

Tema 5: Implantación de contenido multimedia



Ciclo Superior DAW

Asignatura: Diseño de interfaces

Curso 20/21



Introducción

- En este capítulo aprenderemos los siguientes conceptos:
 - Conocer la legislación relativa a derechos de autor
 - Conocer los distintos tipos de imágenes
 - Añadir imágenes a una interfaz
 - **Utilizar y gestionar sonido en una interfaz**



Introducción

- En este capítulo aprenderemos los siguientes conceptos:
 - **Utilizar y gestionar vídeo en una interfaz**
 - **Conocer y gestionar formas de almacenamiento en HTML5**
 - Utilizar geolocalización en documentos Web



Introducción

- Se presentan la inclusión de elementos multimedia como imágenes, sonido, vídeo e incluso almacenamiento y geolocalización.
- Veremos distintos programas para generar y manipular todos esos tipos de medios.



Legislación

- Es posible que al crear las interfaces se tenga que incluir algún recurso ajeno como, por ejemplo, imágenes o fotografías.
- Estos recursos no pueden usarse de cualquier manera
- Se deben comprobar las licencias que tienen para su reutilización



Legislación

- Cualquier obra puede estar sujeta a ciertos derechos y conviene tener claro en qué consiste cada uno de ellos
- Hablaremos de:
 - LPI: Ley de propiedad intelectual
 - Derechos de autor
 - Licencias Copyleft
 - Recursos disponibles



Legislación. Ley de propiedad intelectual

- La ley en vigor en España es la LPI y regula el uso de todo tipo de obras, estableciendo una serie de límites o excepciones.
- La ley otorga el derecho a la copia privada, siempre para uso exclusivamente privado.



Legislación. Ley de propiedad intelectual

- Como ejemplo de aplicación, el artículo 32: entró en vigor en el 2015 y ha cambiado mucho a lo largo de los años. Se incluyó el **caso Google News** con todo lo que supuso.
- En ningún caso se pierden derechos al compartir una obra, lo que se hace es ceder derechos, pero como mínimo se mantiene el derecho a cita.



Legislación. Ley de propiedad intelectual

- Si se utiliza el material de otro estado, la ley que prevalece es la ley del país donde lo utilizas. Por ejemplo, en España no podemos usar la obra Peter Pan, pero en otros países sí.
- En el caso de la red, no hay blanco y negro. Por ejemplo, Google intenta usar la legislación americana, pero si un contenido está orientado a un país concreto se le aplica esa ley. En España, no se puede usar Google News.



Legislación. Derechos de autor

- Los derechos de autor en España caducan a los 50 años
- Si el material es propio, se puede usar siempre
- Si es de dominio público, la restricción es respetar al autor y mantener la integridad de la obra



Legislación. Licencias Copyleft

- Son licencias más abiertas con algunos derechos reservados
- Conocidas como licencias **Creative Commons** o simplemente **CC**
- Estas licencias se desarrollaron por una organización sin ánimo de lucro



Legislación. Recursos disponibles

- Existen muchas plataformas que proporcionan contenidos con licencias CC
- Algunos ejemplos son:
 - Fuentes: www.dafont.com, www.1001fonts.com
 - Imágenes: Flickr, DevianArt



Imágenes

- Para incluir imágenes en la web, es fundamental saber qué formato se va a utilizar. No vale cualquier formato, aunque el navegador pueda reproducirlo.
- Se debe tener en cuenta:
 - La resolución
 - El tamaño del archivo
 - El color
 - Etc



Imágenes. Formato GIF

- Los ficheros GIF son imágenes de colores indexados con un máximo de 256 colores disponibles, ya que utilizan 8 bits para definir los colores.
- Tienen como características:
 - Permite la transparencia
 - Utilizan compresión LZW
 - Permite animaciones, incluyendo un conjunto de imágenes en un único fichero
 - Permiten el entrelazado



Imágenes. Formato GIF

- Si convertimos una imagen con millones de colores a GIF, lo primero es reducir los colores a una paleta de 256 colores o menos.
- Por lo tanto, la imagen puede perder mucha calidad y nitidez, en especial si se trata de una fotografía.



Imágenes. Formato GIF

¿Cuándo utilizar el formato GIF?

- Para imágenes que no necesiten muchos colores o que solo tengan colores planos, como, por ejemplo:
 - Iconos
 - Logotipos
 - Gráficos de datos, etc.



