

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

21-9-2024

# Introducción a los Navegadores Web

DWEC

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Oscar Garcia Sanchez  
2ºDAW

## Contenido

1. Describe las novedades introducidas por HTML5 (HTML versión 5) .....	2
2. Describe las novedades introducidas por el estándar CSS3 (CSS versión 3). ....	4
3. Explica qué es EcmaScript. ¿Qué relación hay entre JavaScript y EcmaScript? Enumera al menos cinco novedades introducidas por las versiones de EcmaScript posteriores a la ES5.....	5
4. Investiga de los siguiente navegadores web: Chrome, Firefox, Edge, Opera y Safari la siguiente información y refléjala en una tabla: .....	5
5. Explica la diferencia entre los protocolos http y https. Averigua también qué número de puerto utilizan. ....	6
6. Explica la función de los siguientes componentes del navegador: .....	6
7. Explica las diferencias entre los lenguajes compilados e interpretados. ¿De qué tipo es JavaScript? ¿Y PHP? Enumera 3 lenguajes compilados y 3 lenguajes interpretados. ....	7
8. Busca información sobre qué es AJAX y para qué sirve. ¿Qué es XML y JSON? ¿Qué relación tienen XML y JSON con la tecnología AJAX? Pon un ejemplo de código XML y otro de JSON .....	7
9. Pon 5 ejemplos de direcciones URLs que incluyan, además de las partes obligatorias, el número de puerto, la parte de la query y la del fragment.....	9
10. ¿Qué es una API Rest? ¿Qué relación tiene con los métodos de petición HTTP? Describe los métodos HTTP usados por las API Rest y su función (refleja la información en una tabla)...	10
11. Debes escribir algún comentario en el foro (qué te han parecido las cuestiones, alguna duda que te ha surgido, etc.). Haz una captura de pantalla demostrativa de tu publicación en el foro.....	11

## 1. Describe las novedades introducidas por HTML5 (HTML versión 5)

Novedades introducidas por HTML5:

- Incorpora etiquetas (canvas 2D y 3D, audio, vídeo) con codecs para mostrar los contenidos multimedia. Actualmente hay una lucha entre imponer codecs libres (WebM + VP8) o privativos (H.264/MPEG-4 AVC).
- Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, Menu y Command. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios. Nuevos tipos de datos (eMail, number, url, datetime...) y facilidades para validar el contenido sin Javascript.
- Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales). En general se deja abierto a poder interpretar otros lenguajes XML.
- Drag & Drop. Nueva funcionalidad para arrastrar objetos como imágenes.
- Añade etiquetas para manejar la Web semántica (Web 3.0): header, footer, article, nav, time (fecha del contenido), link rel="" (tipo de contenido que se enlaza).
- Los buscadores podrán indexar e interpretar esta meta información para no buscar simplemente apariciones de palabras en el texto de la página.
- Permite incorporar a las páginas ficheros RDF / OWL (con meta información) para describir relaciones entre los términos utilizados.
- Además, ofrece versatilidad en el manejo y animación de objetos simples, imágenes etc.
- API para trabajar offline, API de Geolocalización, API Storage, WebSockets API de comunicación bidireccional entre páginas, WebWorkers.

## Resumen en el cambio de etiquetas:

- Añadidas:
  - <a> (atributo media añadido)
  - <article>
  - <aside>
  - <audio>
  - <canvas>
  - <datalist>
  - <details>
  - <dialog>
  - <embed>
  - <figure>
  - <footer>
  - <header>
  - <mark>
  - <meter>
  - <nav>
  - <output>
  - <progress>
  - <ruby>
  - <rp>
  - <rt>
  - <section>
  - <source>
  - <time>
  - <video>
- Eliminadas:
  - <acronym>
  - <applet>
  - <basefont>
  - <big>
  - <center>
  - <dir>
  - <font>
  - <frame>
  - <frameset>
  - <hgroup>
  - <isindex>
  - <noframes>
  - <s>
  - <strike>
  - <tt>
  - <xmp>

