

# **Fundamentos de Aplicaciones y Servicios Telemáticos**

2º Grado en Ingeniería de Tecnologías de Telecomunicación

Departamento de Ingeniería Telemática



## **Tema 01**

### **Programación Web Estática Introducción**

# Bloque I: Temas

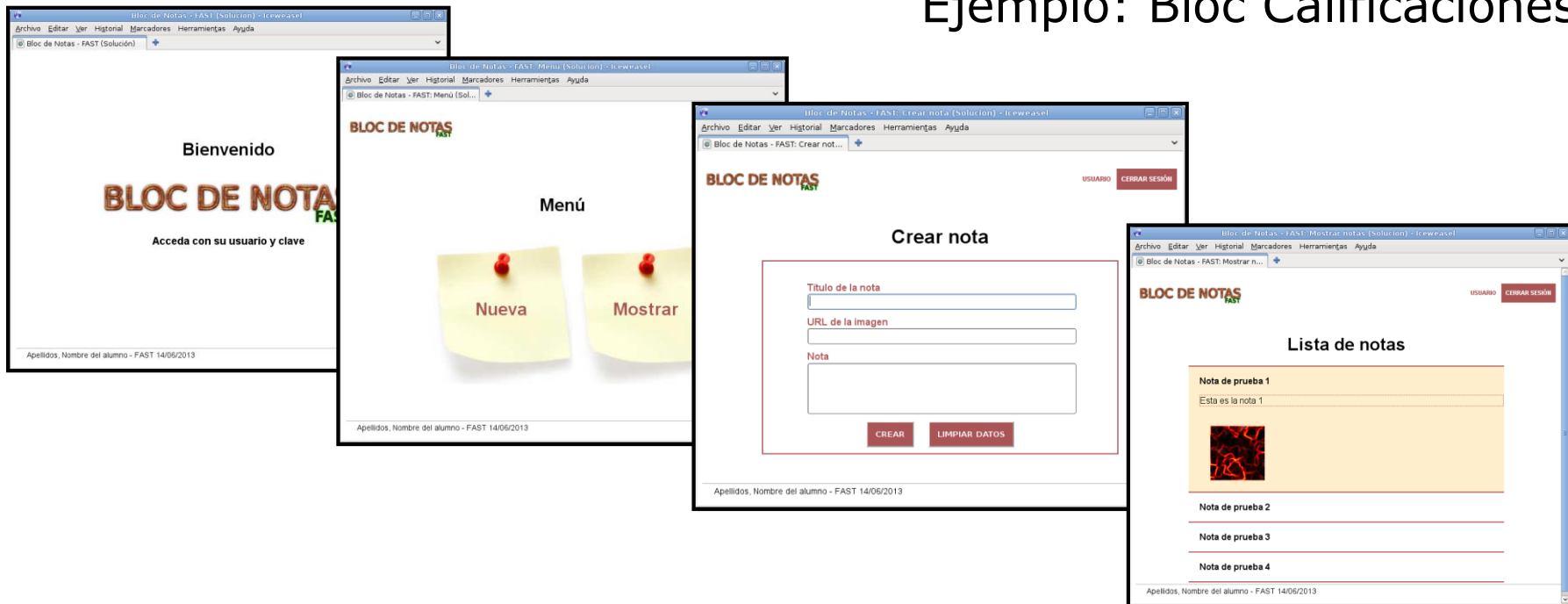
---

Bloque	Tema	Contenido
<b>I.</b> Programación <i>Web estática y dinámica en el cliente</i>	01	<b>Estática</b> (HTTP, HTML, CSS, XML)
	02	<b>Dinámica</b> en el <b>Cliente</b> (ECMAScript)
<b>II.</b> Programación <i>Web dinámica en el servidor y acceso a Bases de Datos</i>	03	<b>Dinámica</b> en el <b>Servidor</b> (Servlets, JSP)
	04	<b>Con Bases de Datos</b> (SQL, PostgreSQL)

# Bloque I: Objetivo

- Aprender a **Desarrollar Aplicaciones Web**
  - Motivación: Conocimiento ampliamente usado.
  - **Muchas Tecnologías** posibles (HTML, JSP, ColdFusion,...)
    - Visión General, Uso básico de las Principales y Aprender a consultar las referencias!.

