

# Lenguajes y estándares para la Web

# Tema 1: Computación en la Web

# Grado en Ingeniería Informática del Software



# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

- **Sitios Web estáticos y dinámicos**
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

# Lenguajes y estándares para la Web

## Sitios Web estáticos y dinámicos

Grado en  
Ingeniería  
Informática  
del Software

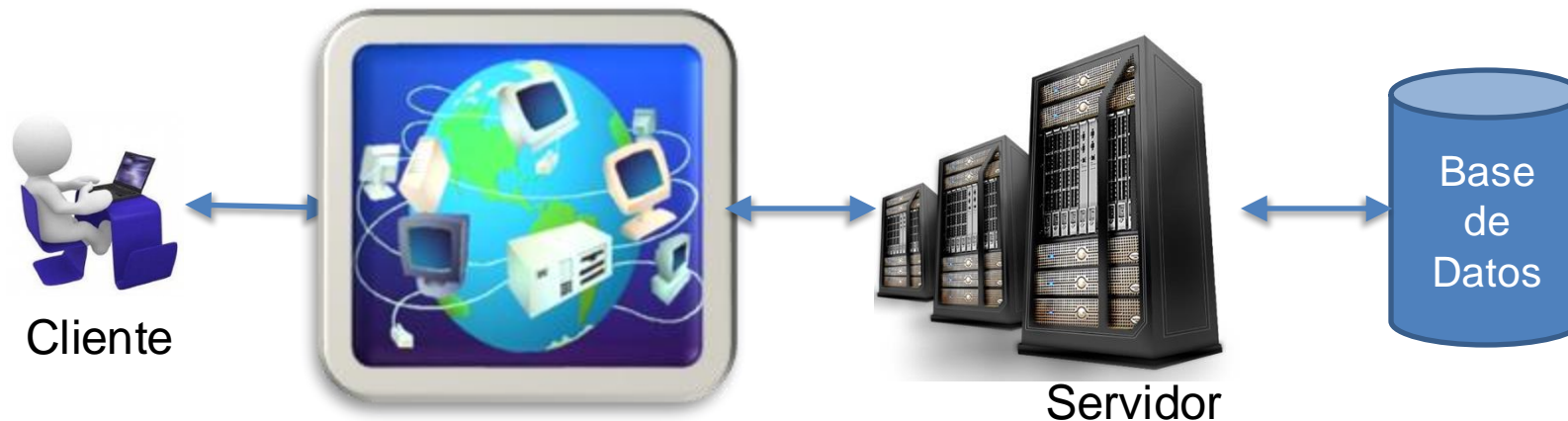
- En el inicio la Web (1989) era estática, solamente usaba HTML
- **Los sitios Web estáticos**
  - Su información permanece invariable desde el momento en que su autor lo crea.
  - Están contruidos principalmente con hipervínculos o enlaces (links) entre las páginas Web que conforman el sitio.
  - No tienen computación
  - No pueden soportar aplicaciones Web como gestores de bases de datos, foros, servicios Web, consultas “*on line*”, ...
  - Esta es una opción más que suficiente para aquellos sitios Web que simplemente ofrecen una descripción de su empresa u organización, quiénes somos, donde estamos, servicios, etc... ideal para entidades que no quieren muchas pretensiones con su sitio Web, simplemente informar a sus clientes de sus productos y su perfil de empresa u organización.
  - Son los sitios Web más baratos, seguros y estables; pero no soportan ningún tipo de información dinámica.

# Lenguajes y estándares para la Web

## Sitios Web estáticos y dinámicos

Grado en  
Ingeniería  
Informática  
del Software

- **Sitios Web dinámicos** o generación dinámica de información
  - La información se genera automáticamente en el momento que alguien solicita su visualización.
  - La información puede cambiar dependiendo de quién lo solicite o en que momento lo haga.
  - Por ejemplo, una sección de noticias o la información del tiempo
  - Es necesario que se realice algún tipo de computación.



# Lenguajes y estándares para la Web

## Sitios Web estáticos y dinámicos

Grado en  
Ingeniería  
Informática  
del Software

- **Ventajas** de los sitios Web dinámicos
  - **Flexibilidad**: La información se adapta a las características del cliente
  - **Eficiencia**: El usuario para interaccionar con las aplicaciones Web no necesita tener la información almacenada en modo local.
- **¿Cómo se pueden hacer sitios Web dinámicos?**
  - Utilizando **computación en el cliente Web**
    - Usando etiquetas **<object>** y **<embed>**
    - Usando etiquetas **<script>**
  - Utilizando **computación en el servidor Web**
    - Usando **CGI** (Common Gateway Interface).
      - CGI es una especificación que permite ejecutar programas en los servidores Web.
      - Los programas se pueden escribir en varios lenguajes: Perl, C, VisualBasic,...
    - Usando código incrustado o **lenguajes de script del lado del servidor** con tecnologías como PHP, ASP (ActiveServer Pages), Java Server Pages, JavaScript en el lado del servidor (Node.js), etc.
    - Generando **Servicios Web** u otros protocolos como **REST**

# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- **Computación el cliente y en el servidor**
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

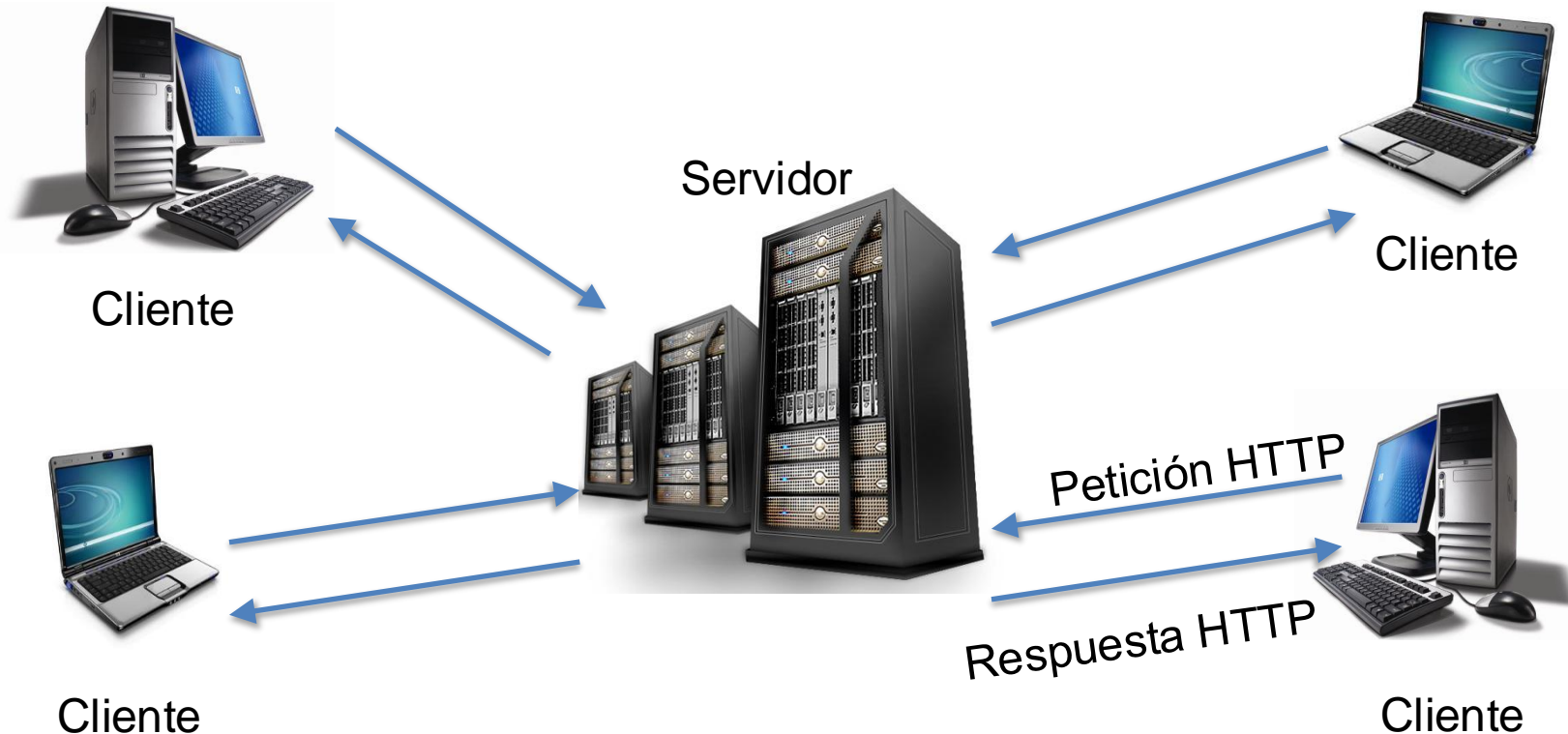


# Lenguajes y estándares para la Web

## Computación en el cliente (**Front-end**)

Grado en  
Ingeniería  
Informática  
del Software

- El ordenador cliente realiza una petición al servidor Web (*http Request*)
- El ordenador servidor Web envía una respuesta en HTML (*HTML Response*).
- La **computación en el cliente** se realiza en la máquina cliente a partir de la respuesta recibida en HTML del ordenador servidor Web.



