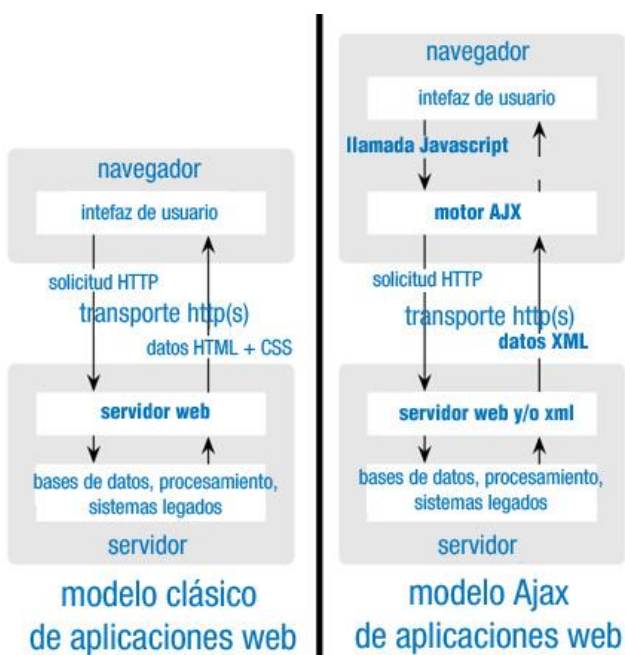


Unidad 6. Programación AJAX en JavaScript. Boletín de actividades

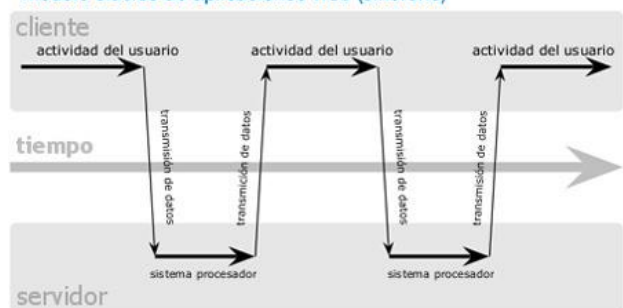
Introducción

AJAX es una técnica que permite intercambiar datos con un servidor en segundo plano. Esta característica posibilita crear páginas web más parecidas a las aplicaciones de escritorio en las que una página no tiene que refrescarse completamente en cada iteración con el servidor. Sus siglas significan *Asynchronous JavaScript and XML* porque habilita una comunicación asíncrona con el servidor y el intercambio de información se puede hacer en formato XML. AJAX es posible gracias al objeto XMLHttpRequest, cuyo uso y aplicaciones estudiaremos en la unidad.

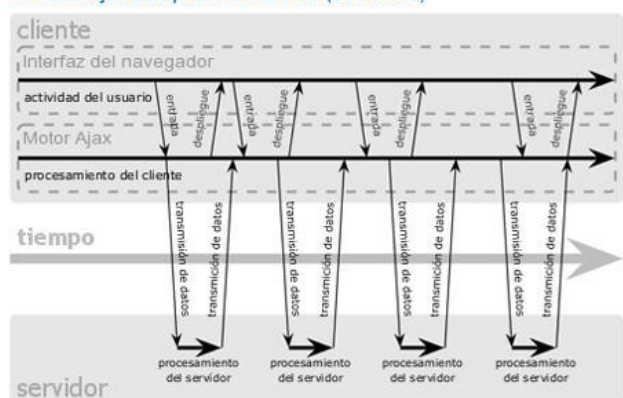
El siguiente gráfico muestra las diferencias entre las aplicaciones web clásicas y las aplicaciones que usan AJAX:



modelo clásico de aplicaciones web (síncrono)



modelo Ajax de aplicaciones web (asíncrono)



Referencias:

- ⑩ Tutorial de AJAX de W3Schools.
- ⑩ Artículos sobre AJAX en la MDN.

Preparación del equipo.

Para realizar las actividades de esta unidad necesitaremos un servidor web sobre el que realizar peticiones. También utilizaremos PHP en el lado del servidor para generar las respuestas dinámicamente.

Actividad 1. Instalación de XAMPP.

La forma más rápida de implantar estos programas en nuestro equipo es instalar el paquete XAMPP. Instala el servidor Apache, el intérprete PHP (versión 7) y el servidor de base de datos MariaDB en tu equipo a través de XAMPP.

Referencias:

- ⑩ Web oficial de XAMPP.
- ⑩ Artículo de instalación de XAMPP en GNU/Linux de Ubuntizando.

