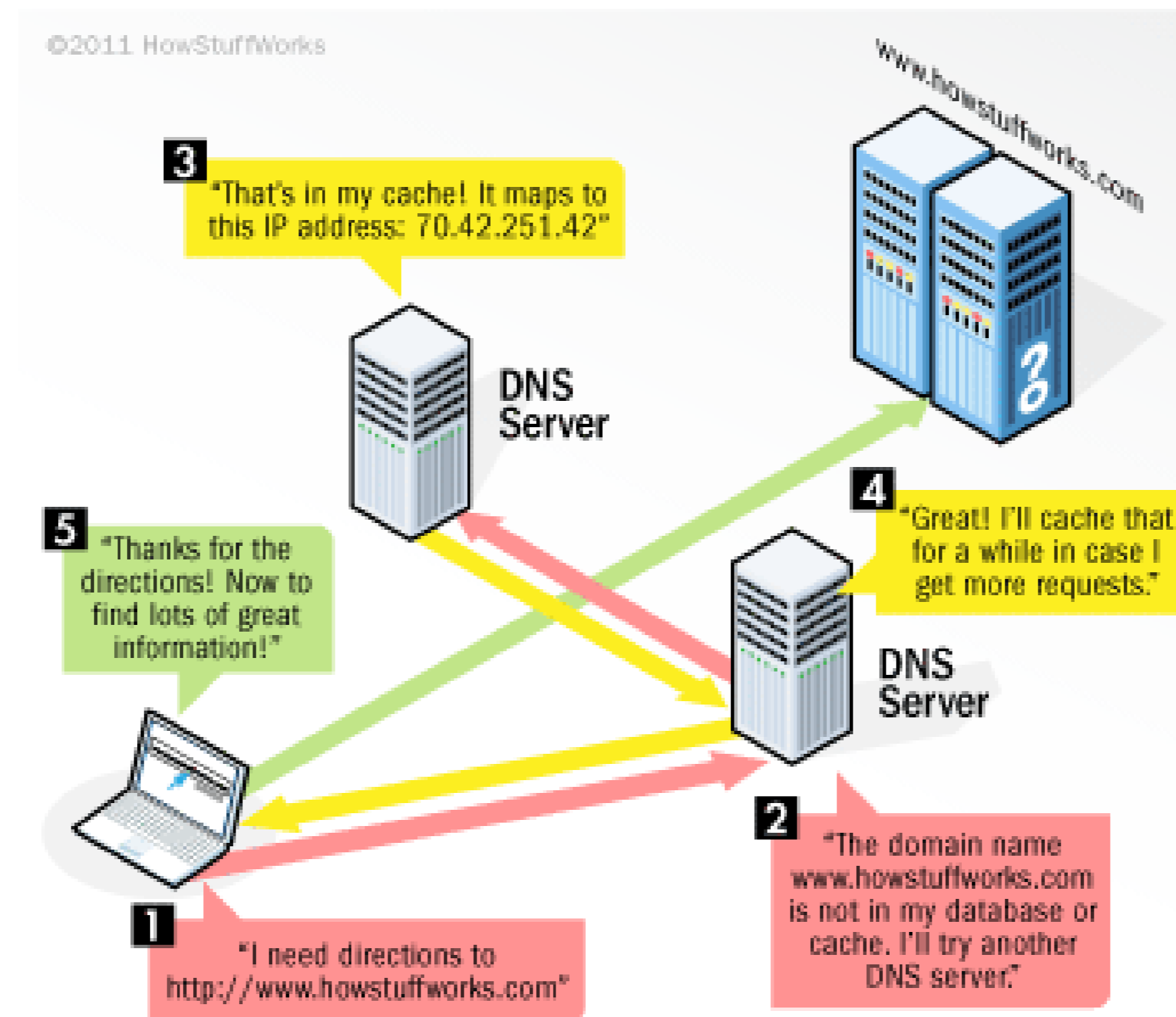


Independiente de los protocolos

Domain Name System

Base de datos de nombres de redes y direcciones IP.

“El directorio de Internet”



¿Qué es la Web (WWW)?

13

Es un **espacio de información** donde documentos de hipertexto y otros recursos web son identificados por una **URL**, ligados entre sí con vínculos y al que se puede acceder **vía Internet**.

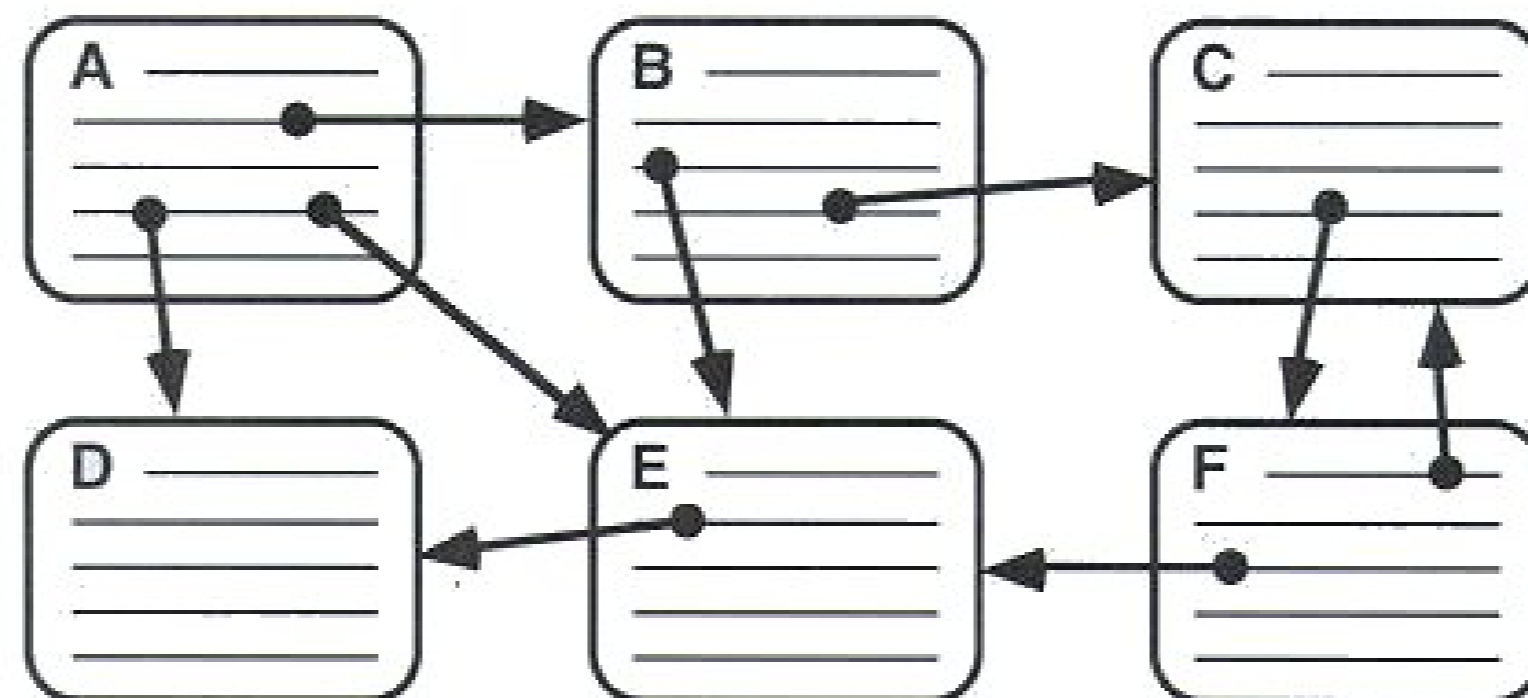


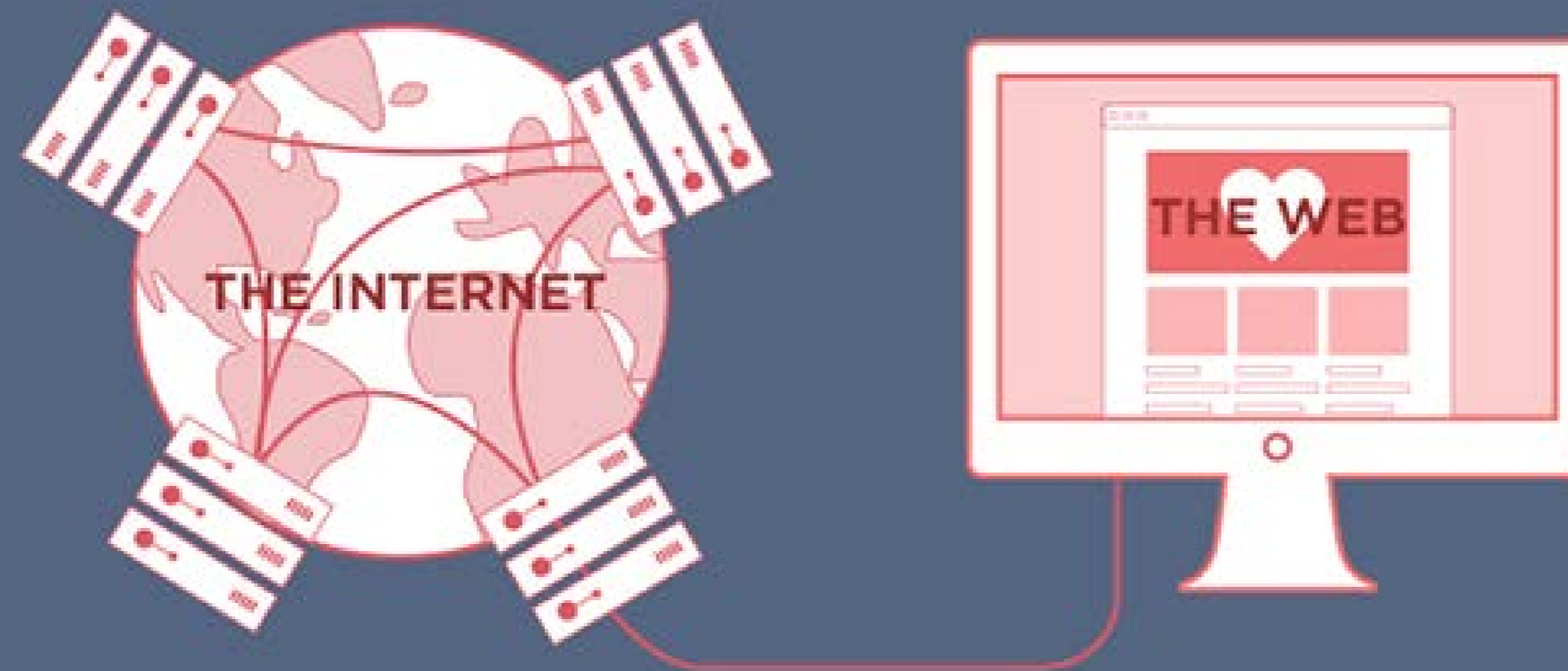
<https://youtu.be/Xlg7tqE1lh8>

¿Qué es la Web (WWW)?

14

- Es un **servicio de internet**, que utiliza principalmente el protocolo **HTTP**, **URLs** y **HTML** para la transferencia de hipertexto.
- Una colección de todos los **sitios web** públicos que hay en Internet.
- Se basa en una serie de interacciones entre dos tipos de sistemas: **clientes** y **servidores**.
- El **hipertexto** es texto estructurado que usa ligas lógicas (**hipervínculos**) entre nodos que contienen texto.





Redes de comunicación interconectadas que utilizan la suite de protocolos de internet

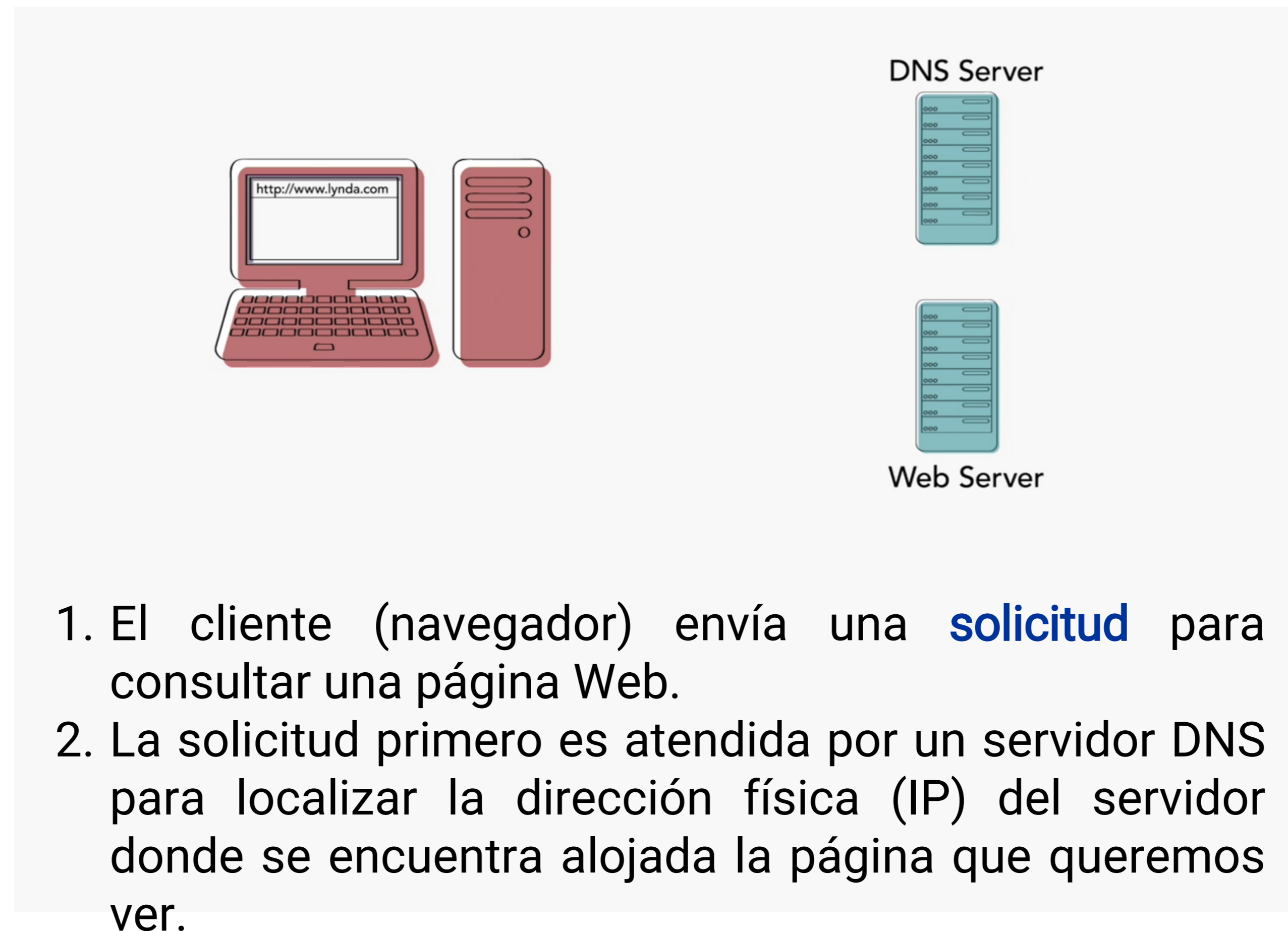
El contenido que viaja en forma de páginas Web a través de Internet / Servicio para consultar el contenido.