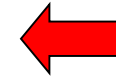
Ejemplo: Bloc Calificaciones



# Contenido del Tema (Teoría + Práctica)

---

## □ Introducción a la Programación Web



### Objetivo

1. Aplicación Web
2. Arquitectura funcional de una Aplicación Web: Lógica/Interfaz
3. Lenguajes de Programación Web
  - Clasificación
  - Uso simultáneo de varios lenguajes

## □ HTTP (Hypertext Transfer Protocol)

## □ HTML (Hypertext Markup Language)

## □ CSS (Cascading Style Sheets)

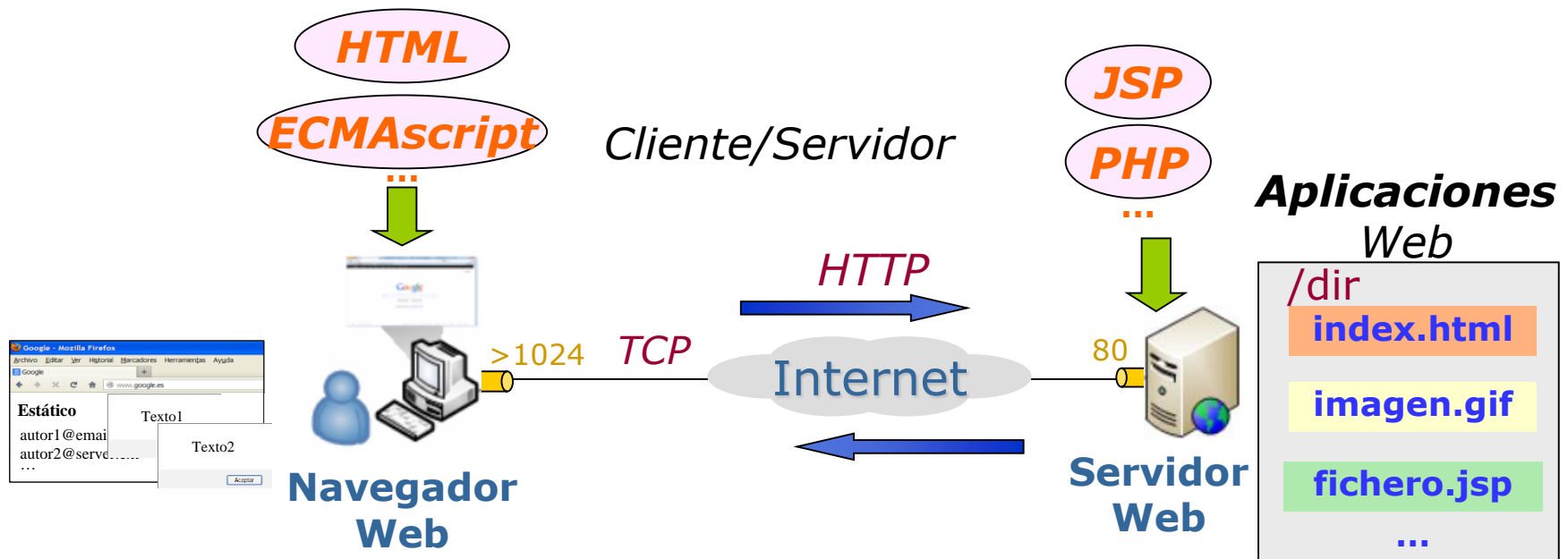
## □ XML (eXtensible Markup Language)

Normalizados por W3C  
(WWW Consortium):

<http://www.w3.org>

# Objetivo de la Introducción a la Programación Web

- **Conocer el Entorno** en que funcionan las Aplicaciones Web.
  - Fundamental para programarlas correctamente.



**URL:**

<http://www.servidor.ext/web/index.html>

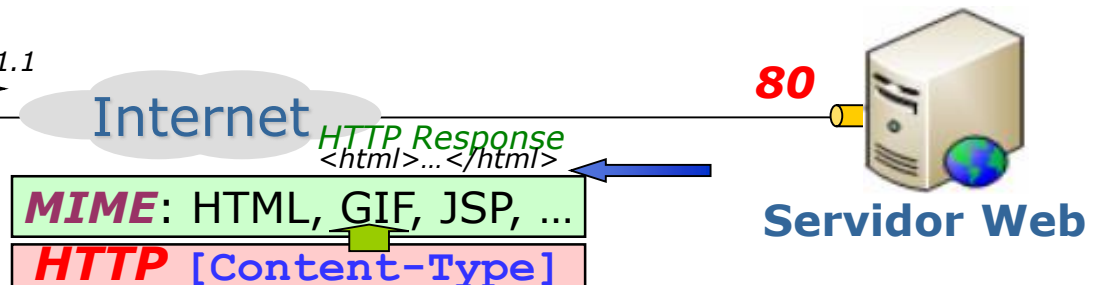
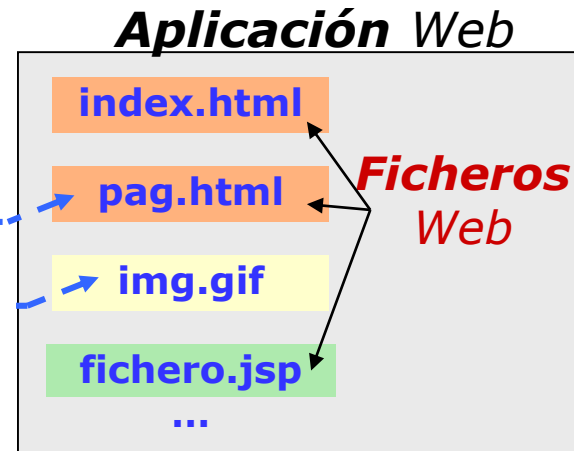
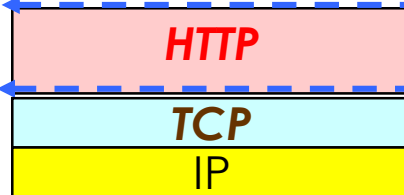
# 1. Aplicación Web

