

# DESARROLLO WEB EN ENTORNO CLIENTE

## TEMA 1: SELECCIÓN DE ARQUITECTURAS Y HERRAMIENTAS DE PROGRAMACIÓN

# Índice

- Evolución y características de los navegadores web
- Funcionamiento del navegador
- Arquitectura de ejecución
- Lenguajes y tecnologías de programación en entorno cliente
- Integración del código JavaScript con las etiquetas HTML

# 1. Evolución y características de los navegadores web

3

## ➤ Navegador web:

- ❑ Componente software que se utiliza en el cliente y que permite acceder al contenido ofrecido por los servidores de Internet sin la necesidad de que el usuario instale un nuevo programa.
- ❑ Aplicación, distribuida habitualmente como software libre.
- ❑ Recursos descritos mediante una dirección URL (Universal Resource Locator).

# 1. Evolución y características de los navegadores web

4

## ➤ Ejemplos:

- ❑ Internet Explorer: Es el navegador de Microsoft.
- ❑ Mozilla Firefox: Se trata de un navegador de código abierto multiplataforma de gran aceptación.
- ❑ Google Chrome: Es el navegador de Google compilado a partir de componentes de código abierto.
- ❑ Safari: Es el navegador por defecto de los sistemas de Apple.
- ❑ Dolphin Browser: Específico para el sistema operativo Android, fue uno de los primeros en incluir soporte para navegación multitáctil.

# 1. Evolución y características de los navegadores web

5



Estadísticas de uso de navegadores

# 1. Evolución y características de los navegadores web

6

- Criterios de clasificación:
  - ❑ Plataforma de ejecución. Sistema operativo.
  - ❑ Características del navegador. Funcionalidades adicionales.
  - ❑ Personalización de la interfaz. Funciones de accesibilidad.
  - ❑ Soporte de tecnologías Web. Grado de soporte de los estándares de la Web.
  - ❑ Licencia de software. Código libre y navegadores propietarios.

## 2. Funcionamiento del navegador

7

- Función principal: solicitar al servidor los recursos web que elija el usuario (HTML, PDF, Word, audio, etc.).
- Usuario especifica la ubicación del recurso mediante el uso de una dirección **URI** (Uniform Resource Identifier).

## 2. Funcionamiento del navegador

8

### ➤ Estructura de una URI:

- ❑ Esquema: identifica el protocolo de acceso al recurso, por ejemplo http:, mailto:, ftp:, etc.
- ❑ Dominio: elemento jerárquico que identifica la autoridad de nombres (por ejemplo: //www.example.com).
- ❑ Ruta: Información usualmente organizada en forma jerárquica, que identifica al recurso en el ámbito del esquema URI (por ejemplo: /ruta/para/recursos.html).



## 2. Funcionamiento del navegador

- Estructura de una URI (cont.):
  - ❑ Solicitud: Información con estructura no jerárquica (usualmente pares "clave=valor") que indica variables que se pasan al recurso Web. El comienzo de este componente se indica mediante el carácter '?'. (por ejemplo: .../recursos.html?variable1=valor1&variable2=valor2)
  - ❑ Fragmento: Permite identificar una parte del recurso principal, o vista de una representación del mismo. El comienzo de este componente se indica mediante el carácter '#'.

## 2. Funcionamiento del navegador

10

- El esquema y el dominio son insensibles a mayúsculas, y por tanto, se generalizan a minúsculas.
- La ruta sí que es sensible, al igual que la solicitud y el fragmento.

