



Universitat
Oberta
de Catalunya

**M4.254 - Desarrollo front-end con frameworks
JavaScript**

PEC 1. Introducción al desarrollo frontend

ÍNDICE:

1. La aparición de HTML5/CSS3/JS.
2. El lenguaje CSS.
3. El flujo de trabajo profesional en front-end.

1. La aparición de HTML5/CSS3/JS ha supuesto el nacimiento del desarrollo front-end moderno.

- **¿Cuál es la ventaja del uso de etiquetas semánticas? Nombra y explica al menos 3 de ellas.**

Las etiquetas semánticas son aquellas que dan un significado a las partes del documento. Son las que indican qué es el contenido que contienen, en lugar de cómo se debe formatear al mostrar el documento HTML en un cliente web.

En HTML5 se han incorporado las etiquetas semánticas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro.

La idea con estas etiquetas es añadir información a la web, como metadatos semánticos que puedan ser interpretados por cualquier sistema informático. Con ello se facilita que cualquier máquina que lea un documento HTML pueda saber a qué parte típica pertenece cada pedazo de código, para entender mejor qué es lo que se está tratando en cada caso.

En lo que respecta a los motores de búsqueda, serán capaz de leer el documento y saber qué parte es la cabecera, que parte contiene los enlaces a las principales secciones del sitio, qué parte contendrá las unidades de contenido, etc.

- `<header>`: se usa para marcar un grupo de elementos de introducción o de navegación dentro de una sección o documento.
- `<nav>`: se usa para marcar una sección del documento cuya función es la navegación por la página web.
- `<article>`: se usa para marcar contenido independiente que tendría sentido fuera del contexto de la página actual y que podría sindicarse, por ejemplo: una noticia, un artículo en un blog o un comentario.
- `<section>`: se usa para marcar una sección genérica de un documento o aplicación. Una sección, en este contexto, es una agrupación temática del contenido, típicamente con un encabezado.
- `<aside>`: se usa para marcar un trozo de contenido que está relacionado con el contenido de la página web, pero que no es parte del mismo.
- `<footer>`: se usa para marcar el pie de una sección o documento y que contiene información sobre el mismo como el autor, licencia, términos de uso, etc.

- **Cita al menos 3 APIs HTML5 y explica brevemente su funcionalidad.**

- **Canvas:** el elemento HTML <canvas> se puede usar para dibujar gráficos mediante scripting en JavaScript.

Esta API nos proporciona un rectángulo en nuestra página web en el que podemos dibujar cualquier cosa que imaginemos usando JavaScript, proporcionándole animación e interactividad. Para ello, la API nos expone un conjunto de métodos o funciones que podemos utilizar para definir y crear formas, gradientes, trazados o aplicar diversas transformaciones.

- **localStorage y sessionStorage:** son propiedades de HTML5 (web storage), que permiten almacenar datos en nuestro navegador web. De manera muy similar a como lo hacen las cookies.

Permiten almacenar entre 5MB y 10MB de información; incluyendo texto y multimedia.

La información está almacenada en la computadora del cliente y no es enviada en cada petición del servidor, a diferencia de las cookies.

Utilizan un número mínimo de peticiones al servidor para reducir el tráfico de la red.

Previenen pérdidas de información cuando se desconecta de la red.

La información es guardada por dominio web.

- **Web Workers:** proveen un medio sencillo para que el contenido web ejecute scripts en hilos en segundo plano. Una vez creado, un worker puede enviar mensajes a la tarea creada mediante envío de mensajes al manejador de eventos especificado por el creador.

El hilo worker puede realizar tareas sin interferir con la interfaz de usuario. Además, pueden realizar I/O usando XMLHttpRequest.

- **Geolocalización:** permite al usuario compartir su ubicación a las aplicaciones web si así lo desea. Por razones de privacidad, al usuario se le pide que confirme el permiso para proporcionar información de ubicación.

La API de geolocalización se publica a través del objeto navigator.geolocation. Si el objeto existe, los servicios de geolocalización están disponibles.

- **Cita qué opción ofrece CSS3 para conseguir que se apliquen diferentes estilos CSS sobre el mismo elemento en su visualización en diferentes dispositivos (diferentes tamaños de pantalla).**

Las media queries son útiles cuando deseas modificar tu página web o aplicación en función del tipo de dispositivo o de características y parámetros específicos (como la resolución de la pantalla o el ancho del viewport del navegador).

Permiten definir estilos condicionales, aplicables únicamente en determinadas situaciones.

- **Cita al menos 4 de las características principales de TypeScript (importante superset de JavaScript que trataremos en el siguiente capítulo).**

- Compilado: al ser un lenguaje compilado podemos detectar muchos de los errores antes de abordar la ejecución.
- Orientación a objetos: soporte para el manejo de clases, interfaces, herencia, que facilita hacer aplicaciones escalables.
- Versatilidad: uso de decoradores, tipos genéricos, unión e intersección de tipos.
- TypeScript se ajusta al estándar marcado por ECMAScript.
- Datos tipados, cosa que no existe en Javascript.