- **Aplicación** ("página") **Web**: conjunto de ficheros con lenguaje/s Web (HTML, PHP, JSP, ...) y recursos adicionales (imágenes, ...).
  - Usada mediante servidor/cliente Web con HTTP (URL).
  - No confundir con Servicios Web (ejecutar código remoto, usan HTTP): Corba, SOAP, ...

**URLs:**

**http://IP/web/pag.html**

**http://IP/web/img.gif**



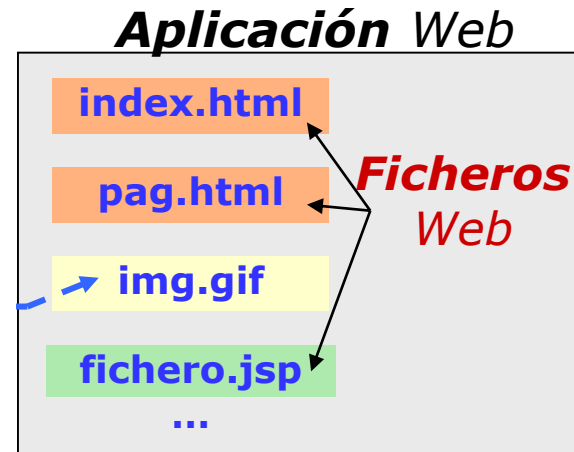
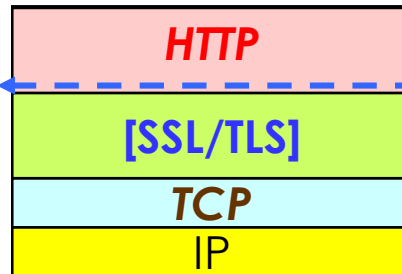
# 1. Aplicación Web "segura"

- HTTPS = HTTP + SSL/TLS (*cifrado, integridad y autenticación*)

URLs:

**http**://IP:443/web/ X

**https**://IP/web/img.gif

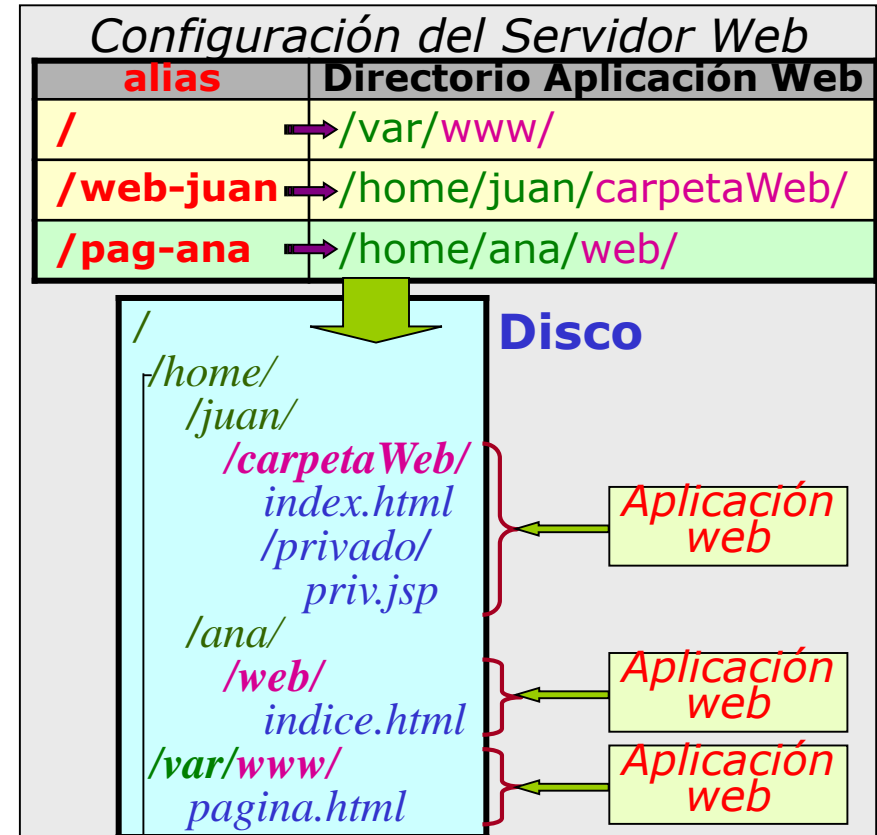
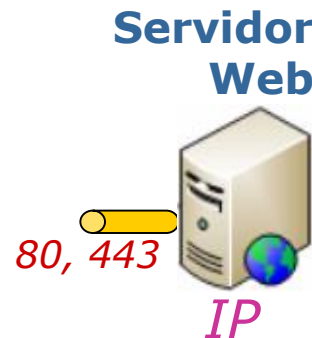