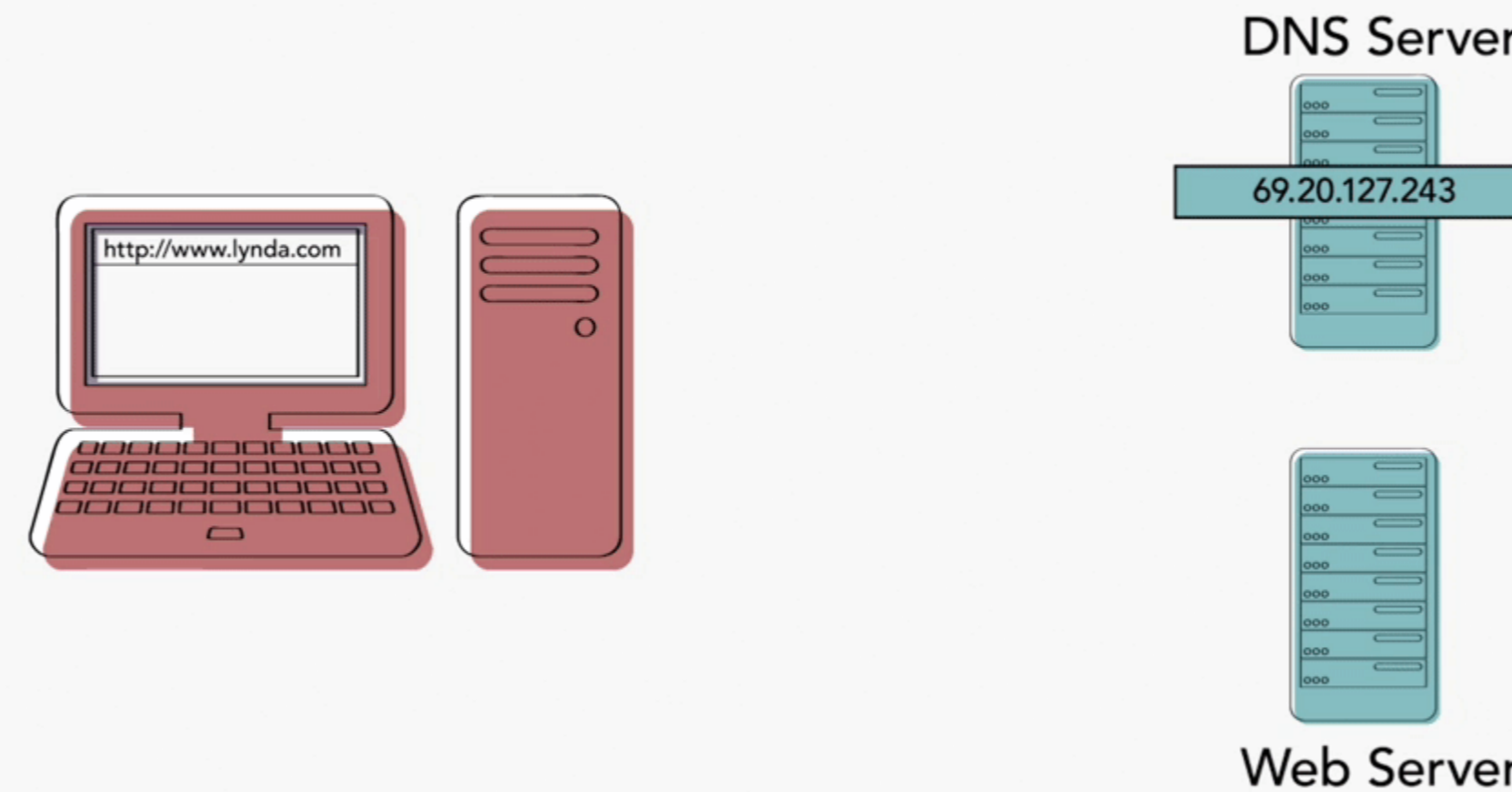


Solicitan y despliegan el contenido Web

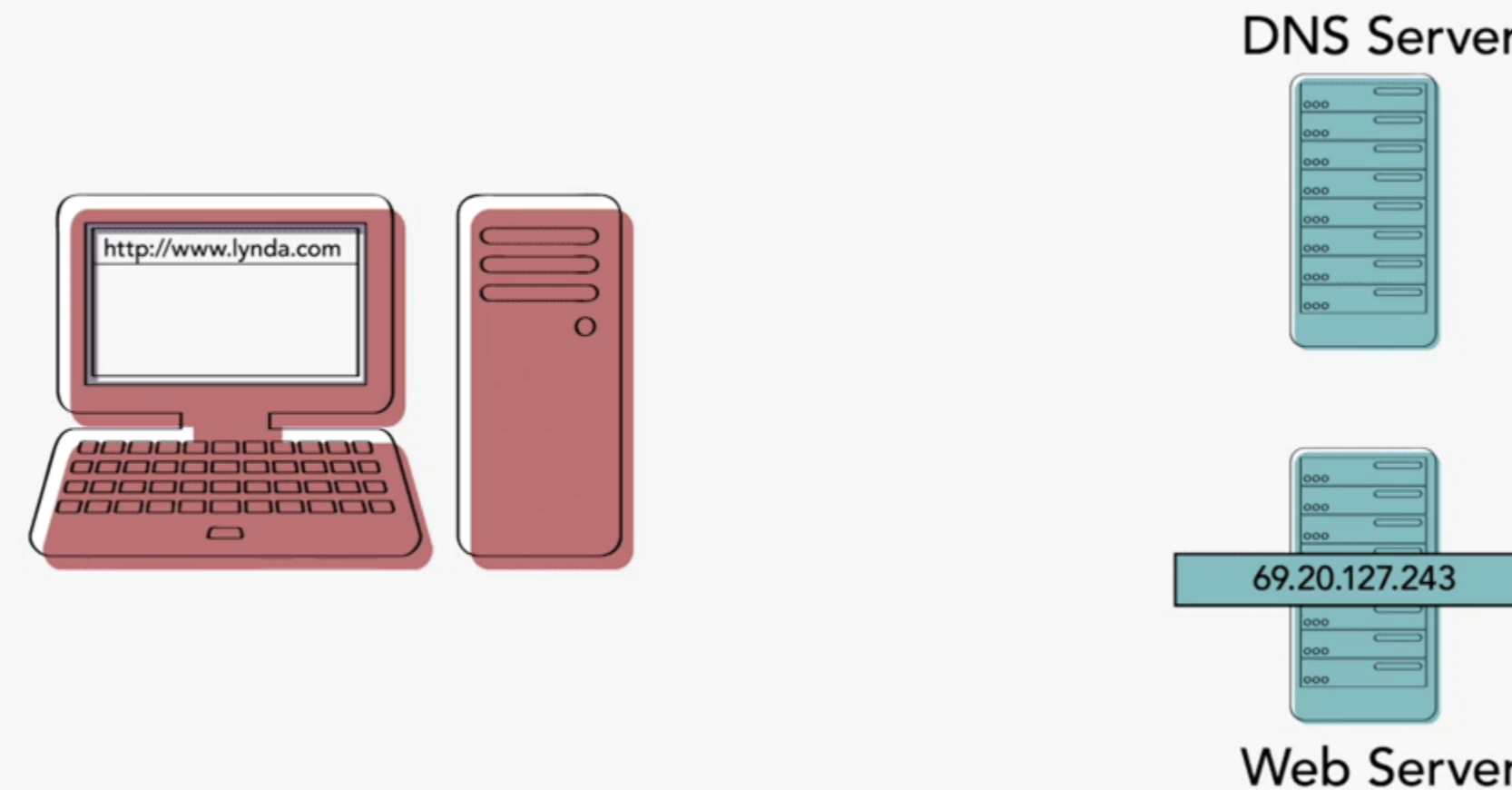
Navegadores, Apps, lectores de pantalla

Aplicaciones que entregan el contenido o servicios Web a los clientes



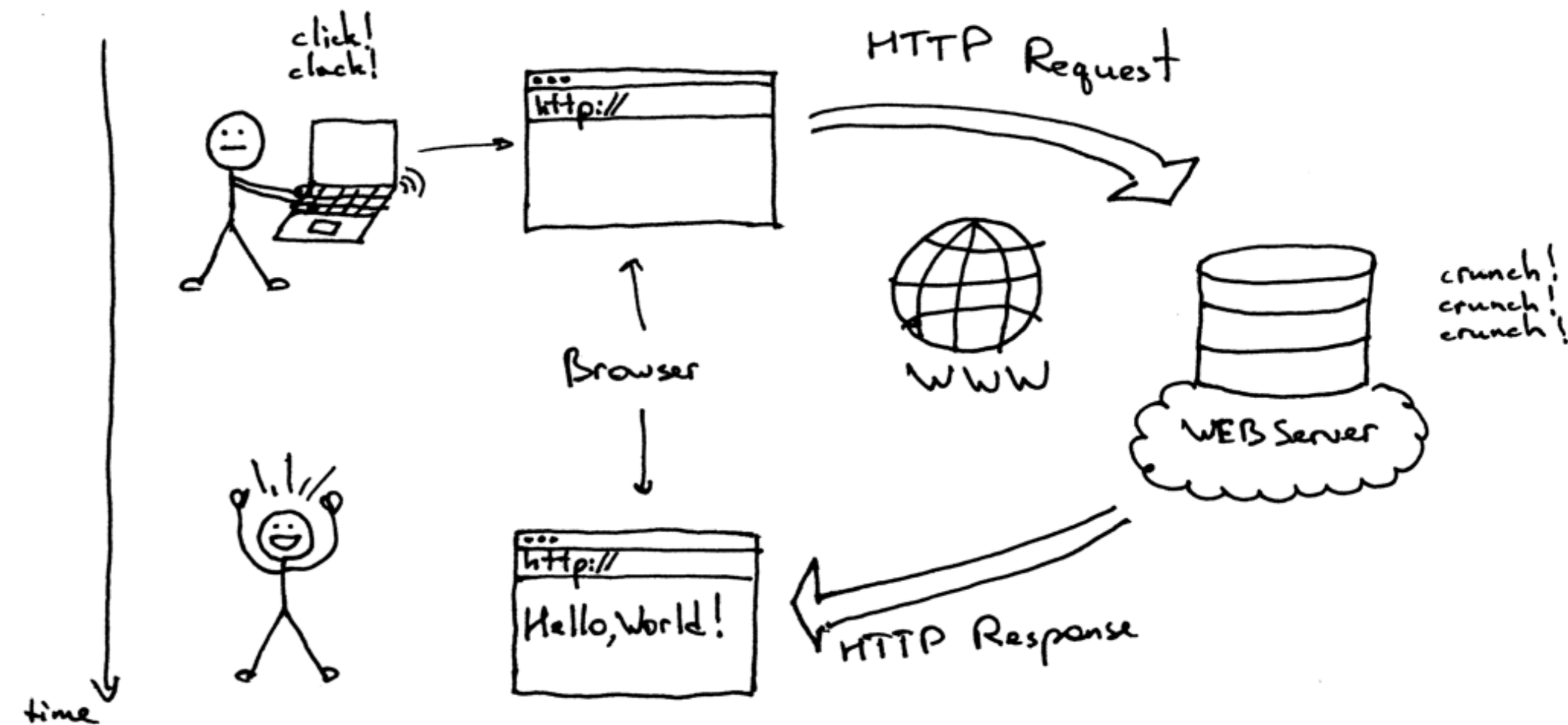


3. El servidor DNS nos devuelve la dirección IP de la página, y el navegador realiza la petición al Servidor Web.

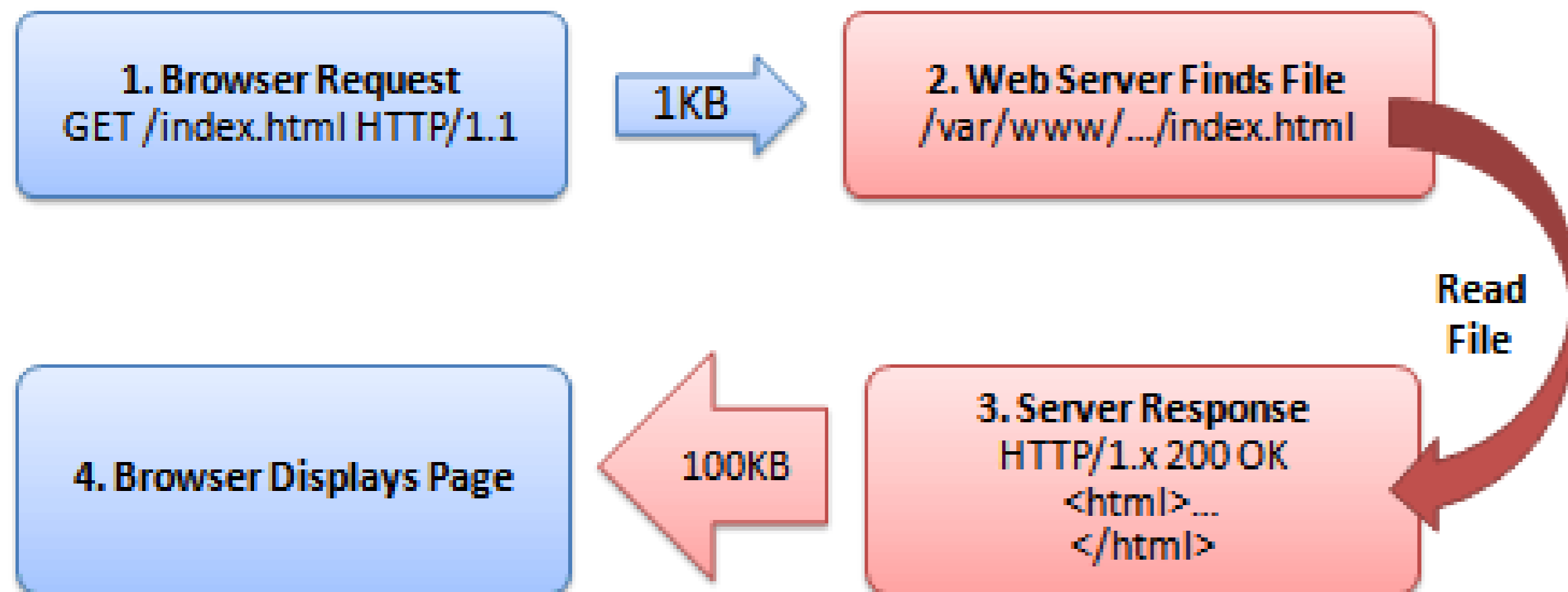


4. El servidor Web atiende la solicitud y devuelve al navegador el contenido de la página Web, incluyendo los recursos que utiliza la misma, como imágenes, audios, videos, hojas de estilo, etc.

- HTTP o **Hypertext Transfer Protocol** es un protocolo de la familia de internet que funciona en la capa de aplicación.
- Sirve para intercambiar o transferir hipertexto.
- HTTP funciona a base de **peticiones** y **respuestas** entre un cliente y un servidor.
- El **navegador** funciona como cliente. El servidor puede ser cualquier computadora que cuente con el software para transmisión de documentos (**servidor Web**).



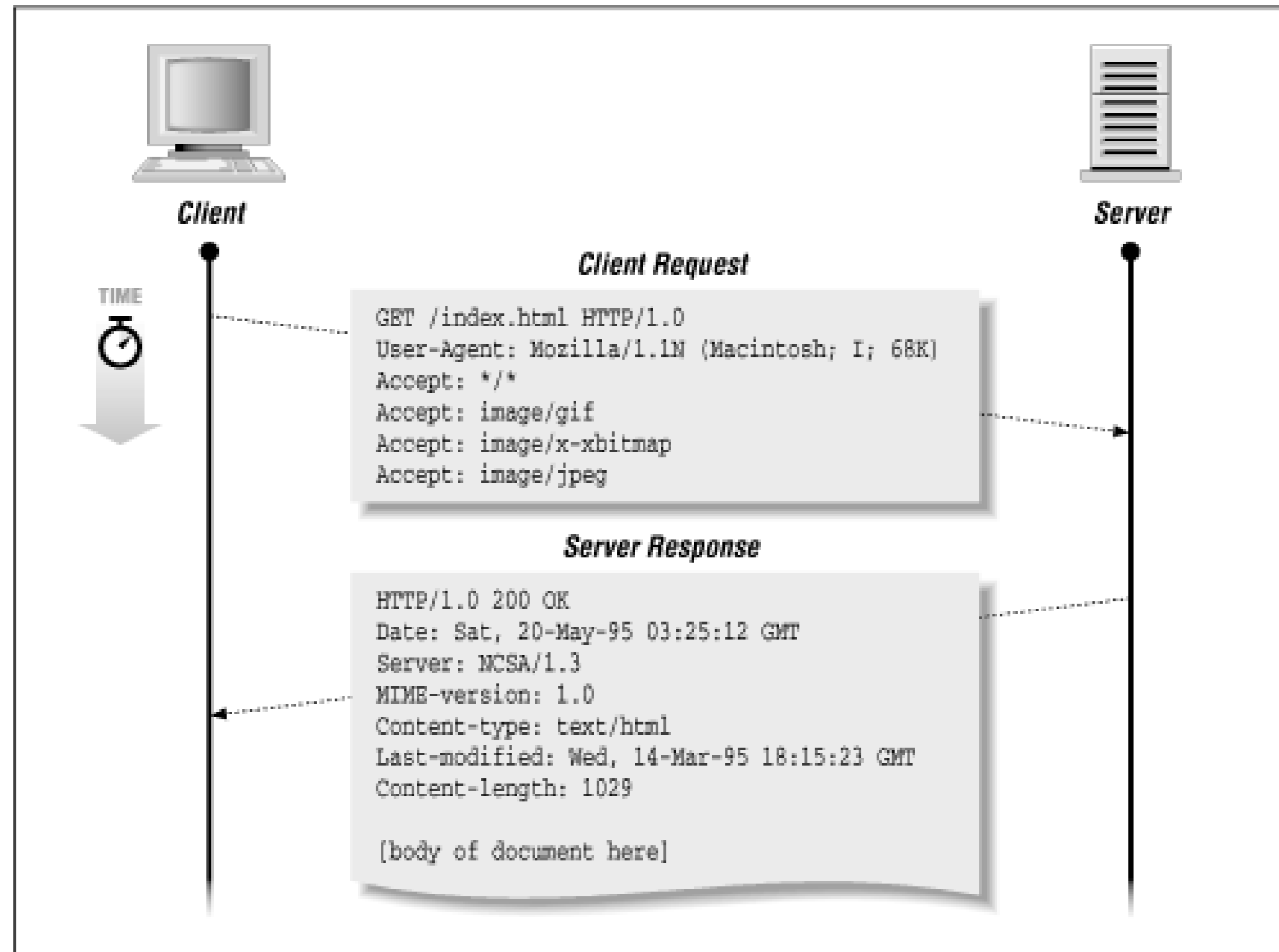
Request - Response



- Una **petición HTTP** es un conjunto de líneas que incluye, entre otras cosas, una dirección URL y un método.
- Una dirección URL puede ser algo como:
<http://www.mipagina.com/index.html>
- La **respuesta** contiene información sobre el estatus de la petición y el contenido solicitado (si aplica).

Request - Response

22



Método	Descripción
GET	Pide una representación del recurso especificado. Por seguridad no debería ser usado por aplicaciones que causen efectos ya que transmite información a través de la URI agregando parámetros a la URL.
HEAD	Pide una respuesta idéntica a la que correspondería a una petición GET, pero en la petición no se devuelve el cuerpo. Esto es útil para poder recuperar los metadatos de los encabezados de respuesta, sin tener que transportar todo el contenido.
POST	Envía los datos para que sean procesados por el recurso identificado. Los datos se incluirán en el cuerpo de la petición. Esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes o ambas cosas.
PUT	Sube, carga o realiza un upload de un recurso especificado (archivo), es el camino más eficiente para subir archivos a un servidor, esto es porque en POST utiliza un mensaje multiparte y el mensaje es decodificado por el servidor. En contraste, el método PUT te permite escribir un archivo en una conexión socket establecida con el servidor.
DELETE	Borra el recurso especificado.
OPTIONS	Devuelve los métodos HTTP que el servidor soporta para un URL específico. Esto puede ser utilizado para comprobar la funcionalidad de un servidor web mediante petición en lugar de un recurso específico.

GET vs POST



	GET	POST
Efecto al usar el botón Atrás o Recargar la página	Debe ser idempotente e inofensivo (Sin efectos secundarios)	Los datos se envían nuevamente.
Se puede agregar a Favoritos	Sí	No
Se puede guardar en caché	Sí	No
Tipo de encoding	application/x-www-form-urlencoded	application/x-www-form-urlencoded o multipart/form-data. Se usa multipart para enviar datos binarios (subir archivos).
Historial del navegador	Los parámetros permanecen en el historial.	Los parámetros no se guardan en el historial.
Restricciones de longitud	Sí, la información enviada se añade a la URL. La longitud de las URL se encuentra limitada a 2048 caracteres en algunos navegadores.	Sin restricciones. La información se envía dentro del cuerpo del mensaje.
Restricciones de tipo de datos	Sólo se permiten caracteres ASCII	Sin restricciones. Se permite también datos binarios.
Seguridad	GET es menos seguro que POST porque los datos forman parte de la URL. ¡Nunca se debe utilizar GET para enviar passwords o información sensible!	POST es un poco más seguro que GET porque los parámetros no se guardan en el historial del navegador ni en las bitácoras del servidor Web.
Visibilidad de los datos	Los datos son visibles en la URL.	Los datos no se muestran en la URL.

