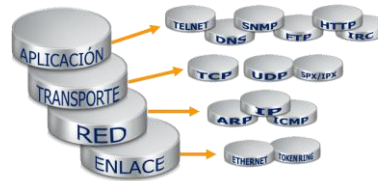


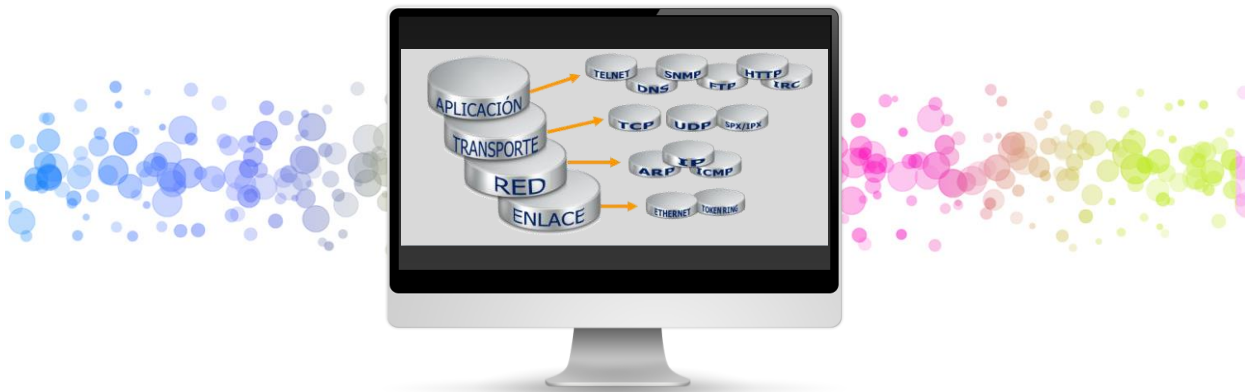
Temas de Hoy

Redes de Comunicación

TCP/IP - HTTP



Arquitectura de Aplicaciones Web - 2°C 2020

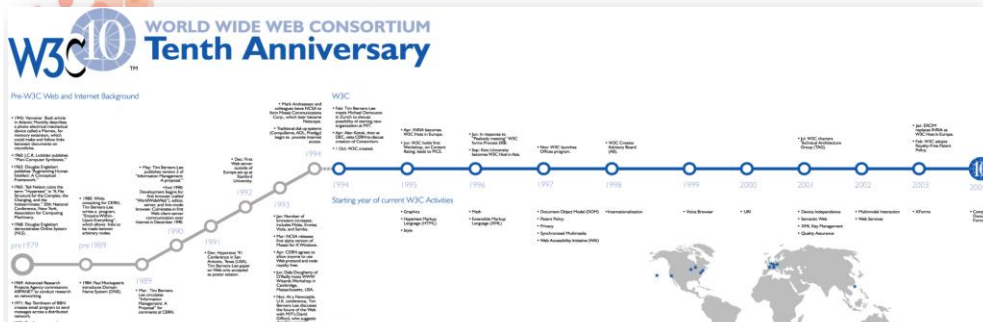


WWW, TCP, HTTP, HTML, etc...

Un poco de historia

• World Wide Web (WWW)

- Inventada por Tim Berners Lee en 1989
- Mientras trabajaba European Organization for Nuclear Research (CERN)
- <http://www.w3.org/Consortium/history.html>
- <http://www.w3.org/History/1989/proposal.html>



Terminología básica

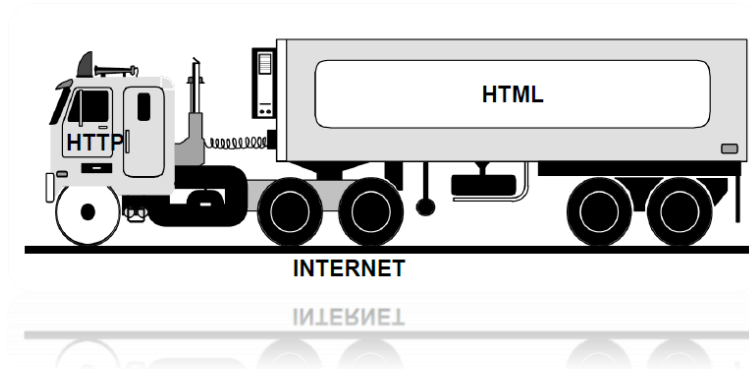
La WWW es una combinación de 4 ideas básicas