Aplicación clásica vs aplicación AJAX.

Para entender qué implica AJAX en una web realizaremos una aplicación completa (lado cliente y lado servidor) de la forma clásica y utilizando AJAX.

Actividad 1. Base de datos de álbumes de música (modelo clásico)

Realiza una aplicación web usando el modelo clásico que muestre una tabla con álbumes de música. Al hacer clic sobre una de las filas, se mostrará información detallada del álbum en cuestión.

Click on a CD to display album information.	
Artist	Title
Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey
Jorn Hoel	Soulsville
Cat Stevens	The very best of
Sam Brown	Stop
T'Pau	Bridge of Spies
Tina Turner	Private Dancer
Kim Larsen	Midt om natten
Luciano Pavarotti	Pavarotti Gala Concert
Otis Redding	The dock of the bay
Simply Red	Picture book
The Communards	Red
Joe Cocker	Unchain my heart

>>>>>>>

Click on a CD to display album information.	
Artist: Many	
Title: 1999 Grammy Nominees	
Year: 1999	
Artist	Title
Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey
Jorn Hoel	Soulsville
Cat Stevens	The very best of
Sam Brown	Stop
T'Pau	Bridge of Spies
Tina Turner	Private Dancer
Kim Larsen	Midt om natten
Luciano Pavarotti	Pavarotti Gala Concert
Otis Redding	The dock of the bay
Simply Red	Picture book
The Communards	Red
Joe Cocker	Unchain my heart

Actividad 2. Base de datos de álbumes de música (modelo AJAX)

Modifica la aplicación anterior para conseguir la misma funcionalidad pero utilizando el modelo AJAX.

AJAX con JSON.

JSON (JavaScript Object Notation) es un formato de texto que sigue la sintaxis de un objeto literal de JavaScript. Ha sido diseñado para el intercambio eficiente de información entre cliente y servidor.

Referencias:

- ⑩ Tutorial de JSON de la web de W3Schools.

Información del servidor con JSON.

Para iniciarnos en JSON crearemos una aplicación AJAX sencilla que recupere 3 datos del servidor y actualice la interfaz de usuario.

Actividad 1. Aplicación del lado del servidor.

Implementa una aplicación PHP que devuelva los siguientes datos en formato JSON:

1. Plataforma desde la que el cliente está realizando la petición. Variable `$_SERVER['HTTP_USER_AGENT']`
2. Versión del intérprete de PHP que está corriendo en el servidor. Función `phpversion`.
3. Fecha actual del servidor. Función `date`.

Referencias:

- ⑩ Artículo PHP JSON de W3Schools.

A continuación, añade los siguientes metadatos a la cabecera de respuesta HTTP. Función `header`.

- ⑩ `Content-Type: application/json; charset=utf-8` para indicar que los datos se envíen en formato JSON.
- ⑩ `Access-Control-Allow-Origin: *` para permitir el acceso a la aplicación desde un origen distinto a localhost (por ejemplo, desde un fichero ubicado fuera del servidor).

Actividad 2. Aplicación del lado del cliente.

Implementa una aplicación JavaScript que recupere los datos JSON del servidor y actualice los 3 elementos `span` del siguiente código HTML:

```
<div class="informacionServidor">
```

La plataforma es `???`. El servidor corre la versión `??? ` de PHP. La hora actual del servidor es `???`.

```
</div>
```

Referencias:

- ⑩ Artículo JSON Parse de W3Schools.

Actividad 2.1. Actualización de datos.

Añade el código necesario a la aplicación JavaScript para que haga una petición de los datos al servidor cada segundo y los actualice en la web.

Actividad 2.2. Monitorización de peticiones.

El objeto XMLHttpRequest permite conocer qué ocurre con una petición HTTP una vez lanzada. Para determinar el estado de una transferencia se pueden escuchar los eventos: progress, load, error, abort y loadEnd.

