

Tema 1

Selección de arquitecturas y herramientas
de programación

Tema 1

Selección de Arquitecturas y de herramientas de programación en entornos cliente

Tema 1

1. Modelos de ejecución de código
2. Lenguajes de programación en entorno cliente
3. Arquitectura de ejecución
4. Características de los lenguajes script
5. Programación de Guiones
6. Integración del código con las etiquetas HTML
7. Herramientas de programación sobre clientes Web

Introducción

- En los orígenes el lenguaje de programación que necesitábamos simplemente en **lenguaje HTML** el cual mediante sus marcas , se creaban sencillos documentos HTML que cualquier navegador podría visualizar
 - Las paginas se encontraban alojadas en servidores conectados 24 horas 365 días al año
 - A través de peticiones de clientes, se transmitía dicha pagina a los equipos que la pedían
- Esto se conoce como **hipertexto** palabras o conjunto de palabras que nos permiten llegar a la información relacionada.
- Con el auge de internet se crean un consorcio para la estandarización en la utilización del lenguaje HTML así como diferentes especificaciones
 - **W3C** es la organización que se responsabiliza de desarrollar y mantener los estándares web

1.- Modelos de ejecución de código

El desarrollo web está dividido en 5 áreas:

- **Contenido**

- Incluye la forma y organización del contenido del sitio.
- Esto abarca desde cómo se escribe un texto hasta como se organiza, presenta y estructura utilizando tecnologías de marca como HTML.

- **Visual**

- Hace referencia a la plantilla empleada en el sitio web.
- Para las plantillas usamos generalmente HTML y CSS combinado con flash
- Se pueden incluir elementos gráficos para la decoración o navegación
- El aspecto visual de la web es la característica más atractiva y por tanto la más importante

1.- Modelos de ejecución de código

- **Tecnología**

- La decisión del lenguaje que utilicemos en la implementación de la web supone una de las características más influyentes
- De ello dependerán las demás características como las bases de datos, el uso de páginas dinámicas etc.

- **Distribución**

- El hardware/software utilizado en la implementación, así como la arquitectura de red utilizada en la conexión de la aplicación web determinarán la velocidad y fiabilidad de la misma durante la distribución en internet o bien en la red interna corporativa.

- **Propósito**

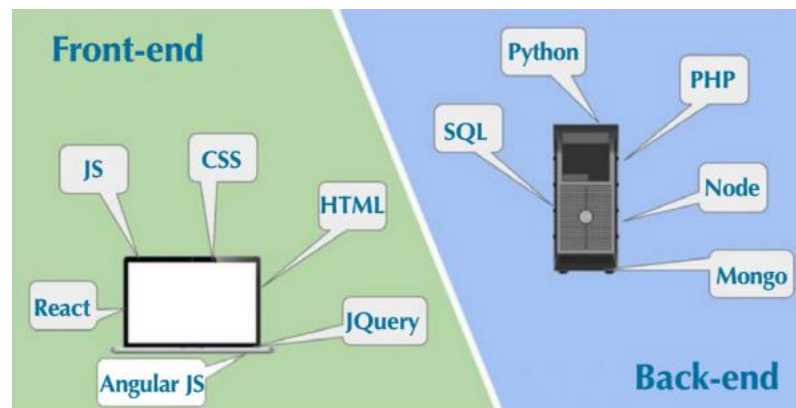
- La principal razón de existencia de un sitio web es su objetivo fundamentalmente económico tratando de conseguir una mayor difusión del producto, de la imagen de la empresa, facilitando la difusión

1.- Modelos de ejecución de código

La arquitectura que siguen los sitios web es el modelo **cliente-servidor**

- Elementos fundamentales:

- Servidor
- Máquina cliente
- Conectividad de red.



1.- Modelos de ejecución de código

Servidor:

- La parte de la aplicación donde tenemos instalados los elementos hardware y software.
- Utilizaremos lenguajes de programación multihilo y multiproceso, que permiten la ejecución de fragmentos de código de una aplicación de forma simultánea.