Internet



# 1. Aplicación Web: URL y alias

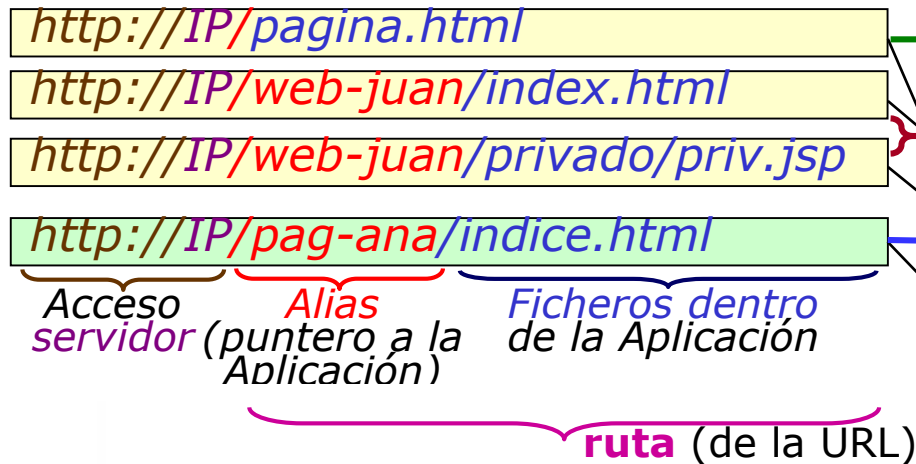
- Un mismo servidor Web puede alojar varias Aplicaciones Web.
- En general, cada Aplicación Web está en un directorio del servidor (con sus ficheros y subdirectorios).
- Servidor Web asocia un "alias" a dicho directorio.



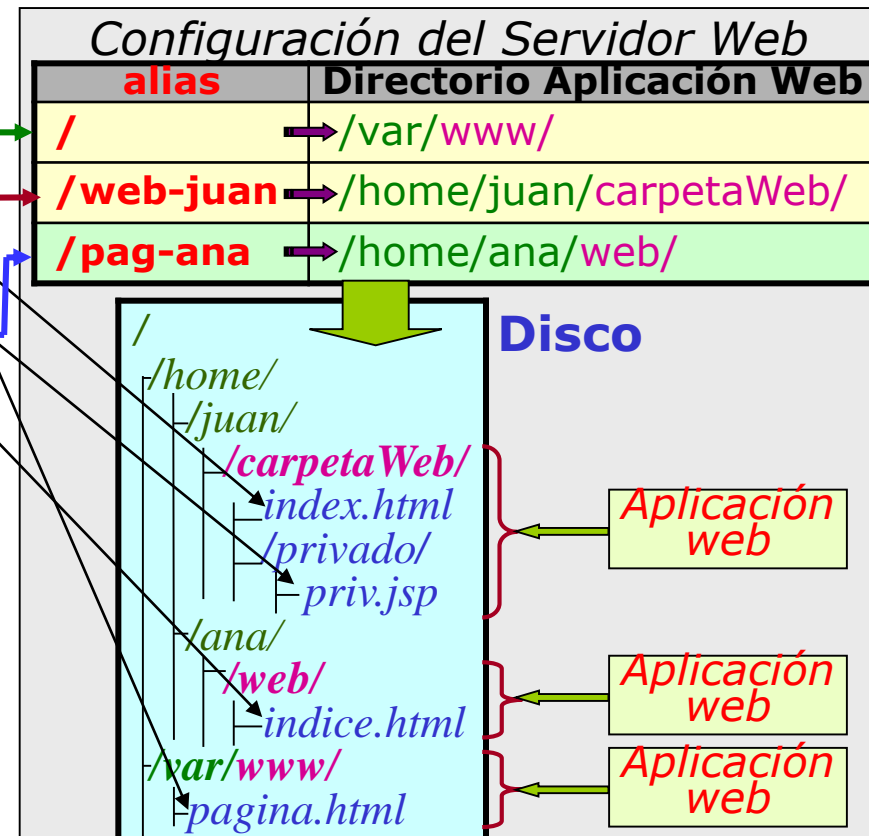
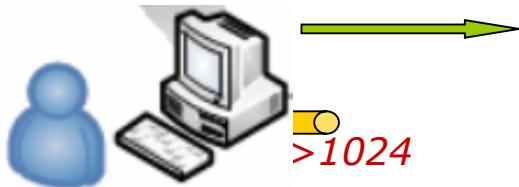