- *Hipertexto*
 - Habilidad de navegar desde un documento a otro a través de conexiones → “hiperenlaces”
- *Identificadores de Recursos*
 - Permite encontrar un recurso particular (un documento, imagen) en la red a través de dicho identificador
 - UniformResourceIdentifier(URI)
 - UniformResourceLocator(URL)
- *Modelo cliente servidor*
 - Un cliente software demanda servicios o recursos a un servidor software
- *Un lenguaje de marcado*
 - Además de texto incluyen conjuntos de caracteres especiales que indican al navegador como presentar dicho texto → HTML

Terminología básica

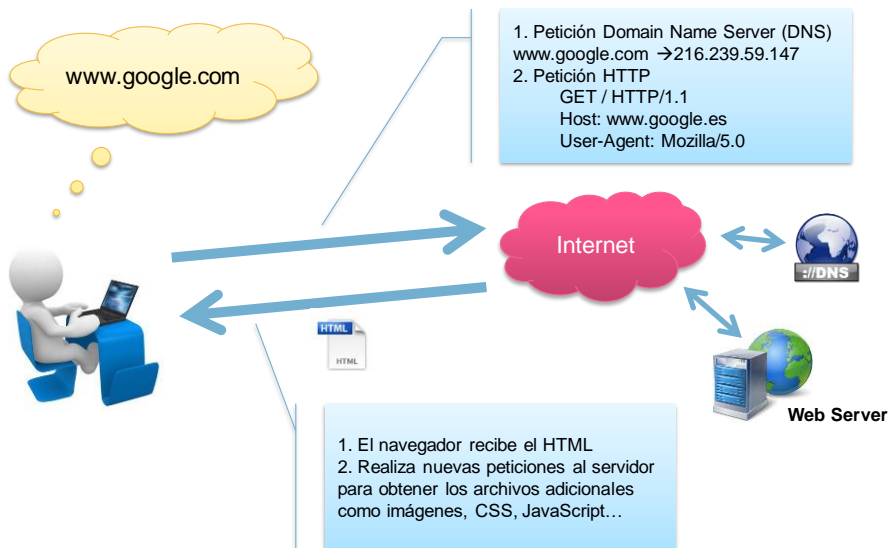
La WWW no es Internet, es un servicio que está montado sobre Internet

Internet (la red) está formado por el conjunto de ordenadores que están interconectados entre sí.



Arquitectura de Aplicaciones Web – 2ºC 2020

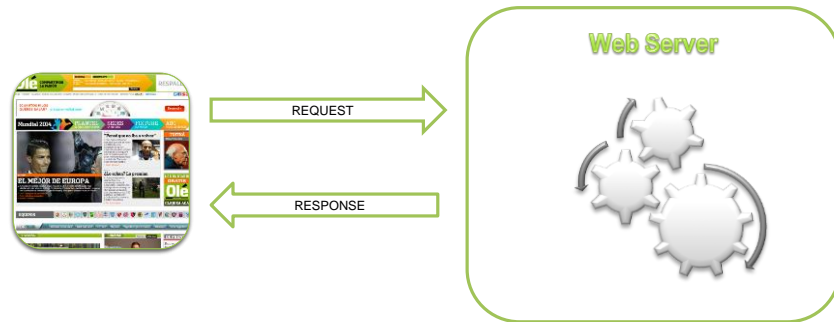
¿Cómo funciona la WEB?