Cliente:

- Es la parte correspondiente a los navegadores web que soportan tecnologías HTML, CSS y lenguajes script como JavaScript y controles ActiveX.
- Permiten crear presentaciones de páginas y proporcionan características interactivas.

1.- Modelos de ejecución de código

Red

- Son los elementos de conectividad, es decir la capacidad que tienen los dispositivos de conectarse unos con otros.
- Está compuesto por protocolos de comunicación y materiales de conexión

2.- Lenguajes de programación en entornos cliente

Los lenguajes web del lado del cliente se basan en 3 pilares

1. Lenguaje HTML:
2. Lenguaje CSS:
3. Lenguaje JavaScript:



2.- Lenguajes de programación en entornos cliente

Lenguaje HTML

- No es un lenguaje de programación, es un lenguaje de marcado
- HTML define que se va a ver en el documento
- El navegador lee e interpreta las etiquetas y lo visualiza al cliente
- Ventaja: cualquier navegador visualiza el contenido HTML de la misma forma y con el mismo aspecto.



2.- Lenguajes de programación en entornos cliente

Lenguaje CSS

- Define la presentación del documento
- Es un lenguaje de diseño gráfico
- Su objetivo es que la página web sea atractiva al usuario
- No modifica el comportamiento ni el contenido, solamente afecta al aspecto de la página web



2.- Lenguajes de programación en entornos cliente

Lenguaje JavaScript

- Agrega el contenido dinámico a las páginas web.
- Es un lenguaje de programación, a diferencia de HTML y CSS.

TABLA DE LAS CUATRO CAPAS DEL DESARROLLO WEB EN EL LADO DEL CLIENTE

Comportamiento (JavaScript)	
Presentación (CSS)	
Estructura (DOM/estructura HTML)	Contenido Estructurado (documento HTML)
Contenido (texto, imágenes, vídeos, etc....)	

2.- Lenguajes de programación en entornos cliente

En la actualidad, las empresas no trabajan directamente con JavaScript, sino que utilizan frameworks basados en JavaScript

- React

- Es un framework creado por Facebook que permite realizar operaciones de dibujo en aplicaciones web de forma rápida y eficiente.
- Permite crear componentes front-end de maneja sencilla y eficaz.
- Se basa en la programación orientada a componentes (que no objetos)
 - Los componentes gestionan sus propios estados y cuando se agrupa una serie de componentes, los programadores son capaces de ir creando las interfaces de usuario.
- La característica más importante de React es que utiliza un DOM virtual capaz de mapear los objetos desde este hasta el DOM del navegador