# 1. Aplicación Web : URL y acceso

## URLs

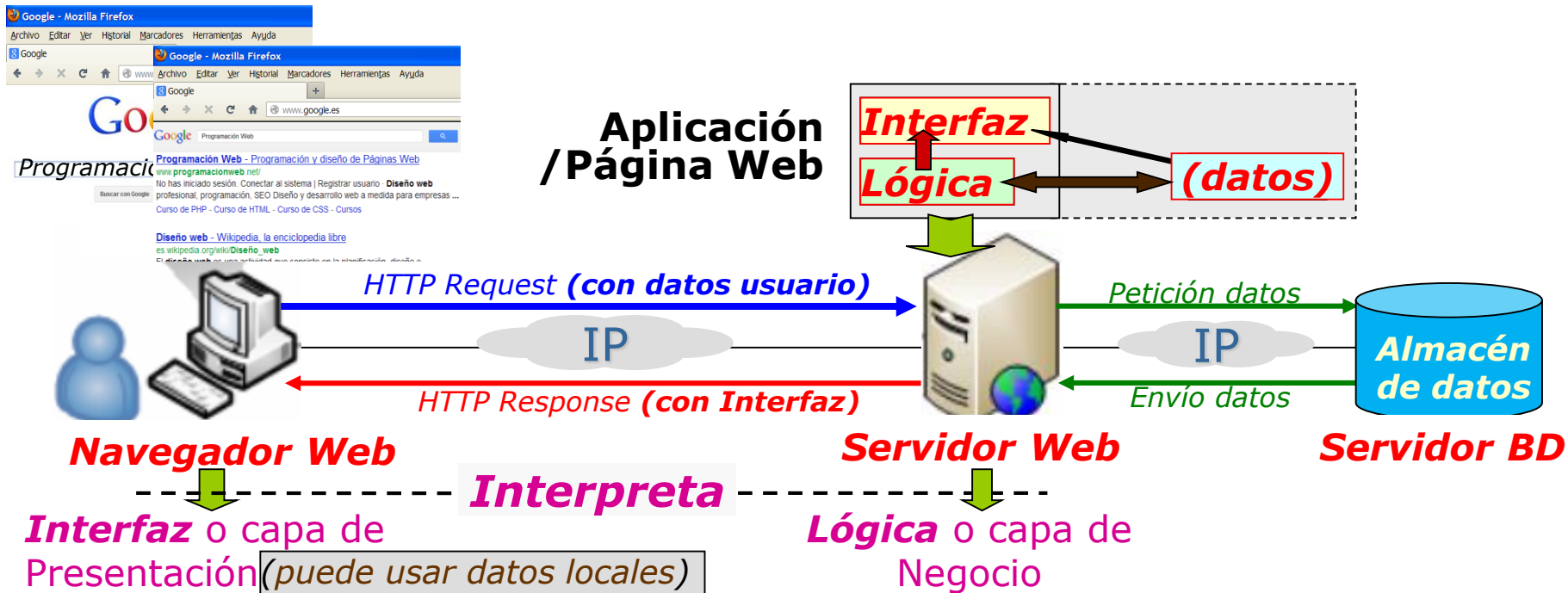


## Navegador Web



## 2. Arquitectura funcional de una Aplicación Web

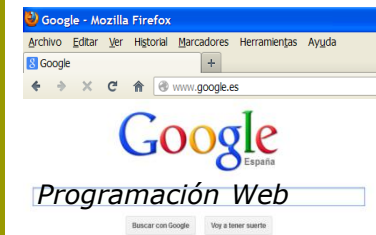
- Contenido de una Aplicación Web: **3 funciones**:
  - Interfaz: muestra información e interactúa con el usuario.
  - Lógica: **núcleo de la funcionalidad**.
  - Almacén de datos: usados por la lógica.
- **Alojamiento** (servidor)  $\neq$  lugar de **Interpretación**.



# 3. Lenguajes de Programación Web: Clasificación

- ❑ Estáticos: sólo información.
- ❑ Dinámicos (2 tipos): **código de programación.**

Nivel funcional	Lenguajes de Programación Web	Objetivo/Funcionalidad	Ejemplo
Interfaz	Estáticos	Contener y ubicar medios (texto, imagen, vídeo...)	HTML, CSS
	Dinámicos con interpretación en el <b>Cliente</b> (Navegad.)	Interactuar con el usuario (comprobar campos, recargar página, efectos dinámicos, ...)	ECMAScript (Javascript)
Lógica	Dinámicos con interpretación en el <b>Servidor</b> Web	Funcionalidad principal (procesado de datos usuario, acceso a BD, cálculos, ...)	JSP, CGI, PHP