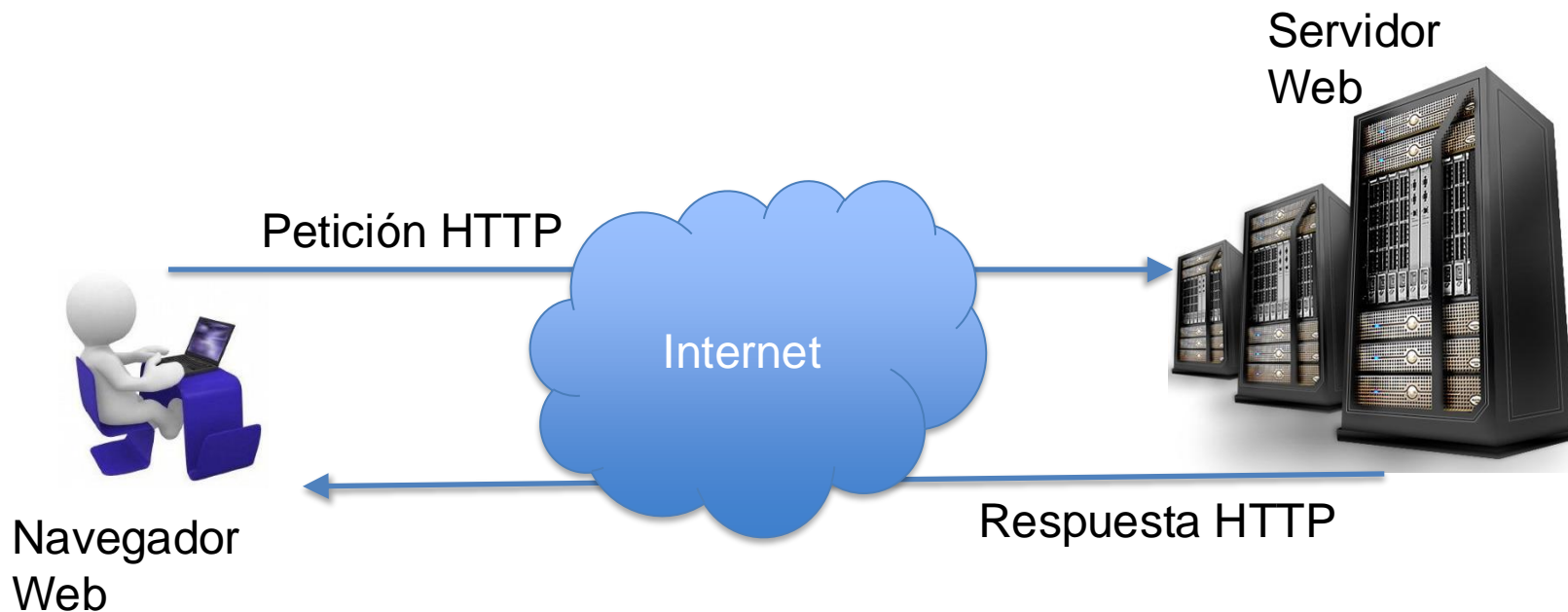


# Lenguajes y estándares para la Web

## ¿Qué es un servidor Web? (I)

Grado en  
Ingeniería  
Informática  
del Software

- Es un programa que procesa **peticiones HTTP** y **envía respuestas HTTP**
- HTTP (**H**ypertext **T**ransfer **P**rotocol) es el protocolo más usado para distribuir información en la Web (World Wide Web)



# Lenguajes y estándares para la Web

## ¿Qué es un servidor Web? (II)

Grado en  
Ingeniería  
Informática  
del Software

1. El navegador pide una página Web

2. El Servidor envía la página y la cookie

Cookie

¡Hola a todos!

3, El navegador solicita otra página  
del mismo servidor

Cookie



Navegador  
Web o  
agente de  
usuario



Servidor  
Web

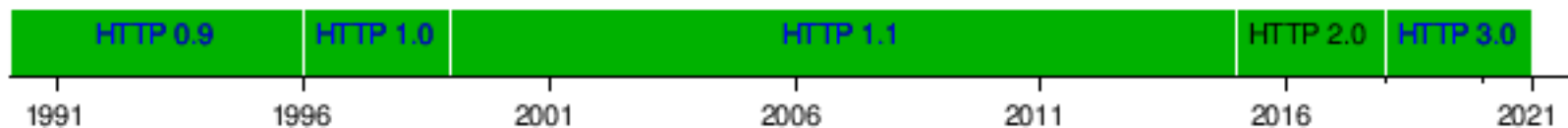
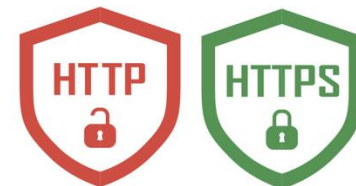
Las **cookies** son archivos que contienen pequeños fragmentos de datos que se intercambian entre un equipo de usuario y un servidor web, su propósito inicial era identificar usuarios específicos y mejorar su experiencia de navegación.

# Lenguajes y estándares para la Web

## Protocolo HTTP (**H**ypertext **T**ransfer **P**rotocol)

Grado en  
Ingeniería  
Informática  
del Software

- HTTP define la **sintaxis** y la **semántica** que utilizan los elementos de software de la **arquitectura web** (clientes, servidores, proxies) **para comunicarse**.
- HTTP es un **protocolo sin estado**, es decir, no guarda ninguna información sobre conexiones anteriores
- **Estándar** HTTP/1.1 del W3C
  - <https://www.w3.org/Protocols/>
- **HTTP/2**
  - <https://es.wikipedia.org/wiki/HTTP/2>
  - <https://datatracker.ietf.org/doc/html/rfc7540>
- **HTTP/3**
  - <https://es.wikipedia.org/wiki/HTTP/3>
- **HTTPS** es una extensión de HTTP para realizar comunicaciones seguras

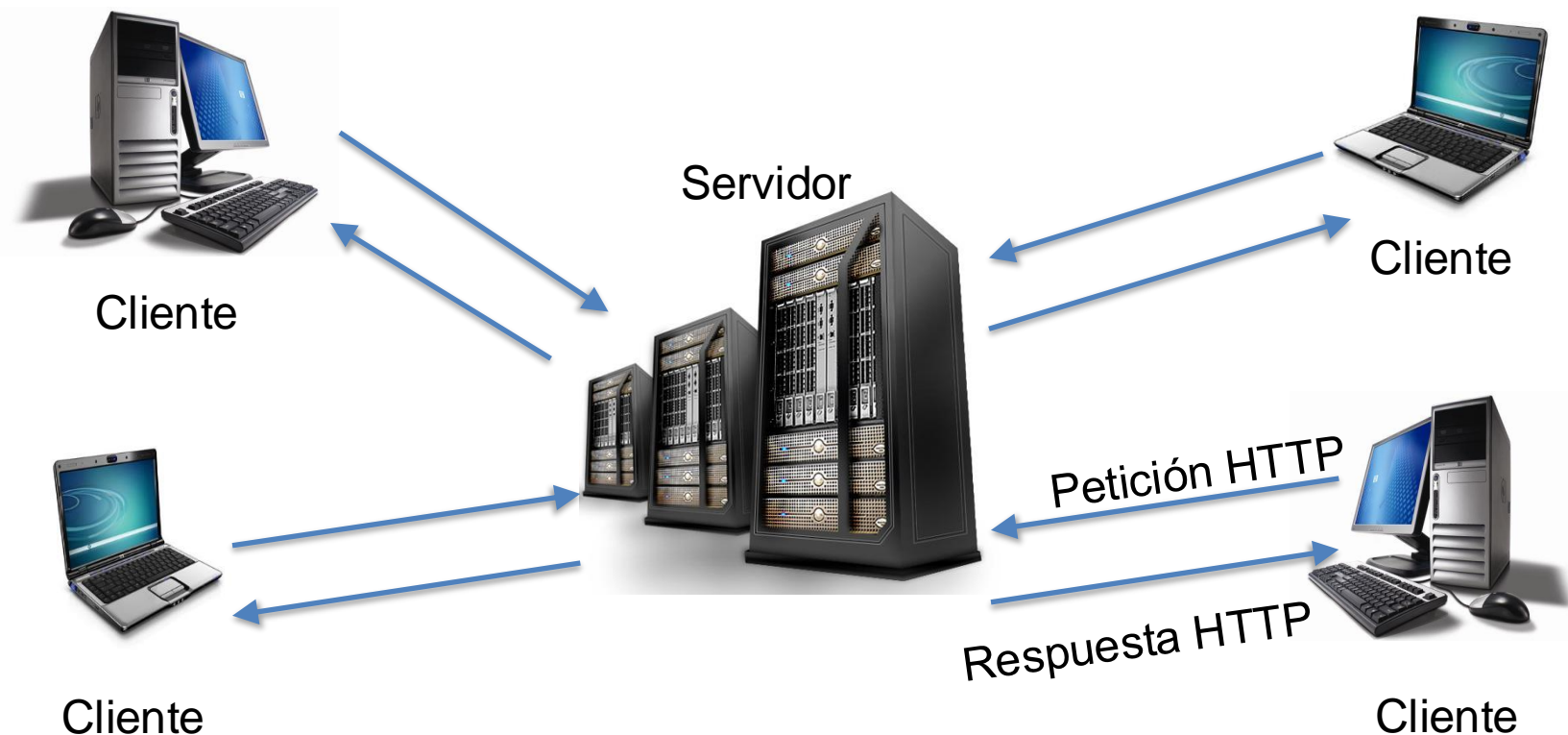


# Lenguajes y estándares para la Web

## Computación en el servidor

Grado en  
Ingeniería  
Informática  
del Software

- El ordenador cliente realiza una petición al servidor Web (*http Request*)
- A partir de esa petición en el servidor Web se realiza la computación
- Una vez finalizada la **computación en el servidor Web** devuelve el resultado al cliente Web en HTML (*HTML Response*).



# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- **Protocolos TCP/IP**
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

# Lenguajes y estándares para la Web

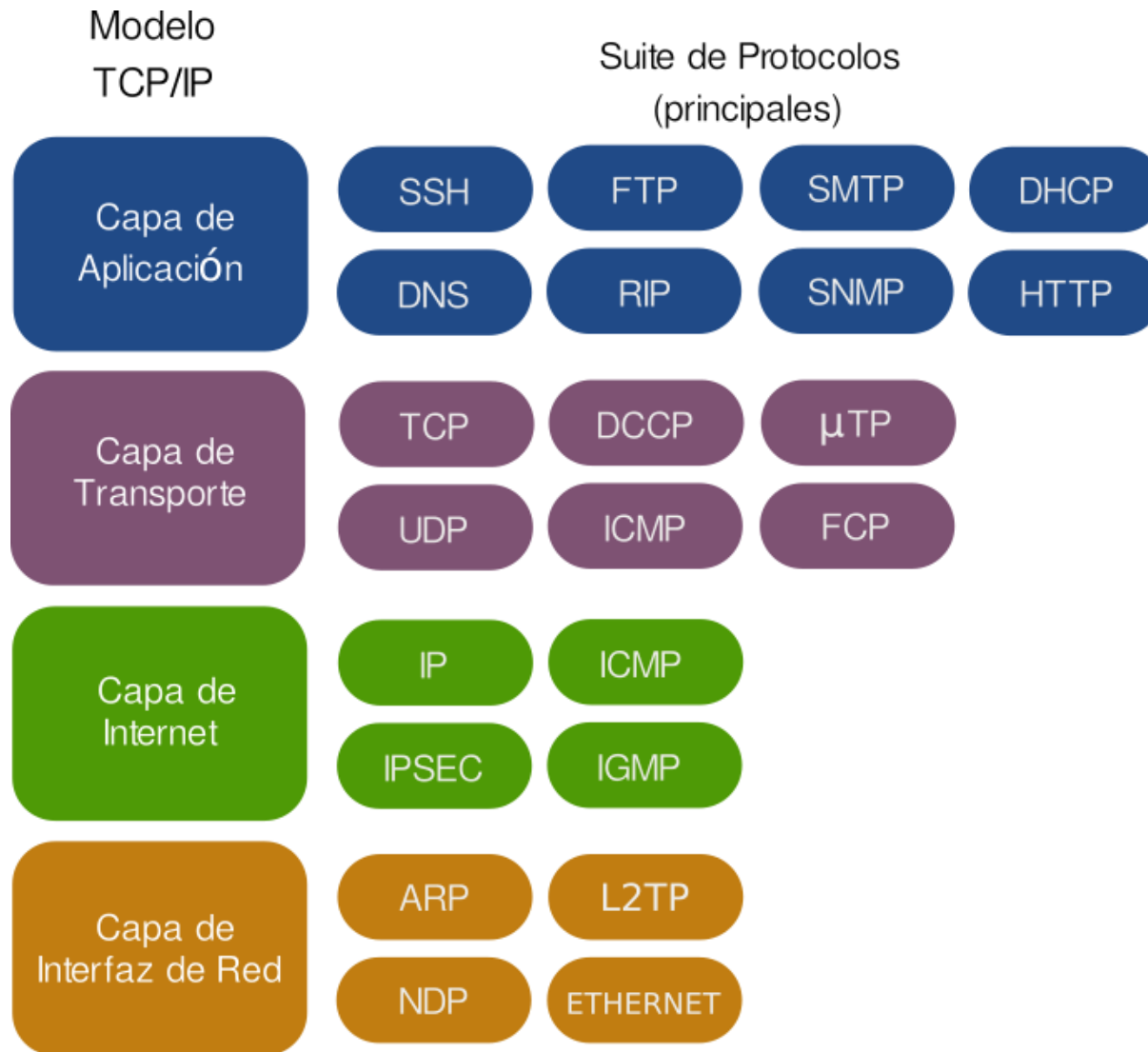
## Protocolos TCP/IP

Grado en  
Ingeniería  
Informática  
del Software

- TCP/IP son las siglas de *Transmission Control Protocol/Internet Protocol*
  - Protocolo de control de transmisión/Protocolo de Internet
  - TCP/IP es un conjunto de reglas estandarizadas que permiten a los equipos comunicarse en una red como Internet
  - TCP e IP son dos protocolos distintos
- **IP** es el protocolo que obtiene la dirección a la que se envían los datos
- **TCP** se encarga de la entrega de los datos una vez hallada dicha dirección IP.

# Lenguajes y estándares para la Web

## Modelo TCP/IP



Grado en  
Ingeniería  
Informática  
del Software



# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- **Protocolo HTTP**
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

# Lenguajes y estándares para la Web

## Protocolo HTTP

Grado en  
Ingeniería  
Informática  
del Software

- HTTP = *HyperText Transfer Protocol*
- <https://www.w3.org/Protocols/>
- Protocolo de nivel de aplicación que articula los intercambios de información entre los clientes web y los servidores HTTP.
  - Petición/Respuesta
  - Soportado sobre los servicios de conexión TCP/IP.
  - En el proceso el servidor escucha por un puerto (por defecto el 80) y espera peticiones de los clientes (en forma de método).

# Lenguajes y estándares para la Web

## Protocolo HTTP: Sintaxis

Grado en  
Ingeniería  
Informática  
del Software

- Sintaxis de la petición

**http://<dirección IP>:<puerto>/<ruta>**

- **<dirección IP>** nombre de dominio de Internet
- **<puerto>** por el que escucha el proceso servidor. Si no se indica, se entiende que es el puerto por defecto (80).
- **<ruta>** Si no se indica ruta, entonces se supone que es / (documento inicial o por defecto).

# Lenguajes y estándares para la Web

## Protocolo HTTP: **Ejemplo**

Grado en  
Ingeniería  
Informática  
del Software

<http://miServidor/index.html>

GET index.html HTTP/1.0

...

**Solicitud**



**Respuesta**

...

```
<html>
<head> ... </head>
<body>
<h1>Mi sitio Web</h1>
```

...

```
</body>
</html>
```



Cliente



Servidor

# Lenguajes y estándares para la Web

## Protocolo HTTP: Recursos

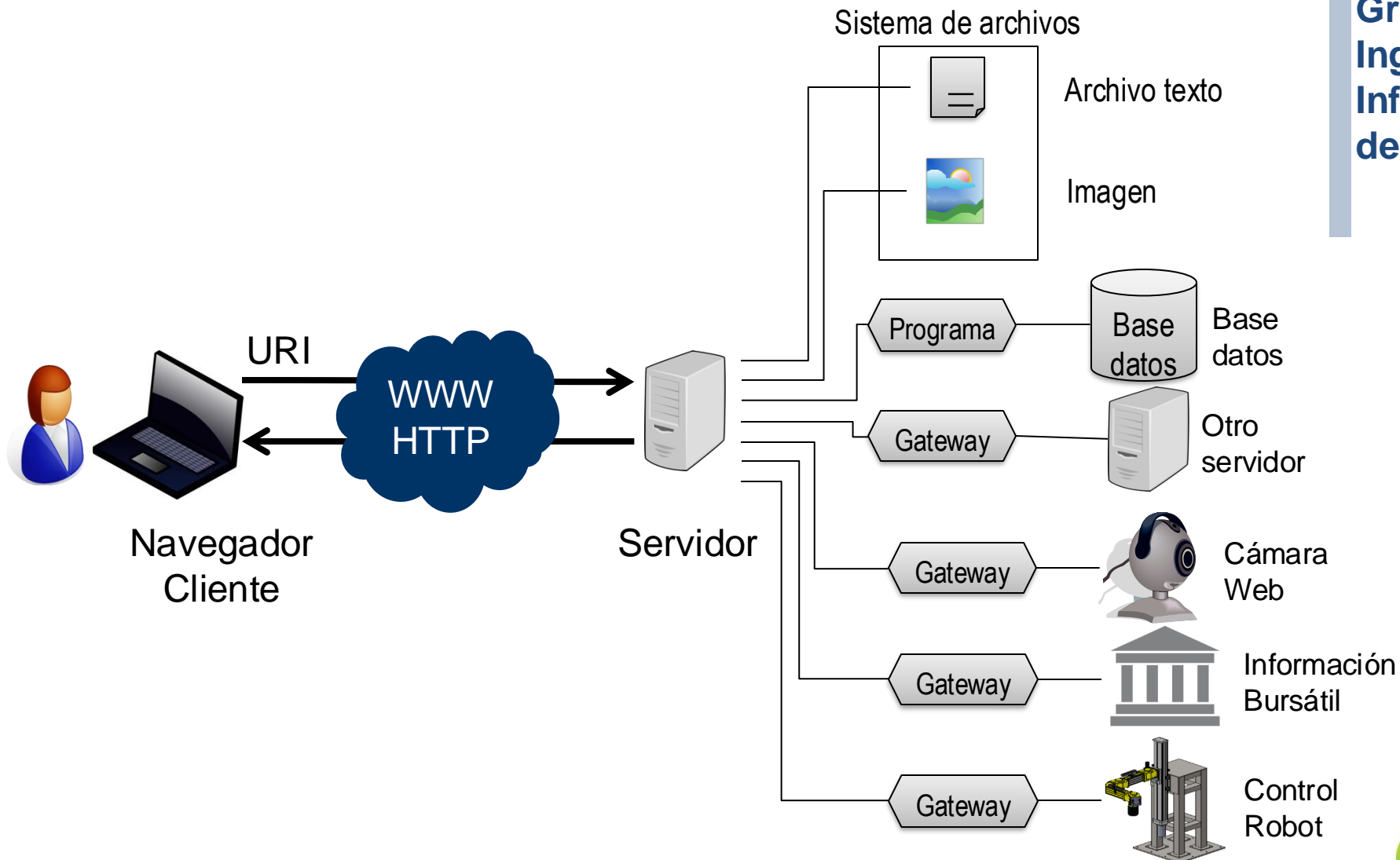
Grado en  
Ingeniería  
Informática  
del Software

- La Web está formada por recursos
  - **Recurso = cualquier fuente de contenido Web**
    - Se identifican mediante URIs  
[https://es.wikipedia.org/wiki/Identificador\\_de\\_recursos\\_uniforme](https://es.wikipedia.org/wiki/Identificador_de_recursos_uniforme)
- Diversas Posibilidades:
  - **Estáticos**: almacenados en Sistema de Ficheros
    - Páginas HTML
    - Otros formatos multimedia: Imágenes, vídeos, sonidos, ...
  - **Dinámicos**: bajo demanda
    - Generado a partir de bases de datos
    - Integrando información de otros servicios Web
    - Información online
- La Web permite transferir **distintos tipos** de recursos
  - El contenido de cada recurso intercambiado está identificado por su clasificación **MIME** (***M**ultipurpose **I**nternet **M**ail **E**xtensions*)