# 2. Funcionamiento del navegador

11

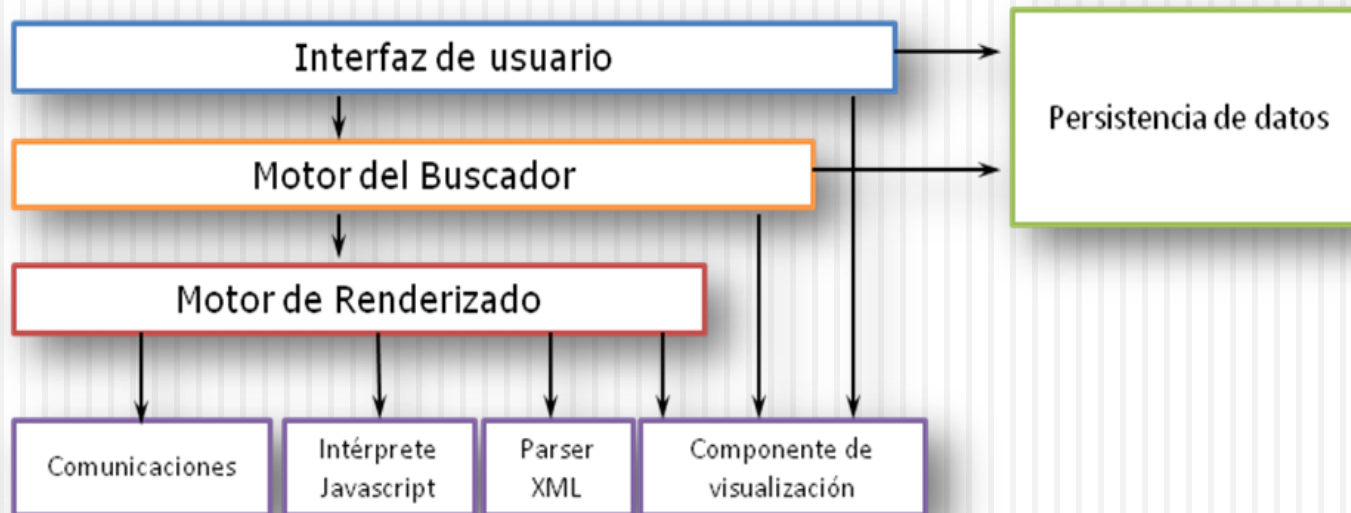
## ➤ **Proceso de petición de un recurso al servidor:**

1. Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo del cliente Web.
2. El cliente descodifica la URL, separando sus diferentes partes: protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional y el objeto requerido.
3. Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente.
4. Se realiza la petición: se envía el comando necesario (GET, POST, HEAD,...), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (HTTP/1.0) y un conjunto variable de información (datos sobre las capacidades del navegador, etc.)
5. El servidor devuelve la respuesta al cliente: consiste en un código de estado y e información de retorno, seguido de la propia información.
6. Se cierra la conexión TCP.

# 3. Arquitectura de ejecución

12

## ➤ Arquitectura de referencia de un navegador web:



# 3. Arquitectura de ejecución

13

- ❑ *Subsistema de interfaz de usuario:* Es la capa que actúa de interfaz entre el usuario y el motor del buscador (o de navegación).
- ❑ *Subsistema del motor del buscador o motor de navegación:* Este subsistema es un componente que ofrece una interfaz de alto nivel para el motor de renderizado.
- ❑ ***Subsistema de renderizado:*** Este componente es el encargado de producir una representación visual del recurso obtenido a partir del acceso a una dirección Web (HTML, XML, CSS, audios, etc.).
- ❑ *Subsistema de comunicaciones:* Es el subsistema encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en Internet (HTTP, FTP, etc.).

# 3. Arquitectura de ejecución

14

- ❑ *Intérprete de JavaScript*: Será el encargado de analizar y ejecutar código JavaScript.
- ❑ *Parser XML*: Módulo que permite cargar en memoria una representación en árbol de la página web (DOM).
- ❑ *Componente de visualización*: Este subsistema ofrece funcionalidades relacionadas con la visualización de los contenidos de un documento HTML en una página web.
- ❑ *Subsistema de persistencia de datos*: Funciona como almacén de diferentes tipos de datos para los principales subsistemas del navegador.

# 4. Lenguajes y tecnologías de programación en entorno cliente

15

- Los lenguajes de programación del entorno de cliente son aquellos que se ejecutan en el navegador Web.
  - ❑ *Lenguajes principales:*
    - ❑ *HTML.*
    - ❑ *XML.*
    - ❑ *XHTML.*
    - ❑ *DHTML.*
  - ❑ *Lenguajes de scripting:*
    - ❑ *JavaScript.*
    - ❑ *VBScript.*
  - ❑ *Otros lenguajes:*
    - ❑ *Ajax.*
    - ❑ *ActionScript.*

# 4. Lenguajes y tecnologías de programación en entorno cliente

16

## ➤ HTML y derivados :

### 1. *HTML*:

- ❑ Es el lenguaje de marcas de texto más utilizado en la World Wide Web.
- ❑ Se basa en la utilización de un sistema de etiquetas cerrado aplicado a un documento de texto.
- ❑ No necesita ser compilado, sino que es interpretado (ejecutado a medida que se avanza por el documento HTML).