Arquitectura de Aplicaciones Web – 2ºC 2020

Primera aproximación

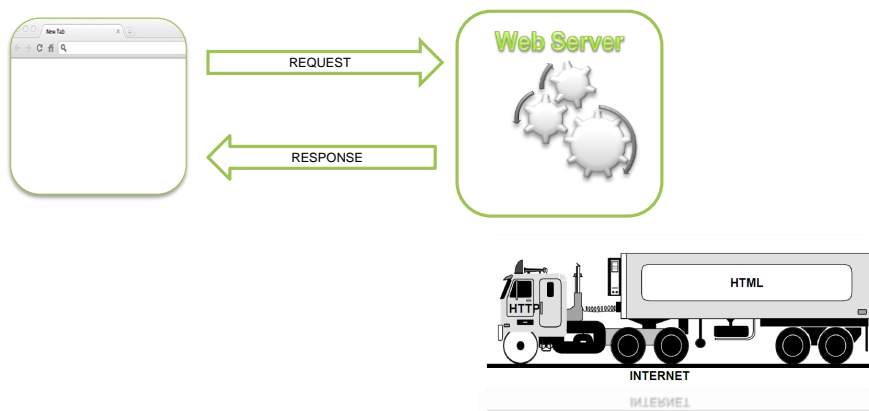
Modelo Cliente/Servidor



Arquitectura de Aplicaciones Web – 2°C 2020

Primera aproximación

- ❖ Protocolo de comunicación HTTP
- ❖ Respuesta contiene contenido HTML



Arquitectura de Aplicaciones Web – 2°C 2020

HTTP

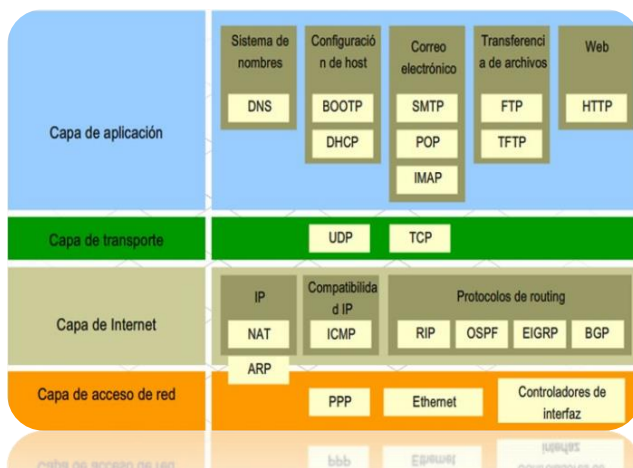
- ❑ Hypertext Transfer Protocol o HTTP (en español *protocolo de transferencia de hipertexto*) es el protocolo usado en cada transacción de la World Wide Web.
- ❑ HTTP define la *sintaxis* y la *semántica* que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.
- ❑ Es un *protocolo orientado a transacciones* y sigue el esquema petición-respuesta entre un cliente y un servidor.
 - Al cliente que efectúa la petición (un navegador web) se lo conoce como "user agent" (agente del usuario).
 - A la información transmitida se la llama recurso y se la identifica mediante un localizador uniforme de recursos (URL)



Aplicaciones Web – 2°C 2020

Modelo de Referencia TCP/IP

“Es un protocolo de transporte orientado a la conexión, lo que significa que antes de que un dato pueda ser transportado se debe tener una conexión fiable y segura que permita realizar el transporte.”



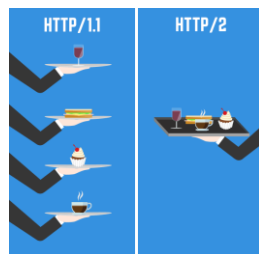
Arquitectura de Aplicaciones Web – 2°C 2020

HTTP

❖ TCP port 80 (443 secure)

❖ Estándares

- RFC 1945 (HTTP/1.0, 1996)
- RFC 2616 (HTTP/1.1, 1999)
- RFC 7540 (HTTP/2.0, 2015)



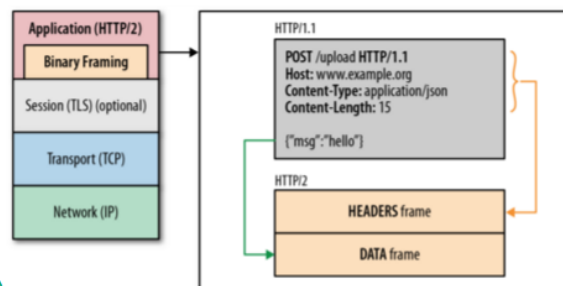
Arquitectura de Aplicaciones Web – 2°C 2020

HTTP/2



1. **One TCP connection**
2. **Request → Stream**
 - Streams are multiplexed
 - Streams are prioritized
3. **Binary framing layer**
 - Prioritization
 - Flow control
 - Server push
4. **Header compression (HPACK)**

Header compression (HPACK)



Arquitectura de Aplicaciones Web – 2°C 2020

URI = URL + URN

- Uniform Resource Identifier
 - Es una cadena de caracteres que identifica los recursos de una red de forma unívoca.
- Uniform Resource Name
 - Es una cadena de caracteres que identifican recursos en la web, pero no indican exactamente dónde se encuentra ese objeto.
- Uniform Resource Locator
 - Es una secuencia de caracteres que determina cómo encontrar el recurso.