# Lenguajes y estándares para la Web

## Protocolo HTTP: Tipos de Recursos

Grado en  
Ingeniería  
Informática  
del Software

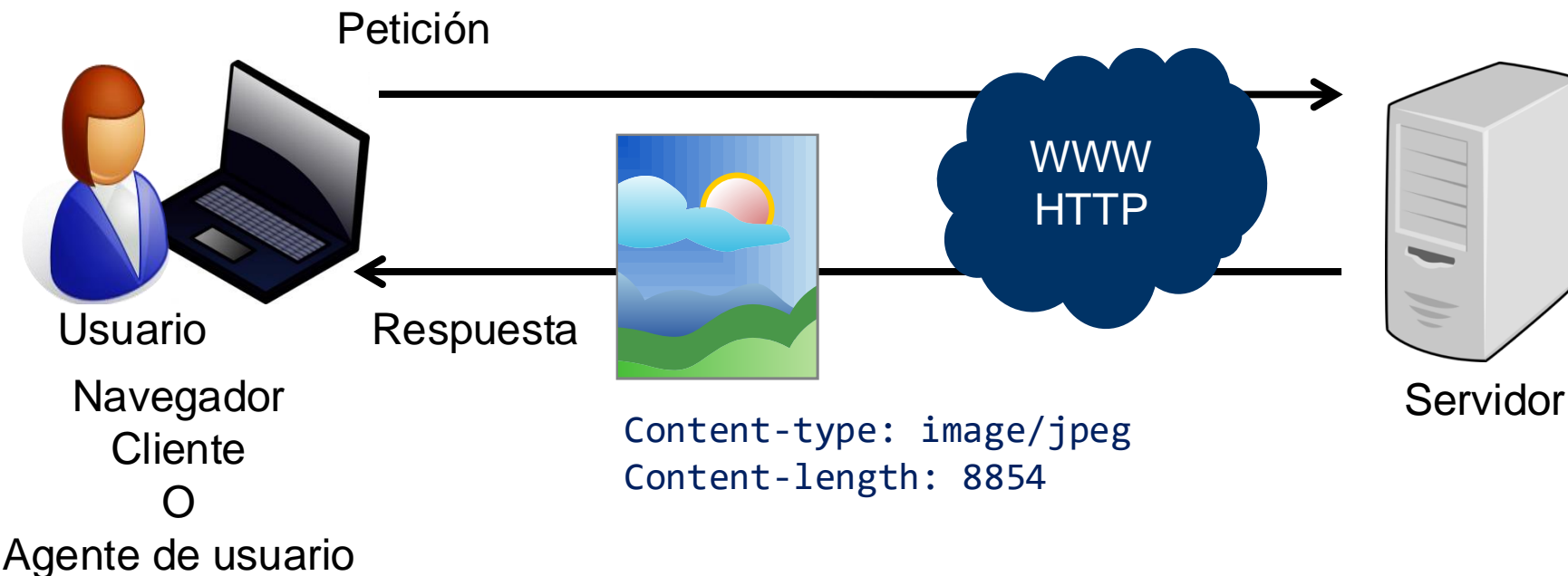


# Lenguajes y estándares para la Web

## Protocolo HTTP: Asociación de tipos de recursos

Grado en  
Ingeniería  
Informática  
del Software

- Los servidores asocian un tipo a cada recurso
  - Tipos **MIME** (**M**ultipurpose **I**nternet **M**ail **E**xtensions)
  - Ejemplos:
    - `text/plain`, `text/html`, `application/xml`, `image/jpg`,...
  - El cliente decide qué hacer con dichos tipos





# Lenguajes y estándares para la Web

## Protocolo HTTP: **Formato de mensajes o peticiones**

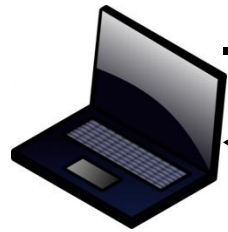
Grado en  
Ingeniería  
Informática  
del Software

Modo texto:

```
línea inicial
cabecera del mensaje*
cuerpo del mensaje ?
```

**Petición**

```
GET /index.html HTTP/1.1
Host: www.uniovi.es
User-Agent: Mozilla/5.0 ...
Accept: text/html
Accept-language: es, en
```



Navegador  
Cliente



Servidor

**Respuesta**

```
HTTP/1.1 200 OK
Date: Thu, 12 Oct 2013 09:36:05 GMT
Server: Apache
Content-length: 80554
Content-type: text/html; charset=utf-8
<html>
  <head>
    <title>Curso XML</title>
  <head>
    ...
</html>
```

# Lenguajes y estándares para la Web

## Protocolo HTTP: **Métodos o verbos**

Grado en  
Ingeniería  
Informática  
del Software

- Normalmente la comunicación la inicia un cliente realizando una **petición**.
- Toda petición incluye un **método o verbo** que se aplica al recurso:
  - **GET** para solicitar un recurso por su URI (también se puede utilizar para enviar datos al servidor).
  - **POST** permite enviar información al servidor para que sea procesada para el recurso identificado.
  - **HEAD** similar a GET pero obtiene únicamente la cabecera
  - **PUT** crea un recurso
  - **DELETE** elimina un recurso
  - **TRACE** pide la solicitud que se envió al servidor
  - **CONNECT** convierte la petición en un túnel TCP/IP
    - Facilita la comunicación a través de SSL
  - **OPTIONS** solicita los métodos que soporta el servidor

# Lenguajes y estándares para la Web

## Protocolo HTTP: **No mantiene estado**

Grado en  
Ingeniería  
Informática  
del Software

- Cada operación HTTP implica una conexión con el servidor que es liberada al final de la misma (ciclo petición/respuesta).
  - En una operación solo se puede recoger un único recurso
- **No mantiene estado**
  - Cada petición de un cliente a un servidor no está influenciada por las transacciones previas
  - El servidor trata cada operación de forma totalmente independiente del resto
  - Por eso se usan **cookies**, para almacenar información de las visitas

# Lenguajes y estándares para la Web

## Protocolo HTTP: Transacciones o mensajes

Grado en  
Ingeniería  
Informática  
del Software

- **Transacciones o mensajes HTTP**

- Consiste en una petición de un cliente al servidor y en la respuesta de éste al cliente.
- Las peticiones y respuestas pueden ser simples o completas.
  - Las primeras se mantienen por compatibilidad con la versión inicial del protocolo (HTTP/0.9).
- Una petición de un cliente siempre comienza por el método a aplicar a un recurso, seguido de la URI de éste.

# Lenguajes y estándares para la Web

## Protocolo HTTP: Mensajes de petición

Grado en  
Ingeniería  
Informática  
del Software

### Formato general:

```
<method> <url> <version>  
<headers> *  
<entity-body>
```

<headers> = parejas de la forma:  
nombre1: valor1  
nombre2: valor2

### Ejemplos:

```
GET /index.html HTTP/1.1  
Host: www.uniovi.es  
User-Agent: Mozilla/5.0 ...  
Accept: text/html  
Accept-language: es, en
```

```
POST /admin HTTP/1.1  
Host: www.uniovi.es  
User-Agent: Mozilla/5.0 ...  
Accept: text/html  
Accept-language: es, en  
...datos POST...
```

# Lenguajes y estándares para la Web

## Protocolo HTTP: Cabeceras en la petición (**headers**)

Grado en  
Ingeniería  
Informática  
del Software

**Accept:** Tipos de representaciones aceptables

**Accept-charset:** Conjunto de caracteres aceptable

**Accept-encoding:** Codificación de caracteres aceptable

**Accept-language:** Idiomas aceptables

**Authorization:** Indicar credenciales de autorización

**Cache-control:** Especificar directivas para controlar la cache

**Connection:** Tipo de conexión preferida

**Cookie:** Cookie enviada previamente por el servidor

**Content-length:** Longitud de la petición

**Content-type:** Tipo MIME del cuerpo de la petición

**Date:** Fecha/hora de la solicitud

**If-Modified-Since:** Permite enviar código 304 No modificado si no se ha modificado el contenido desde una fecha

**If-None-Match:** Permite enviar código 304 No modificado (ETag)

**User-Agent:** Identifica el tipo agente de usuario utilizado

# Lenguajes y estándares para la Web

## Protocolo HTTP: Mensajes de respuesta

Grado en  
Ingeniería  
Informática  
del Software

### Formato general:

```
<version> <status> <reason-phrase>  
<headers> *  
<entity-body>
```

<Status>

Códigos estándar:

2\*\*: Variaciones de OK

3\*\*: redirecciones

4\*\*: Problemas del cliente

5\*\*: problemas del servidor

### Ejemplos:

```
HTTP/1.1 200 OK  
Date: Thu, 12 Oct 2013 09:36:05 GMT  
Server: Apache  
Content-length: 80554  
Content-type: text/html; charset=utf-8  
<html>  
  <head>  
    <title>Curso XML</title>  
  </head>  
  ...  
</html>
```

```
HTTP/1.1 404 Not found  
Content-length: 0
```



# Lenguajes y estándares para la Web

## Protocolo HTTP: Cabeceras en la respuesta (**headers**)

Grado en  
Ingeniería  
Informática  
del Software

**Content-type:** Tipo MIME de la respuesta

**Cache-control:** Especificar directivas para controlar la cache

**Content-encoding:** Tipo de codificación utilizado en el mensaje

**Content-language:** Idioma utilizado en el mensaje

**Content-length:** Tamaño del mensaje

**Content-location:** Localización alternativa de los datos devueltos

**Date:** Fecha/hora de la respuesta

**ETag:** Identificador de la versión de un recurso

**Expires:** Fecha a partir de la cual el contenido puede eliminarse de la caché

**Server:** Identifica el tipo de servidor

**Set-cookie:** Activa una cookie en el cliente

**WWW-Authenticate:** Indica el esquema de autenticación a utilizar

# Lenguajes y estándares para la Web

## Protocolo HTTP: Peticiones y respuestas (simples y completas)

Grado en  
Ingeniería  
Informática  
del Software

- Peticiones y respuestas completas
  - Se envían cabeceras (*metadatos*) y un contenido, separando ambos por una línea en blanco.
- Peticiones simples
  - Sólo se puede utilizar el método GET y no hay contenido.
- Respuestas simples
  - Sólo constan de contenido.