Imágenes. Formato GIF. GIFs animados

- Pese a que los GIF animados puedan parecer cómicos, **su uso se desaconseja totalmente en el diseño de interfaces** por el hecho obvio de que distraen fácilmente la atención del usuario.
- Hoy en día vuelven a utilizarse en ocasiones cuando se necesita explicar algo breve de forma visual sin tener que hacer un vídeo o una animación Flash.



Imágenes. Formato JPEG

- Los ficheros JPEG son ficheros de imágenes que contienen 24 bits de información para los colores, lo que da lugar a millones de colores y, por lo tanto, lo convierte en un buen formato para las imágenes fotográficas
- Sus principales características son:
 - Imágenes de colores 24-bits
 - Compresión con pérdida de calidad
 - Soportan el entrelazado



Imágenes. Formato JPEG

- JPEG tiene una ventaja añadida para la Web, y es que aplica compresión a las imágenes, aunque con cierta pérdida.
- La pérdida y la calidad de un JPEG es ajustable en el momento de crearla, pero en cualquier caso la relación que ofrece entre calidad/compresión es bastante buena.



Imágenes. Formato JPEG

- JPEG no utiliza una paleta de colores limitada como el GIF, lo que permite imágenes detalladas e incluso escalas de grises de mucha calidad.
- Por lo tanto, para imágenes fotográficas o, en general, los gráficos con colores degradados y, en especial, con imágenes con las que nos podamos permitir cierta pérdida, **el formato JPEG es el ideal.**



Imágenes. Formato JPEG. Compresión

- Las ratios de compresión que puede conseguir JPEG son muy buenos, pudiendo ser desde 10:1 hasta 20:1
- **En cuanto a rendimiento, la compresión del JPEG requiere que antes de visualizarse la imagen se tenga que descomprimir, por lo que una interfaz configurada con muchas imágenes de cierta resolución puede provocar una carga más lenta.**



Imágenes. Formato PNG

- Se sitúa entre el GIF y el JPEG, pero no por su calidad, sino porque sus cualidades pueden ser variadas según lo que precisemos:
 - Soporte para imágenes de colores indexados en 8 bits.
 - Permite transparencia, que además puede ser gradual.
 - Permite imágenes de escalas de grises de 16 bits.
 - Imágenes de colores de 24 bits y 48 bits con una compresión sin pérdida.
 - Se pueden mostrar de forma progresiva



Imágenes. Formato PNG

- Hasta hace bien poco había algunos problemas de compatibilidad con algunos navegadores y el PNG, aunque, afortunadamente, eso ya ha sido superado.
- En principio **PNG debería sustituir al formato GIF** en la mayoría de los casos, **pero no así al formato JPEG** debido a que su compresión no tiene pérdida y, por tanto, no consigue optimizar el tamaño de los ficheros.



Imágenes. Formato PNG. Transparencia

- Una de las cualidades que aporta versatilidad al PNG es la **posibilidad de crear imágenes con transparencia.**
- Además de la transparencia binaria (se ve o no), permite una transparencia gradual a través un canal extra añadido a los valores RGB: el canal alfa, el cual es capaz de contener una información de 8 bits para 256 niveles de transparencia.



Imágenes. ¿Qué formato seleccionar?

ESCENARIO	FORMATO	MOTIVACIÓN
Fotografía o colores degradados	JPEG	Mejor resultado a tamaño reducido
Gráfico con colores planos	GIF, PNG 8 bits	Para gráficos de tonos planos es mejor usar formatos de colores indexados
Combinación de arte fotográfico, tipografías y colores planos	GIF, PNG 8 bits	Se descarta JPEG por no ser eficiente
Requiere animación	GIF	No hay más opciones
Se necesita transparencia	GIF, PNG	Únicos formatos permitidos
Se precisa transparencia gradual	PNG	No hay más opciones



Actividad 1

“La mejor forma de animar una página es introduciendo un GIF animado, en vez de un vídeo”

¿Qué opináis de estas declaraciones? ¿Verdadero o falso?

De los formatos que vimos anteriormente, ¿cuál elegiríais para vuestra página y por qué?



Imágenes. Mapa de bits

- Los **bitmaps** son un formato de imagen muy simple que consiste en un conjunto de bits que de manera similar a un mosaico forman una imagen.
- Una imagen en formato bitmap no tiene ningún tipo de optimización.
- **No resultan eficientes para la web.**
- La más conocida es **BMP**



Imágenes. Imagen vectorial

- Una **imagen vectorial** puede ampliarse o reducirse todo lo que haga falta y mantiene exactamente la misma nitidez.
- Esto es posible gracias a que la imagen vectorial está formada de definiciones de líneas, curvas, figuras geométricas básicas, colores y toda clase de combinaciones y transformaciones entre ellas.