Arquitectura de Aplicaciones Web – 2°C 2020

URIs

- URI
 - *Protocolo (http, ftp, news)*
 - *Host name (name.domain name)*
 - *Puerto (usualmente 80)*
 - *Ruta al recurso*
 - *Nombre del Recurso*
- `http://www.myplace.com/www/index.html`
- `http://www.myplace.com:80/cgi-bin/t.exe`

Arquitectura de Aplicaciones Web – 2°C 2020

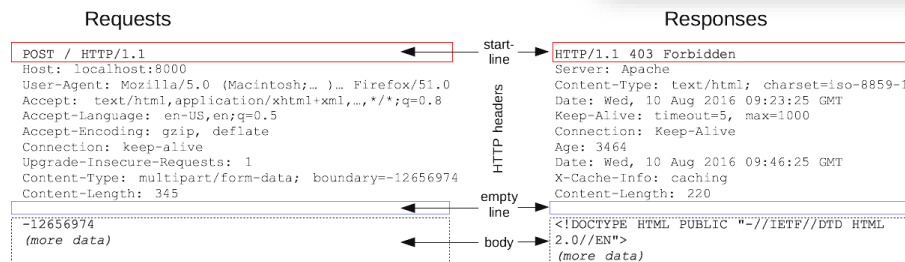
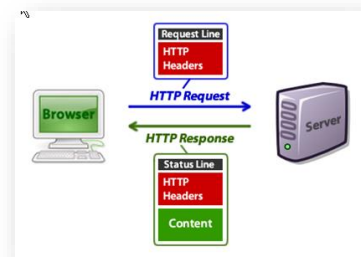
Mensajes HTTP

- Pueden ser
 - *Solicitudes (Request)*
 - *Respuestas (Response)*
- Tanto las solicitudes como las respuestas utilizan el formato genérico de e-mails (RFC-822)
- Ambos tipos de mensajes consisten de
 - *Una línea inicial*
 - *Cero o más encabezados (headers)*
 - *Una línea en blanco*
 - *Un cuerpo del mensaje (opcional, ej. archivo, datos de una consulta).*

Arquitectura de Aplicaciones Web – 2°C 2020

HTTP – Request & Response

- Poseen las siguientes partes:
 - *Línea Inicial:*
 - *Determina la naturaleza del mensaje.*
 - *Encabezados HTTP (Headers):*
 - Por ejemplo: *Accept-Language: es*
 - *Línea vacía*
 - *Cuerpo del mensaje (opcional)*



Arquitectura de Aplicaciones Web – 2°C 2020

HTTP – Request & Response

- Línea Inicial

- *Solicitudes*

Tiene tres partes separadas entre sí por un espacio.

- *El método (GET, PUT, POST, OPTIONS, TRACE, DELETE,...)*
- *El identificador del recurso (URI).*
- *La versión del protocolo HTTP en uso.*

Ejemplo: GET /images/logo.png HTTP/1.1

- *Respuestas*

- Tiene tres partes separadas entre sí por un espacio.

- *Versión de HTTP*
- *Código de estado*
- *Frase explicativa (legible por humanos)*

Ejemplo: HTTP/1.0 200 OK HTTP/1.0 404 Not Found

Arquitectura de Aplicaciones Web – 2°C 2020

HTTP – Métodos de petición

Método	Significado
GET	Devuelve el recurso identificado en la URL pedida.
HEAD	Funciona como el GET, pero sin que el servidor devuelva el cuerpo del mensaje. Es decir, sólo se devuelve la información de cabecera.
POST	Indica al servidor que se prepare para recibir información del cliente. Suele usarse para enviar información desde formularios.
PUT	Envía el recurso identificado en la URL desde el cliente hacia el servidor.
OPTIONS	Pide información sobre las características de comunicación proporcionadas por el servidor. Le permite al cliente negociar los parámetros de comunicación.
TRACE	Inicia un ciclo de mensajes de petición. Se usa para depuración y permite al cliente ver lo que el servidor recibe en el otro lado.
DELETE	Solicita al servidor que borre el recurso identificado con el URL.
CONNECT	Este método se reserva para uso con proxys. Permitirá que un proxy pueda dinámicamente convertirse en un túnel. Por ejemplo para comunicaciones con SSL.