## 4. Lenguajes y tecnologías de programación en entorno cliente

17

2. *XML*: lenguaje de etiquetado extensible cuyo objetivo principal es describir datos para su transferencia eficiente y no mostrarlos, como es el caso de HTML.
3. *XHTML*: adaptación de HTML al lenguaje XML.
4. *HTML Dinámico (DHTML)*: No es un formato como tal. Se refiere a integración de HTML con lenguajes de scripting (JavaScript), hojas de estilo personalizadas (CSS) y la identificación de los contenidos de una página Web en formato de árbol (DOM).

# 4. Lenguajes y tecnologías de programación en entorno cliente

18

- *CSS (Cascade Style Sheets)*: sirve para separar el formato que se quiere dar a la página Web de la estructura de la página Web y las demás instrucciones.
- *JavaScript*: lenguaje de programación de scripting embebido en un documento HTML.
- *Applets de Java*: pequeños componentes (objetos independientes) integrados en una página Web y programados en Java.
- *AJAX (Asynchronous JavaScript And XML)*: conjunto de técnicas y métodos de desarrollo Web para la creación aplicaciones Web interactivas y asíncronas.
- *Adobe Flash*: tecnología de animación actualmente bajo licencia de Adobe y que utiliza ActionScript como lenguaje principal (→ HTML5).

# 5. Integración del código JavaScript con las etiquetas HTML

19

- **JavaScript en el mismo documento HTML:**
  - Uso de unas etiquetas predefinidas para marcar el texto (`<script>` y `</script>`).
  - Puede incluirse en cualquier parte del documento, aunque se recomienda que se defina dentro de la cabecera del documento HTML.
    - Esta técnica suele utilizarse cuando se definen instrucciones que se referenciarán desde cualquier parte del documento o cuando se definen funciones con fragmentos de código genéricos.

# 5. Integración del código JavaScript con las etiquetas HTML

20

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=utf-8" />
    <title>Ejemplo 1</title>
    <script type="text/javascript">★
      alert("Prueba de JavaScript");
    </script>
  </head>
  <body>
    <h1>Ejemplo 1: código embebido</h1>
  </body>
</html>
```

# 5. Integración del código JavaScript con las etiquetas HTML

21

## ➤ **JavaScript en un archivo externo:**

- Las mismas instrucciones de JavaScript que se incluyen entre un bloque `<script></script>` pueden almacenarse en un fichero externo con extensión `.js`.
- La forma de acceder y enlazar esos ficheros `*.js` con el documento HTML es a través de la propia etiqueta `<script>`.
- No existe un límite en el número de ficheros `.js` que pueden enlazarse en un mismo documento HTML/XHTML.

# 5. Integración del código JavaScript con las etiquetas HTML

22

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=utf-8" />
    <title>Ejemplo 2</title>
    <script type="text/javascript"
      src="js/mensaje.js"></script>
  </head>
  <body>
    <h1>Ejemplo 2: fichero externo</h1>
  </body>
</html>
```

```
alert("Prueba de JavaScript");
```



# 5. Integración del código JavaScript con las etiquetas HTML

23

## ➤ JavaScript en elementos HTML:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=utf-8" />
    <title>Ejemplo 3</title>
  </head>
  <body>
    ★ <p onclick="alert('Prueba de JavaScript');">
      Ejemplo 3: código en atributos
    </p>
  </body>
</html>
```

# 5. Integración del código JavaScript con las etiquetas HTML

24



A.1.1. Crear un fichero HTML vacío y añadir el siguiente código JavaScript en el cuerpo de la página:

```
<script type="text/javascript">  
    document.bgColor = "red";  
    alert('Segundo Mensaje: cambiamos el color de la página');  
</script>
```