Imágenes. Imagen vectorial

- Aparentemente es una imagen normal, pero no se almacena como un fichero binario con un mapa de bits, sino como un conjunto de formas geométricas; por eso, el formato de un **fichero SVG de dibujo vectorial es un fichero de texto con etiquetas XML**



Actividad 2

Busca en Google un ejemplo de mapa de bits y uno de una imagen vectorial.

¿Qué observas en cada uno de ellos? ¿Los pondrías en tu web?



Imágenes. Software de edición de imágenes

- Existen muchos programas que permiten crear y procesar imágenes antes de subirlas a la web.
- Veremos algunos ejemplos a continuación:



Imágenes. Software de edición de imágenes

- **Photoshop**

- Photoshop de Adobe es el programa de edición de imágenes más famoso. El formato de un fichero Photoshop es el PSD, pero obviamente puede importar y exportar para todos los tipos de formatos de imágenes.
- El único inconveniente en definitiva es que es un software de pago y que quizá, según para lo que se necesite no es necesario un programa con un número tan elevado de prestaciones.



Imágenes. Software de edición de imágenes

- **Gimp**

- Una buena alternativa para la edición de imágenes es el Gimp, un clon de Photoshop que podemos ejecutar en cualquier sistema operativo. Gimp cubre todas las necesidades esenciales de la edición de imágenes.
- Gimp tiene un reducido tamaño y a se puede descargar de forma totalmente gratuita.



Imágenes. Software de edición de imágenes

- **Adobe Illustrator**
 - Herramienta de edición de dibujo vectorial de Adobe. Orientada a dibujar, crear logotipos, botones, iconos, vectorial etc. Utiliza su propio formato de fichero, que es el AI.



Imágenes. Software de edición de imágenes

- **Inkscape**

- Inkscape es la mejor alternativa libre que existe para crear dibujo vectorial. También disponible para todas las plataformas y con un tamaño mucho más reducido que los programas de Adobe.
- Además, Inkscape es capaz de abrir ficheros en formato AI de Illustrator.



Imágenes. Software de edición de imágenes

- **ImageMagick**

- Permite manipular archivos de imagen desde la línea de comandos y llevar a cabo todo tipo de operaciones de forma automatizada.
- ImageMagick puede hacer de todo: convertir formatos, escalar imágenes, extraer una parte, etc. para las operaciones más típicas nos puede bastar con echar un ojo a los comandos.



Gráficos HTML5

- **Los formatos más comunes para web son JPEG, PNG y GIF**
- En menor medida también está el formato SVG, aunque esta se muestra de forma totalmente distinta.
- Para añadir imágenes usaremos las etiquetas:
 - ****
 - **<figure>**



Gráficos HTML5. Etiqueta

- **Etiqueta img**
- La manera más simple de incluir una imagen es a través de la conocida etiqueta:

```

```

- Debe hacer referencia a un fichero externo que obviamente debe existir para que esa imagen se muestre.



Gráficos HTML5. Etiqueta <figure>

- **Etiqueta figure**
- Esta es la nueva manera de incluir imágenes que ha introducido HTML5.
- Es un elemento que nos permite introducir una imagen y su pie de imagen correspondiente mediante la etiqueta hija <figcaption>.



Gráficos HTML5. Etiqueta <figure>

- Su formato es el siguiente:

```
<figure>
  
  <figcaption>
    Caption de la imagen
  </figcaption>
</figure>
```




Actividad 3

Descarga dos imágenes diferentes y añádelas a una página web utilizando las dos etiquetas existentes.

¿Qué diferencias ves entre las dos?



Optimización de imágenes para la Web

- El tamaño de las imágenes es un aspecto que se debe cuidar para que una página o cualquier tipo interfaz pueda cargarse de forma ágil.
- En general, siempre **debemos procurar que la imagen tenga la escala que se precisa y nunca cambiarle el tamaño con atributos de etiqueta **



Optimización de imágenes para la Web

Optimización de GIF

- Gracias a que el formato GIF utiliza la indexación, ya supone una optimización por sí misma. De hecho, si vamos a utilizar una imagen que no tenga muchos colores, basta con indicar ese conjunto en el índice de colores de ese GIF



Optimización de imágenes para la Web

Optimización de JPEG

- En el caso de tener que utilizar una imagen de tipo fotográfico, debemos ajustar la calidad del mismo. Con un programa de edición de imágenes podemos pasar del 100% a un 90%, 80% de calidad y conseguiremos reducir el tamaño de fichero considerablemente sin que esto afecte mucho a la nitidez de la imagen.