Arquitectura de Aplicaciones Web – 2°C 2020

HTTP – Códigos de estados

- El código de estado es un entero de 3 dígitos

Rango	Descripción
1XX	Informativos
2XX	Éxito
3XX	Redirección
4XX	Error de cliente
5XX	Error de servidor

Arquitectura de Aplicaciones Web – 2°C 2020

HTTP – Códigos de estados

- Los más comunes

Código	Descripción
200	OK Solicitud exitosa. La respuesta se envía en el cuerpo
404	Not Found El recurso no existe.
303	See Other El recurso se ha movido a otra URL (Dada en el header Location)
500	Server Error Error no esperado en el servidor.

Arquitectura de Aplicaciones Web – 2°C 2020

HTTP – Códigos de estados

- 1xx: Mensaje informativo
- 2xx: Éxito
 - 200 OK
 - 201 Created
 - 202 Accepted
 - 204 No Content
- 3xx: Redirección
 - 300 Multiple Choice
 - 301 Moved Permanently
 - 302 Found
 - 304 Not Modified
- 4xx: Error del cliente
 - 400 Bad Request
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
- 5xx: Error del servidor
 - 500 Internal Server Error
 - 501 Not Implemented
 - 502 Bad Gateway
 - 503 Service Unavailable

Arquitectura de Aplicaciones Web – 2°C 2020

Http - Encabezados

- Formato de los encabezados
 - Nombre : Valor
- Clasificación
 - Genéricos: cliente y servidor
 - Exclusivos de la petición (información del cliente)
 - Exclusivos de la respuesta (información del Servidor)
 - Entidad del cuerpo del mensaje
- HTTP 1.0 define 16 headers (ninguno es obligatorio).
- HTTP 1.1 define 46 headers (solo Host: es obligatorio).

Arquitectura de Aplicaciones Web – 2°C 2020

Encabezados genéricos

Encabezado	Significado
Cache-Control	Permite especificar distintas directivas para controlar la caché, tanto el cliente como de servidores proxy.
Connection	Especifica opciones de la conexión de red.
Date	Envía una fecha en la representación estándar definida en el protocolo.
Pragma	Transporta información no HTTP a un receptor que sea capaz de entenderla.
Trailer	Indica que ciertas cabeceras HTTP pueden encontrarse en el final de un mensaje con múltiples partes (multipart).
Upgrade	Información sobre protocolos adicionales soportados por el cliente.
Via	Añadidos al mensaje de proxys o gateways para indicar que pasó por ellos.
Warning	Información adicional sobre un estado o transformación de un documento que podría no estar reflejado en el cuerpo del mismo. Por ejemplo, transformaciones que se hacen en servidores caché.

Manejan información que puede ser utilizada tanto por clientes como por servidores, ya que se aplican a una sesión completa de comunicación

Arquitectura de Aplicaciones Web – 2°C 2020

Encabezados de Solicitud

Encabezado	Significado
Accept	Listado de tipos MIME que el cliente soporta. Hay otros encabezados relacionados como Accept-Charset , Accept-Encoding , Accept-Language .
Authorization	Indica las credenciales de acceso a un recurso que presenta el usuario.
Expect	Indica que comportamiento del servidor necesita el cliente.
From	Dirección de correo que controla el cliente (navegador).
Host	Nombre o IP del host desde donde se conecta el cliente.
If-Match	Un cliente que tiene recursos en cache puede verificar si están actualizados incluyendo este encabezado. Hay otros encabezados que también tienen que ver con la caché. If-Modified-Since , If-None-Match , If-Range , If-Unmodified-Since .
Max-Forwards	Cuántas veces la petición del cliente puede ser reenviada por proxies.
Proxy-Authorization	Indica las credenciales de acceso a un proxy que presenta el usuario.
Range	Indica que porción de recurso (rango de bytes) recuperar.

Arquitectura de Aplicaciones Web – 2°C 2020

Encabezados de Solicitud

Encabezado	Significado
Referer	Es el URI del recurso desde donde la petición se ha realizado (generalmente por provenir de un enlace HTML).
TE	Que codificaciones de transferencia está dispuesto a recibir el cliente.
User-Agent	Información sobre el agente de usuario (generalmente navegador) que origina la petición.