# Lenguajes y estándares para la Web

## Protocolo HTTP: **Herramientas**

Grado en  
Ingeniería  
Informática  
del Software

curl <http://curl.haxx.se/>

Redbot <http://redbot.org>

Web-sniffer <http://websniffer.cc/>

RestClient <https://github.com/wiztools/rest-client>

Agentes de usuario

(navegadores) Pueden visualizar el protocolo HTTP

Desarrollo propio Por ejemplo en Python

# Lenguajes y estándares para la Web

Protocolo HTTP: Herramienta <http://redbot.org/>

Grado en  
Ingeniería  
Informática  
del Software

**REDbot**

<https://www.uniovi.es>

add a request header

```
HTTP/1.1 200
Cache-Control: private, no-cache, no-store, must-revalidate
Expires: Thu, 01 Jan 1970 00:00:00 GMT
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1
Set-Cookie: JSESSIONID=F657123A763C2AC947E2B3025E2E8E02; Path=/; Secure;
HttpOnly
Pragma: no-cache
Set-Cookie: COOKIE_SUPPORT=true; Max-Age=31536000; Expires=Sun, 03-Nov-2024
23:37:29 GMT; Path=/; Secure; HttpOnly
Set-Cookie: GUEST_LANGUAGE_ID=es_ES; Max-Age=31536000; Expires=Sun,
03-Nov-2024 23:37:29 GMT; Path=/; Secure; HttpOnly
Liferay-Portal: Liferay Community Edition Portal 7.3.7 CE GA8 (Athanasius /
Build 7307 / April 13, 2021)
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Date: Sat, 04 Nov 2023 23:37:29 GMT
Set-Cookie: NSC_XfcVojpwj_TTM_443=fffffffff9f2bf76d45525d5f4f58455e445a4a4237
8b; path=/; secure; httponly
```

response headers: 847 bytes

content: 171,374 bytes

transfer overhead: 171 bytes

# Lenguajes y estándares para la Web

Protocolo HTTP: Herramienta <http://websniffer.cc/> (a)

Grado en  
Ingeniería  
Informática  
del Software

## WebSniffer

### View HTTP Request and Response Headers

HTTP(S)-URL:    
Request type:  HTTP version:  User agent:

### Share This Query

Share this query on a website or a forum by copying the following link:

**<https://websniffer.com/?url=http://www.uniovi.es/>**

Do you have a **feature request** or **found a bug**? Please, **let us know** and we'll try our best to fix or implement it just within **1 week!**

We have some interesting information about the domain **uniovi.es** and the IP address **156.35.233.101**. Make sure to check it out:

- **[uniovi.es](http://uniovi.es)**
- **156.35.233.101**

# Lenguajes y estándares para la Web

Protocolo HTTP: Herramienta <http://websniffer.cc/> (b)

Grado en  
Ingeniería  
Informática  
del Software

## HTTP Request Header

Connect to **156.35.233.101** on port **80** ... ok

```
GET / HTTP/1.1
User-Agent: WebSniffer/1.1 (+http://websniffer.com/)
Host: www.uniovi.es
Accept: */*
Referer: https://websniffer.com/
Connection: Close
```

## HTTP Response Header

Name	Value
<b>HTTP/1.1 301 Moved permanently</b>	
<b>Location:</b>	<a href="https://www.uniovi.es/">https://www.uniovi.es/</a>
<b>Connection:</b>	close
<b>Cache-Control:</b>	no-cache
<b>Pragma:</b>	no-cache

# Lenguajes y estándares para la Web

Protocolo HTTP: **Edge – ...>Más herramientas>Herramientas de desarrollo > Red**

The screenshot shows a web browser displaying the University of Oviedo website (https://www.uniovi.es). The page features a large image of two people in protective suits in a laboratory setting. Below the image, the text reads: "Motor de investigación de Asturias" and "La Universidad de Oviedo realiza el 80% de la investigación que se realiza en el Principado". The Chrome DevTools Network tab is open on the right, showing a list of network requests. The table below represents the data visible in the Network tab:

Nombre	Es...	Tipo	Iniciador	Ta...	Hora	C...	Cascada
6f671aeb-d4a...	200	jpeg	Otros	1...	29...		
c40174f2-c52...	200	jpeg	Otros	90...	17...		
c901732e-b41...	200	jpeg	Otros	15...	55...		

At the bottom of the Network tab, it shows: "3 solicitudes 2.8 MB transferidos 2.8 MB recursos". The bottom of the browser window shows the word "Estudia".



# Lenguajes y estándares para la Web

Protocolo HTTP: **Firefox – ...>Desarrollador Web > Red**

Univer: X

+

←

→

↻


https://www.uniovi.es

☆

🔒

📄

☰



Universidad de Oviedo

Intranet

Búsqueda 🔍

ES ▼

☰

Elige Universidad de Oviedo

*Pública y de calidad, UniOvi te ofrece más de 120 títulos en una completa oferta académica.*

✓

Inspector

Consola

Depurador

↕ Red

🛠 Editor de estilos

📈 Rendimiento

🧠 Memoria

💾 Almacenamiento

♿ Accesibilidad

🧩 Aplicación

🗑 Filtrar las URL

|| + 🔍 ⌛

Todos HTML CSS JS XHR Tipografía Imágenes Medios WS Otros

☐ Desactivar caché

Sin limitación ⌵ ⚙

Estado	Método	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño	0 ms		40,96 s
200	GET	www.uniovi.es	/combo/?browserId=firefox&minifierType=&languageId=es_ES&b=	combo:495 (script)	js	cacheado	395 B	0 ms		
200	GET	www.uniovi.es	/combo/?browserId=firefox&minifierType=&languageId=es_ES&b=	combo:495 (script)	js	cacheado	5,03 KB	0 ms		
200	GET	www.uniovi.es	/combo/?browserId=firefox&minifierType=&languageId=es_ES&b=	combo:77 (script)	js	cacheado	14,79 KB	0 ms		
200	GET	www.uniovi.es	648db59f-58a0-3152-6556-9a1329312d10	FaviconLoader.sys.mjs:...	vnd.micr...	cacheado	1,15 KB	0 ms		
304	GET	www.uniovi.es	194fc4e6-a7fb-61b5-7279-5c5138cdec49?t=1698133398926	combo:11 (img)	jpeg	cacheado	143,03 KB			122 ms
304	GET	www.uniovi.es	de204f82-6451-e7be-e71a-90f50fa15af9?t=1676979028327	combo:11 (img)	jpeg	cacheado	1,01 MB			129 ms

🕒 68 solicitudes

6,65 MB / 1,02 MB transferido

Finalizado: 50,71 s

DOMContentLoaded: 362 ms

load: 561 ms

# Lenguajes y estándares para la Web

Protocolo HTTP: Chrome – ...>Más herramientas>Herramientas de desarrollo> Red

Recibidos (358) - cuevalovel

Google Calendar - Semana d

Universidad de Oviedo - Univ

uniovi.es

Bookmarks

EVENTOS GUÍA |...

CANALES TV onli...

VUELO

CANALES TV onli...

Library Genesis

TRABAJO SEMAN...

Gmail

YouTube

Maps

Universidad de Oviedo

Motor de investigación de Asturias

La Universidad de Oviedo realiza el 80% de la investigación que se realiza en el Principado

Estudia

Elementos

Consola

Fuentes

Red

Rendimiento

>>

Mantener registro

Inhabilitar caché

Sin limitaciones

Filtrar

Invertir

Ocultar URLs de datos

Ocultar URLs de extensiones

Todo

Documento

JS

Fetch/XHR

CSS

Fuente

Img

Multimedia

Archivo de manifiesto

Cookies de respuesta bloqueadas

Solicitudes bloqueadas

Solicitudes externas

20 ms

40 ms

60 ms

80 ms

100 ms

Nombre	Est...	Tipo	Iniciador	Ta...	Hora	Cascada
194fc4e6-a7fb-6...	304	jpeg	Otros	24...	31 ...	
de204f82-6451-...	304	jpeg	Otros	24...	30 ...	

Solicitudes: 2

Se ha transferido 498 B

Recursos: 1.2 MB

Consola

Problemas

Novedades

Highlights from the Chrome 119 update

Improved @property section in Elements > Styles

You can now edit the @propery rule in Elements > Styles.