Optimización de imágenes para la Web

Generación de Sprites o mapas

- Además de la optimización de cada fichero por separado y de centrarnos en la optimización específica del formato, otro tipo de optimización que se puede aplicar es la de unir todos los ficheros de imagen de una web en una única imagen.



Optimización de imágenes para la Web

Generación de Sprites o mapas

- Gracias a eso el navegador no necesita hacer una solicitud para cada una de las imágenes que supone un peso muy considerable. Para poder mostrar imágenes que están metidas en un mismo fichero, debe utilizarse como fondo y mostrarse la parte que nos interesa aplicando propiedades de CSS.



Optimización de imágenes para la Web

Generación de Sprites o mapas

- Esta es una técnica similar a la que se utiliza en los videojuegos, donde las imágenes se consideran sprites y se extraen de una única imagen que contiene todos los elementos visuales del juego.



Optimización de imágenes para la Web

Generación de Sprites o mapas

- Podemos utilizar para generar estos mapas con:
 - **sprity**: típica herramienta de consola que podemos descargar y manejar a través de npm
 - **Compass**, un framework para generar CSS que funciona junto con SASS y que dispone de mecanismos para crear sprites: <http://compass-style.org/help/tutorials/spriting>



Actividad 4

Genera o descarga un sprite y aplícalo en tu web utilizando CSS.

¿Merece la pena? ¿Por qué?

Echa un vistazo a este enlace. ¿Qué opinas?

<https://kadumweb.com/crear-sprite-css-ejemplo-mejora-seo/>



Audio HTML5

- Los formatos de audio más conocidos son **mp3**, **wav**, **ogg** o **au**
- De todos que ellos **el que más conviene utilizar es mp3**, ya que sigue siendo el formato que mejor garantía ofrece.
- A continuación, veremos las características de cada formato:



Audio HTML5. Formatos

FORMATO	DESCRIPCIÓN
Mp3	<p>Es el más popular. Añade compresión a los ficheros para optimizar su tamaño y ofrecer una buena relación calidad/bytes.</p> <p>Se puede guardar en distintas opciones:</p> <ul style="list-style-type: none">• 128kbps• 192kbps• 320kbps
Ogg	<p>Formato abierto de audio y vídeo alternativo a Mp3</p>
WAV	<p>Formato de audio sin pérdida y sin compresión. Calidad perfecta pero tamaño excesivo.</p>



Audio HTML5. Inserción de contenidos

- Para insertar audio en una página web debemos utilizar **la nueva etiqueta**

<audio> que en su forma más básica tendría el siguiente aspecto:

```
<audio id="audio" preload="auto" tabindex="0" controls="" >  
  <source src="audio/prueba.mp3" type="audio/mp3">  
  <source src="audio/prueba.ogg" type="audio/ogg">  
</audio>
```



Audio HTML5. API de audio HTML5

- A continuación vemos el API de JavaScript para audio:

MÉTODOS	PROPIEDADES	EVENTOS
play()	currentSrc	play
pause()	currentTime	pause
load()	startTime(readonly)	progress
canPlayType()	duration (readonly)	error
	ended (readonly)	timeupdate
	error	ended
	paused (readonly)	abort
		empty
		emptied



Actividad 5

Agrega un archivo de audio a tu página web. Intenta reproducirlo.

¿Funciona? ¿Qué formato has puesto?



Vídeo HTML5

- Hasta la llegada de HTML5 una de las escasas formas de introducir un vídeo era mediante un reproductor FLASH
- La inserción de vídeo en HTML5 se ha estandarizado con una etiqueta obvia: **<video>**
- Podemos verla en el siguiente ejemplo:



Vídeo HTML5

```
<video id="video" controls="controls" >  
  <source src="video/prueba.mp4" type="video/mp4">  
  <source src="video/prueba.avi" type="video/ogg">  
  Su navegador no soporta el elemento video. Puede descargar los videos aquí  
  <a href="video/prueba.mp4">MP4</a>  
  <a href="video/prueba.ogg">OGG</a>  
</video>
```




Vídeo HTML5

- Debe tenerse en cuenta lo siguiente:
 - Se mostrarán unos controles básicos para manejar el vídeo: play, stop, volumen y progreso.
 - El navegador tratará de reproducir el primer fichero que pueda.
 - En caso de no poder reproducir ningún fichero mostrará el contenido alternativo, en este caso además de un mensaje se ofrecen los enlaces a los dos formatos de vídeo.
 - La etiqueta no deja de ser un elemento más del DOM, por lo tanto, se le pueden aplicar estilos.