Los utiliza el cliente para enviar (en sus peticiones de servicio) información adicional al servidor

Arquitectura de Aplicaciones Web – 2°C 2020

Encabezados de Respuesta

Encabezado	Significado
Accept-Ranges	Indica las unidades en las que el servidor acepta peticiones de rangos.
Age	Tiempo estimado por el servidor para cumplir la petición.
Etag	Valor actual de la etiqueta de entidad solicitada.
Location	Contiene un URI al que el cliente debe ser redireccionado.
Proxy-Authenticate	Indica el esquema de autenticación que acepta un servidor proxy.
Retry-After	Cuanto tiempo se espera que el servicio no esté disponible.
Server	Información del software servidor.
Vary	Indica que un recurso tiene múltiples fuentes que pueden variar de acuerdo a la lista de encabezados de petición.
WWW-Authenticate	Indica que el recurso solicitado necesita de credenciales de autorización.

Los utiliza el servidor para enviar información adicional al cliente.

Arquitectura de Aplicaciones Web – 2°C 2020

Encabezados de Entidad

Encabezado	Significado
Allow	Informa al cliente de los métodos válidos asociados con el recurso.
Content-Type	Indica el tipo MIME de los contenidos. Hay otros encabezados muy relacionados como Content-Language , Content-Length , Content-Location , Content-MD5 , Content-Range o Content-Encoding .
Expires	Indica la fecha y hora en la que el recurso se considerará obsoleto.
Last-Modified	Indica la fecha y hora en la que el recurso original fue modificado por última vez.

Contienen información relacionada directamente con el recurso que se le va a proporcionar al cliente.

Arquitectura de Aplicaciones Web – 2°C 2020

Custom Header

- ☐ Son headers que no pertenecen al estándar.
- ☐ Tradicionalmente, se los denominaba con el prefijo X-
- ☐ El uso del prefijo x- quedo *deprecated* en 2012.

Arquitectura de Aplicaciones Web – 2°C 2020



HTTP - Request

```
GET / HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0;
  Windows NT; DigExt)
Host: www.yahoo.com
Connection: Keep-Alive
Cookie: B=2td79o0sjlf5r&b=2
```

Arquitectura de Aplicaciones Web – 2°C 2020



HTTP - Cuerpo

- Luego de las líneas de encabezado un mensaje HTTP puede contener un cuerpo (body).
- En las respuestas el cuerpo es la sección en donde se envía el recurso solicitado.
- En las solicitudes el cuerpo se utiliza para subir datos que ingresó el usuario o para transferir archivos hacia el servidor.
- Las líneas de encabezado más comunes que definen el cuerpo son:
 - Content-Type: (Da el tipo MIME de los datos del cuerpo, ejemplo: text/html image/gif).
 - Content-Length: (Especifica el número de bytes en el cuerpo).

Arquitectura de Aplicaciones Web – 2°C 2020



HTML

- El *HiperText Markup Language* (HTML) es el lenguaje utilizado para diseñar páginas web.
- Es el language comprendido por los *browsers*.
- Una página HTML consiste en texto delimitado por *tags de marcado* que describen la apariencia, el formato y la ubicación del contenido.
- Puede ser editado utilizando cualquier procesador de texto.

Arquitectura de Aplicaciones Web – 2°C 2020



HTML

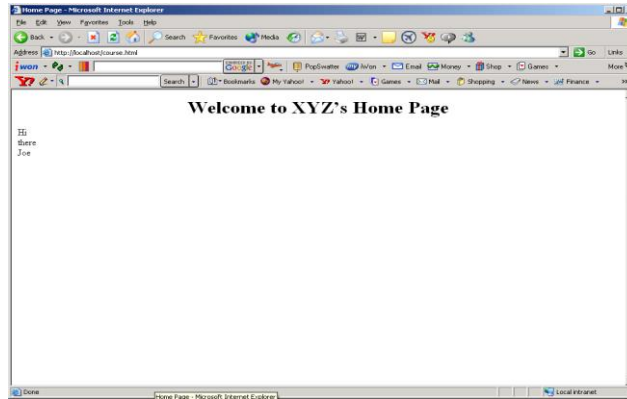
```

<HTML>
  <HEAD>
    <TITLE> Home Page</TITLE>
  </HEAD>
  <BODY>
    <H1 ALIGN=CENTER>
      Welcome to XYZ's Home Page
    </H1>
    Hi<BR>there<BR>Joe<BR>
  </BODY>
</HTML>

```

Arquitectura de Aplicaciones Web – 2°C 2020

HTML



Más adelante retomaremos este tema...

