



---

# Tema 2

**Lenguajes y herramientas  
de programación en clientes  
web**

---

# Lenguajes y herramientas de programación en clientes web

Lo que vamos a aprender:

- Identificar las capacidades y entender los mecanismos de ejecución de los navegadores web
- Conocer los principales lenguajes de programación en entorno cliente
- Repasar las características de los lenguajes de marcas

# Navegadores

- El punto de entrada del sistema será siempre la interfaz gráfica, es decir, lo que mostremos en el navegador.
- Para poder construir un front-end que se adapte a las necesidades de los usuarios debemos conocer cómo funciona el navegador, qué tareas realiza y cómo las realiza.
- El navegador se encarga de realizar la visualización del código recibido desde el servidor tras la petición HTTP.
- En ocasiones sucede que diferentes navegadores realizan interpretaciones distintas del código recibido.
- Hay que realizar adaptaciones para diferentes navegadores (\$\$)

# Motor de ejecución de los navegadores

- 5 Navegadores principales: Chrome, IE/Edge, Firefox, Safari y Opera. En dispositivos móviles se basan en el framework Webkit
- Componentes principales del navegador
  - Interfaz gráfica
  - Motor de navegación
  - **Motor de visualización**
  - Intérprete de JavaScript
  - Manejador de red
  - Gestor de almacenamiento

# Motor de visualización del navegador

- Este componente se encarga de realizar las transformaciones necesarias para poder visualizar el código HTML recibido desde el servidor y convertirlo en una página web. Además, debe interpretar las reglas de estilo (CSS) para generar una representación válida.
- Dependiendo de la interpretación realizada, el resultado visual puede cambiar entre los diferentes navegadores.
- Por suerte en entornos web existe un gran conjunto de estándares que nos sirven para garantizar un comportamiento homogéneo.

# Motor de visualización del navegador

Cada uno de los principales navegadores dispone de un motor de visualización:

- Chrome. Inicialmente se basaba en WebKit. Sin embargo, actualmente utiliza un motor propio llamado Blink.
- Firefox. Se basa en el motor Gecko.
- Safari. Su motor se basa en WebKit.
- Edge. Posee su propio motor de visualización llamado EdgeHTML.
- Opera: Poseía un motor propio Presto hasta el 2013 y a partir de ahí Blink

# Desarrollo web en entorno cliente (front-end)

- Se encarga principalmente de gestionar la interacción con el usuario, pues se convierte en la interfaz gráfica que, en muchas ocasiones, funciona como la única puerta de entrada a una aplicación web.
- La tecnología front-end es aquella que se ejecuta siempre en el dispositivo del usuario, es decir, en su navegador web.
- El desarrollador de front-end debe ser capaz de realizar las peticiones oportunas al servidor (back-end)
- Es necesario crear diferentes formas de visualización que contemplan diversos tamaños de pantalla (móviles).

# Entornos de programación

En las tareas de desarrollo, normalmente se utilizan IDEs (Integrated Development Environment) para poder facilitar todas las tareas involucradas. Las herramientas que se utilizan en front-end no requieren de grandes recursos hardware. Realmente con un editor en texto plano bastaría. Editores más utilizados:

- Sublime Text: Reconocimiento de múltiples lenguajes (no solo de cliente). Está disponible para los principales sistemas operativos de propósito general.
- Notepad++. Es un editor de código abierto que reconoce la sintaxis de múltiples lenguajes (actualmente 27).
- Vim. Se trata de un editor de texto de propósito general disponible para sistemas UNIX.
- Visual Studio Code. Se trata de un editor de código abierto desarrollado por Microsoft. Se encuentra disponible para los sistemas operativos de propósito general.



---

# Instalemos Visual Studio Code

Instamos Visual Studio Code y las siguientes extensiones de utilidad para el desarrollo

- Live Server
- Quokka
- Quokka Statusbar Buttons
- ESLint

# Lenguajes más utilizados en clientes web

- **HTML** (Hyper Text Markup Language). Se trata de un lenguaje de marcado estándar para la creación de páginas web. Con este lenguaje se puede definir la estructura de una página web, entendiendo por estructura la distribución del contenido.
- **CSS** (Cascade Style Sheet). Se trata de hojas de estilo en cascada que permiten cambiar la representación visual de la estructura HTML creada previamente.
- **JavaScript**. Es un lenguaje sencillo y ligero que el navegador interpreta para permitir la interacción y el control de componentes web. Se puede utilizar en contextos diferentes a entornos web. Con la evolución de este lenguaje surge TypeScript, un lenguaje de alto nivel basado en JavaScript.