- Modificadas:
  - o <b> (Atributos globales)
  - o <cite> (Atributos globales)
  - o <hr>
  - o <i>
  - o <input> (Añadidos 13 elementos a type)
  - o <small> (Atributos globales)

<https://es.wikipedia.org/wiki/HTML5>

<https://desarrolloweb.com/articulos/nuevas-etiquetas-html5.html>

<https://html.spec.whatwg.org/multipage/>

## 2. Describe las novedades introducidas por el estándar CSS3 (CSS versión 3).

Mientras que en CSS2 se definía toda la especificación en tan solo un documento, CSS3 se ha modulado, y cada modulo añade nuevas funcionalidades a las especificaciones de CSS2, manteniendo así la compatibilidad.

Algunas de las novedades mas importantes serian:

- Esquinas redondeadas.
- Sombras.
- Transparencias de color.
- Texto en varias columnas.
- Nuevo modelo de cajas flexibles.
- Nuevos selectores.
- Nuevas pseudo-clases.
- Animaciones.

<https://www.jesustovar.es/veteasabertu/tipos-de-selectores-en-css1-css2-y-css3>

<https://master-diseño.com/las-nuevas-y-antiguas-versiones-de-css/>

<https://designicode.com/2023/03/26/la-evolucion-de-css-en-la-web/>

### 3. Explica qué es EcmaScript. ¿Qué relación hay entre JavaScript y EcmaScript? Enumera al menos cinco novedades introducidas por las versiones de EcmaScript posteriores a la ES5.

EcmaScript es el estándar encargado de definir el funcionamiento de lenguajes como por ejemplo JavaScript. Fue creado por Ecma International, una organización que desarrolla estándares técnicos y de información.

La relación entre EcmaScript y JavaScript es la necesidad de estandarización para conseguir una adecuada interoperabilidad entre los diferentes navegadores de la época.

Novedades introducidas tras ES5:

- ES6 ECMAScript 2015 → Añadido `let` y `const`, `Array.find()`, `Array.findIndex()`.
- ES6 ECMAScript 2016 → Añadido operador exponencial `(**)` y `Array.includes()`.
- ES6 ECMAScript 2017 → Añadido `Object.entries()`, `Object.values()`, funciones de `async`, memoria compartida, etc.
- ES6 ECMAScript 2018 → Añadido iteración asincrónica, Sumas a `RegExp`, `Promise.finally()`, `rest / spread properties`.
- ES6 ECMAScript 2019 → Añadido `String.trimStart()`, `String.trimEnd()`, `Array.flat()`, `Object.fromEntries`, `Optional catch binding`.

[https://www.w3schools.com/js/js\\_versions.asp](https://www.w3schools.com/js/js_versions.asp)

### 4. Investiga de los siguiente navegadores web: Chrome, Firefox, Edge, Opera y Safari la siguiente información y refléjala en una tabla:

	Engine	Cuota 2024	Última Vers.	EcmaScript
<b>Chrome</b>	Blink	65%	129.0.6668.58/59	ES6 2016
<b>Firefox</b>	Gecko	6.5%	130.0.1	ES6 2016
<b>Edge</b>	Blink	13%	127.0.2651.74	ES6 2016
<b>Opera</b>	Blink	2%	112.0.5197.60	ES6 2016
<b>Safari</b>	WebKit	9%	17.5	ES6 2016

[https://en.wikipedia.org/wiki/Comparison\\_of\\_browser\\_engines](https://en.wikipedia.org/wiki/Comparison_of_browser_engines)

<https://statcounter.com/> (Cuota de mercado según statcounter).