2.- Lenguajes de programación en entornos cliente

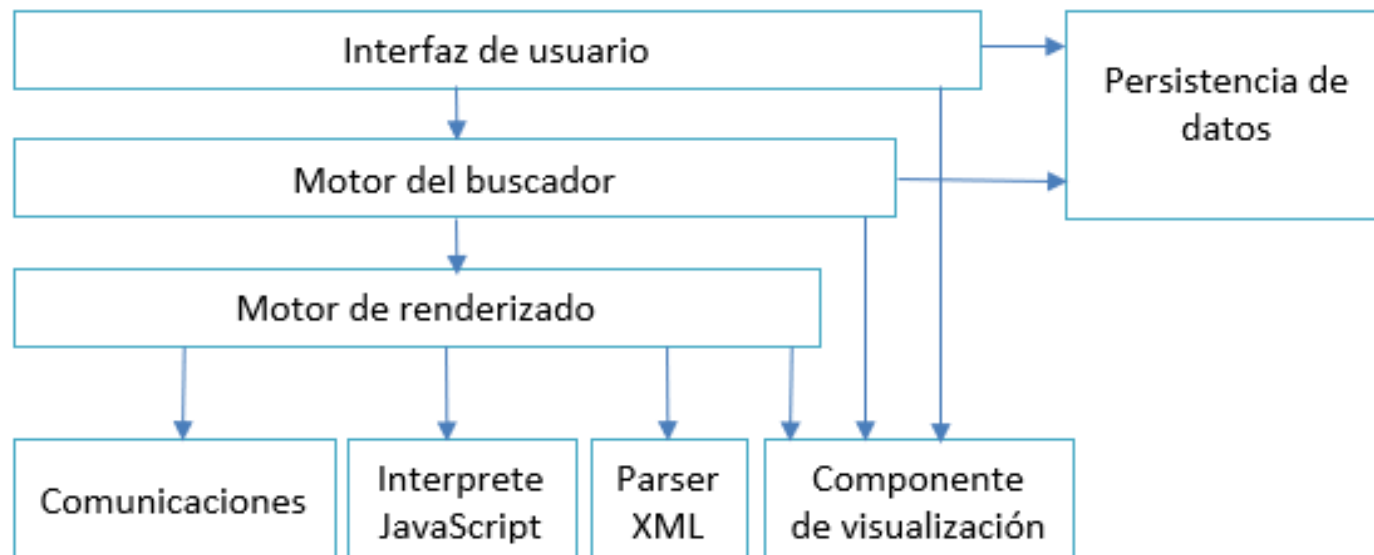
- Angular
 - Lenguaje creado por Google.
 - La primera versión fue denominada AngularJS y ofrece una fácil integración con JavaScript.
 - En versiones sucesivas, se denominan Angular a secas y han dejado de ser una simple librería para pasar a ser una plataforma de desarrollo.
 - Su principal inconveniente es la pronunciada curva de aprendizaje,
 - Angular está basado en TypeScript, que es un conjunto de JavaScript desarrollado por Microsoft y que utiliza el patrón reactivo RxJS.
- Vue.js
 - Su principal característica es la ligereza y velocidad de ejecución.
 - Su principal ventaja es que su curva de aprendizaje no es tan pronunciada como Angular o React

3.- Arquitectura de ejecución

- Cada navegador web tiene su propia forma de interpretar la interacción con el usuario.
 - El resultado de esta interacción, en cualquier caso, se inicia en el usuario indicando una dirección del recurso
 - Finaliza con la visualización por parte del navegador en la pantalla
- La forma de realizar este proceso depende del propósito del navegador y de la configuración del mismo.
 - El navegador puede estar más centrado en ofrecer una respuesta más rápida, en mostrar una respuesta más fiel, priorizar aspectos de la seguridad de las comunicaciones con el servidor etc.
- Para llevar a cabo este proceso cada navegador está formado por una serie de elementos y componentes determinados que conforman lo que se denomina **arquitectura del navegador**.
 - Cada navegador tiene su propia arquitectura, aunque coinciden en componentes básicos comunes entre ellos.

3.- Arquitectura de ejecución

Componentes básicos en arquitectura cliente:



3.- Arquitectura de ejecución

- **Interfaz de usuario**

- Es la capa que actúa entre el usuario y el motor del buscador.
- Sus funcionalidades son:
 - La visualización de barras de herramientas
 - Progreso de carga de página
 - Gestión inteligente de las descargas
 - Preferencias de configuración de usuario e impresión.
- En algunos casos mantiene comunicación con el SO para el manejo de sesiones y preferencias de visualización o configuración

- **Motor del buscador**

- Este subsistema es un componente que ofrece una interfaz de alto nivel para el motor de renderizado (pintado)
- Su principal función es cargar una dirección determinada(URL o URI) y soportar mecanismos básicos como volver a la página , siguiente, recarga
- Es el componente que carga las alertas de JavaScript

3.- Arquitectura de ejecución

- **Motor de Renderizado**

- Es el encargado de producir una representación visual del recurso obtenido a partir del acceso a una dirección web
- Este interpretará el código de la página web
- En función de los lenguajes utilizados, este módulo será capaz de mostrar documentos HTML, XML, CSS e incluso contenido embebido en la página (video/audio) e imágenes

- **Subsistema de comunicaciones**

- Encargado de implementar los protocolos de transferencia de ficheros y documentos utilizados en internet.