- Es una referencia a un **recurso Web** que especifica su **ubicación** dentro de una red así como el **mecanismo para obtenerlo**.
- Es un **tipo de URI**, o identificador de recursos.
- Su uso más común es para **direcciones de páginas Web** (http), pero también se usan para transferencia de archivos (ftp), correo electrónico, acceso a bases de datos, entre otros.
- Una URL típica como <http://example.org/index.html>, indica el protocolo a utilizar (http), un nombre de dominio (example.org) y un archivo (index.html)

`scheme://[user:password@]host[:port][/]path[?query][#fragment]`

`scheme:[//[user:password@]host[:port]][/]path[?query][#fragment]`

`http://www.example.org/`

`http://example.org:80/index.html`

`http://example.org/about/index.html`

`http://example.org/index.html?v=4587&p=kwjdh6yetgh`

`http://example.org/index.html#news`

Query String

- Parte de la URL que contiene datos o parámetros asociados a la petición.
- Visibles en la URL en peticiones realizadas con método GET.
- Cuando la petición se realiza con método POST, el query string se envía dentro del cuerpo o contenido y no como parte de la URL.
- La manera más común de generar query strings es con campos en formularios HTML `<form></form>`
- Se pueden agregar también a ligas `<a href>`

Query String

- Compuesto de una serie de **pares campo-valor**.
- Dentro de cada par, el nombre del campo y el valor se separan con signo de igual **=**
- Cada par se separa con el carácter ampersand **&**
- El query string inicia siempre con un signo de interrogación **?**

<http://example.org/index.html?v=4587&p=kwjdh6yetgh>

- Analizar la petición y respuesta de distintos sitios Web utilizando las Herramientas del desarrollador de Google Chrome (pestaña Network):

<http://www.unam.mx>

<http://www.google.com>

<http://www.apple.com>

<http://rouxacademy.com>

- <https://developers.google.com/web/tools/chrome-devtools>

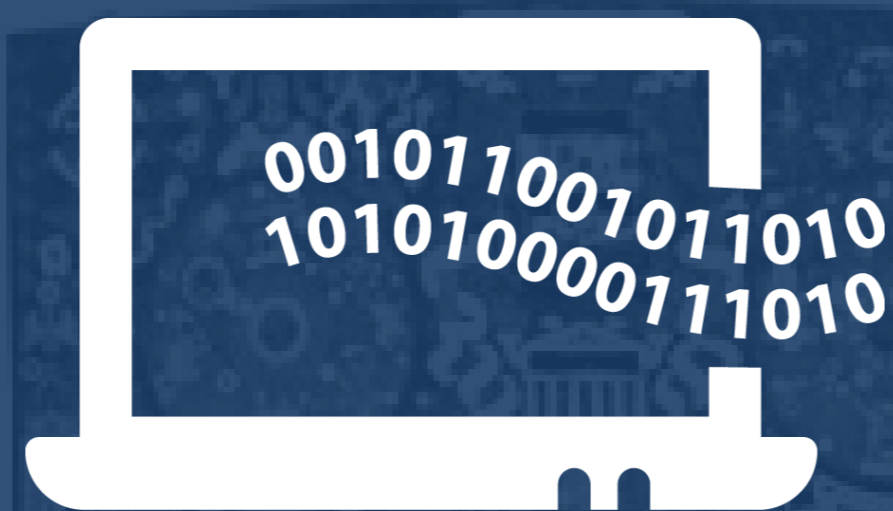
- Crear una página HTML:
 - inicio.html: Esta página deberá tener un hipervínculo a la página otra.html, el hipervínculo debe enviar por query string los valores: nombre = [su nombre] y correo = [su correo]
 - otra.html: Esta página deberá ???
- Crear una página HTML:
 - formulario.html: Esta página deberá tener un formulario con 2 campos: nombre y correo y enviar los datos a registro.html
 - registro.html: Esta página deberá ???
- Analizar las diferencias al enviar el formulario HTML con método GET y con método POST.

- Manipulación de headers con PHP

- Es importante recordar que esta conversación de petición-respuesta vía HTTP es el **proceso fundamental** que guía toda la comunicación en la Web.

Sin importar el lenguaje de programación que se utilice, el tipo de aplicación que se construya (Web, Móvil, API) o la filosofía de desarrollo que se siga al desarrollar aplicaciones Web, todo se reduce a esto:

- comprender y procesar cada petición y
- crear y devolver la respuesta apropiada.



Fundamentos de aplicaciones Web

Karla A. Fonseca Márquez



- Los navegadores usan HTTP para **comunicarse con servidores Web** y solicitar páginas Web y contenido.
- El navegador después usa su **motor interno de renderización** para analizar la página y mostrarla en pantalla.
- Cualquier contenido adicional, como imágenes o video, se solicita y muestra **dependiendo** de si el navegador soporta el tipo de archivo o si se encuentran instalados los plugins necesarios.
- Cualquier **código del lado del cliente** como Javascript, se ejecuta en el navegador cuando sea invocado.
- Aunque cada uno es diferente, los navegadores generalmente están conformados por los mismos **componentes básicos**.

Componentes básicos

4



Motores de renderización

5

Trident



Internet Explorer

Windows Mobile

AOL Browsers

Gecko



Mozilla Firefox

Firefox OS

Presto



Opera 13

Opera Mini

Opera Mobile

WebKit



Safari

webOS

Android

Blink



Chrome

Opera 14+

Android 4.4+

Guerra de navegadores

6

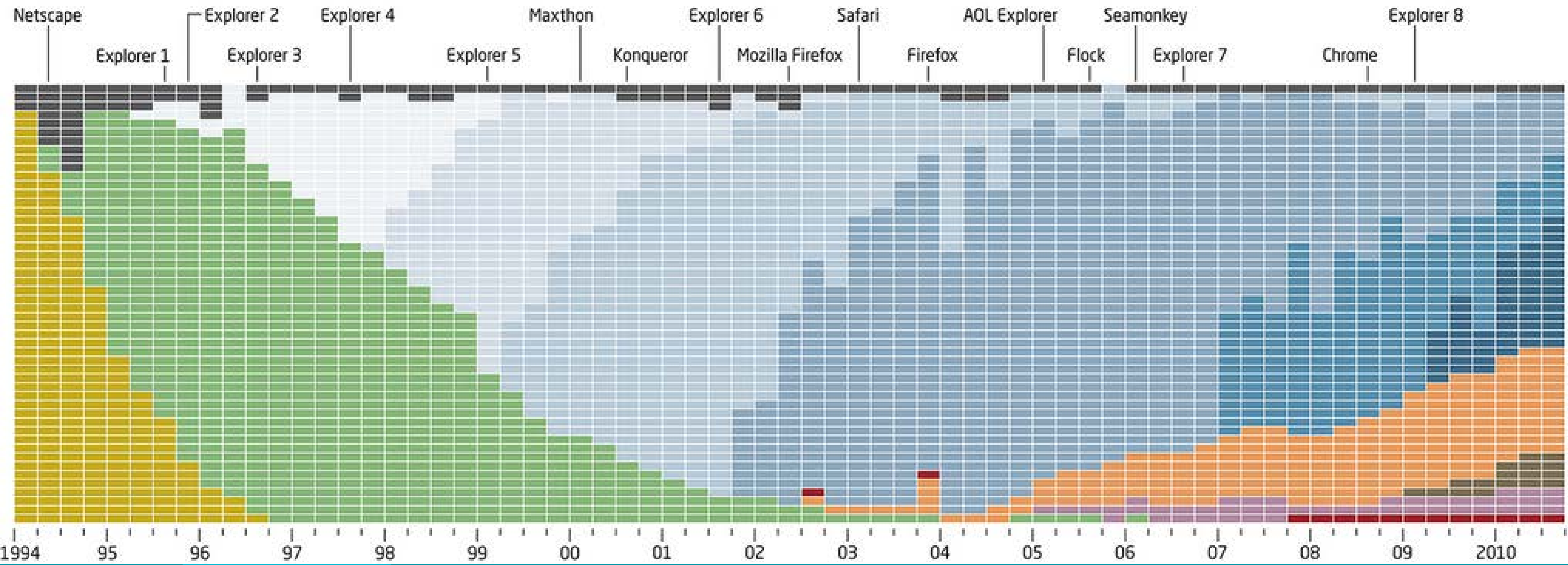
Browsers wars

In 1991 Berners-Lee unveils at CERN World Wide Web, the first Web browsing software. 20 years later the struggle keeps on.

Publish date and market share (2% increments)

Browsers legend (with at least 2% of market share)


- Internet Explorer 1-2-3
- Internet Explorer 4
- Internet Explorer 5
- Internet Explorer 6
- Internet Explorer 7
- Internet Explorer 8
- Firefox
- Chrome
- Safari
- Opera
- Mosaic
- Netscape
- Other



Guerra de navegadores



Conversation Between Browsers

What are we?	Browsers!	...	Browsers!	Browsers!
				
What do we want?	Faster!	...	Faster!	Faster!
				
When do we want it?	Now!	...	Now!	Now!
				
	Browsers!			
				

Ejemplos de diferencias

8

- http://www.browsera.com/web_sites

W3C

- Creado en octubre de 1994 por Tim Berners-Lee (creador de la WWW).
- El objetivo del Consorcio World Wide Web (W3C) es “**guiar la Web hacia su máximo potencial**”.
- Desarrolla **protocolos y directrices** que **aseguren el crecimiento de la Web a largo plazo**.
- Los estándares del W3C definen las partes claves que hacen **que la World Wide Web funcione**.
- Los estándares son **abiertos**, es decir, todos los estándares están disponibles de forma pública, sin costo.

- Compatibilidad con múltiples navegadores
- Reducir los costos de mantenimiento
- Cargar más rápido, consumiendo menos ancho de banda
- Mejorar el posicionamiento en los buscadores
- Adaptación a múltiples plataformas
- Accesibilidad

- HTML – HyperText Markup Language
- XML – Extensible Markup Language
- XHTML
- CSS – Cascading Style Sheets
- DOM – Document Object Model
- ECMAScript (JavaScript estandarizado)

Para empezar....

12



- HTML significa **Hypertext Markup Language**
- Es el lenguaje de marcado estándar para páginas Web.
- Es uno de los conocimientos **imprescindibles** para cualquier diseñador o desarrollador Web.
- Al día de hoy, **HTML5** se considera como la versión vigente.
- HTML5 **retro-compatible** con las versiones anteriores, y también incorpora nuevas etiquetas y soporte de APIs
- HTML es un **lenguaje estructural**: identifica elementos y define la **estructura semántica** de una página Web y no cómo debe presentarse ni su formato.
- <https://www.w3.org/TR/html5/>

- Publicar documentos en línea con encabezados, texto, tablas, listas, fotos, etc.
- Obtener información en línea utilizando vínculos de hipertexto, con el clic de un botón.
- Diseñar formularios para realizar transacciones con servicios remotos, búsquedas de información, reservaciones, compras, etc.
- Incluir video, audio y otras aplicaciones directamente en los documentos.

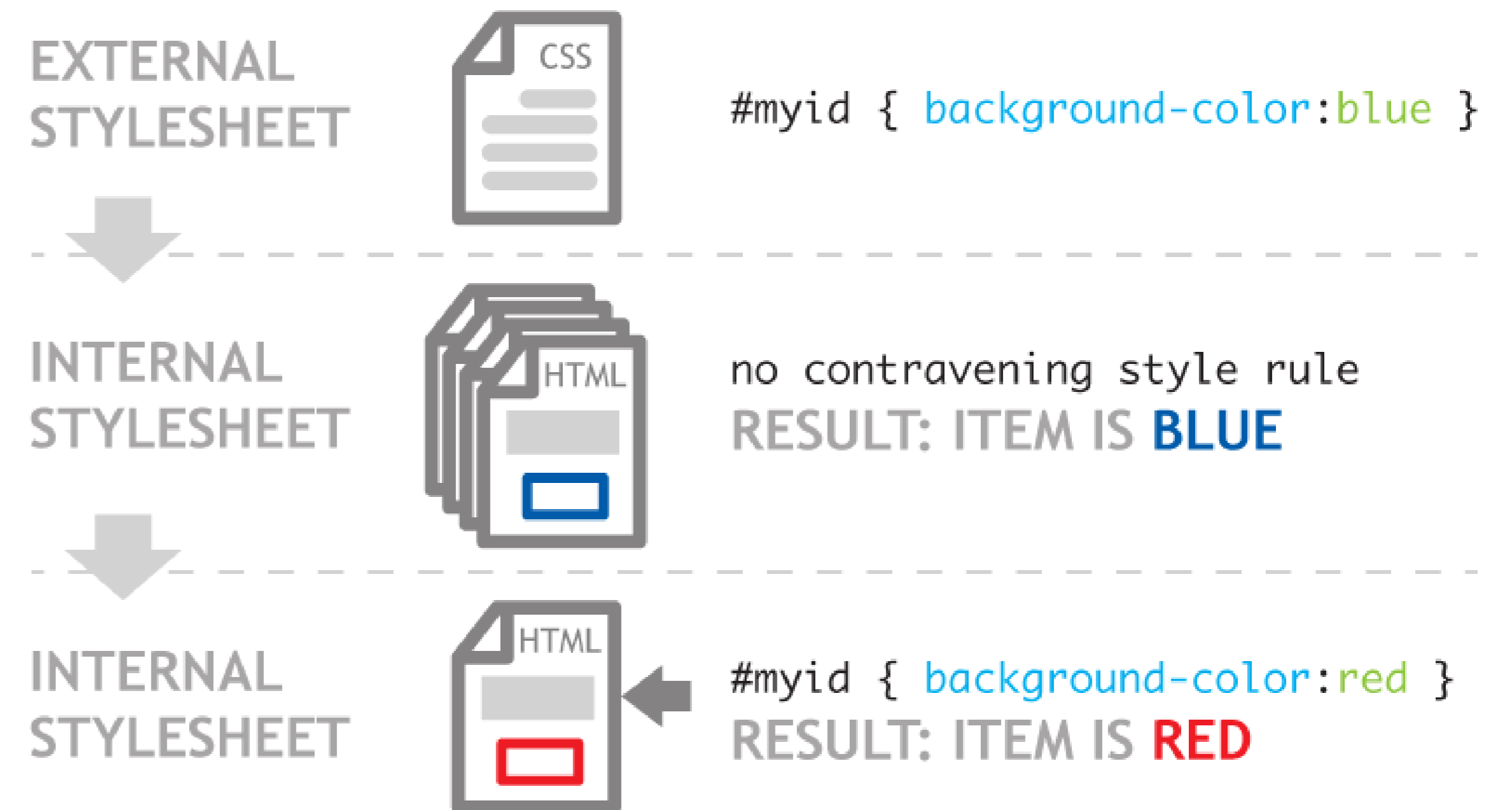
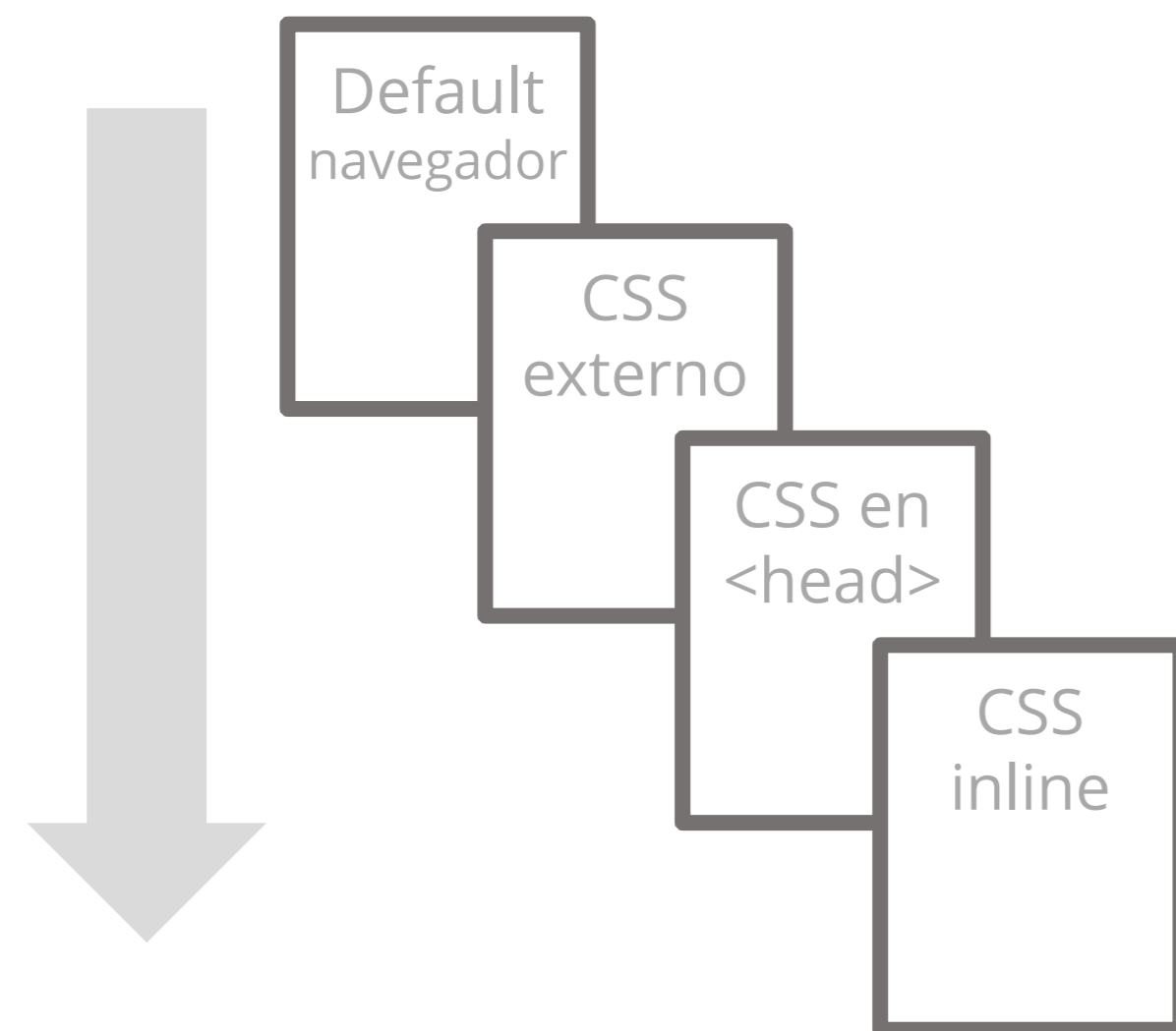
- Consiste en un **conjunto de etiquetas** con el cual se “marca” el contenido:

Ejemplos:

<html></html>	describen el documento HTML
<head></head>	proporciona información sobre el documento
<title></title>	proporciona el título del documento
<body></body>	encierran el contenido visible de la página Web
<h1></h1>	describen un encabezado de primer nivel
<p></p>	describe un párrafo
<form></form>	describe un formulario

http://www.w3schools.com/html/html_examples.asp

- **CSS**, o Cascading Style Sheets, es un lenguaje de **hojas de estilo**.
- Sirve para **controlar la presentación** de documentos HTML.
- En CSS se pueden definir varios aspectos de la presentación como colores, fuentes, distribución, entre otros.
- Permite la **separación** del contenido y del formato lo cual **reduce la complejidad** de una página y brinda **más flexibilidad**.
- Permite definir diferentes presentaciones **de acuerdo al medio** en que se presente, por ejemplo, en pantalla, al imprimir, en dispositivos móviles, dispositivos de sólo texto, etc.
- Especifica un **esquema de prioridades** para determinar qué hacer en caso de que al mismo elemento le apliquen dos o más reglas. Por eso se le llama “en cascada”.