[https://www.w3schools.com/js/js\\_versions.asp](https://www.w3schools.com/js/js_versions.asp)

## 5. Explica la diferencia entre los protocolos http y https. Averigua también qué número de puerto utilizan.

La principal diferencia entre ambos protocolos estaría a nivel de seguridad. Siendo el https una versión más segura al cifrar los datos transferidos, protegiéndolos así mientras que http no lo realiza. Para ello utiliza certificados SSL/TLS.

El protocolo http utiliza el puerto 80, mientras el https hace uso del puerto 443.

## 6. Explica la función de los siguientes componentes del navegador:

- **Browser Engine:** Es el motor del navegador, encargándose de transformar el html y los diferentes recursos en lo que vamos a visualizar en la pantalla y va a ver el usuario.
- **Rendering Engine:** Es el encargado de renderizar y pintar las imágenes y contenidos de la página web.
- **Intérprete JavaScript:** su función será la de interpretar/“traducir” y ejecutar el código JavaScript.
- **Data Persistence:** Es la capa encargada de la persistencia de los datos, sería como tener un MongoDB en el navegador, pero sin su capacidad ni potencia. (cookies, localStorage, WebSQL, etc)
- **UI Backend:** Es la interfaz que aprovecha los recursos del sistema operativo para hacer peticiones en segundo plano y no atascar el hilo principal.

<https://miso-4208-labs.gitlab.io/book/chapter3/31-la-arquitectura-de-un-browser.html>

## 7. Explica las diferencias entre los lenguajes compilados e interpretados. ¿De qué tipo es JavaScript? ¿Y PHP? Enumera 3 lenguajes compilados y 3 lenguajes interpretados.

Los lenguajes compilados generan un código que posteriormente será ejecutado, que puede darse creando un archivo ejecutable con el código máquina, mientras que los interpretados son lenguajes cuyas instrucciones se traducen en el mismo momento de la ejecución sin ningún tipo de código intermedio.

Tanto PHP como JavaScript son lenguajes interpretados.

Lenguajes compilados → C, C++, Visual Basic, Pascal, Fortran, Rust.

Lenguajes interpretados → PHP, JavaScript, Pyth, Perl, Java.

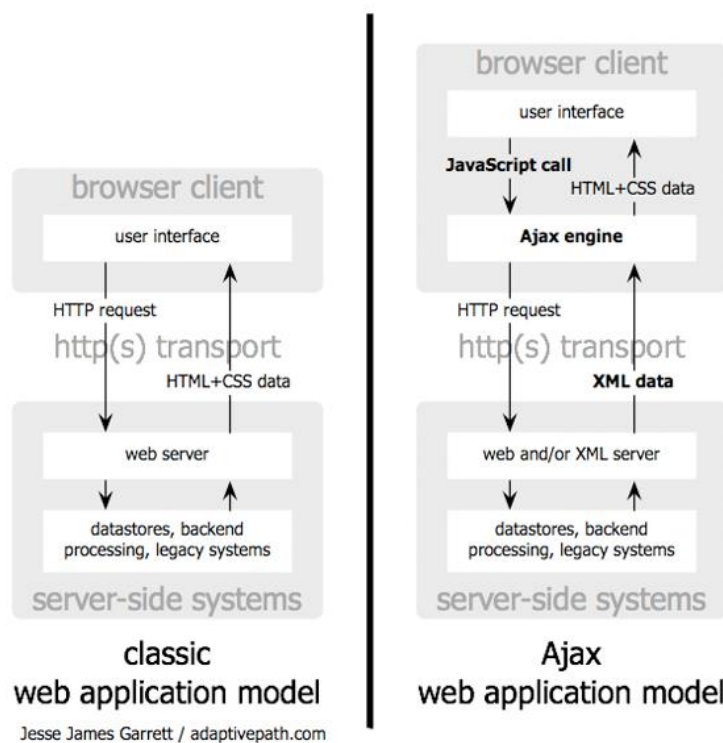
Fuente: Apuntes Desarrollo del Software 1ºDAW.