Arquitectura de Aplicaciones Web – 2°C 2020

Session & Cookies

- HTTP es un protocolo *sin estado*
- Existen diversos *mecanismos* útiles para recordar el estado:
 - HTTP Headers
 - FAT URLs
 - Session
 - Cookies
 - Entre otros...

Arquitectura de Aplicaciones Web – 2°C 2020



Session

- ❑ Una sesión (session) es un período de tiempo dentro del cual, todas las actividades realizadas por un cliente (user-agent) se consideran asociadas a él.
- ❑ Session tracking: es un mecanismo que permite almacenar en el servidor, durante el período de tiempo que dure la sesión, cierta información a través de los subsiguientes solicitudes realizadas por el usuario.

Arquitectura de Aplicaciones Web – 2°C 2020



Cookies

- Pequeños archivos de texto almacenados en los browsers.
- Se submitean en cada request.
- Permiten contar con trazabilidad intersesión.

Arquitectura de Aplicaciones Web – 2°C 2020

Session & Cookies

Session	Cookie
Server Side	Client Side
Al cerrar el browser se da por terminada la session	Soporta múltiples sesiones.
Permite almacenar objetos	Permite almacenar Strings

Arquitectura de Aplicaciones Web – 2°C 2020

¿Qué es Javascript?

- Es un lenguaje *Script*
- Extiende las capacidades de las páginas Web
- El código está integrado en el HTML
- Se interpreta en el ordenador que recibe el HTML, no se compila
- Ejecución dinámica
 - Los programas y funciones no se chequean hasta que se ejecutan
- Tiene programación orientada a objetos
- Trabaja con los elementos del HTML
- No se declaran los tipos de variables

Arquitectura de Aplicaciones Web – 2°C 2020



Generalidades de *JavaScript*

- Modelo orientado al WWW
- Elementos de una página HTML pueden causar un evento que ejecutará una acción
- Esa acción se ejecutará a través de una serie de sentencias *JavaScript*
- Comandos de *JavaScript*:
 - Variables
 - Expresiones
 - Estructuras de control
 - Funciones (bloques de sentencias)
 - Clases, objetos y arrays (agrupaciones de datos)

Arquitectura de Aplicaciones Web – 2°C 2020



¿Qué se puede hacer con *JavaScript*?

- Chequear Formularios
 - Comprobar que se han rellenado correctamente antes de enviarlos y que el servidor de error
- Realizar cálculos simples
- Hacer interactiva una página web
- Juegos...
- Muchas cosas!

Arquitectura de Aplicaciones Web – 2°C 2020

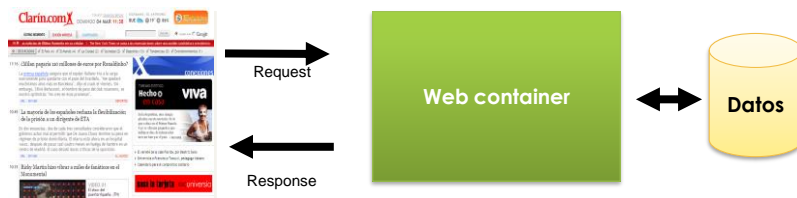
Ajax

- Asynchronous JavaScript and XML
- Inicialmente no fue un elemento *First Class*
- Surgió como esfuerzo de recortar el gap entre usabilidad e interactividad, entre las aplicaciones web y las aplicaciones desktop.
- No fue un lenguaje de programación
- No fue una nueva tecnología

Arquitectura de Aplicaciones Web – 2°C 2020

Ajax

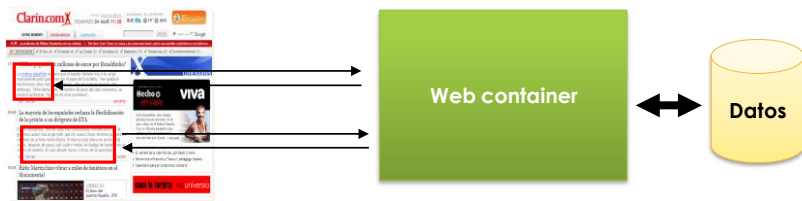
Otra manera de pensar la programación Web



Arquitectura de Aplicaciones Web – 2°C 2020

Ajax

Otra manera de pensar la programación Web



Arquitectura de Aplicaciones Web – 2°C 2020



Arquitectura de Aplicaciones Web – 2°C 2020