2. El lenguaje CSS es muy rígido, poco práctico y ordenado a la hora de programar. Para evitar este problema se han creado los preprocesadores CSS, que ofrecen evidentes ventajas.

- **Cita al menos 2 de estos preprocesadores.**

- LESS (que significa Leaner Style Sheets) es una extensión de lenguaje compatible con el CSS. LESS sigue un patrón de estilo declarativo por naturaleza. Esto implica que especificas lo que quieres ver y no cómo quieres que se haga. Esto se debe principalmente a sus similitudes con la programación funcional, lo que lo hace más legible y fácil de entender.
- SASS, que son las siglas de Syntactically Awesome Style Sheets (Hojas de Estilo Sintácticamente Impresionantes), fue creado por Hampton Catlin. Está igualmente diseñado para ser compatible con

todas las versiones de CSS. Sigue un patrón de estilo imperativo que implica que tienes que especificar cómo se hacen las cosas.

- **Cita al menos 4 ventajas que ofrecen estos preprocesadores.**

- Posibilidad de añadir funcionalidades adicionales que no son posible de utilizar en archivos CSS tradicionales, como son el uso de variables y lógica condicional.
- Dispondremos de una hoja de estilos más limpia, consiguiendo a la vez que sea más eficiente y fácil de mantener.
- Ofrece la posibilidad de reutilizar mucho código.
- Los cambios serán mucho más rápidos, ya que únicamente deberemos cambiar el valor de alguna variable.
- Ayuda a crear código compatible con todos los navegadores, lo que supone una gran ayuda para los diseñadores de sitios web.
- Se crea una capa de abstracción, donde no trabajaremos directamente sobre nuestro archivo CSS, ofreciendo una mayor seguridad.
- Posibilidad de separación absoluta del proceso de desarrollo y producción

- **Explica brevemente en qué consisten los sourcemaps.**

Los sourcemaps consisten en un mapeo que se realiza entre el código original y el código transformado, tanto para archivos JavaScript como para archivos CSS.

Existen diferentes tipos de source maps:

- **Inline:** En los que la información de este mapeo es agregada directamente a los ficheros generados, es decir, va a estar todo incluido en el mismo bundle, con lo cual también aumenta un poco el tamaño del mismo.
- **Separate:** En este caso el module bundler va a emitir ficheros .map específicos para estos mapeos del código generado, por lo que están separados del bundle, por lo que él mismo va a tener un menor tamaño.

- **Explica qué es un transpilador.**

Transpilador es un tipo especial de compilador que traduce de un lenguaje fuente a otro fuente también de un nivel de abstracción parecido. Se diferencia de los compiladores tradicionales en que estos reciben como entrada ficheros conteniendo código fuente y generan código máquina del más bajo nivel.

Un ejemplo de transpilador en Javascript es Babel, una herramienta que nos permite transformar nuestro código JS de última generación (o con funcionalidades extras) a JS que cualquier navegador o versión de Node.js entienda.

3. El flujo de trabajo profesional en front-end.

- **Cita al menos dos sistemas de control de versiones.**

Un sistema de control de versiones es el software que gestiona los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

- Git
- Subversion
- Mercurial.
- Perforce.

- **Cita y explica al menos 3 comandos de Git.**

- **git init:** crea localmente un repositorio con GIT y así utilizar todo el funcionamiento que GIT ofrece. Basta con estar ubicados dentro de la carpeta donde tenemos nuestro proyecto y ejecutar el comando. Cuando agreguemos archivos y un commit, se va a crear el branch master por defecto.
- **git add -A:** agrega al repositorio todos archivos y carpetas que estén en nuestro proyecto, los cuales GIT no está siguiendo.
- **git commit -m "mensaje" + archivos:** hace commit a los archivos que indiquemos, de esta manera quedan guardados nuestras modificaciones.
- **git checkout NombreDeBranch:** sirve para moverse entre branches, en este caso vamos al branch que indicamos en el comando.
- **git push origin NombreDeBranch:** después de hacer commit, si estamos trabajando remotamente, este comando va a subir los

archivos al repositorio remoto, específicamente al branch que indiquemos.

- **git pull origin NombreDeBranch:** hace una actualización en nuestro branch local, desde un branch remoto que indicamos en el comando.

- **Cita y explica brevemente las características más definitorias de dos build systems.**

Un build system es un programa encargado de automatizar procesos informáticos que, sin su uso, deberían realizarse manualmente.

Al optimizar el flujo de trabajo, se tarda menos en realizar estas tareas y en consecuencia se dispone de más tiempo para dedicárselo a otras. Los dos aspectos básicos que diferencian a las diferentes soluciones son, por un lado, el modo en que se programan y, por otro, la forma de trabajar internamente. Al mismo tiempo, el usuario va a tener siempre el control sobre las fases del trabajo que se quieren automatizar y las que no.

Características:

- Gestión de dependencias.
- Ejecución de tareas.
- Conversión de formatos.
- Servidor de desarrollo.
- Carga y uso de módulos de todo tipo.