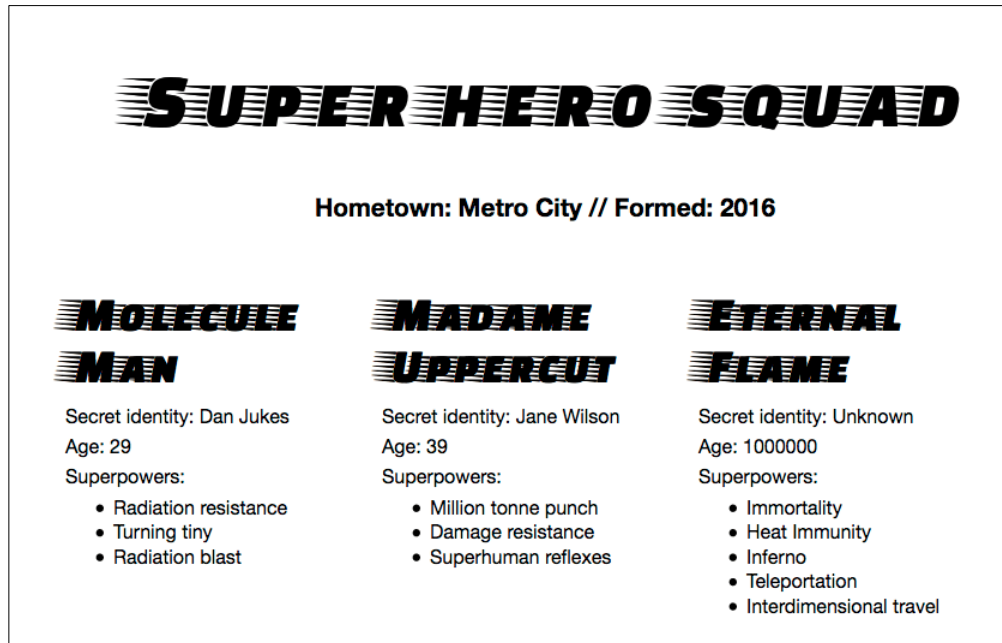
Modifica la aplicación JavaScript para conocer en cada momento el estado de las peticiones.

Referencias:

- ⑩ Apartado Monitoring progress del artículo Using XMLHttpRequest de la MDN.

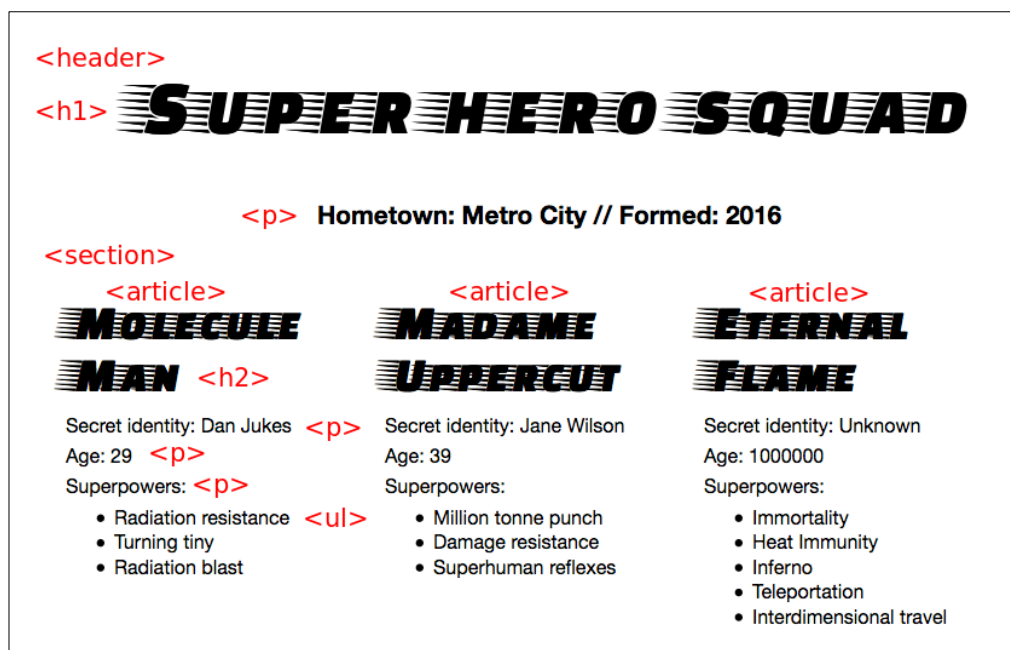
Aplicación de Super héroes con JSON.

En esta actividad vamos a crear una aplicación Ajax que nos permitirá profundizar en el formato JSON. Implementa una aplicación que recupere los datos de un escuadrón de super héroes del servidor de la MDN y forme la siguiente web:



Parte de los siguientes ficheros HTML y CSS. Los datos de los super héroes están en la siguiente dirección: <https://mdn.github.io/learning-area/javascript/ojs/json/superheroes.json>

Para conseguir el mismo aspecto de la imagen, inserta los datos siguiendo esta estructura HTML:



Recursos: artículo Trabajando con JSON de MDN.