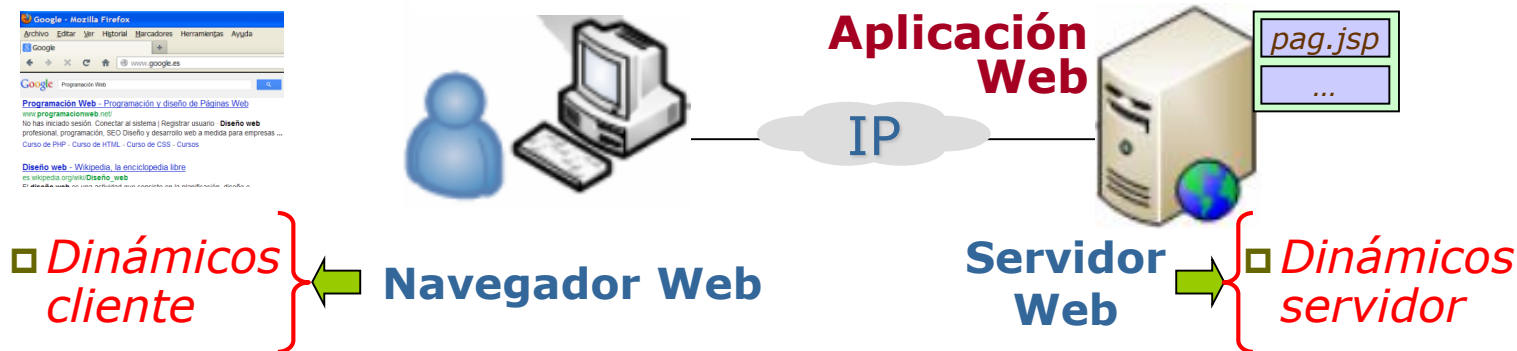


- ❑ Estáticos
- ❑ Dinámicos cliente
- ❑ Dinámicos servidor

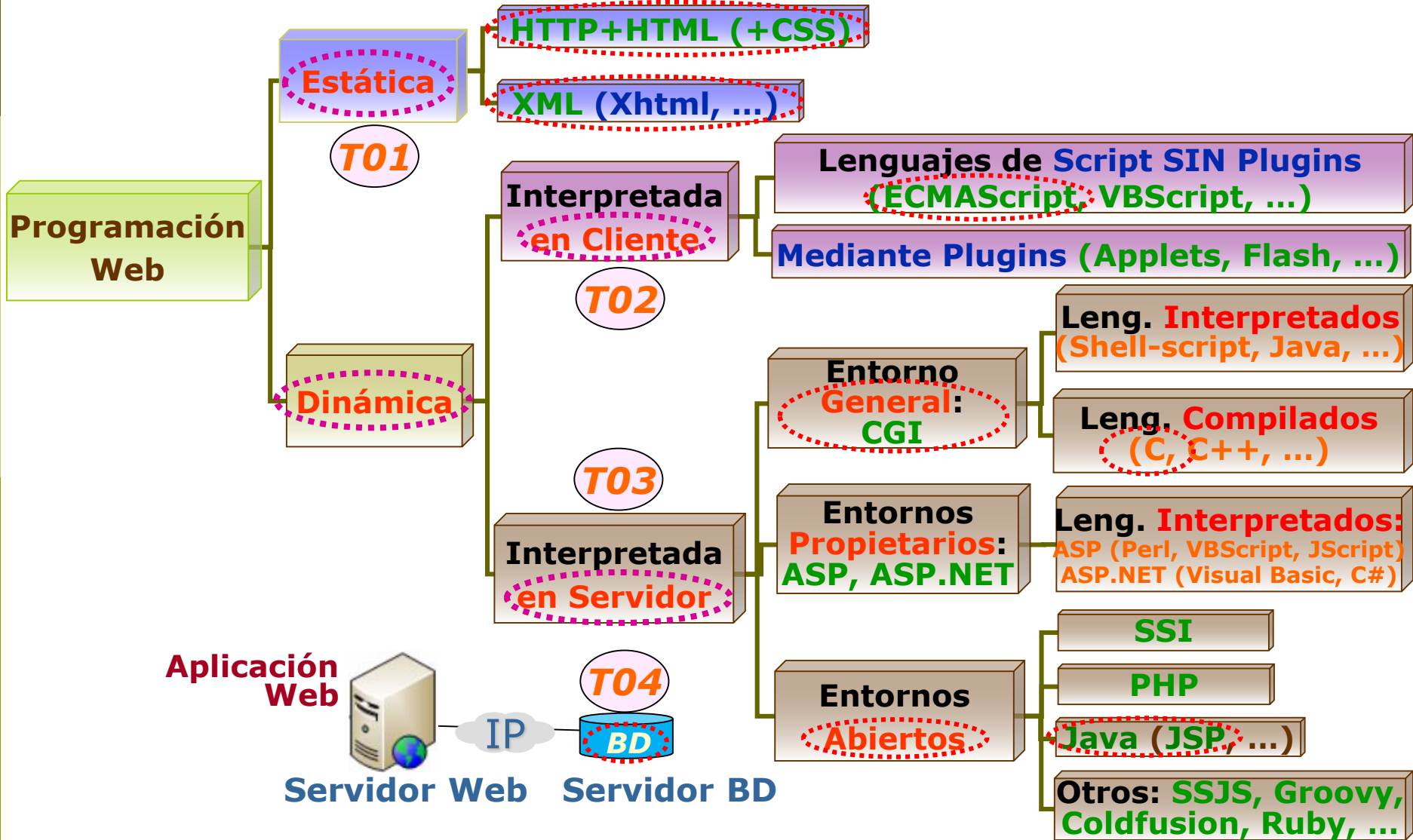
# 3. Lenguajes Programación Web: Dinámicos