# Lenguajes y estándares para la Web

## Protocolo HTTP con Python: 02200-HTTP.py (I)

Grado en  
Ingeniería  
Informática  
del Software

```
# 02200-HTTP.py
# # -*- coding: utf-8 -*-
"""
Procesamiento del protocolo HTTP

@version 1.1 16/Mayo/2020
@author: Juan Manuel Cueva Lovelle. Universidad de Oviedo
"""

import http.client
import pprint

def verHTTP(sitioWeb):
    """Función verHTTP(sitioWeb)
    Visualiza por pantalla el protocolo HTTP

    Version: 1.1 16/Mayo/2020
    Author: Juan Manuel Cueva Lovelle. Universidad de Oviedo
    """
    try:

        conexion = http.client.HTTPConnection(sitioWeb)
        conexion.request("GET", "/")
        respuesta = conexion.getresponse()
```

# Lenguajes y estándares para la Web

## Protocolo HTTP con Python: 02200-HTTP.py (II)

Grado en  
Ingeniería  
Informática  
del Software

```
except:
    print ('No se puede conectar al sitio Web ', sitioWeb)
    exit()

print("\nInformación del protocolo HTTP")

if respuesta.version==10:
    versionHTTP="HTTP/1.0"
if respuesta.version==11:
    versionHTTP="HTTP/1.1"

print("Versión           = ", versionHTTP)
print("Puerto HTTP       = ", http.client.HTTP_PORT)
print("Puerto HTTPS       = ", http.client.HTTPS_PORT)
print("Tipo de contenidos = ", respuesta.headers['Content-type'])
print("Fecha               = ", respuesta.headers['Date'])
print("Status              = ", respuesta.status)
print("Reason              = ", respuesta.reason)

print("\nLista de headers como diccionario:")
headers = dict(respuesta.getheaders())
pprint.pprint(headers)
```

# Lenguajes y estándares para la Web

## Protocolo HTTP con Python: 02200-HTTP.py (III)

Grado en  
Ingeniería  
Informática  
del Software

```
print("\nContenidos:")
contenidos = respuesta.read()
print("len(contenidos) = ", len(contenidos), " bytes")
archivo = open("contenidos-"+sitioWeb+".html", "wb")
archivo.write(contenidos)
archivo.close()
print("Creado el archivo = ", "contenidos-"+sitioWeb+".html")
```

```
def main():
    """Prueba de la función verHTTP()"""

    print(verHTTP.__doc__)

    miSitioWeb = input("Introduzca el sitio Web = ")

    verHTTP(miSitioWeb)

if __name__ == "__main__":
    main()
```



# Lenguajes y estándares para la Web

## Protocolo HTTP con Python: 02200-HTTP.py (IV)

```
Python-Win-02000-Servicios-Web — -zsh — 106x28
(base) juanmanuelcuevalovelle@MacBook-Air-de-Juan Python-Win-02000-Servicios-Web % python 02200-HTTP.py
Función verHTTP(sitioWeb)
Visualiza por pantalla el protocolo HTTP

Version: 1.1 16/Mayo/2020
Author: Juan Manuel Cueva Lovelle. Universidad de Oviedo

Introduzca el sitio Web = www.uniovi.es

Información del protocolo HTTP
Versión           = HTTP/1.1
Puerto HTTP      = 80
Puerto HTTPS     = 443
Tipo de contenidos = None
Fecha            = None
Status           = 301
Reason           = Moved permanently

Lista de headers como diccionario:
{'Cache-Control': 'no-cache',
 'Connection': 'close',
 'Location': 'https://www.uniovi.es/',
 'Pragma': 'no-cache'}

Contenidos:
len(contenidos)    = 0 bytes
Creado el archivo = contenidos-www.uniovi.es.html
(base) juanmanuelcuevalovelle@MacBook-Air-de-Juan Python-Win-02000-Servicios-Web %
```

# Lenguajes y estándares para la Web

## Protocolo HTTP con Python: 02200-HTTP.py (V)

Grado en  
Ingeniería  
Informática  
del Software

```
Python-Win-02000-Servicios-Web — zsh — 106x44
(base) juanmanuelcueva Lovelle@MacBook-Air-de-Juan Python-Win-02000-Servicios-Web % python 02200-HTTP.py
Función verHTTP(sitioWeb)
Visualiza por pantalla el protocolo HTTP

Version: 1.1 16/Mayo/2020
Author: Juan Manuel Cueva Lovelle. Universidad de Oviedo

Introduzca el sitio Web = www.google.com

Información del protocolo HTTP
Versión          = HTTP/1.1
Puerto HTTP     = 80
Puerto HTTPS    = 443
Tipo de contenidos = text/html; charset=ISO-8859-1
Fecha           = Sun, 05 Nov 2023 01:32:50 GMT
Status          = 200
Reason          = OK

Lista de headers como diccionario:
{'Accept-Ranges': 'none',
 'Cache-Control': 'private, max-age=0',
 'Content-Security-Policy-Report-Only': "object-src 'none';base-uri "
                                         "'self';script-src "
                                         "'nonce-8q_2rAqOtCwHJaQRb2ilIA' "
                                         "'strict-dynamic' 'report-sample' "
                                         "'unsafe-eval' 'unsafe-inline' https: "
                                         "'http;report-uri "
                                         "'https://csp.withgoogle.com/csp/gws/other-hp',
 'Content-Type': 'text/html; charset=ISO-8859-1',
 'Date': 'Sun, 05 Nov 2023 01:32:50 GMT',
 'Expires': '-1',
 'Server': 'gws',
 'Set-Cookie': 'AEC=Ackid1QrdosTFNMcgDYTHhzN35gojB2DrPJFvLUQagHkPDFe2Wp75DzfIg; '
               'expires=Fri, 03-May-2024 01:32:50 GMT; path=/; '
               'domain=.google.com; Secure; HttpOnly; SameSite=lax',
 'Transfer-Encoding': 'chunked',
 'Vary': 'Accept-Encoding',
 'X-Frame-Options': 'SAMEORIGIN',
 'X-XSS-Protection': '0'}

Contenidos:
len(contenidos) = 52715 bytes
Creado el archivo = contenidos-www.google.com.html
(base) juanmanuelcueva Lovelle@MacBook-Air-de-Juan Python-Win-02000-Servicios-Web %
```

# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- **Protocolo HTTPS**
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos



# Lenguajes y estándares para la Web

## Protocolo HTTPS (I)

Grado en  
Ingeniería  
Informática  
del Software

- **Hypertext Transfer Protocol Secure** o HTTPS
- El **protocolo seguro de transferencia de hipertexto** es una extensión del protocolo HTTP, destinado a la transferencia segura de datos de hipertexto en la Web
- Es la versión segura de HTTP
- El sistema HTTPS utiliza un cifrado basado en la seguridad de textos SSL/TLS para crear un canal cifrado
  - **SSL = Secure Sockets Layer**
  - **TLS = Transport Layer Security**
- El nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente

# Lenguajes y estándares para la Web

## Protocolo HTTPS (II)

Grado en  
Ingeniería  
Informática  
del Software

- De este modo se consigue que la información sensible (habitualmente nombre de usuario y clave) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará muy difícil de descifrar.
- El puerto estándar para este protocolo es el **443**.
  - También se usa el 4433
  - [https://es.wikipedia.org/wiki/Protocolo\\_seguro\\_de\\_transferencia\\_de\\_hipertexto](https://es.wikipedia.org/wiki/Protocolo_seguro_de_transferencia_de_hipertexto)

# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- **Tipos de computación en el cliente**
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

# Lenguajes y estándares para la Web

## Tipos de computación en el cliente

Grado en  
Ingeniería  
Informática  
del Software

- El agente de usuario o navegador es la interfaz del cliente Web con el servidor Web
- La respuesta que recibe el cliente es un archivo HTML
- El navegador interpreta el archivo HTML
- En el archivo HTML pueden venir **dos tipos** de etiquetas que definen la computación en el cliente:
  - Etiquetas **<object>** y **<embed>**
  - Etiqueta **<script>**
- El navegador ejecuta las instrucciones para realizar la computación en la máquina cliente Web
- La computación en el cliente Web también se denomina *computación del lado del cliente* (“*client-side programming*”)

# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- **Computación con las etiquetas <object> y <embed>**
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- Ejercicios propuestos

# Lenguajes y estándares para la Web

## Computación con las etiquetas <object> y <embed> (I)

Grado en  
Ingeniería  
Informática  
del Software

- En HTML la etiqueta **<object>** indica que existe un recurso externo.
- La etiqueta **<embed>** también indica que existe un recurso externo.
  - Se ha incluido en el estándar de HTML5 para que sea compatible con algunas versiones antiguas.
  - Anteriormente estaba implementada en la mayor parte de los navegadores, pero no era estándar.
- Dependiendo del tipo de recurso puede ser tratado como
  - una imagen
  - un navegador incrustado en el contexto
  - un recurso externo que debe ser procesado con un “*plug-in*”

# Lenguajes y estándares para la Web

## Computación con las etiquetas <object> y <embed> (II)

Grado en  
Ingeniería  
Informática  
del Software

- La especificación de las etiquetas **<object>** y **<embed>** en HTML5 se puede encontrar en W3C
  - <https://www.w3.org/TR/html52/semantics-embedded-content.html#the-object-element>
  - <https://www.w3.org/TR/html52/semantics-embedded-content.html#the-embed-element>
- También puede verse la evolución y **parámetros** de <object> y <embed> desde versiones anteriores de HTML en los sitios web de los desarrolladores de los navegadores
  - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/object>
  - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/embed>

# Lenguajes y estándares para la Web

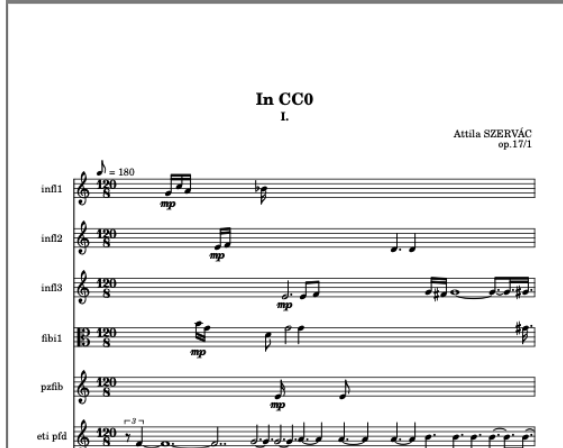
## Computación con las etiquetas <object> y <embed> (III)

- Ejemplo de <object>

Grado en  
Ingeniería  
Informática  
del Software

HTML Demo: <object>

RESET

HTML	CSS	OUTPUT
<pre>1 &lt;object type="application/pdf" 2 data="/media/examples/In-CC0.pdf" width="250" height="200"&gt;&lt;/object&gt;</pre>		