- **Interprete de JavaScript**

- Será el encargado de analizar y ejecutar dicho código como pueden ser eventos del ratón o teclado u otro intercalado para ciertas funciones

3.- Arquitectura de ejecución

- **Parse XML**

- Facilita el acceso de los contenidos definidos en un documento HTML
- El módulo **Parse** permite cargar en memoria una representación en árbol (**DOM**) de la página
 - Así se facilita y agiliza el acceso a los elementos de la página

- **Componente de visualización**

- Ofrece funcionalidades relacionadas con la visualización de contenidos de un documento HTML

- **Subsistema de persistencia de datos**

- Almacena diferentes tipos de datos para los principales subsistemas del navegador,:
 - historiales de navegación
 - mantenimiento de sesiones de usuario
 - preferencias de configuración del navegador
 - Almacenamiento de certificados de seguridad y cookies

4.- Características de los lenguajes Script

- Los lenguajes de programación para clientes web no son un reemplazo de la programación del lado del cliente
- Cualquier web que reaccione dinámicamente a interacciones del usuario o que almacene datos, está gestionada por lenguajes script del lado del cliente
- Usaremos JavaScript en el cliente para mejorar la experiencia de usuario
- JavaScript por si mismo no puede escribir ficheros en el servidor
 - Puede ayudar al usuario a elegir opciones incluso a preparar los datos, pero solamente cede datos al lenguaje del servidor encargado de su actualización.
- No todos los clientes web ejecutan JavaScript
 - Algunos lectores, dispositivos móviles buscadores etc en ciertos contextos no pueden realizar llamadas javascript

4.- Características de los lenguajes Script

- Importante: la web debe ser completamente funcional con JavaScript desactivado:
 - Utilizaremos JavaScript para tratar de conseguir una experiencia de navegación web más rápida, segura y fiable.
- AJAX:
 - Es la integración de la programación cliente con la programación servidor.
 - El proceso “asíncrono” de AJAX se ejecuta en el navegador del cliente y emplea JavaScript
 - Este proceso se encarga de solicitar datos XML o enviar datos al lenguaje del servidor y todo ello de forma transparente en background
 - Los datos devueltos por el servidor pueden ser examinados por JavaScript en el lado del cliente, para actualizar secciones o partes de la página web

4.- Características de los lenguajes Script

• Objetivos del uso de JavaScript

- Conseguir que nuestra página web responda o reaccione directamente a la interacción del usuario con elementos de formulario y enlaces hipertexto.
- La distribución de pequeños grupos de datos y proporcionar una interfaz amigable para esos datos.
- Controlar múltiples ventanas o marcos de navegación, plugins, o applets Java basados en las elecciones que ha hecho el usuario en el documento HTML.
- Pre-procesar datos en el cliente antes de enviarlos al servidor.
- Modificar estilos y contenido en los navegadores de forma dinámica e instantáneamente, en respuesta a interacciones del usuario.
- Solicitar ficheros del servidor, y enviar solicitudes de lectura y escritura a los lenguajes de servidor.

4.- Características de los lenguajes Script

- **Diferencias entre los lenguajes de script y los lenguajes de programación**

- Los lenguajes de script son interpretados, mientras que, muchas veces, los lenguajes de programación se compilan.
- Los lenguajes de script utilizan componentes ya preexistentes, mientras que los lenguajes de programación, en ocasiones, empiezan a desarrollarse desde cero.
- A veces, los scripts se incrustan dentro de otros programas (como el JavaScript se integra en HTML).
- Los programas de lenguajes de programación se pueden ejecutar de forma independiente, mientras que un lenguaje de script se ejecuta muchas veces con otro programa.
- Los scripts se ejecutan línea a línea, con lo cual pueden producirse muchos errores en ejecución y por tanto no necesitan ser compilados
- Los lenguajes de script no generan un fichero ejecutable.
- Los lenguajes de script han sido diseñados para ser fáciles de utilizar y programar.

5.- Programación de guiones

- Los guiones se pueden colocar en uno de dos sitios posibles de la página HTML:
 - Entre las etiquetas <head> y </head> (parte denominada **encabezado del guion**),
 - Entre las etiquetas <body> y </body> (o **cuerpo del guion**).
- La etiqueta contenedora HTML, que designa los guiones, empieza por <script> y termina por </script>.