## □Dinámicos en **Navegador** vs en **Servidor**:

Interpretado en	Ventajas	Inconvenientes
<b>Navegador</b> (usualmente simples)	<ul style="list-style-type: none"> <li>□ Independencia servidor.</li> <li>□ No sobrecargan servidor.</li> <li>□ Sin problemas seguridad en el servidor.</li> </ul>	<ul style="list-style-type: none"> <li>□ Variaciones entre navegadores.</li> <li>□ Dependen configuración navegador.</li> <li>□ Acceso limitado a servicios.</li> </ul>
<b>Servidor</b> (suelen funcionalidad principal)	<ul style="list-style-type: none"> <li>□ Independencia navegador.</li> <li>□ Acceso a múltiples servicios.</li> </ul>	<ul style="list-style-type: none"> <li>□ Sobrecarga (un servidor, muchos usuarios).</li> <li>□ Seguridad (usuarios externos).</li> </ul>



### 3. Lenguajes Prog. Web: Principales Lenguajes



### 3. Lenguajes Prog. Web: Varios Lenguajes juntos

- Es habitual un **mismo fichero** Web contenga **varios lenguajes**:

	Estático	Dinámico cliente	Dinámico servidor
Nº lenguajes	1	0, 1 ó varios	0 ó 1

HTML {

JavaScript {

VBScript {

JSP {

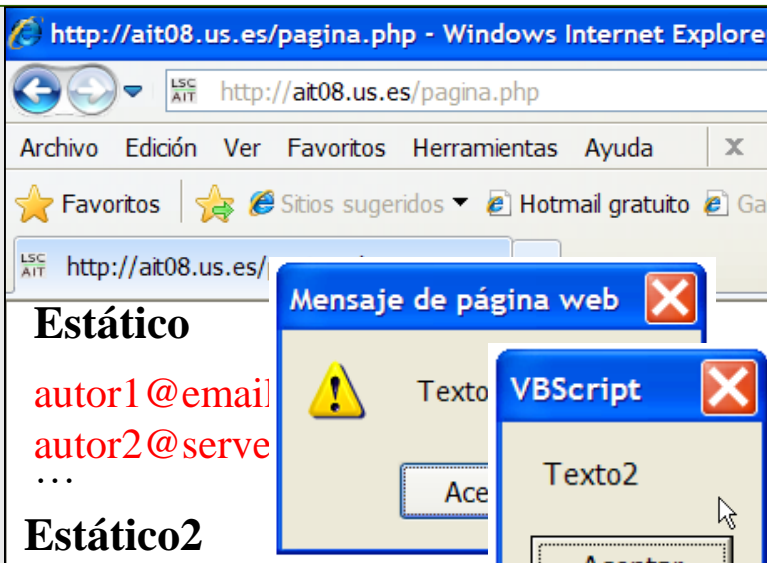
HTML {

```

<!DOCTYPE html ...>
<html><body>
  <h1>Estatico</h1>
  -----
  <script type="text/javascript">
    alert('Texto1');
  </script>
  <script type="text/vbscript">
    MsgBox ("Texto2")
  </script>
  -----
  <%
    Class.forName("org.postgresql.Driver").newInstance
    Connection con=DriverManager.getConnection(...);
    ... %>
  -----
  <h1>Estatico2</h1>
</body></html>

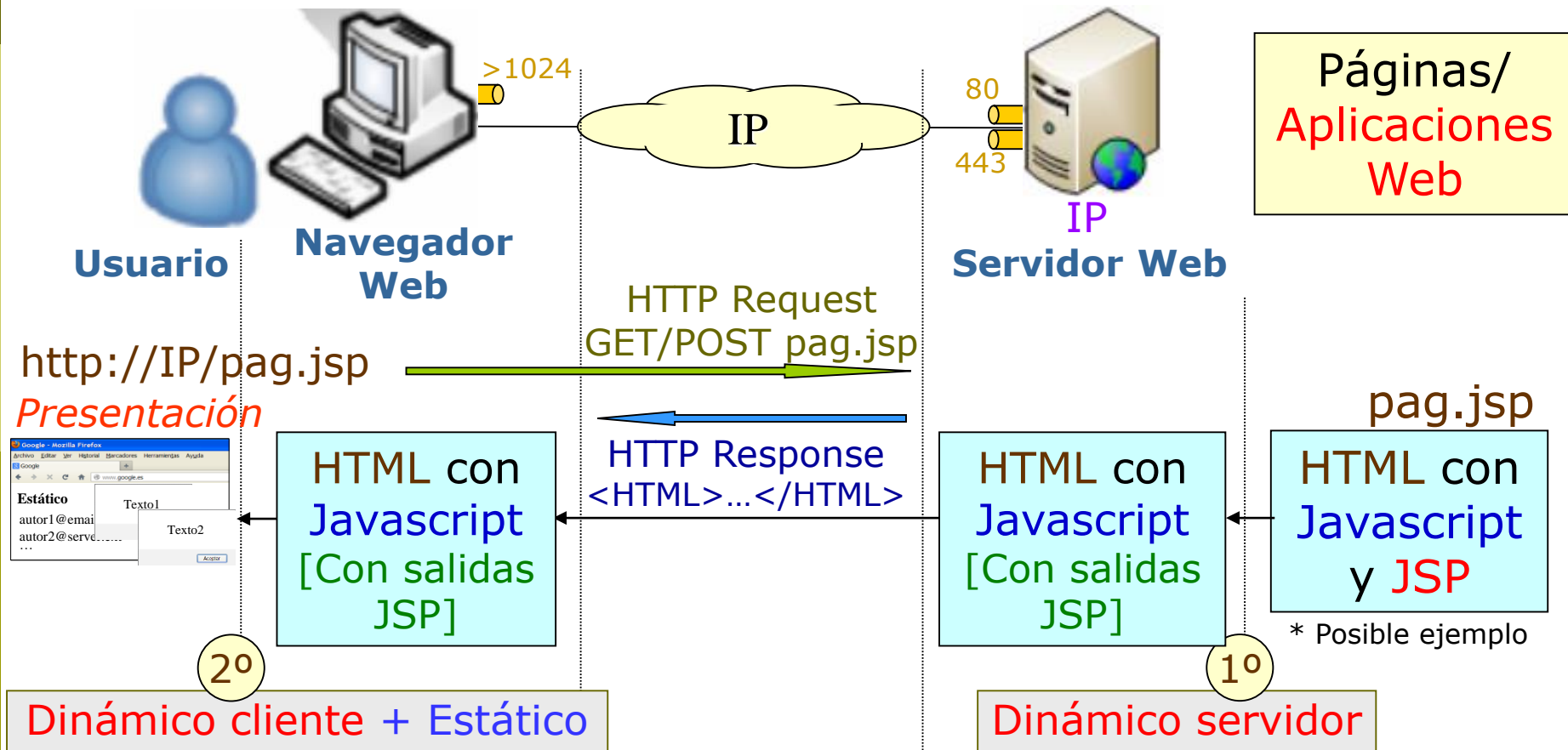
```