# Lenguajes y estándares para la Web

## Computación con las etiquetas <object> y <embed> (IV)

Grado en  
Ingeniería  
Informática  
del Software

- Ejemplo de <object>

```
<!-- Incluir un archivo flash -->
```

```
<object data="película.swf" type="application/x-shockwave-flash">
```

```
</object>
```

```
<!-- Incluir un archivo flash con parámetros -->
```

```
<object data="película.swf" type="application/x-shockwave-flash">
```

```
<param name="sonido" value="estereo" />
```

```
</object>
```

# Lenguajes y estándares para la Web

## Computación con las etiquetas <object> y <embed> (V)

Grado en  
Ingeniería  
Informática  
del Software

- Ejemplo de <embed>

```
<embed type="video/quicktime" src="pelicula.mov" width="640" height="480" />
```

# Lenguajes y estándares para la Web

## Computación con la etiqueta <object>: plug-ins

Grado en  
Ingeniería  
Informática  
del Software

- En HTML la etiqueta **<object>** permite incluir elementos computacionales
- El navegador reconoce el tipo de elemento y lo ejecuta
- Es necesario instalar “**plug-ins**” para ejecutar los elementos computacionales.
- Ejemplo:
  - Se ejecuta un juego escrito en el lenguaje de programación Python si está instalado el “*plug-in*” de Python.

```
<html>
<head><title>Ejemplo de juego en Python</title></head>
<body>
<p>
<object data="juego.py" type="application/x-python"
        title="Juego de lógica">
</object>
</p>
</body>
</html>
```

# Lenguajes y estándares para la Web

## Computación con la etiqueta <object>: applet

Grado en  
Ingeniería  
Informática  
del Software

- Históricamente el navegador Netscape 2.0 introdujo la primera etiqueta para incluir elementos computacionales
- La etiqueta se denominaba **<applet>** y permitía ejecutar los componentes de código denominados Applets desarrollados en el lenguaje Java.
- Era necesario tener instalado un “plug-in”: la Máquina Virtual de Java (**JVM**).
- Hoy en día la etiqueta <applet> está desaprobadada (“**deprecated**”) y se ha sustituido por **<object>**
- **Ejemplo** de applet usando la etiqueta <object> y mensaje que se genera si el “plug-in” no está instalado

```
<object type="application/x-java-applet">  
<param name="code" value="MiAppletJava" />  
<p>No tiene activada la máquina virtual de Java</p>  
</object>
```

# Lenguajes y estándares para la Web

## Computación con la etiqueta <object>: ActiveX

Grado en  
Ingeniería  
Informática  
del Software

- **ActiveX** es una especificación desarrollada por **Microsoft**
- Permite que programas desarrollados en el sistema operativo Windows se ejecuten en una página Web.
- Los ActiveX se pueden desarrollar en Visual C++ o Visual Basic
- Una vez compilados **los ejecutables se instalan en el servidor Web**
- Los ActiveX se descargan en la máquina cliente y se ejecutan por el navegador (al igual que los Java applets)
- A diferencia de los Java applets **se instalan permanentemente** en la máquina cliente, eliminando la necesidad de volver a ser descargados.
- Su principal ventaja e inconveniente es que pueden hacer cualquier cosa en la máquina cliente, pudiendo generar **problemas de seguridad**.
- Los ActiveX incluyen **firma digital** indicando su procedencia.

# Lenguajes y estándares para la Web

## Computación con la etiqueta <object>: flash

Grado en  
Ingeniería  
Informática  
del Software

- La etiqueta **<object>** también se puede utilizar para incluir animaciones o elementos multimedia creados con *flash* (producto de Adobe).
- Ejemplo para incrustar *flash* con <object> en HTML

```
<object data="pelicula.swf" type="application/x-shockwave-flash"></object>
```

```
<!-- Incrustar una pelicula flash con parámetros -->
```

```
<object data="pelicula.swf" type="application/x-shockwave-flash">
```

```
<param name="code" value="miPelicula" />
```

```
</object>
```

- Para que funcione es necesario tener instalado el “*plug-in*” de *flash*

# Lenguajes y estándares para la Web

## Computación con la etiqueta <object>: silverlight

Grado en  
Ingeniería  
Informática  
del Software

- La etiqueta **<object>** también se puede utilizar para incluir animaciones o elementos multimedia creados con **silverlight** (producto de Microsoft).
- Ejemplo para incrustar *silverlight* con <object> en HTML

```
<object width="230" height="380"  
  data="data:application/x-silverlight-2,"  
  type="application/x-silverlight-3" >  
  <param name="source" value="ClientBin/SilverlightWeatherControl.xap"/>  
</object>
```

- Para que funcione es necesario tener instalado el “*plug-in*” de *silverlight* disponible en
  - <http://www.microsoft.com/silverlight/>

# Lenguajes y estándares para la Web

## Computación con las etiquetas <object> y <embed>: pros y contras

Grado en  
Ingeniería  
Informática  
del Software

- **Ventajas** de la computación en el cliente Web con las etiquetas <object> y <embed>
  - Reducción de la carga computacional en el servidor Web
  - Reducción del ancho de banda en las comunicaciones en el momento de la computación.
  - Se baja una vez el elemento a ejecutar y el resto de las veces ya se ejecuta en modo local en el cliente Web.
- **Inconvenientes** de la computación en el cliente Web con las etiquetas <object> y <embed>
  - Dependencia de las capacidades de computación del cliente
  - Necesidad de instalar el “plug-in” adecuado
  - **Problemas de seguridad** al permitirse la instalación de “plug-in” maliciosos con *malware* (virus, troyanos,...) a usuarios no técnicos que los confunden con los “plug-in” correctos.
  - Habitualmente los administradores del sistema restringen las capacidades de los usuarios para instalar “plug-in” y por defecto dejan instalados los básicos.
  - En algunos no se garantiza la accesibilidad para personas con necesidades especiales.
  - No es buena idea construir un “plug-in” propio y distribuirlo. Salvo las grandes empresas u organizaciones no se genera la confianza necesaria para que se instale.



# Lenguajes y estándares para la Web

## Computación con las etiquetas <object> y <embed>: pros y contras

Grado en  
Ingeniería  
Informática  
del Software

- **Ventajas** de la computación en el cliente Web con las etiquetas <object> y <embed>
  - Reducción de la carga computacional en el servidor Web
  - Reducción del ancho de banda en las comunicaciones en el momento de la computación.
  - Se baja una vez el elemento a ejecutar y el resto de la vez se ejecuta en modo local en el cliente Web.
- **Inconvenientes** de la computación en el cliente Web con las etiquetas <object> y <embed>
  - Dependencia de las capacidades de computación del cliente
  - Necesidad de instalar el plug-in adecuado
  - **Problemas de seguridad** permitirse la instalación de “plug-in” maliciosos con *malware* (virus, trojan, ...) a usuarios no técnicos que los confunden con los “plug-in” correctos.
  - Habitualmente los administradores del sistema restringen las capacidades de los usuarios para instalar “plug-in” y por defecto dejan instalados los básicos.
  - En algunos no se garantiza la accesibilidad para personas con necesidades especiales.
  - No es buena idea construir un “plug-in” propio y distribuirlo. Salvo las grandes empresas u organizaciones no se genera la confianza necesaria para que se instale.

# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- **Computación con la etiqueta <script>**
- Referencias
- Bibliografía
- Ejercicios propuestos

# Lenguajes y estándares para la Web

## Computación con la etiqueta <script>

Grado en  
Ingeniería  
Informática  
del Software

- La etiqueta **<script>** permite incluir código fuente directamente en el archivo HTML que es interpretado y ejecutado por el navegador.
- El lenguaje de programación habitual es un dialecto del estándar **ECMAScript**.
- El código fuente siempre es visible y evita que se introduzcan aspectos maliciosos en el cliente.
- **No necesita** la instalación de “**plug-in**”, el navegador se ocupa de todo.
- Solamente es necesario tener instalado un navegador que interprete y ejecute el dialecto de **ECMAScript**.
- Solamente los navegadores (y algunos de dispositivos móviles) muy antiguos no soportan ECMAScript

# Lenguajes y estándares para la Web

## Computación con la etiqueta <script>

Grado en  
Ingeniería  
Informática  
del Software

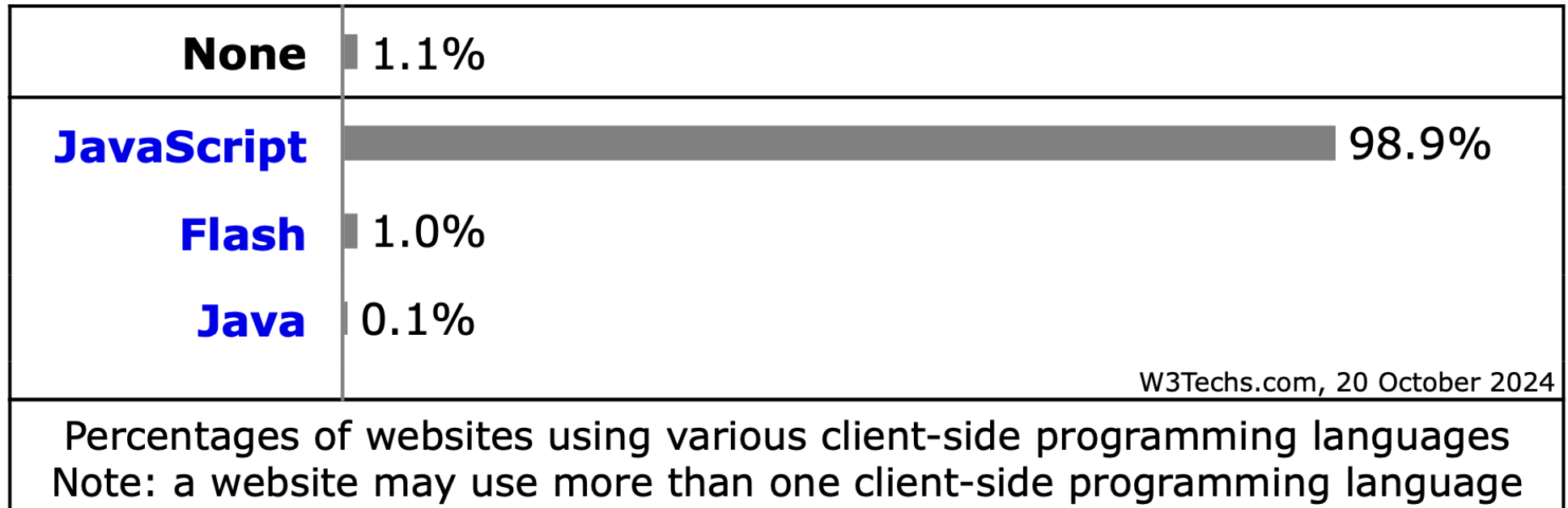
- El uso de la etiqueta <script> ha generado lo que se denominó DHTML.
- **DHTML** (*Dynamic HTML*) **OBSOLETO**
  - designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático, un lenguaje interpretado en el lado del cliente (como JavaScript), el lenguaje de hojas de estilo en cascada (CSS) y la jerarquía de objetos de un DOM (Modelo Objeto Documento).
  - El principal problema es que cada navegador desarrolló su propio DHTML
  - La solución ha sido el desarrollo del estándar DOM del W3C