CÓDIGO:

```
<script type="text/javascript">  
    document.write("¡Hola, mundo!");  
</script>
```


5.- Programación de guiones

- `<script type="text/javascript">`
 - Script es la etiqueta de apertura de guion.
 - Esta le comunica al navegador que espere JavaScript en lugar de HTML
 - El atributo `type="text/javascript"` dice que el guion está en formato de sólo texto, organizado como JavaScript.
- `document.write("¡Hola, mundo!");`
 - Escribimos "¡Hola, mundo!" en la web que estamos ejecutando.
 - El punto y coma al final de la línea (;), el cual, indica al intérprete de JavaScript del navegador que la línea ha terminado.
- `</script>`
 - La etiqueta de cierre de script indica al navegador que quede a la espera de encontrar nuevamente código HTML.

6.- Integración del código con las etiquetas HTML

- Los navegadores aceptan la integración del código JavaScript en documentos HTML de dos formas:
 - Integrar el código JavaScript dentro de archivos HTML
 - Tener separado el código JavaScript en archivos externos con extensión .js



6.- Integración del código con las etiquetas HTML

- Para integrar código JavaScript en HTML empezaremos con las marcas `<script>` `</script>` y el atributo `type` para indicar el tipo de lenguaje que estamos utilizando:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Título de la WEB</title>
  <script>
    function saludo(){
      alert("Hola Mundo");
    }
    saludo();
  </script>
</head>
<body>
</body>
</html>
```

6.- Integración del código con las etiquetas HTML

- La manera más recomendable código JavaScript en un documento HTML es de forma externa en un fichero,
 - De esta forma podemos considerar de manera independiente los diferentes códigos implementados en distintos lenguajes correspondientes a distintas partes de la página web (estructura, formato y tratamiento).
 - Tendremos la opción de compartir el mismo fichero para varios archivos HTML, facilitar la depuración de los errores, así como la claridad en el entendimiento del código de nuestro portal web.
 - Entre sus ventajas son la modularidad, seguridad y rapidez a la hora de cargar la web.

`<script type="text/javascript" src="tucodigo.js"></script>`

- Para cargar más de un fichero tendremos que repetir la etiqueta `<script>` `</script>`, pero cambiando el nombre del fichero fuente.

6.- Integración del código con las etiquetas HTML

- Podemos utilizar:
 - **ruta relativa**
 - empezaremos a direccionar la ruta desde el directorio donde nos encontramos
 - **ruta absoluta**
 - empezaremos a direccionar la ruta desde el directorio raíz o directorio de la unidad en cuestión
 - **ruta remota**
 - donde se comenzará con el prototipo a utilizar http

CÓDIGO:

Ejemplos:

```
<script type="text/javascript" src="http://www.tudominio.com/ejemplo.js"></script>
```

```
<script type="text/javascript" src="../js/ejemplo.js"></script>
```

(el fichero ejemplo.js se encuentra en el directorio anterior (../) al actual, dentro de la carpeta js/)

6.- Integración del código con las etiquetas HTML

- Fichero HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Título de la WEB</title>
  <script src="script1.js"></script>
</head>
<body>
</body>
</html>
```

- Fichero script1.js

```
function saludo(){
  alert("Hola Mundo");
}
saludo();
```

6.- Integración del código con las etiquetas HTML

- En aquellos casos que el navegador no soporte los scripts que hemos implementado en JavaScript, debemos de informar al usuario de lo sucedido.
- La marca que utilizar para esta situación es
 - `<noscript>Texto informativo </noscript>`.

7.- Herramientas de programación sobre clientes web

- Un simple editor de texto puede ser una herramienta para desarrollar el código, tanto HTML como JavaScript.
- Actualmente, se dispone en el mercado de una gran variedad de editores para codificar en JavaScript y HTML, aunque la elección del editor depende del desarrollador.
- Uno de los factores importantes que tienes que tener en cuenta a la hora de elegir un editor, es ver la facilidad con la que se pueden grabar los ficheros con extensión .html.
 - Independientemente del sistema operativo que estés utilizando cualquier programa que te permita grabar ficheros directamente con la extensión .html te evitaría un gran número de problemas.