## 8. Busca información sobre qué es AJAX y para qué sirve. ¿Qué es XML y JSON? ¿Qué relación tienen XML y JSON con la tecnología AJAX? Pon un ejemplo de código XML y otro de JSON

El término **AJAX** (Asynchronous JavaScript And XML) es una técnica de desarrollo web, que permite comunicar el navegador del usuario con el servidor, en un segundo plano. De esta forma, se podrían realizar peticiones al servidor sin tener que recargar la página, y podríamos gestionar esas respuestas, que nos permitirían actualizar los contenidos de nuestra página, sin tener que realizar recargas.

Cada vez que se realiza una petición al servidor, el usuario lo único que puede hacer es esperar, ya que muchas veces la página cambia a otra diferente, y hasta que no reciba todos los datos del servidor, no se mostrará el resultado, con lo que el usuario no podrá interactuar de ninguna manera con el navegador. Con AJAX, lo que se intenta evitar, son esencialmente esas esperas. El cliente podrá hacer solicitudes al servidor, mientras el navegador sigue mostrando la misma página web, y cuando el navegador reciba una respuesta del servidor, la mostrará al cliente y todo ello sin recargar o cambiar de página





**XML** → XML es un metalenguaje que permite crear lenguajes de marcas que se usan elaborar ficheros para almacenar datos de forma legible.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <breakfast_menu>
3    <food>
4      <name>Belgian Waffles</name>
5      <price>$5.95</price>
6      <description>Two of our famous Belgian Waffles with plenty
7      of real maple syrup</description>
8      <calories>650</calories>
9    </food>
10   <food>
11     <name>Strawberry Belgian Waffles</name>
12     <price>$7.95</price>
13     <description>Light Belgian waffles covered with
14     strawberries and whipped cream</description>
15     <calories>900</calories>
16   </food>
17 </breakfast_menu>

```

**JSON** → es un formato ligero para el intercambio de datos. Es un subconjunto de la notación literal de objetos de JavaScript. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX.