# Lenguajes y estándares para la Web

## Estadística de uso de tecnologías

Grado en  
Ingeniería  
Informática  
del Software

- Estadística de uso de tecnologías en la computación en el cliente Web
  - [http://w3techs.com/technologies/overview/client\\_side\\_language/all](http://w3techs.com/technologies/overview/client_side_language/all)



# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- **Referencias**
  - Bibliografía
  - Ejercicios propuestos

# Lenguajes y estándares para la Web

## Referencias (I)

Grado en  
Ingeniería  
Informática  
del Software

- W3C: Tecnologías de script
  - <http://www.w3.org/standards/webdesign/script>
- Sobre el lenguaje Python
  - <http://www.python.org>
- Computación en el cliente Web con Python
  - <https://wiki.python.org/moin/WebClientProgramming>
- Tutorial para escribir applets en Java
  - <http://docs.oracle.com/javase/tutorial/deployment/applet/index.html>
- Desarrollo de aplicaciones con flash
  - <http://www.adobe.com/devnet/flashplayer.html>

# Lenguajes y estándares para la Web

## Referencias (II)

Grado en  
Ingeniería  
Informática  
del Software

- Microsoft Silverlight
  - <http://www.microsoft.com/silverlight/>
- Tutorial sobre <object> y <embed>
  - [http://www.w3schools.com/html/html\\_object.asp](http://www.w3schools.com/html/html_object.asp)
- Estadística de uso de tecnologías en la computación en el cliente Web
  - [http://w3techs.com/technologies/overview/client\\_side\\_language/all](http://w3techs.com/technologies/overview/client_side_language/all)
- Protocolos HTTP y HTTPS
  - [https://en.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
  - <https://en.wikipedia.org/wiki/HTTPS>



# Lenguajes y estándares para la Web

## Esquema

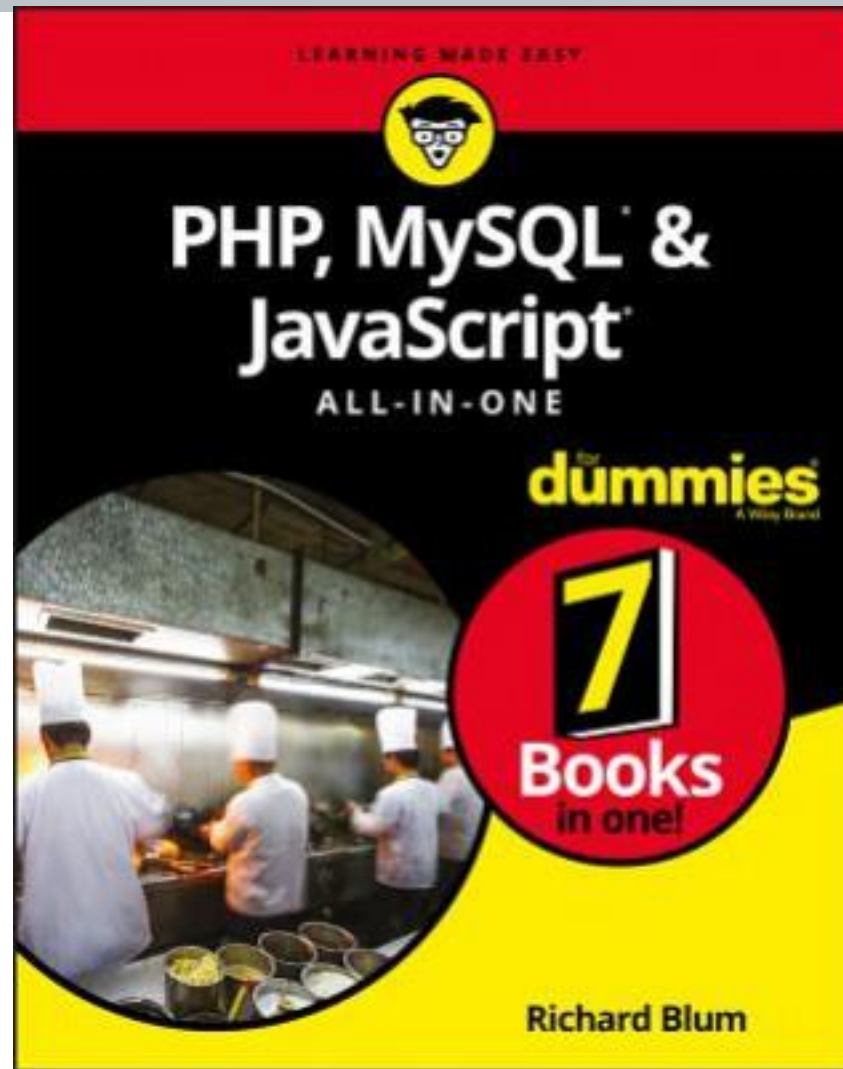
Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- **Bibliografía**
- Ejercicios propuestos

# Lenguajes y estándares para la Web

## Bibliografía

- **“PHP, MySQL, JavaScript. ALL-IN-ONE FOR DUMMIES”**
- Richard Blum
- John Wiley & Sons, Inc (2018).
- 800 páginas
- Presenta una visión general de las tecnologías Web



Grado en  
Ingeniería  
Informática  
del Software

# Lenguajes y estándares para la Web

## Esquema

Grado en  
Ingeniería  
Informática  
del Software

- Sitios Web estáticos y dinámicos
- Computación el cliente y en el servidor
- Protocolos TCP/IP
- Protocolo HTTP
- Protocolo HTTPS
- Tipos de computación en el cliente
- Computación con las etiquetas <object> y <embed>
- Computación con la etiqueta <script>
- Referencias
- Bibliografía
- **Ejercicios propuestos**

# Lenguajes y estándares para la Web

## Ejercicios propuestos (I): OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Desarrollar una aplicación en **Python** para ser ejecutada en el cliente usando un “*plug-in*”
- Probar la aplicación haciendo una instalación en el servidor Web y ejecutarla desde un cliente Web
- Escribir un documento en PDF comentando las **advertencias de seguridad** que recibe el usuario y los peligros potenciales que le pueden ocurrir. Incluir las capturas de pantalla con las advertencias. Indicar las versiones del lenguaje Python y su “*plug-in*”.

# Lenguajes y estándares para la Web

## Ejercicios propuestos (II): OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Desarrollar una aplicación en **Java (applet)** para ser ejecutada en el cliente usando la **JVM (Java Virtual Machine)**
- Probar la aplicación haciendo una instalación en el servidor Web y ejecutarla desde un cliente Web
- Escribir un documento en PDF comentando las **advertencias de seguridad** que recibe el usuario y los peligros potenciales que le pueden ocurrir. Incluir las capturas de pantalla con las advertencias. Indicar las versiones del lenguaje Java y la JVM

# Lenguajes y estándares para la Web

## Ejercicios propuestos (III): OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Desarrollar una aplicación creando un componente **ActiveX** para ser ejecutada en el cliente Web
- Probar la aplicación haciendo una instalación en el servidor Web y ejecutarla desde un cliente Web
- Escribir un documento en PDF comentando las **advertencias de seguridad** que recibe el usuario y los peligros potenciales que le pueden ocurrir. Incluir las capturas de pantalla con las advertencias. Indicar el lenguaje y la versión con la que se ha creado el ActiveX.

# Lenguajes y estándares para la Web

## Ejercicios propuestos (IV): OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Desarrollar una aplicación creando un componente **flash** para ser ejecutada en el cliente Web
- Probar la aplicación haciendo una instalación en el servidor Web y ejecutarla desde un cliente Web
- Escribir un documento en PDF comentando las **advertencias de seguridad** que recibe el usuario y los peligros potenciales que le pueden ocurrir. Incluir las capturas de pantalla con las advertencias. Indicar la versión con la que se ha creado el archivo *flash*.

# Lenguajes y estándares para la Web

## Ejercicios propuestos (V): OPTATIVO

Grado en  
Ingeniería  
Informática  
del Software

- Desarrollar una aplicación creando un componente **silverlight** para ser ejecutada en el cliente Web
- Probar la aplicación haciendo una instalación en el servidor Web y ejecutarla desde un cliente Web
- Escribir un documento en PDF comentando las **advertencias de seguridad** que recibe el usuario y los peligros potenciales que le pueden ocurrir. Incluir las capturas de pantalla con las advertencias. Indicar la herramienta y la versión con la que se ha creado el archivo *silverlight*.



# Tema 1: Computación en la Web

# Grado en Ingeniería Informática del Software