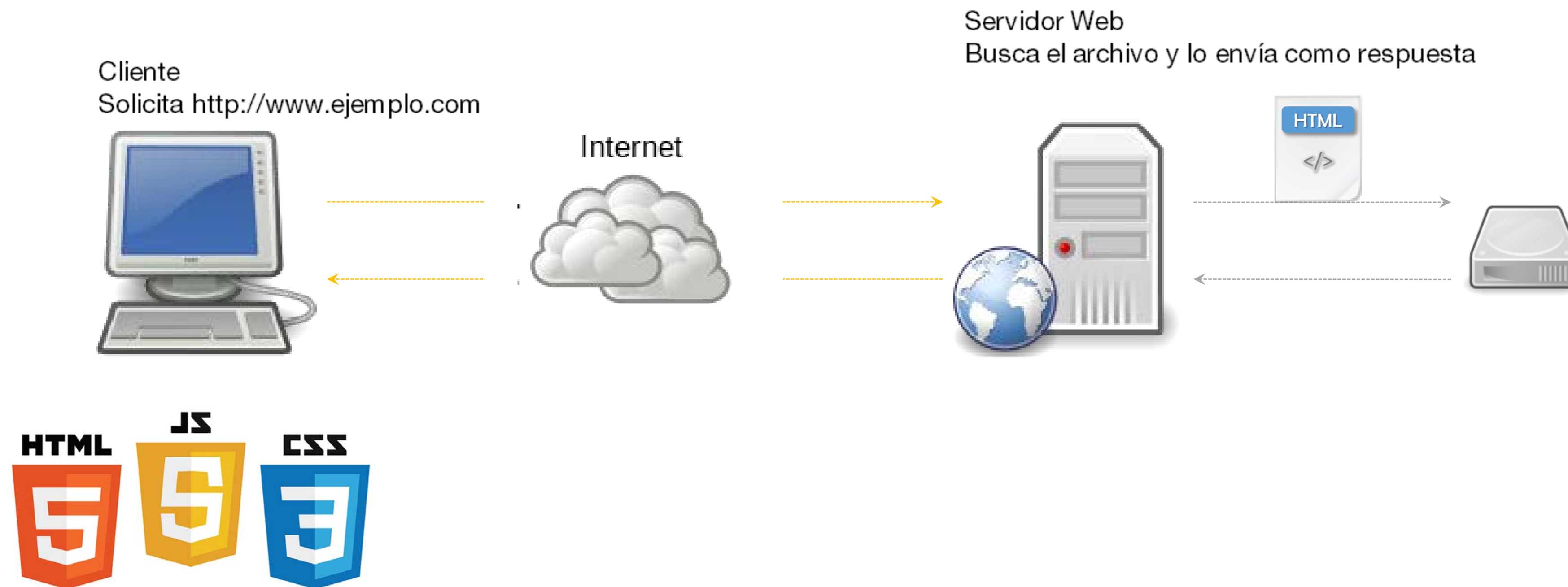
<http://www.rouxacademy.com>

- JavaScript es un lenguaje de **scripting** que se usa para agregar interactividad y funcionalidad a sitios y aplicaciones Web.
- Por ejemplo, permite crear elementos interactivos, menús dinámicos, abrir nuevas ventanas del navegador, actualizar datos, etc.
- Se ejecuta en el navegador, es decir, del **lado del cliente**. Eso tiene la ventaja de que la página no tiene que recargarse para que ocurran cambios.
- Aunque es ampliamente soportado por los diversos navegadores, es importante señalar que cada uno tiene su propio intérprete de Javascript.
- Además, el usuario puede **deshabilitar Javascript** en su navegador, por lo que es importante definir una estrategia para cuando se accede a un sitio desde un navegador sin Javascript.

http://www.w3schools.com/js/js_intro.asp



Contenido Estático



Contenido Estático

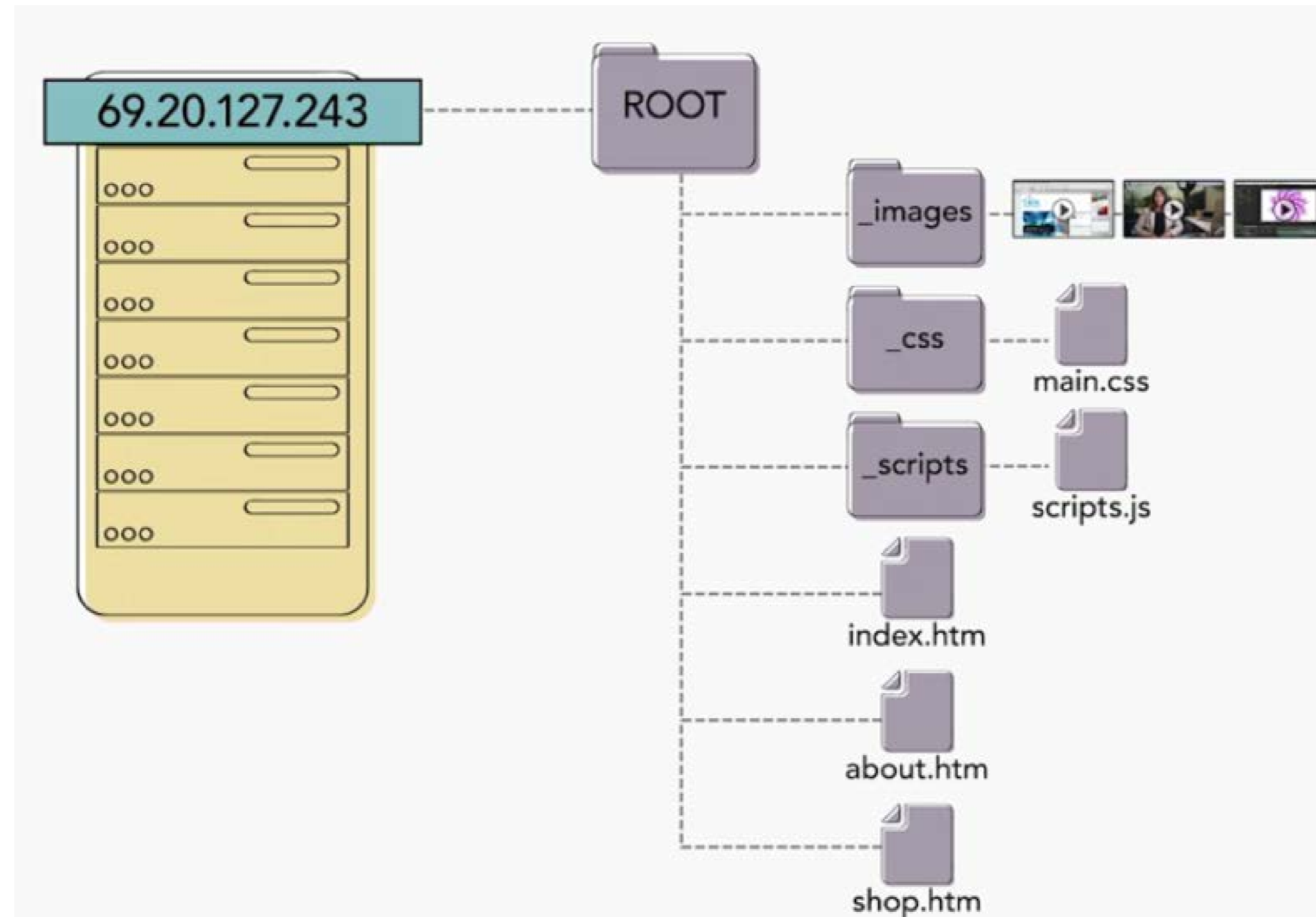
- Las páginas se muestran tal y como están almacenadas en el servidor
- Muestran la misma información para todos los usuarios.
- No hay procesamiento del lado del servidor.
- Desventajas:
 - Cualquier personalización e interacción debe realizarse del lado del cliente.
 - Es difícil e impráctico dar mantenimiento a un gran número de páginas.



- Los **sitios Web** se alojan en **Servidores Web**.
- Un servidor Web puede ser **cualquier computadora** que cuente con un conjunto especializado de **software y componentes**.
- Todos los servidores deben tener una **dirección IP**, y los sitios que aloje deben tener asignado un **nombre de dominio**.
- Muchos servidores Web utilizan hosting virtual que permite alojar varios sitios (con dominios distintos) bajo la misma dirección IP.



- Los sitios se almacenan en el servidor en archivos y directorios.



- Conjunto de software y componentes disponibles en el servidor Web, que permiten procesar y servir páginas Web.
- Generalmente consisten en:

Sistema operativo

Servidor Web

Servidor de Base de datos*

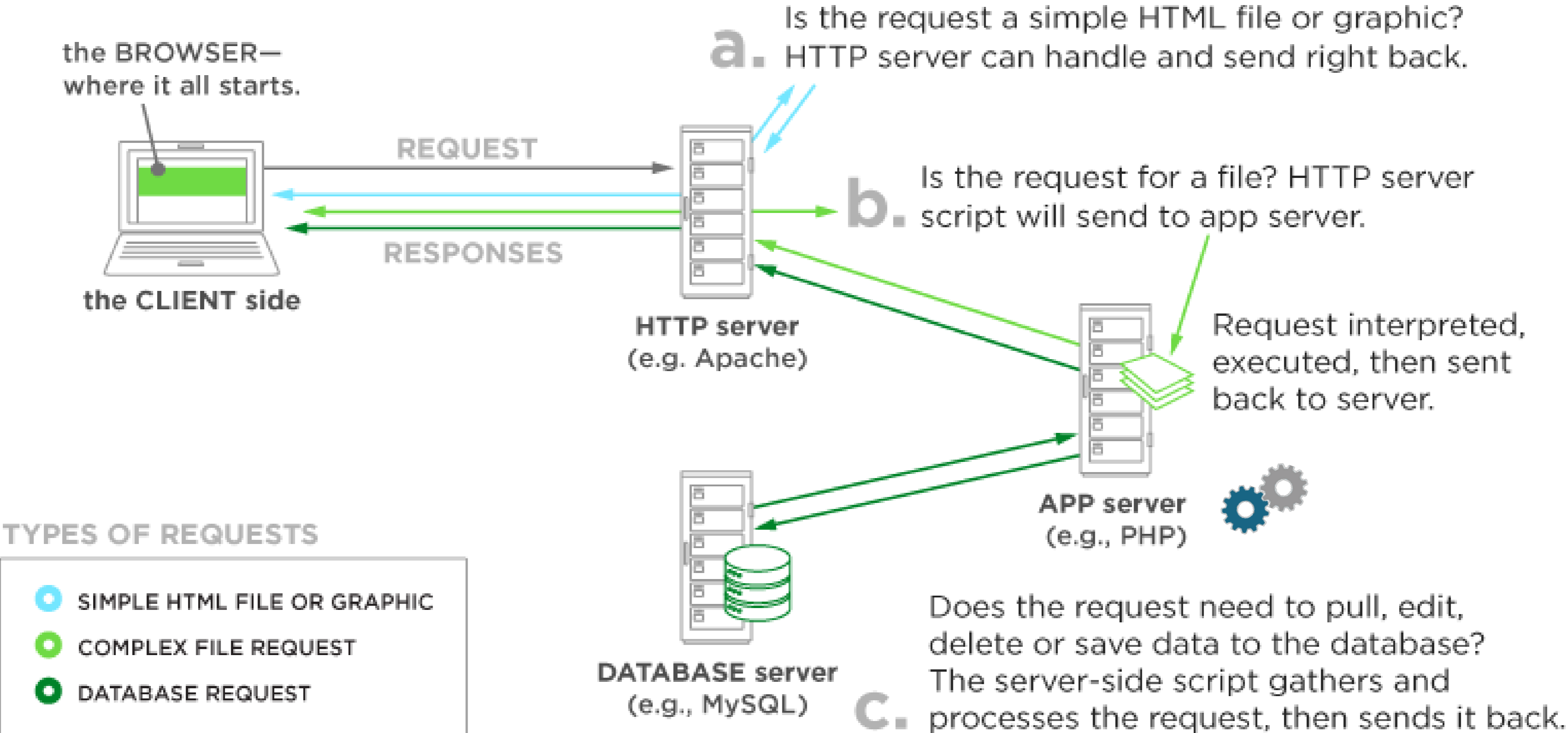
Lenguaje de programación*

* Opcionales, para sitios Web con páginas Web estáticas.

- **LAMP**: Linux, Apache, mySQL, PHP (o Python o Perl)
- **LAPP**: Linux, Apache, PostgreSQL, PHP
- **WAMP**: Windows, Apache, mySQL, PHP
- **WISA**: Windows, IIS, MS SQL, ASP.NET
- **MARS**: mySQL, Apache, Ruby, Solaris

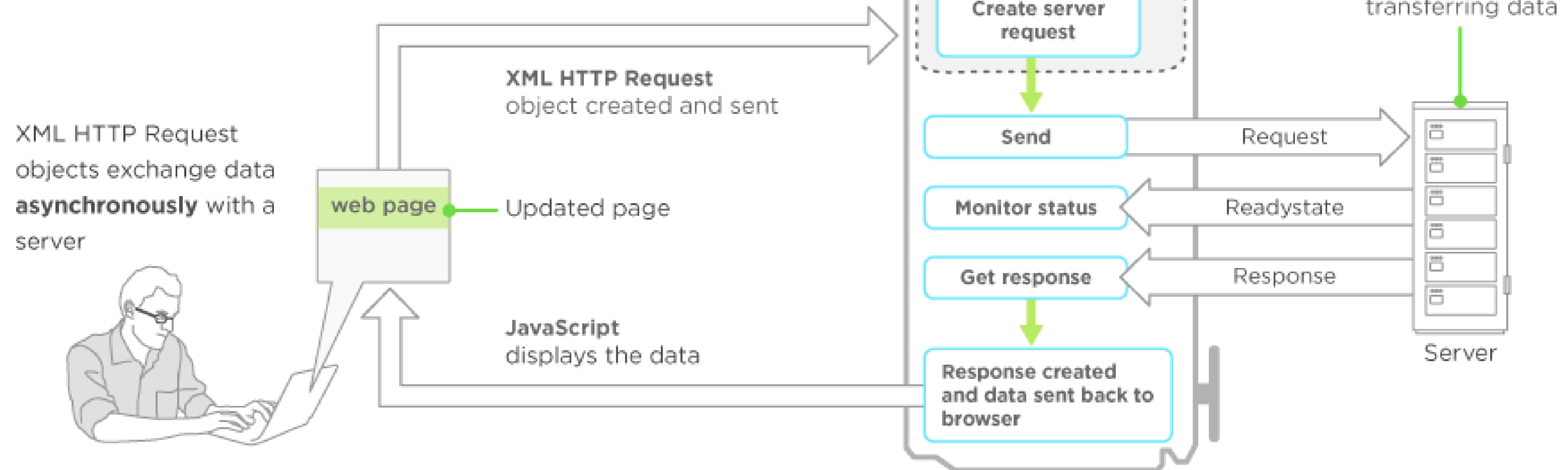
- Permiten realizar procesamientos del lado del servidor, por ejemplo:
 - Obtener o almacenar información en una base de datos.
 - Realizar cálculos.
 - Seleccionar el contenido que se deben presentar en la página.
- Existen muchos lenguajes para crear aplicaciones Web:
 - PHP
 - Ruby
 - Python
 - Java
 - .NET
 - Cold Fusion

- Los scripts se ejecutan en el servidor antes de enviar la respuesta al navegador.
- Esto puede ocasionar cargas excesivas en el servidor.
- El procesamiento de datos usualmente requiere que se refresque la página.
- AJAX y otros mecanismos del lado del cliente ayudan a mejorar la usabilidad, aprovechando las ventajas de lenguajes de programación del lado del servidor.



HOW AJAX CREATES A SEAMLESS USER EXPERIENCE

The AJAX engine runs in the background, initiating quick server requests and returning responses that refresh parts of a page that's already loaded into a browser. It allows parts of a page to update without reloading the entire site's code.