7.- Herramientas de programación sobre clientes web

- También hay que tener en cuenta la codificación que emplea ese programa para grabar los ficheros.
 - En Microsoft Word cuando intentamos guardar archivos éste intenta almacenarlos en un formato binario de Word, algo que los navegadores web no pueden cargar.
 - Para grabar ese archivo con extensión .html requiere moverse un poco más por el menú de diálogo "Guardar como", lo cual es una desventaja.

7.- Herramientas de programación sobre clientes web

- La idea es decantarse por editores que posean características que te faciliten la programación web, como, por ejemplo:
 - Sintaxis con codificación de colores. Que resalte automáticamente en diferente color o tipos de letra los elementos del lenguaje tales como objetos, comentarios, funciones, variables, etc.
 - Verificación de sintaxis. Que te marque los errores en la sintaxis del código que estás escribiendo.
 - Que te permita diferenciar los comentarios del resto del código.
 - Que genere automáticamente partes del código tales como bloques, estructuras, etc.
 - Que disponga de utilidades adicionales, tales como cliente FTP para enviar tus ficheros automáticamente al servidor, etc.

7.- Herramientas de programación sobre clientes web

- Dependiendo del sistema operativo que utilices en tu ordenador dispones de múltiples opciones de editores. Cada una es perfectamente válida para poder programar en JavaScript. Algunos ejemplos de editores web gratuitos son:
 - Para **Windows** tienes: Notepad++, Visual Studio Code, Atom, Eclipse, NetBeans, etc.
 - Para **Macintosh** tienes: Aptana Studio, Bluefish, Eclipse, KompoZer, Nvu, etc.
 - Para **Linux** tienes: KompoZer, Amaya, Quanta Plus, Bluefish, codetech, etc.

7.- Herramientas de programación sobre clientes web

- Para programar código en JavaScript no es necesario la conexión a internet, solo necesitamos un navegador que soporte este plugin.
 - Por tanto, se puede desarrollar el código desde cualquier equipo y en cualquier lugar.
 - Algunos ejemplos de los navegadores más habituales son: Mozilla Firefox, Google Chrome, Safari, Opera, etc.
- Una recomendación muy interesante es el disponer de 2 o 3 tipos de navegadores diferentes, ya que así se puede comprobar la compatibilidad de tu página web y ver si tu código fuente de JavaScript se ejecuta correctamente en todos ellos.

7.- Herramientas de programación sobre clientes web

- Es recomendable poder ejecutar tu editor web y tu navegador de forma simultánea, ya que el flujo típico de trabajo en JavaScript va a ser:
 1. Introducir HTML, JavaScript y CSS en el documento original en el editor web.
 2. Guardarlo en disco.
 3. Cambiarte al navegador web.
 4. Realizar una de las siguientes tareas:
 - Si es un nuevo documento, abrirlo a través de la opción Abrir del menú Archivo > Abrir Archivo.
 - Si el documento ya está cargado en el navegador pues simplemente recargar la página.
- Los pasos 2 al 4 son acciones que se van a ejecutar muy frecuentemente
 - Esa secuencia grabar cambiar-recargar la realizarás tantas veces cuando estés escribiendo y depurando tu script, que llegará a ser prácticamente un acto reflejo.

7.- Herramientas de programación sobre clientes web

- Otro aspecto muy importante es la validación.
 - Se debe asegurar de que el código HTML es válido.
 - Si el código HTML contiene imperfecciones, hay muchas posibilidades de que el JavaScript o CSS no funcionen de la manera esperada, ya que ambos dependen de los elementos HTML y sus atributos.
 - Cuanto más ajustado este el código HTML a las especificaciones del estándar, mejor resultado obtendrás entre los diferentes tipos de navegadores.
 - La estandarización del consorcio W3C es el que rige las validaciones y, por tanto, en su dirección web es donde podemos acceder para esta operación.
 - La dirección del Validador W3C es: <http://validator.w3.org/>