**pag.jsp**



### 3. Lenguajes Prog. Web: Varios Lenguajes (2)

- Cada lenguaje es interpretado en su correspondiente equipo:



\* El navegador va interpretando "Estático y Dinámico" conforme los va encontrando

## Resumen/Conclusiones

---

- ❑ Aplicación Web, HTTP/S, URL
- ❑ Clasificación de los Lenguajes de Programación Web
  - Estáticos, Dinámicos cliente, Dinámicos servidor
  - Varios en el mismo fichero



# Siguientes contenidos del tema:

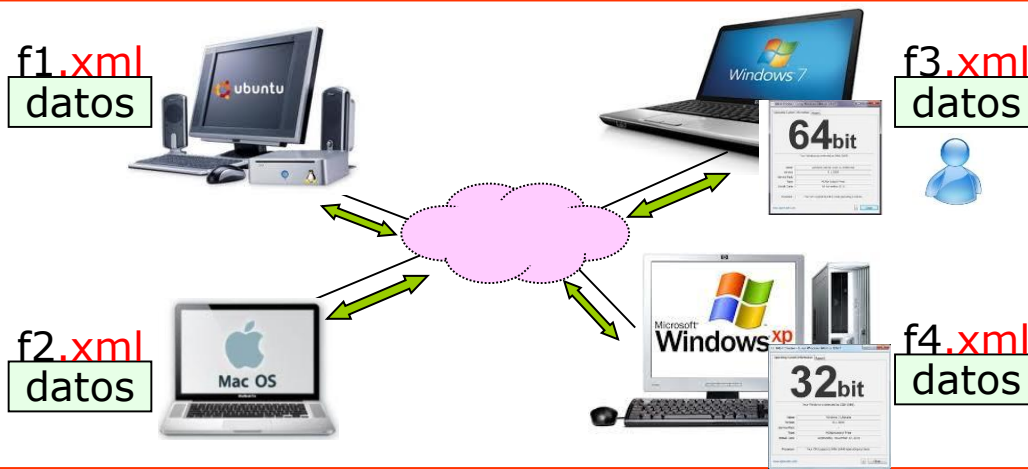
- Aprender a Diseñar Aplicaciones Web estáticas.



□ HTTP (protocolo de transferencia)

□ Sólo HTML (usa estilo predeterminado del navegador)

□ Con CSS: estilo (formato de presentación) personalizado



□ XML: Formato de datos

- "universal" (cualquier equipo/aplicación)
- legible (jerárquico, etiquetas personalizadas)
- XHTML