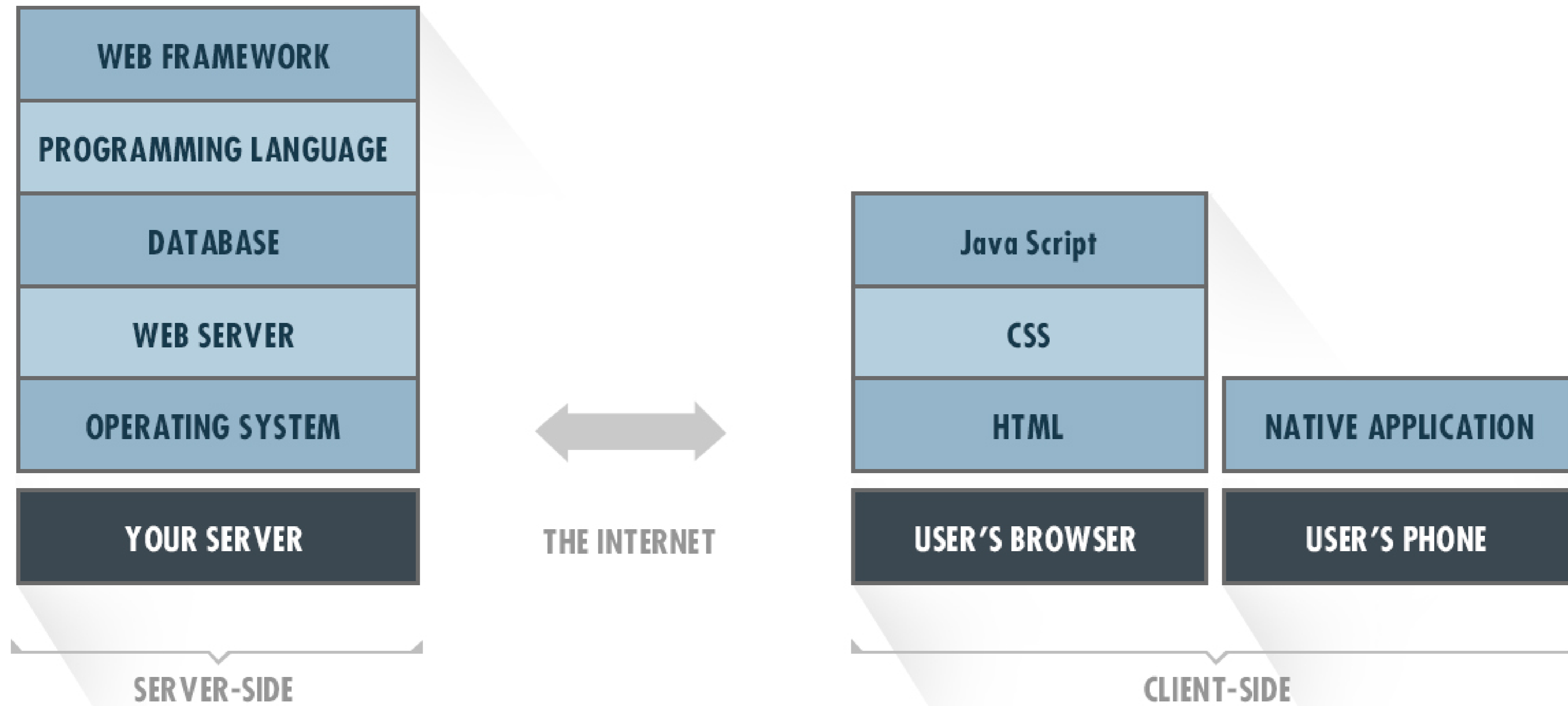


http://www.w3schools.com/ajax/ajax_examples.asp

Server side vs Client side

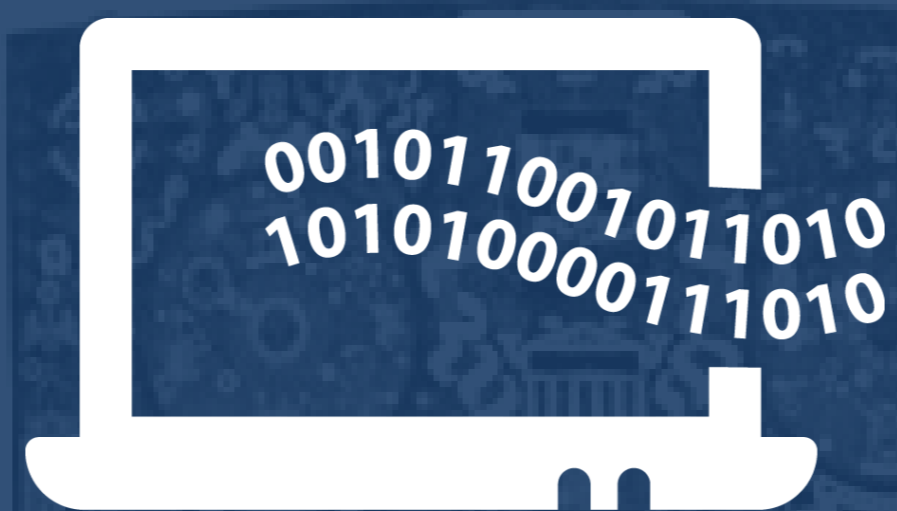
(AKA: Back-end vs Front-end)

34



Tecnologías Web





Fundamentos de aplicaciones Web

Karla A. Fonseca Márquez

APLICACIONES WEB



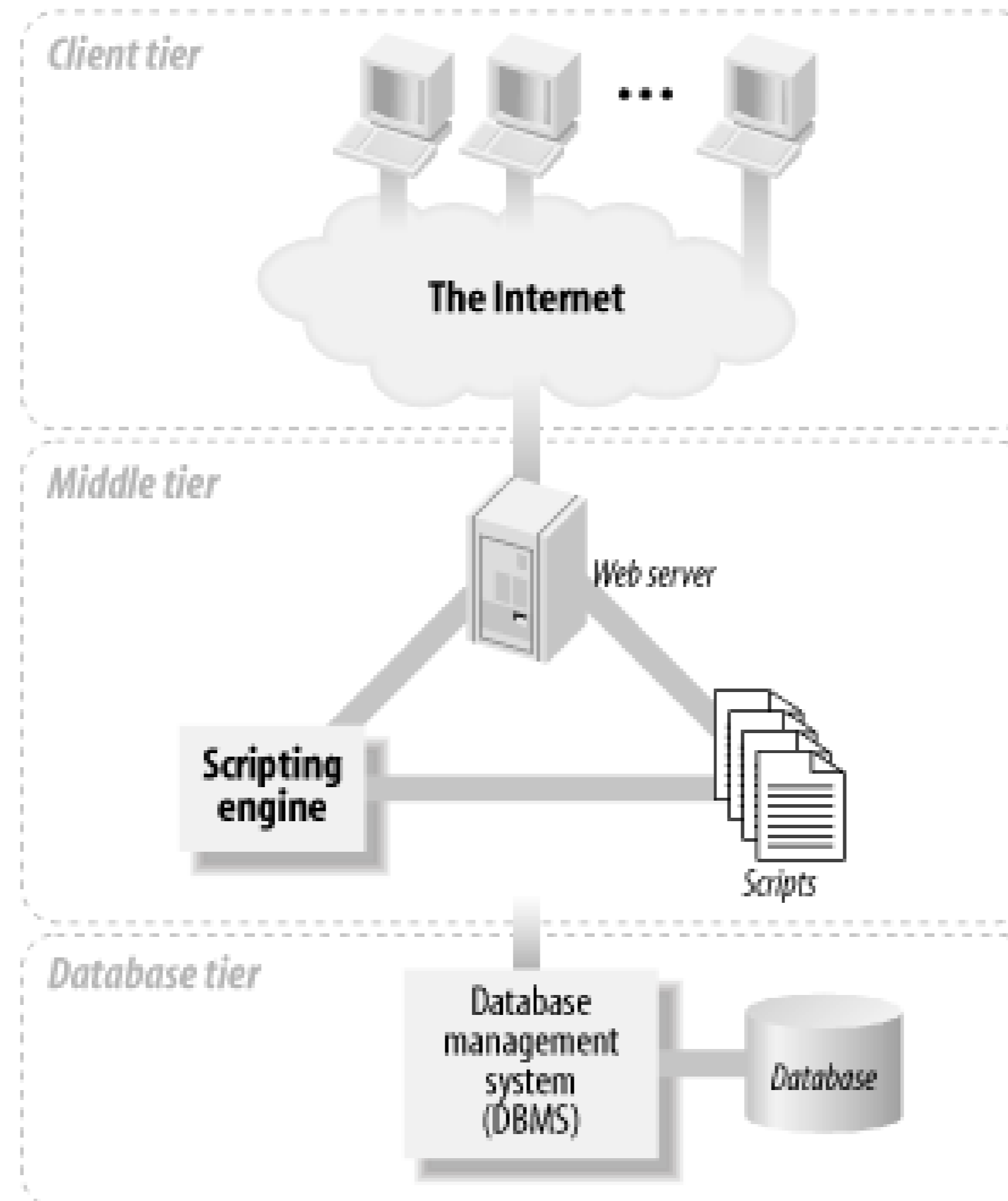
Aplicaciones Web

- Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor Web a través de Internet, mediante una interface o cliente, tradicionalmente un navegador.
- Aplicaciones que permiten capturar, procesar, almacenar y/o transmitir información para su uso inmediato y recurrente.
- Ejemplos típicos de aplicaciones Web: correo Web, páginas de autenticación o login, carritos de compra, sistemas manejadores de contenido, formularios de registro



Modelo 3 capas

4



Modelo 3 capas

5

- **Capa de presentación.** Esta capa es la responsable de proveer la interfaz de usuario y de controlar la interacción entre el usuario y el sistema. En aplicaciones Web usualmente se refiere al browser.
- **Capa de negocio.** Esta es la capa que implementa la funcionalidad principal del sistema y encapsula la lógica de negocio relevante. Esta capa permite la comunicación de las otras dos capas.
- **Capa de datos.** La capa de datos provee el acceso a los datos del sistema, así como a datos en sistemas relacionados.

Ventajas de las Aplicaciones Web

- Cualquier navegador sirve como cliente.
- Accesibles desde cualquier ubicación y dispositivo.
- Independencia del sistema operativo.
- Facilidad de actualización.
- No requieren instalación del lado del cliente.
- Se puede distribuir a miles de usuarios.
- Se puede acceder a datos de modo interactivo e inmediato.

Hoy en día, son fundamentales en la mayoría de organizaciones para potencializar su presencia en Web y para lograr relaciones ventajosas con clientes, usuarios y proveedores.

- **Basadas en Web (Navegador)**

Aplicaciones que se utilizan mediante un navegador, y que utilizan tecnologías como HTML, CSS y Javascript del lado del cliente, y lenguajes de programación, bases de datos y otros servicios del lado del servidor.

- **Basadas en Cliente**

Aplicaciones que no se ejecutan en un navegador, sino utilizando algún otro cliente que debe instalarse en el equipo del usuario. En ejecución, el cliente se conecta a un servidor Web para sus funciones.

- **Aplicaciones móviles**

Aplicaciones nativas desarrolladas específicamente para plataformas de dispositivos móviles, como IOS y Android.

- **Servicios Web**

Aplicaciones que intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web.

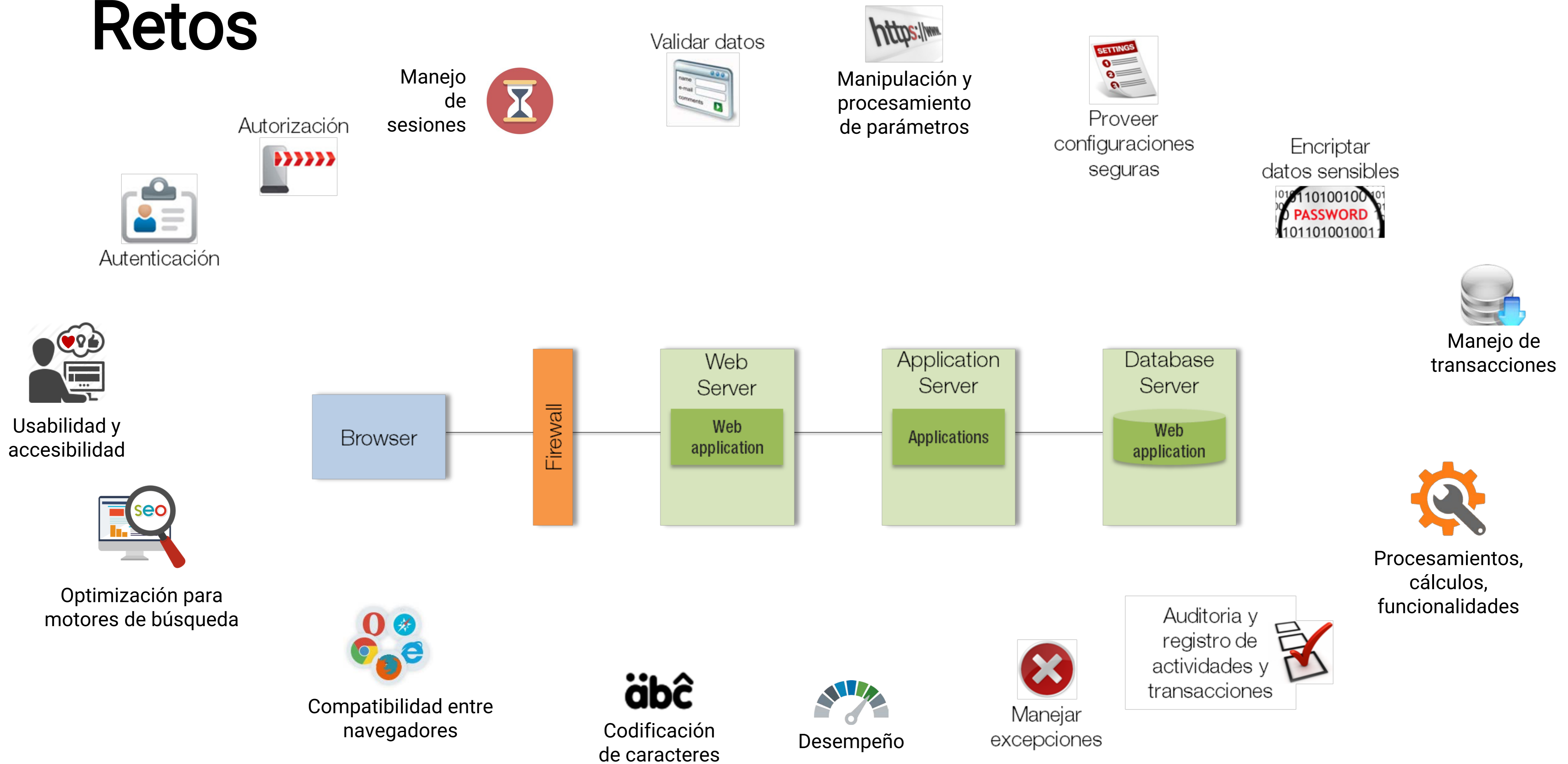
Atributos de calidad en aplicaciones Web

8



Tarea:
Reporte de la descripción de cada uno de estos atributos de calidad en el contexto de aplicaciones Web.

Retos



- HTTP es un protocolo sin estado.
- El servidor Web trata cada solicitud HTTP de página como una solicitud independiente.
- De forma predeterminada, el servidor no retiene información alguna sobre los valores de las variables que se utilizan durante las solicitudes realizadas.
- Las páginas Web no persisten datos dinámicos por si mismas.
- En varias ocasiones es necesario persistir los datos entre diferentes páginas y/o módulos de una aplicación Web.

- Las aplicaciones Web, dependiendo de la plataforma y el lenguaje de programación, poseen un mecanismo que permite lograr la persistencia de datos entre solicitudes que se realicen entre un mismo cliente y el servidor
- Una sesión es la secuencia de páginas que un usuario visita en un sitio web. Desde que entra en nuestro sitio, hasta que lo abandona.
- El manejo de sesiones es un mecanismo que nos permite preservar información de un visitante a lo largo de sus diferentes accesos a nuestras aplicaciones.

- Las sesiones son almacenadas en el servidor de tal manera que al ser solicitadas por el cliente puedan ser recuperadas.
- Cada cliente tiene un identificador de su propia sesión

- Una cookie son datos que se almacenan en el cliente.
- Se trata de pares de nombre-valor que se acompañan de una fecha de caducidad.
- Los datos de una cookie se pasan al servidor como parte de cada solicitud HTTP, de forma que la aplicación Web tiene acceso a esos datos.
- La aplicación puede manipular los datos que le interesen en cada momento.
- Nada nos garantiza que los datos se almacenen de forma segura. Usualmente se guardan sin ningún tipo de protección.
- No se debe almacenar información sensible en una cookie.

Principios para el desarrollo de software

- **KISS** – Keep It Simple Stupid!

Muchos sistemas funcionan mejor si se mantienen simples en lugar de hacerlos más complejos.

La simplicidad debe ser una meta en el diseño y se debe evitar complejidad innecesaria.

Una solución simple es mejor que una compleja, incluso si la solución parece estúpida.

- **DRY** – Don't Repeat Yourself

Cada pieza de conocimiento debe tener una única representación en un sistema y ésta debe ser no ambigua y confiable.

Duplicar aumenta las posibilidades de error. En caso de un cambio, todas las instancias de una pieza duplicada debe cambiarse. Esto implica el riesgo de olvidar modificar alguno de los duplicados o modificarlo erróneamente.

- **YAGNI** – You Ain't Gonna Need It

No implementes una característica o funcionalidad hasta que exista la necesidad.

Las funcionalidades que parecen útiles pero no son realmente necesarias en el momento es mejor dejarlas para cuando se demuestre que son realmente necesarias.