Envío de datos al servidor.

Hasta ahora las aplicaciones se han centrado en recuperar información del servidor. En este apartado veremos cómo enviar datos.

El protocolo HTTP cuenta con varias formas de enviar datos a un servidor, las más comunes son el método GET y el método POST. Antes de introducir estos métodos, ten en cuenta que una petición HTTP tiene dos partes:

- ⑩ Una **cabecera** que contiene metadatos sobre los datos.
- ⑩ Un **cuerpo** que contiene datos en sí.

Método GET

El método GET envía los datos en la cabecera de la petición, concretamente en la URL. Por ejemplo al enviar este formulario:

```
<form action="http://servidor.com" method="get">
  <input name="id" value="12">
  <input name="seccion" value="javascript">
```

Se formará una URL como la siguiente:

`www.servidor.com?id=12&seccion=javascript`

El principal uso del método GET es pedir información al servidor en función de unos parámetros. Por ejemplo solicitar un artículo de una revista en función de su identificador.

Método POST

El método POST envía los datos en el cuerpo de la petición. Por ejemplo para este formulario:

```
<form action="http://servidor.com" method="post">
  <input name="nombre" value="juan">
  <input name="edad" value="20">
```

Se generará una petición como la siguiente:

```
Host: servidor.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 19

nombre=Juan&edad=20
```

Por defecto el método POST codifica los datos siguiendo las reglas de formato de una URL. (Content-Type: application/x-www-form-urlencoded) las que también usa el método GET. Siguiendo este formato los datos se envían como pares clave-valor separados por un símbolo &. Cada clave se separa de su valor por el símbolo =. Tanto en las claves como en los valores se reemplazan los espacios en blanco por el carácter + así como los caracteres que no sean alfanuméricos siguiendo otras reglas del formato.

Por ejemplo, los siguientes pares:

Name: Gareth Wylie
Age: 24
Formula: a + b == 13%

son codificados como:

Name=Gareth+Wylie&Age=24&Formula=a+%2B+b+%3D%3D+13%25%21

El método POST cuenta con otra posible codificación: multipart/form-data utilizada para enviar datos binarios como ficheros.

Referencias:

- ⑩ Artículo Sending form data de la MDN.
- ⑩ Artículo HTTP Request Methods de W3Schools.

Al final del enlace anterior de W3Schools puedes ver una comparativa entre los métodos GET Y POST.

Envío de datos con AJAX

A continuación crearemos aplicaciones AJAX que envíen datos al servidor utilizando distintos métodos. La funcionalidad de las aplicaciones será muy simple para que podamos centrarnos en los métodos de envío. Siempre realizarán la misma tarea:

- 1) Al pulsar el botón de envío, la aplicación AJAX recuperará los datos de los campos del formulario y los enviará al servidor utilizando el método indicado.
- 2) La aplicación del servidor recogerá los datos y los devolverá tal cuál en texto plano.
- 3) La aplicación AJAX mostrará los datos recibidos por consola.

Referencias:

- ⑩ Artículo [Short XHR Examples](#) en Github sobre métodos de envío con AJAX.
- ⑩ Artículo [XMLHttpRequest RESTful](#) en Github sobre envío y procesamiento de respuesta.

Actividad 1. Envío de datos con AJAX. Método GET.

Seleccionar Artículo

Capítulo:

Sección:

Actividad 2. Envío de datos con AJAX. Método POST.

Formulario AJAX

Nombre:

Apellidos:

Sexo: ☐ Hombre ☐ Mujer

Contraseña:

¿Qué prefieres?:

☐ Tengo una bici.

☐ Tengo un coche.

Describe a ti mismo:

Referencias:

- ⑩ Apartado *Building an XMLHttpRequest manually* del artículo *Sending forms through JavaScript* de la MDN.

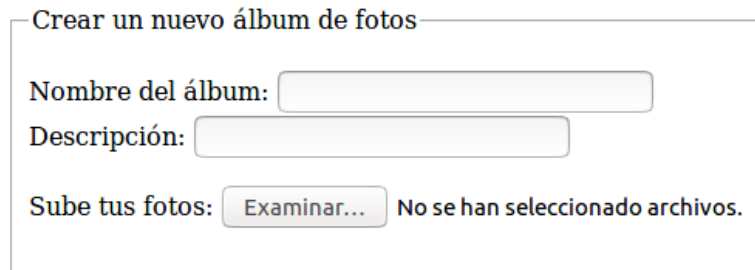
Actividad 3. Envío de datos con AJAX. Método POST y JSON.