# Para el intercambio de datos

- **JSON** (JavaScript Object Notation). Se trata de un formato sencillo para facilitar el intercambio de datos que puede interpretarse directamente a través de JavaScript. Los datos en JSON se organizan en pares de clave-valor. A diferencia de XML no tiene etiquetas de apertura y cierre.

```
{
  "nombre": "Juan",
  "edad": 30,
  "casado": false,
  "hijos": ["Ana", "Luis"],
  "direccion": {
    "calle": "Calle Falsa 123",
    "ciudad": "Ciudad Ejemplo"
  }
}
```

# Para el intercambio de datos

- **XML** (Extensible Markup Language). Se trata de un lenguaje de marcado (basado en SGML) para el intercambio de datos. Permite representar datos de manera estructurada para permitir su intercambio en la red. Existe un gran conjunto de tecnologías que rodea a esta representación, por ejemplo: XPath, XQuery, XSLT, etc.

---

# Frameworks más utilizados en desarrollo cliente

- Un framework es un conjunto de herramientas, bibliotecas y convenciones de desarrollo que proporciona una estructura predefinida para construir aplicaciones de software.
- Su propósito es facilitar el desarrollo, promoviendo la reutilización de código y asegurando consistencia y calidad en los proyectos
- Proporciona bibliotecas y componentes reutilizables que resuelven problemas comunes (como manejo de bases de datos, autenticación, o gestión de vistas), reduciendo la necesidad de escribir código desde cero.
- Los desarrolladores pueden concentrarse más en la lógica del negocio, en lugar de en detalles técnicos de bajo nivel.

## Frameworks más utilizados en desarrollo cliente

- **React**: Sigue un enfoque de desarrollo reactivo y basado en JavaScript. React fue desarrollado por Facebook para crear interfaces de una manera más sencilla e intuitiva. Permite a los programadores realizar aplicaciones web de forma rápida y eficiente renderizando (dibujando) los componentes del front-end de una manera sencilla y eficaz
- **Angular**: Creado y mantenido por Google. Ofrece diferentes mecanismos para crear, no solo interfaces gráficas, sino aplicaciones completas para entornos de escritorio y móvil.
- **Vue.js**: Permite crear interfaces de usuario de una manera sencilla y progresiva. Ligereza y velocidad de ejecución

# Integración de código mediante etiquetas

- Para poder crear un front-end correctamente es necesario trabajar, en primer lugar, con **HTML**. A través de este lenguaje basado en etiquetas podremos crear la distribución del contenido en la web. El lenguaje está formado por un conjunto de etiquetas que siempre deben abrirse y cerrarse.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

---

# Lenguajes de script en entornos web

El aumento de la interacción de usuario ha sido posible gracias al motor de JavaScript incluido en los navegadores. Entre las tareas que nos permite realizar JavaScript, encontramos:

- Aumento de la interacción del usuario
- Realizar comprobaciones antes de enviar información al servidor.
- Funcionamiento asíncrono mientras el usuario realiza otras tareas.



# Ejemplo Javascript – Hola Mundo

```
<> index.html > html > body > script > miFuncion
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-sc
6      <title>Hola Mundo</title>
7  </head>
8  <body>
9      <h2>Boton de alerta Javascript</h2>
10     <button onclick="miFuncion()">Pulsa</button>
11     <script>
12         function miFuncion() {
13             alert ("Hola Mundo")
14         }
15     </script>
16 </body>
17 </html>
```

# Ejemplo Javascript - Consola

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Log en Consola</title>
5    </head>
6
7    <body>
8      <h2>JavaScript Alert</h2>
9      <script>
10       console.log ('Mi primer log')
11     </script>
12
13   </body>
14
15 </html>
```