Always code as if the guy who
ends up maintaining your code
will be a violent psychopath who
knows where you live. - Martin Golding

ANY FOOL

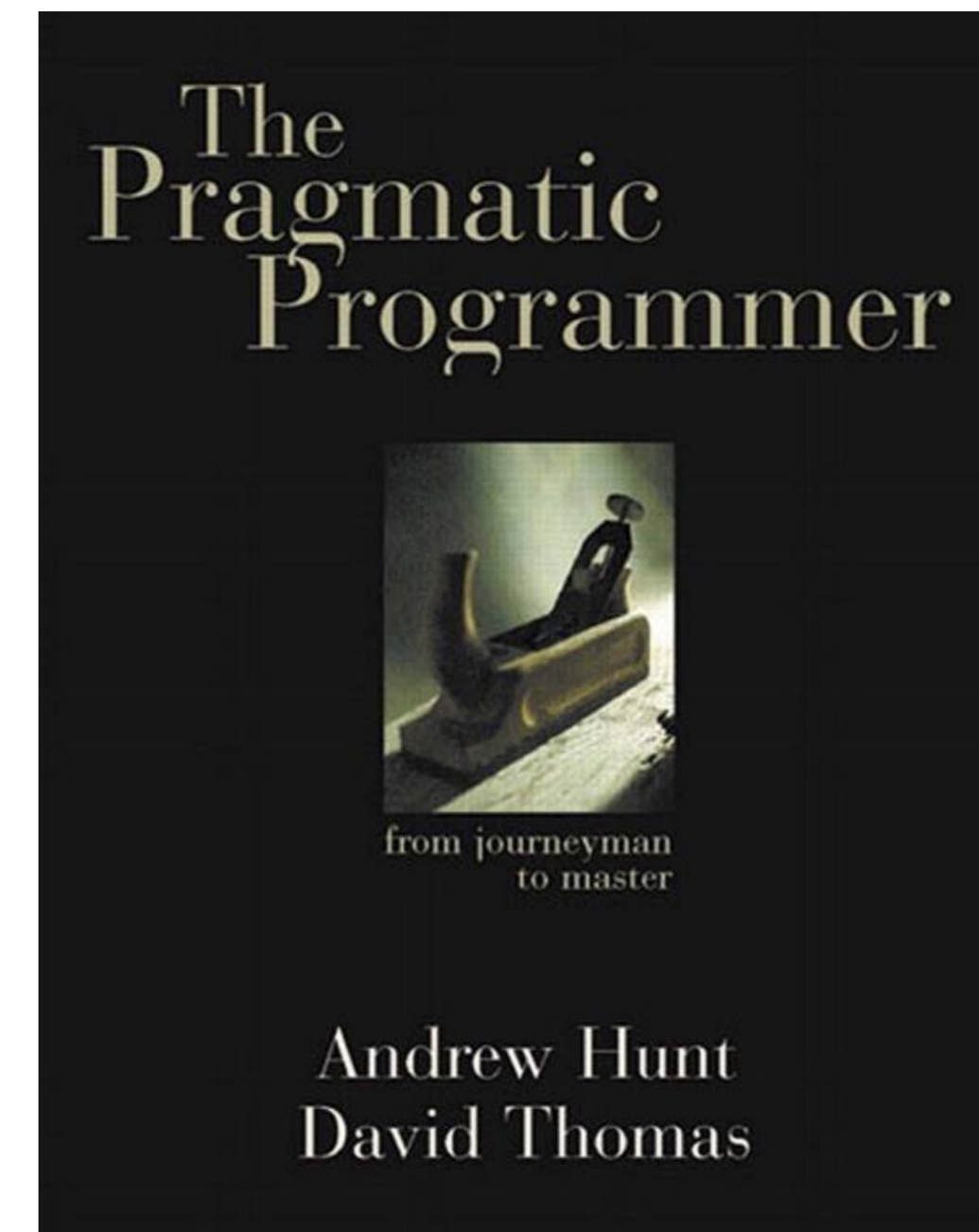
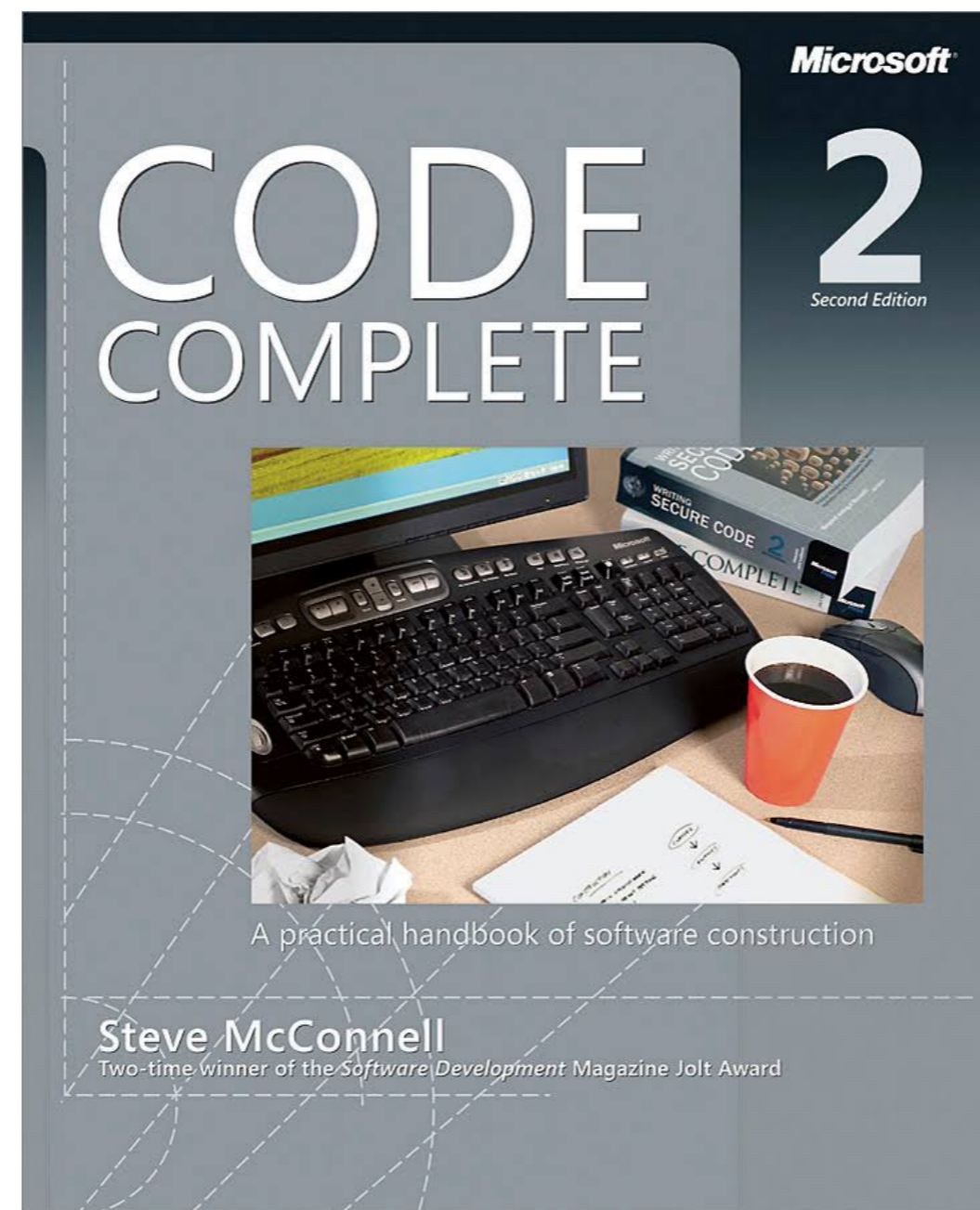
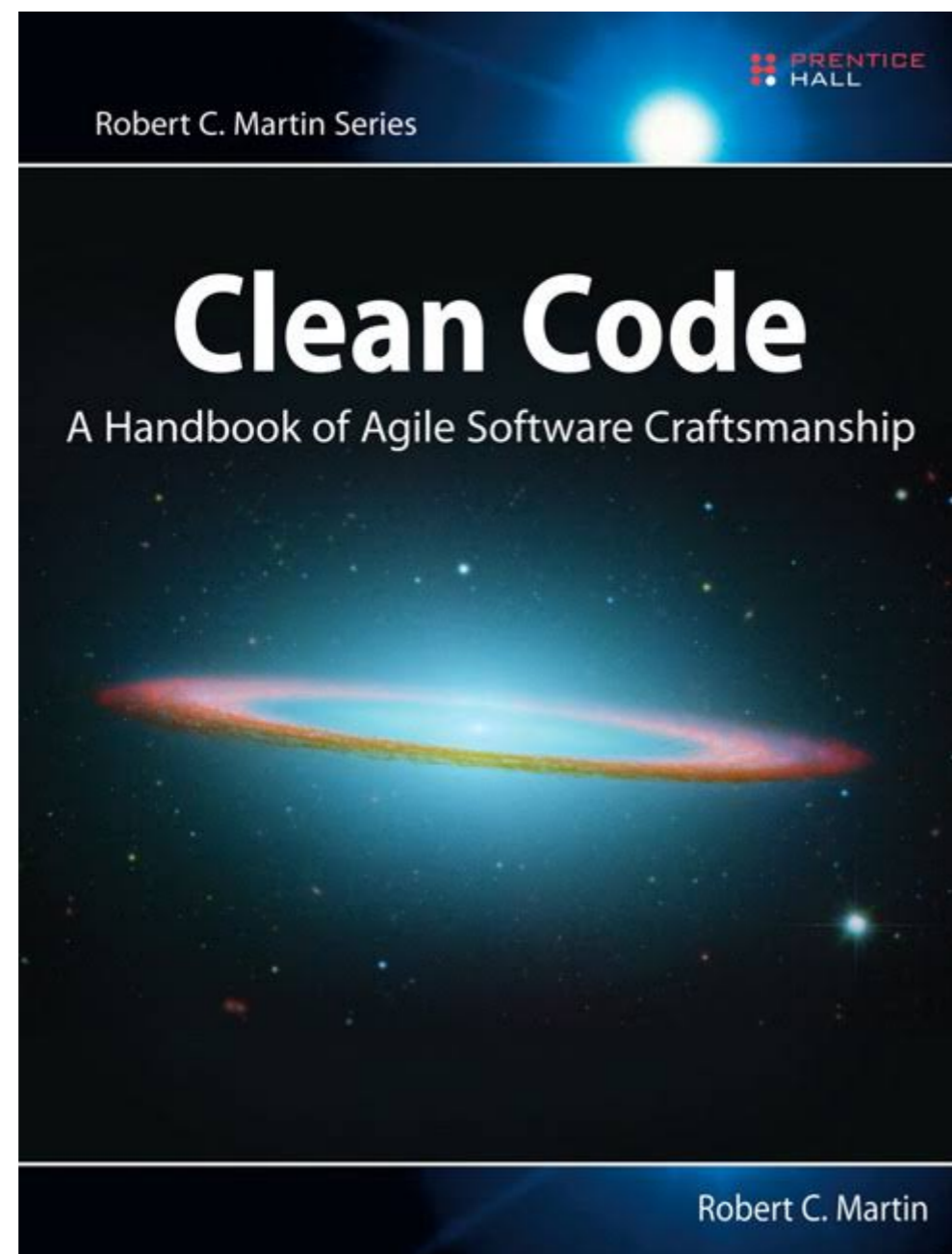
can write code that a computer
can understand.



GOOD PROGRAMMERS

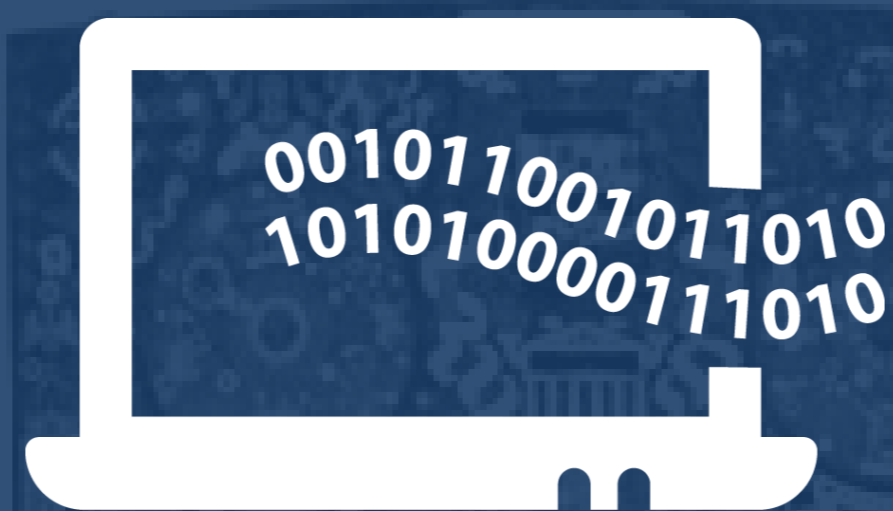
write code that humans
can understand.

– Martin Fowler



APLICACIONES WEB





Fundamentos de aplicaciones Web

Karla A. Fonseca Márquez



Qué es un framework

- “es una abstracción en la cual un software que provee funcionalidad genérica puede ser modificado para agregar código generado por el usuario, con el objetivo de crear una aplicación específica”
- “es un ambiente universal y reutilizable que provee funcionalidad particular como parte de una plataforma de software para facilitar el desarrollo de aplicaciones de software, productos y soluciones.”
- Generalmente incluyen programas, compiladores, librerías de código, herramientas e interfaces de programación (API)

 duda

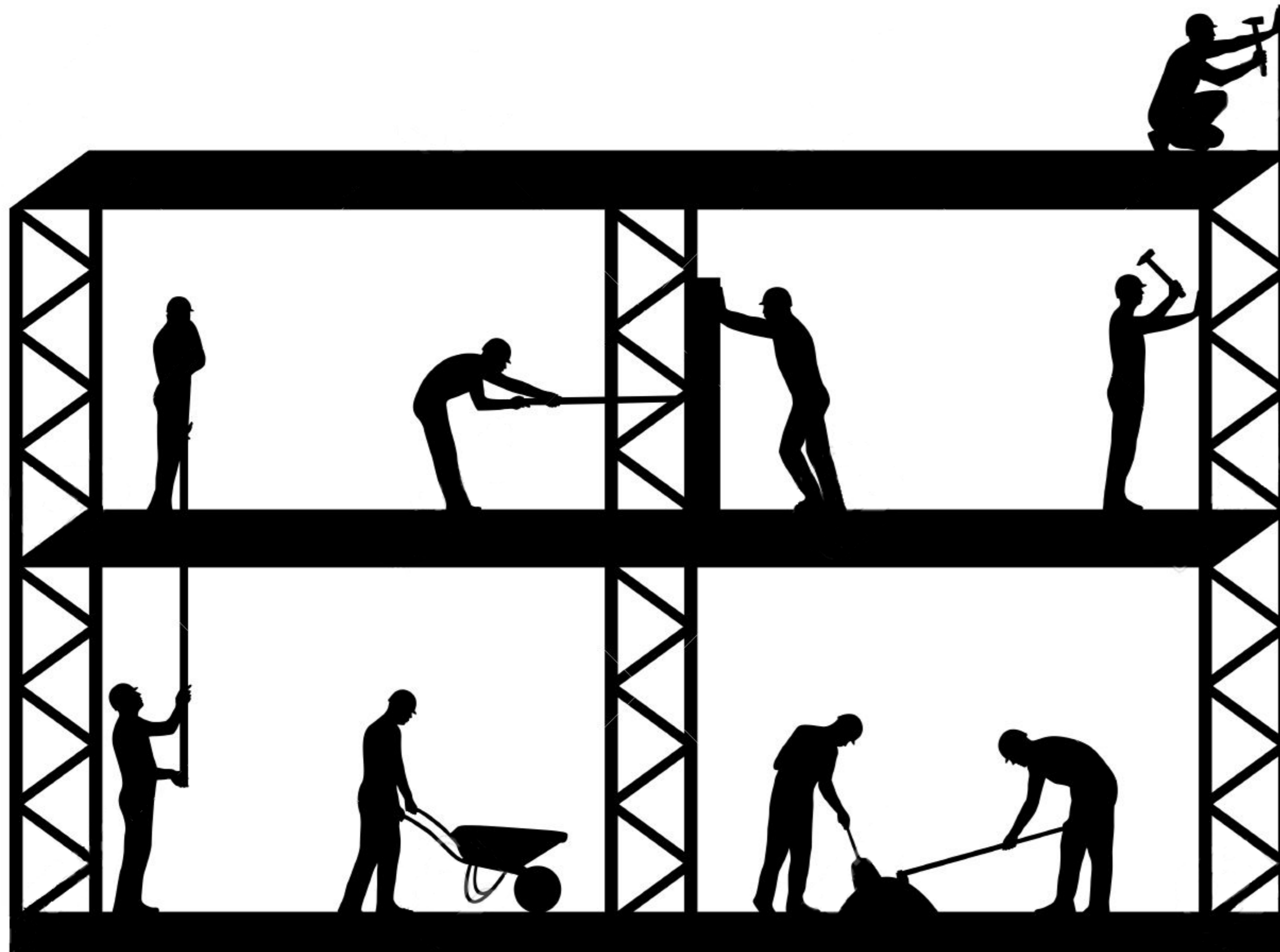
¿cuál es la diferencia entre una librería y un framework?

Librerías

- Se pueden ocupar individualmente o en conjunto
- Sólo se utilizan cuando es necesario.
- No hay restricciones, fuera de la sintaxis



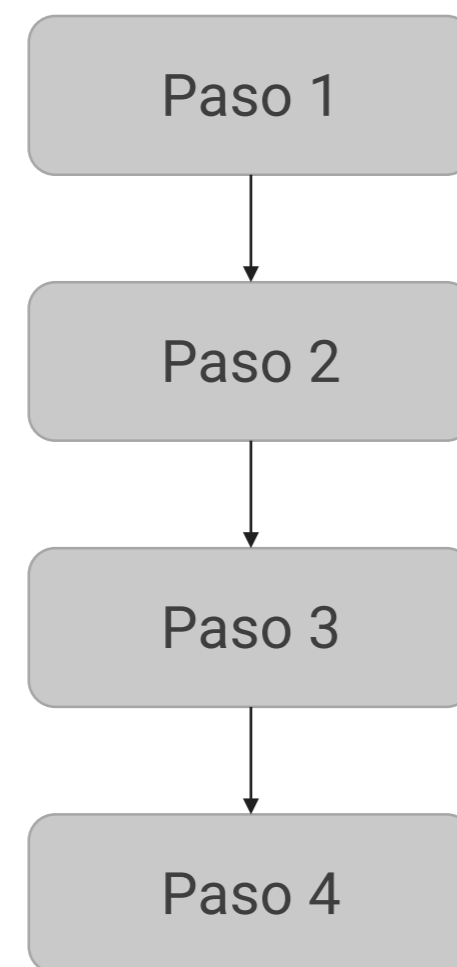
Framework



- Define la estructura del software
- Desarrollo rápido
- Fomenta la reutilización de código y el uso de patrones.
- Facilidad de mantenimiento
- Inversión de control
- Extensibilidad
- Contienen utilerías y librerías
- Consideran aspectos de seguridad
- Promueve la división de tareas
- Enfoque en el proyecto