Modifica la aplicación anterior para enviar los datos en formato JSON.

Envío de datos con AJAX. Ficheros.

Para enviar ficheros al servidor se puede utilizar el método POST con el tipo de codificación `multipart/form-data` o utilizar la nueva API `FormData` de JavaScript. Estos métodos pueden utilizarse aunque el formulario no tenga que enviar ficheros al servidor.

Actividad 1. Envío de datos con AJAX. POST y enctype: multipart/form-data.



Crear un nuevo álbum de fotos

Nombre del álbum:

Descripción:

Sube tus fotos: No se han seleccionado archivos.

Referencias:

- ⑩ Apartado Dealing with binary data del artículo Sending forms through JavaScript de la MDN.

Actividad 2. Envío de datos con AJAX. Objeto FormData.

Modifica la aplicación anterior para utilizar el objeto `FormData` en el envío de datos al servidor.

Para conocer el funcionamiento de este método revisa el

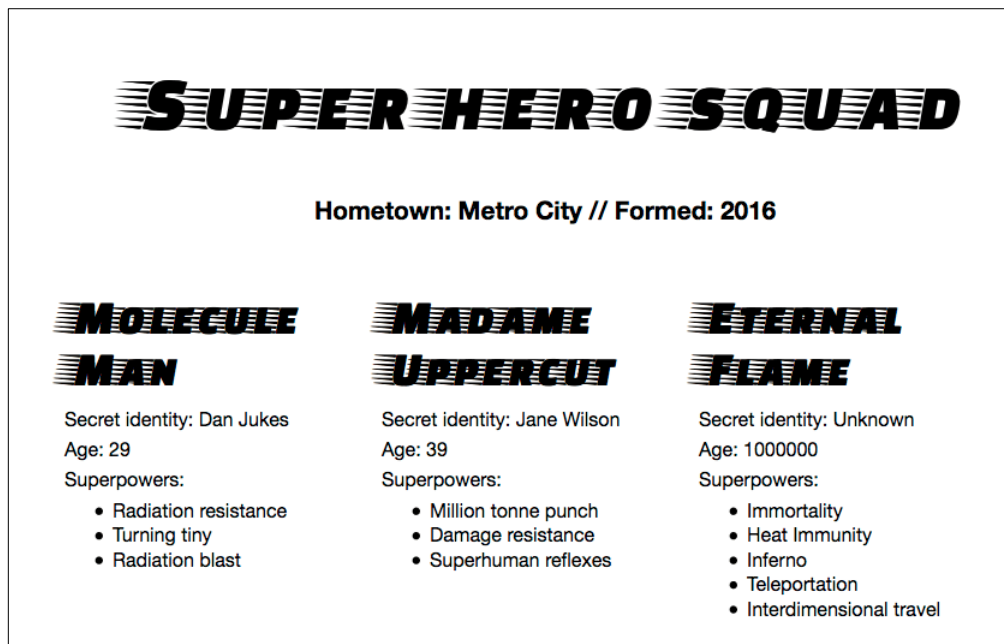
Referencias:

- ⑩ Artículo Usando objetos `FormData` de la MDN.
- ⑩ Artículo Uploading files using `xhr.send(FormData)` to PHP server en Github.

Aplicación de Super Héroes con jQuery y JSON.

Para iniciarnos en el uso de jQuery como librería cross-browser vamos a realizar la aplicación de super héroes con algunos cambios.

Ésta es la aplicación original:



Actividad 1. Aplicación del lado del servidor.

Implementa una aplicación en PHP que devuelva en formato JSON los datos de la cabecera, los del superhéroe *MoleculeMan*, los del superhéroe *MadameUppercut* o los del superhéroe *EternalFlame* en función de un objeto JSON enviado al servidor con el siguiente formato:

`superhero:booleanValue, noOfSuperHero:integerValue`

Por ejemplo, el objeto JSON: `superhero:true, noOfSuperHero:1` devolvería los datos del superhéroe *Madame Uppercut* en formato JSON.

Actividad 2. Aplicación del lado del cliente.

Implementa una aplicación con jQuery que muestre la cabecera de la página y dos botones: **Prev** y **Next**. Al pulsar sobre **Next** se pedirán al servidor los datos del siguiente superhéroe y se mostrarán en pantalla. El botón **Prev** mostrará el superhéroe anterior.

Referencias:

⑩ <https://gabrieleromanato.name/jquery-sending-json-data-to-php-with-ajax>