```
{ "Fruteria":
  [
    { "Fruta":
      [
        { "Nombre": "Manzana", "Cantidad": 10 },
        { "Nombre": "Pera", "Cantidad": 20 },
        { "Nombre": "Naranja", "Cantidad": 30 }
      ]
    },
    { "Verdura":
      [
        { "Nombre": "Lechuga", "Cantidad": 80 },
        { "Nombre": "Tomate", "Cantidad": 15 },
        { "Nombre": "Pepino", "Cantidad": 50 }
      ]
    }
  ]
}
```

**Fuente:** Apuntes de Lenguajes de Marca y Sistemas de Gestión de la Información 1º DAW.

9. Pon 5 ejemplos de direcciones URLs que incluyan, además de las partes obligatorias, el número de puerto, la parte de la query y la del fragment.

Protocolo Dominio Puerto Ruta Query Fragmento

1. <https://ciclos.iesruizgijon.es:443/course/view.php?id=30&section=8#tabs-tree-start>
2. [https://es.wikipedia.org:443/wiki/Basilisco\\_de\\_Roko?utm\\_source=example#Cr%C3%ADticas](https://es.wikipedia.org:443/wiki/Basilisco_de_Roko?utm_source=example#Cr%C3%ADticas)
3. <https://www.youtube.com:443/watch?v=7Qp5vcuMilk&list=PLFgquLnL59alW3xmYiWRaoz0oM3H17Lth#t=30s>
4. <https://www.amazon.com:443/gp/product/B07FZ8S74R?ie=UTF8&th=1#productDetails>
5. [https://fama.us.es:443/discovery/search?query=any,contains,anatomia&vid=34CBUA\\_US:VU1&tab=all\\_data\\_not\\_idus&bulkSize=10&highlight=true&dum=true&displayField=all&pcAvailabilityMode=true&search=anatomia&search\\_scope=all\\_data\\_not\\_idus#:~:text=Prometheus%20%3A%20texto%20y,Sch%C3%BCnke%2C%20Michael%2C%20autor](https://fama.us.es:443/discovery/search?query=any,contains,anatomia&vid=34CBUA_US:VU1&tab=all_data_not_idus&bulkSize=10&highlight=true&dum=true&displayField=all&pcAvailabilityMode=true&search=anatomia&search_scope=all_data_not_idus#:~:text=Prometheus%20%3A%20texto%20y,Sch%C3%BCnke%2C%20Michael%2C%20autor)

## 10. ¿Qué es una API Rest? ¿Qué relación tiene con los métodos de petición HTTP? Describe los métodos HTTP usados por las API Rest y su función (refleja la información en una tabla).

Una API Rest, es una interfaz de programación de aplicaciones ajustada a los límites de la arquitectura REST, permitiendo la interacción con los servicios web de RESTful. Siendo una API un conjunto de definiciones y protocolos. Puede explicarse como un contrato entre un proveedor de información y el usuario. Por ejemplo, una API de servicio meteorológico, le requiere al usuario una localización y que el productor de una respuesta por ejemplo de la temperatura máxima y mínima. Por ello podemos considerar a la API como un mediador entre el usuario y los recursos/información o los servicios web que quieran obtener.

La relación existente entre la API Rest y HTTP es que cuando el usuario realiza la solicitud a través de la API Rest, la información será entregada por medio de HTTP en formato JSON, HTML, XLT, etc.

<https://www.redhat.com/es/topics/api/what-is-a-rest-api>

<https://www.ibm.com/es-es/topics/rest-apis>

Métodos HTTP	Descripción	Función Principal
GET	Con este método se solicita una representación de un recurso específico sin modificar su estado.	Recuperar/Consultar información o recursos. (Ejemplo obtener una lista de vehículos o un vehículo específico).
POST	Envía datos al servidor para crear un nuevo recurso.	Creación de un nuevo registro. (Ejemplo crear un nuevo vehículo).
PUT	Actualizar por completo un registro ya existente.	Reemplazar un registro o recurso ya existente (Ejemplo actualizar los datos de un vehículo).
PATCH	Actualiza de forma parcial un registro existente.	Modificar un registro o recurso ya existente (Ejemplo modificar el color de un vehículo).
DELETE	Elimina por completo un registro existente.	Borrar un registro o recurso ya existente. (Ejemplo borrar un vehículo existente).

HEAD	Obtiene información de un registro/recurso, pero sin obtener el cuerpo del recurso en sí.	Obtener metadatos del recurso sin descargar el contenido completo.
OPTIONS	Describe las opciones de comunicación con el servidor	Mostrar los métodos http permitidos. (Ejemplo comprueba si el servidor permite el GET).
TRACE	Devuelve la ruta que sigue una solicitud hasta el servidor.	Mostrar la ruta de la solicitud.
CONNECT	Establece una conexión con el servidor. Normalmente no es necesario usarlo.	Permitir conexiones.

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

<https://www.oscarblancarteblog.com/2018/12/03/metodos-http-rest/>

- Debes escribir algún comentario en el foro (qué te han parecido las cuestiones, alguna duda que te ha surgido, etc.). Haz una captura de pantalla demostrativa de tu publicación en el foro.

### Foro sobre Navegadores Web

Comentario sobre la Tarea Navegadores Web

CONFIGURACIONES

PRÁCTICA OBLIGATORIA: NAVEGADORES WEB

Mostrar respuestas anidadas



**Comentario sobre la Tarea Navegadores Web**  
de Óscar García Sánchez - miércoles, 25 de septiembre de 2024, 12:31

Las tareas en general me han parecido adecuadas para consolidar los conceptos de la Unidad 1 sobre navegadores explicada en clase. En un principio no me han surgido dudas.

ENLACE PERMANENTE   EDITAR   BORRAR   RESPONDER

PRÁCTICA OBLIGATORIA: NAVEGADORES WEB