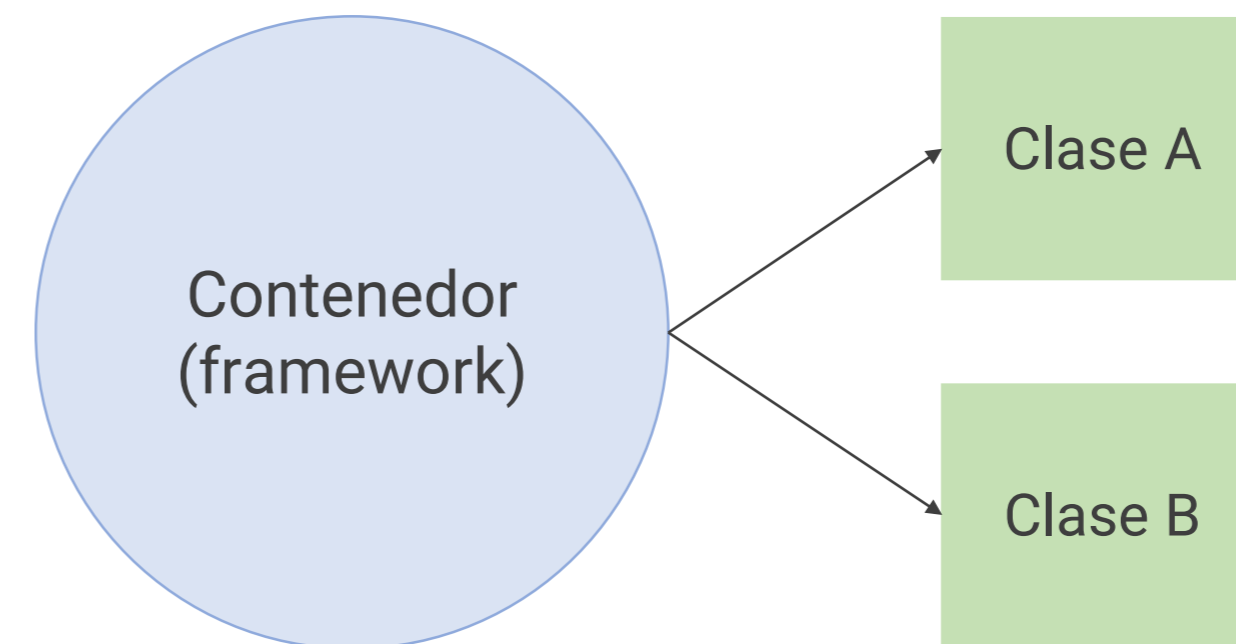
Inversión de control

Técnica de programación donde el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales



tradicional

VS

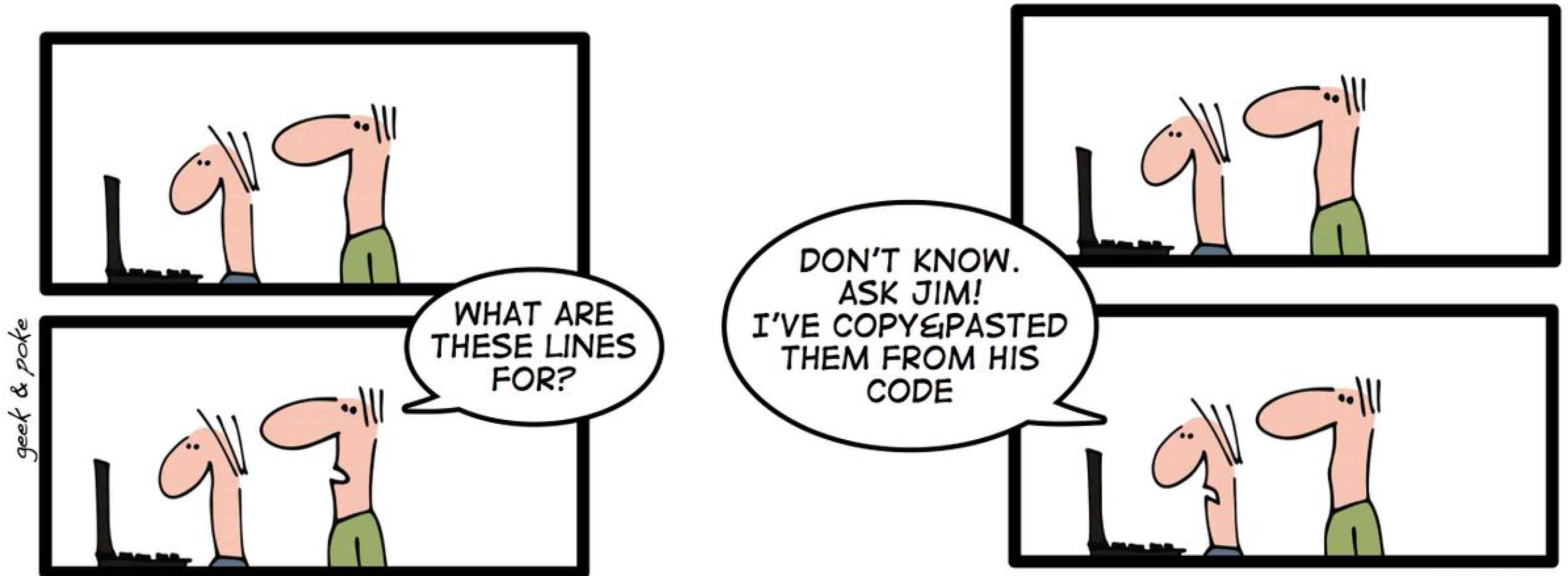


IoC

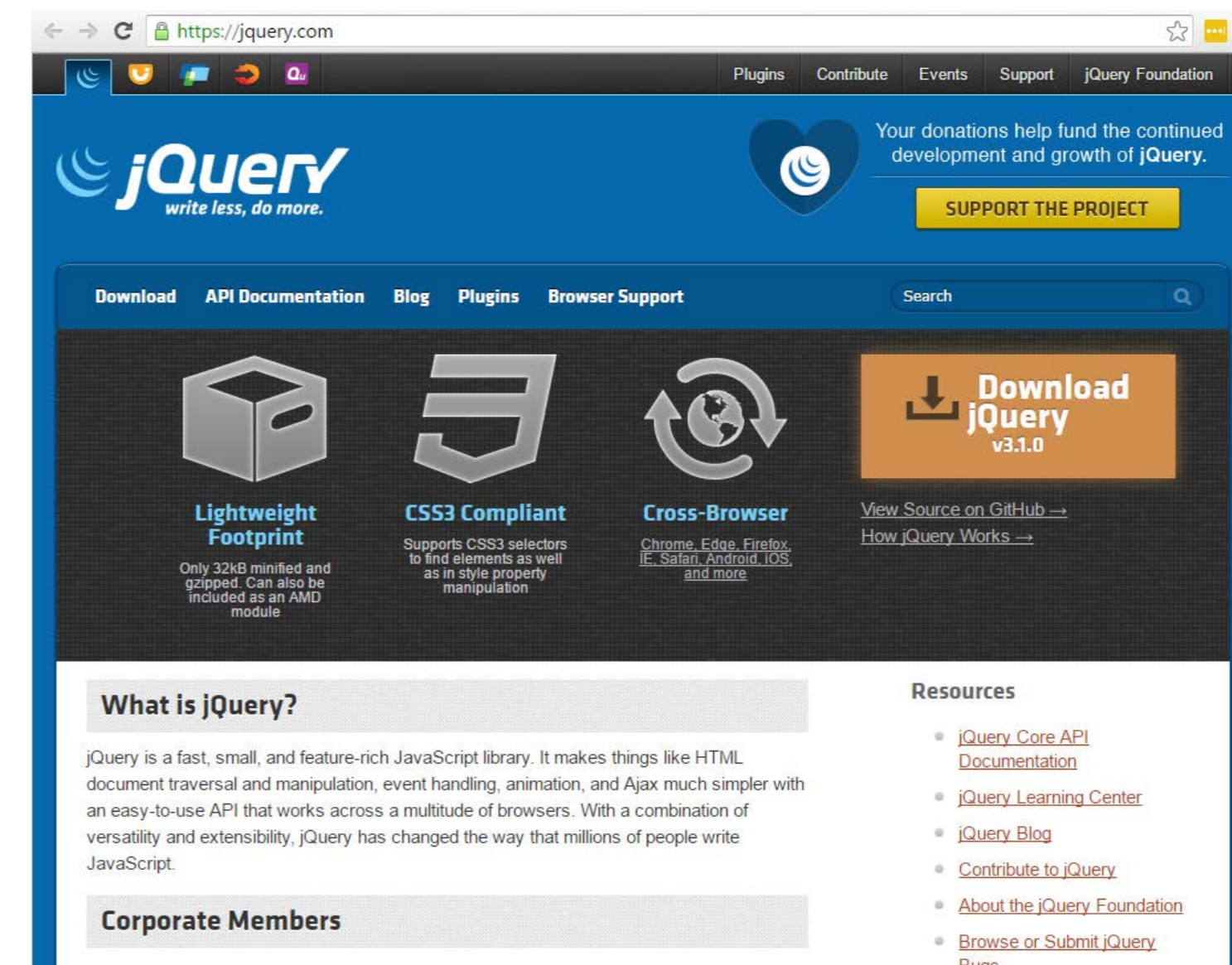


reuse

SIMPLY EXPLAINED



- Colección de métodos y funciones javascript que facilitan la ejecución de diversas tareas.
- Extienden la funcionalidad de javascript nativo.
- Hay librerías desde las más simples como Date.js hasta las que abarcan aspectos muy diversos como JQuery.



How many roads must a man walk down?



The answer is
blowing in the wind

Native JavaScript

```
...if (center2centerDistance < radiusSum)
{
  // tip is left of select
  if ((objectOffsets.get('leftCenter') <=
selectOffsets.get('leftCenter') &&
objectOffsets.get('right') < selectOffsets.get('left')) ||
  // tip is right of select
  (objectOffsets.get('leftCenter') >
selectOffsets.get('leftCenter') && objectOffsets.get('left')
> selectOffsets.get('right')) ||
  // tip is above select
  (objectOffsets.get('topCenter') <=
selectOffsets.get('topCenter') &&
objectOffsets.get('bottom') < selectOffsets.get('top')) ||
  // tip is below select
  (objectOffsets.get('topCenter') >
selectOffsets.get('topCenter') && objectOffsets.get('top') >
selectOffsets.get('bottom'))
  {
    thisElement.hideList.remove(in_object.id);
    if (!thisElement.hideList.length)
    {
      thisElement.style.visibility = 'visible';
      if (domLib_isKonq)
      {
        thisElement.style.display = '';
      }
    }
  }
  else
  {
    thisElement.hideList.set(in_object.id, true);
    thisElement.style.visibility = 'hidden';
    if (domLib_isKonq)
    {
      thisElement.style.display = 'none';
    }
  }
}....
```


jQuery

```
$(.tooltip).fadeIn('slow');
```

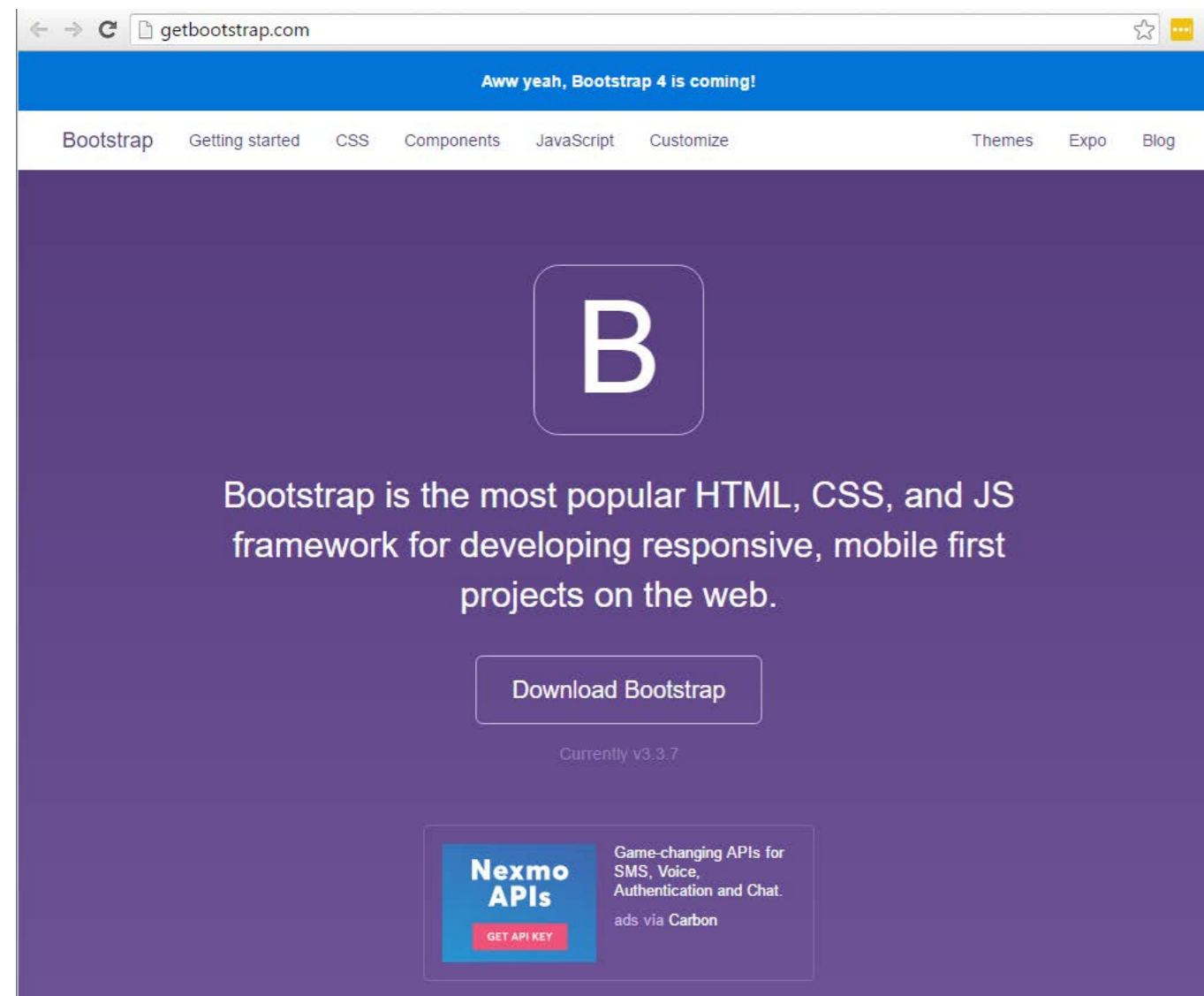
How many roads must a man walk down?



The answer is
blowing in the wind

Frameworks Front-End

12



getbootstrap.com

Aww yeah, Bootstrap 4 is coming!

Bootstrap Getting started CSS Components JavaScript Customize Themes Expo Blog

B

Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.3.7

Nexmo APIs Game-changing APIs for SMS, Voice, Authentication and Chat. ads via Carbon

Designed for everyone, everywhere.

Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes.



Preprocessors

Bootstrap ships with vanilla CSS, but its source code utilizes the two most popular CSS preprocessors, *Less* and *Sass*. Quickly get started with precompiled CSS or build on the source.



One framework, every device.

Bootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries.

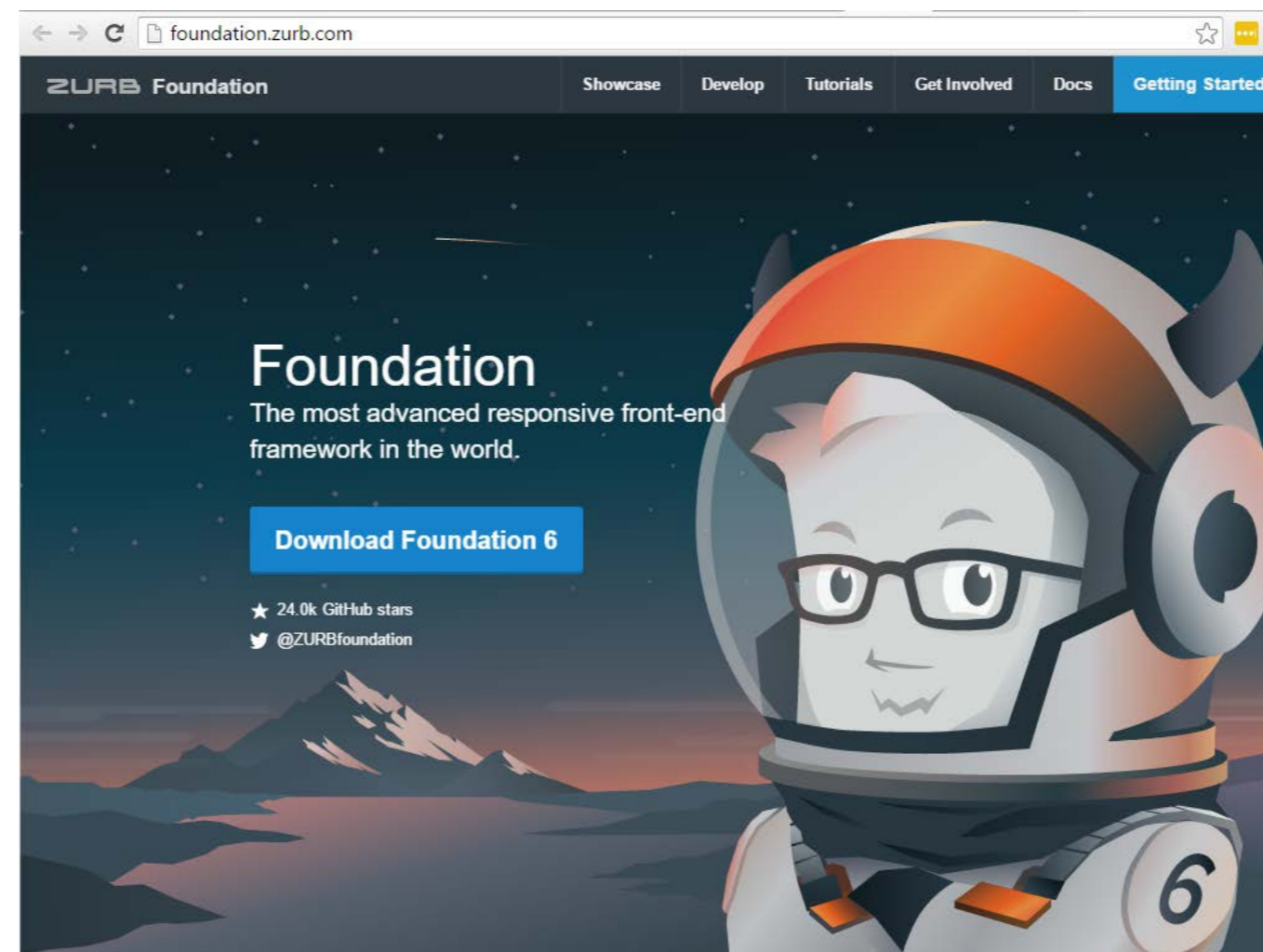


Full of features

With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.

Bootstrap is open source. It's hosted, developed, and maintained on GitHub.

[View the GitHub project](#)



foundation.zurb.com

ZURB Foundation Showcase Develop Tutorials Get Involved Docs Getting Started

Foundation

The most advanced responsive front-end framework in the world.

[Download Foundation 6](#)

★ 24.0k GitHub stars
@ZURBfoundation

Build products, apps and services

Foundation is the professional choice for designers, developers and teams.



Semantic

Everything is semantic. You can have the cleanest markup without sacrificing the utility and speed of Foundation.



Mobile First

You can build for small devices first. Then, as devices get larger and larger, layer in more complexity for a complete responsive design.



Customizable

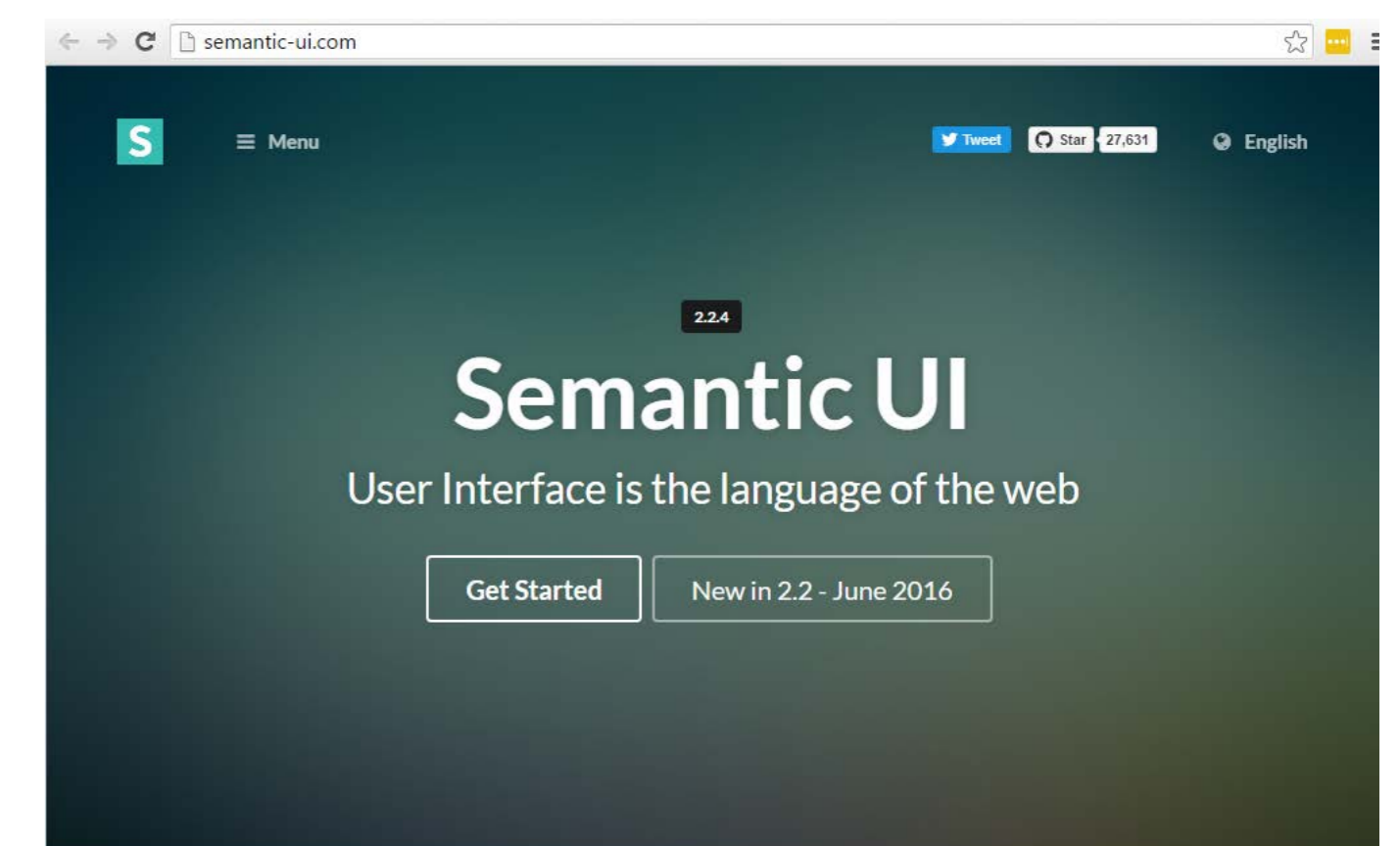
You can customize your build to include or remove certain elements, as well as define the size of columns, colors, font size and more.



Professional

Millions of designers and developers depend on Foundation. We have business support, training and consulting to help grow your product or service.

[Learn more about why Foundation fits in your company →](#)



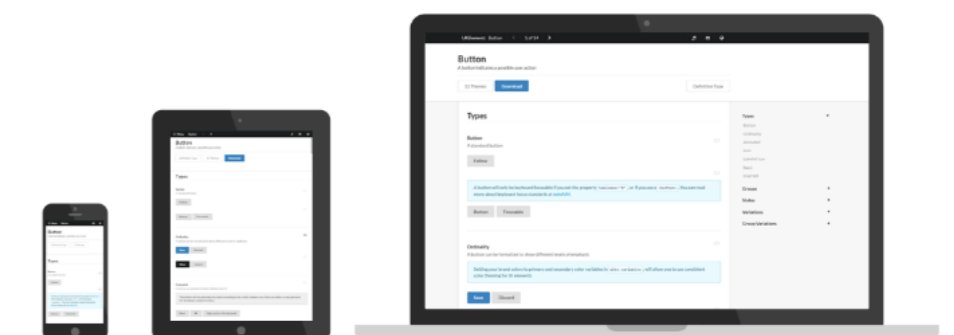
semantic-ui.com

S Menu [Tweed](#) [Star](#) 27,631 English

Semantic UI

User Interface is the language of the web

[Get Started](#) [New in 2.2 - June 2016](#)



Design Beautiful Websites Quicker

Semantic is a development framework that helps create beautiful, responsive layouts using human-friendly HTML.

Frameworks Javascript

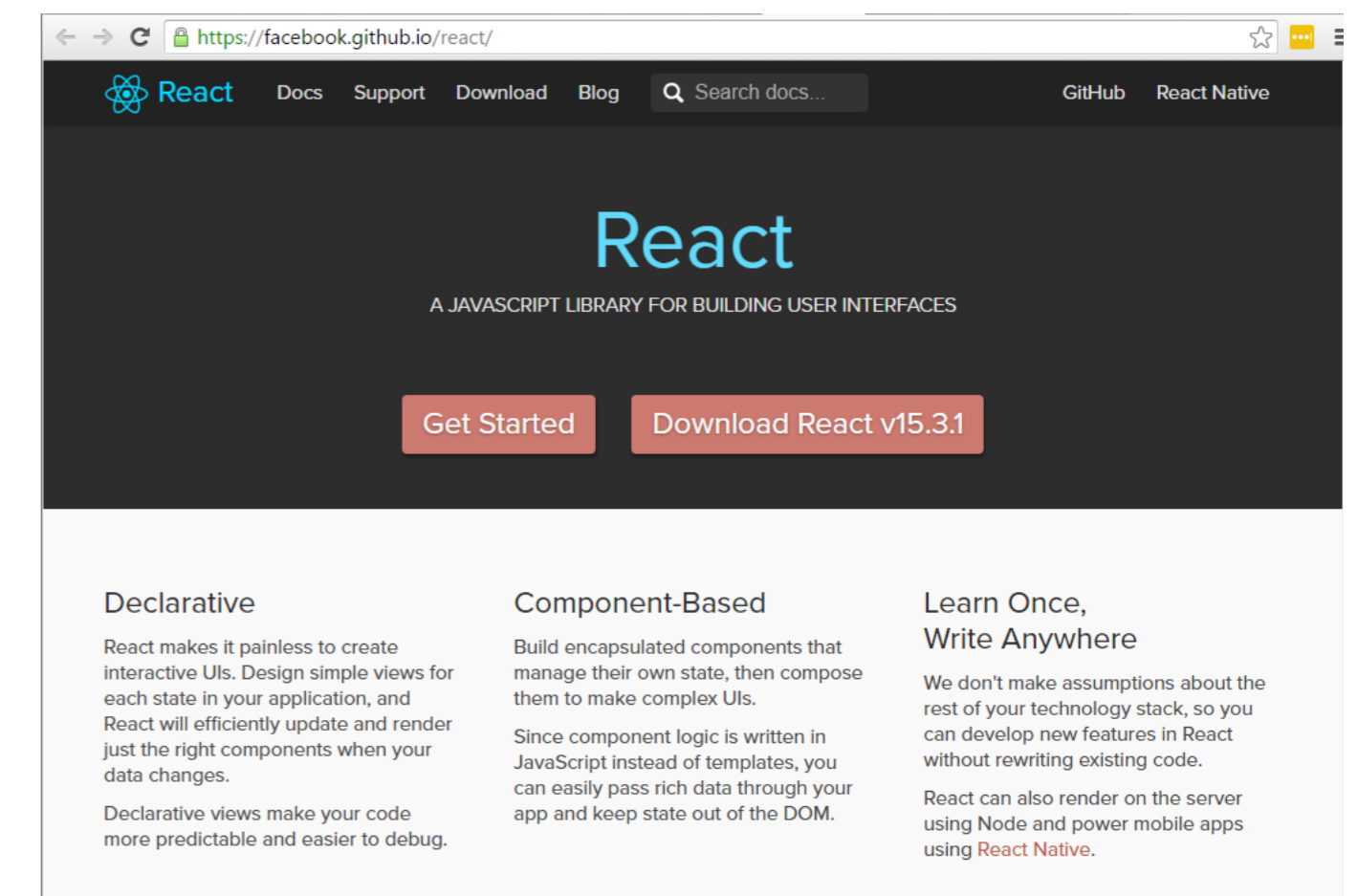
13



The screenshot shows the AngularJS website at <https://angularjs.org>. The page features the AngularJS logo with the text "by Google" and the tagline "HTML enhanced for web apps!". Below this, there are two main buttons: "Download AngularJS 1 (1.5.8 / 1.2.30)" and "Try the new Angular 2". There are also links for "View on GitHub" and "Design Docs & Notes". At the bottom, there are social media follow buttons for GitHub, Twitter (@angularjs), and a "Learn Angular in your browser for free!" button.



The screenshot shows the Backbone.js website at <https://backbonejs.org>. The page features the Backbone.js logo and the text "BACKBONE.JS". Below the logo, there is a description: "Backbone.js gives structure to web applications by providing **models** with key-value binding and custom events, **collections** with a rich API of enumerable functions, **views** with declarative event handling, and connects it all to your existing API over a RESTful JSON interface." There are also links for "Getting Started", "Events", and "Model". At the bottom, there is a "Downloads & Dependencies" section with three buttons: "Development Version (1.3.3)", "Production Version (1.3.3)", and "Edge Version (master)".



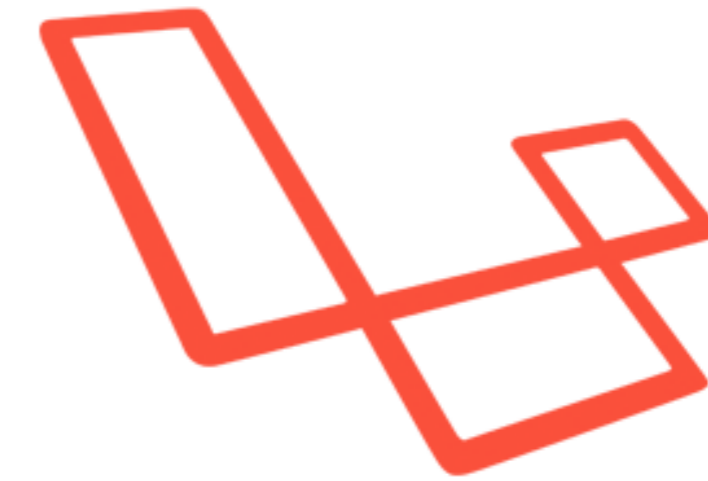
The screenshot shows the React website at <https://facebook.github.io/react/>. The page features the React logo and the text "React A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES". Below this, there are two main buttons: "Get Started" and "Download React v15.3.1". At the bottom, there are three columns of text: "Declarative", "Component-Based", and "Learn Once, Write Anywhere".

Frameworks PHP

CakePHP



CodeIgniter



Laravel

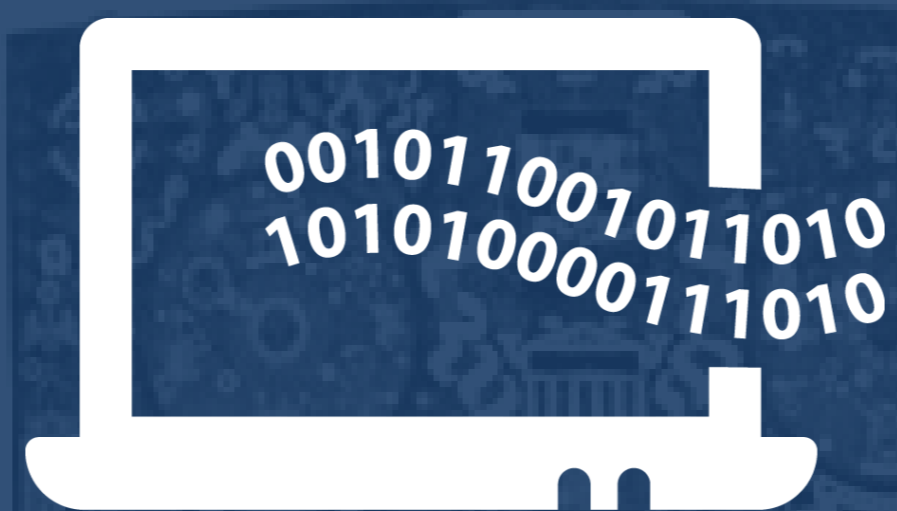


yii framework



Symfony





Fundamentos de aplicaciones Web

Karla A. Fonseca Márquez

Tipos de contenido en internet

2

- Texto plano
- Estándares Web
- Lenguajes del lado del servidor
- Formatos que requieren otras aplicaciones o plugins

- Los archivos de texto plano contienen **caracteres imprimibles** que pueden ser leídos y comprendidos por humanos (a diferencia de los archivos binarios)

`Fiqwdl9%467abe` vs `DrawingType=UMLActivityDrawing`

- El texto plano **carece** completamente de formato tipográfico (Fuente, negritas, subrayados, colores, etc).
- Los archivos están compuestos por bytes que representan caracteres como letras, números, signos de puntuación y algunos caracteres de control como tabulaciones, saltos de línea y retornos de carro.
- Los caracteres se pueden codificar de distintos modos dependiendo de la lengua usada. Algunas codificaciones son: ASCII, UTF-8, ISO8859-1 (o Latin 1)

- El **texto** está conformado por **caracteres**. Por ejemplo, la letra latina á o el ideograma chino 讀 o del carácter hindú ह.
- Los caracteres que sirven al mismo propósito se agrupan en un **set de caracteres** (character set)
- A cada caracter dentro de un set se le asigna un numero o código para evitar ambigüedades.
- Un caracter se almacena en una computadora **como uno o más bytes**.
- Una **codificación de caracteres** (o character encoding) actúa como una llave para “decifrar” el código. Se refiere a un mapeo entre los bytes en la computadora y los caracteres en un set. Sin la llave, los datos parecen basura.

Author: Guðrún Guðmundsdóttir. Title: Introduction to character encoding (文字符号化入門). Copyright © 2004-2007 W3C® (MIT, ERCIM, Keio).

Author: Guðrún Guðmundsdóttir. Title: Introduction to character encoding (æ—ǰ—ç—|ǰ—ǰœ—ǰ...¥é—€). Copyright © 2004-2007 W3C® (MIT, ERCIM, Keio).

Codificación de caracteres en HTML

- Actualmente, lo recomendado es siempre escoger la codificación de caracteres **UTF-8** para los datos. Es el default para HTML5.
- Es necesario declarar la codificación utilizada en los documentos Web (por ejemplo en HTML)

```
<head>  
<meta charset="UTF-8">  
<meta name="description" content="Free Web tutorials">  
<meta name="keywords" content="HTML,CSS,XML,JavaScript">  
<meta name="author" content="Hege Refsnes">  
</head>
```

- Es importante aclarar que al declarar una codificación en el documento no modifica los bytes; se necesita **guardar el archivo** en la misma codificación también.
- Se debe revisar en la configuración de los editores de código que se utilicen la codificación en el que se guardan los archivos.
- También es importante revisar la **configuración del servidor Web**, ya que en ocasiones están configurados para enviar headers con una cierta codificación.
- Otro punto a considerar es la revisión de las **comunicaciones entre componentes** del sistema. Por ejemplo, hay que revisar que la conexión a la base de datos, se realice con la misma codificación; las tablas y datos de la base de datos deben crearse con la codificación correcta.

Ejemplo con Notepad++

8

```
<!DOCTYPE html>
<html>
<head>

    <meta charset="UTF-8">

</head>
<body>

    <h1>Prueba de charset</h1>

    <p>áéíóúñü</p>

</body>
</html>
```

- Delimitados. Por caracteres como comas o tabulaciones.

field 1	2	3	4	5
1 6 - J A N - 1 9 9 4	0 0	/	1 :	2 7 8 . 9
1 6 - F E B - 1 9 9 4	0 0	/	2 :	2 8 0 . 0
1 6 - M A R - 1 9 9 4	0 0	/	3 :	2 7 8 . 9

- De ancho fijo. Cada columna mide un determinado numero de caracteres

field 1	2	3	4	5
1 6 - J A N - 1 9 9 4	0 0	/	1 :	2 7 8 . 9
1 6 - F E B - 1 9 9 4	0 0	/	2 :	2 8 0 . 0
1 6 - M A R - 1 9 9 4	0 0	/	3 :	2 7 8 . 9

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

- CSV
 - Es un formato especial para archivos de texto plano.
 - Cada registro se encuentra en una línea.
 - Los campos se encuentran delimitados por comas.
 - Si un campo contiene comas, el valor del campo debe encerrarse entre comillas dobles.
 - Si un campo contiene comillas dobles, las comillas deben “escaparse” usando doble dobles comillas:

3829,The Great Gatsby,"Texto que contiene una coma, y además ""Comillas""")

- Existen formatos especiales o notaciones basadas en texto y que tienen como gran ventaja que son independientes de la posición del texto en el archivo:
 - XML
 - JSON
- Fueron creados para facilitar el intercambio de información entre distintas aplicaciones.
- Los datos se encuentran estructurados, aportando significado y facilidad de procesamiento.

https://www.w3schools.com/js/js_json_xml.asp

- Una expresión regular es una secuencia de caracteres que forma un patrón de búsqueda.
- Su utilidad principal es buscar y manipular textos.
- En programación un uso común es validar ciertos formatos de textos.

<http://regexr.com/>