Vídeo HTML5. Formatos

- Pese a que la etiqueta se llegó a estandarizar no ocurrió lo mismo con el formato de vídeo.
- Se sigue ofreciendo la posibilidad abierta de tener más de uno.
- Se puede convertir entre formatos utilizando **FFMPEG**
- A continuación, vemos los formatos y sus principales características:



Vídeo HTML5. Formatos

NOMBRE	DESCRIPCIÓN
MP4	Formato de vídeo con muy buen ratio de compresión/calidad
OGG	Formato de vídeo basado en un estándar libre y abierto
WEBM	Formato creado para la web



Vídeo HTML5. API de vídeo HTML5

MÉTODOS	PROPIEDADES	EVENTOS
play()	currentSrc	play
pause()	currentTime	pause
load()	startTime(readonly)	progress
canPlayType()	videoWidth	error
	videoHeight	timeupdate
	duration (readonly)	ended
	ended (readonly)	abort
	error	empty
	paused (readonly)	emptied
	muted	Waiting
	seeking	loadedmetadata
	volumen	
	height	
	width	
	seekable (readonly)	
	played (readonly)	



Actividad 6

Agrega un archivo de vídeo a tu página web. Intenta reproducirlo.

¿Funciona? ¿Qué formato has puesto?



Almacenamiento en HTML5

- Para almacenar información persistente a través del navegador, HTML5 introdujo una serie de mejoras y novedades
- La más llamativa es la **posibilidad de almacenamiento local en el cliente**, que viene en dos formas:
 - **Dos arrays relacionales con límite de 5Mb**
 - **Base de datos en el cliente (IndexedDB)**



Almacenamiento en HTML5

- ¿Para qué nos iba a interesar guardar datos en el cliente?
- Con HTML5 se aspira a crear aplicaciones y programas que funcionen incluso sin necesidad de conectarse a la red.
- La **posibilidad de guardar datos en el cliente** nos facilitaría el desarrollo de esas aplicaciones, en las que, por ejemplo, se **podrían guardar configuraciones básicas, perfiles de usuario, etc.**



Almacenamiento en HTML5. Arrays relacionales

- Dentro de JavaScript disponemos de esos **dos objetos** que básicamente no son más **que arrays relacionales o hashtables**, es decir, **arrays cuyos índices no son numéricos, sino palabras clave que no se pueden repetir** (también tienen métodos).



Almacenamiento en HTML5. Arrays relacionales

- La diferencia entre los dos es que lo que guardamos en **localStorage** está disponible a pesar de cerrar el navegador mientras que en el caso de **sessionStorage** no:

```
//Para guardar datos
localStorage["usuario"] = 'Martin Garcia';
sessionStorage["clave"] = '611254';

//Para recuperarlos
Console.log(localStorage["usuario"]);
```



Almacenamiento en HTML5. IndexedDB

- Desde la llegada de HTML5 se suponía que el nuevo estándar iba a soportar algún tipo de gestión de base de datos dentro del navegador.
- El modelo **IndexedDB** da paso a los **ObjectStores**, bases de datos que poco tienen que ver con el modelo relacional.



Almacenamiento en HTML5. IndexedDB

Diferencias entre relacional y NoSQL

- En las **BD no relacionales o NoSQL** tenemos algo parecido a tablas y registros, pero su **estructura es mucho más laxa**.
- **Cada uno de los registros de una tabla, a la que se suele llamar documento u objeto puede tener campos variables.**
- **Además, no existen las relaciones entre esos documentos**



Almacenamiento en HTML5. IndexedDB

Conexión y estilo

- La **forma de interactuar con IndexedDB es JavaScript** y, en el caso de los métodos en torno a IndexedDB, funcionan **de forma asíncrona**.
- Es decir, al dar la orden de abrir la BD el programa no se detiene para hacerlo, sino que deja una petición que debe ser recogida por un handler (onsuccess, onerror, etc).



Almacenamiento en HTML5. IndexedDB

Conexión y estilo

- Por otro lado, **operaciones como la de crear el ObjectStore sólo se pueden llevar a cabo dentro de onupgradeneeded**, de lo contrario te salta una excepción de illegal state.



Almacenamiento en HTML5. IndexedDB

Patrones básicos

- El patrón básico que IndexedDB propone es:
 1. **Abrir una base de datos.**
 2. **Crear un objeto de almacenamiento en la base de datos.**
 3. **Iniciar una transacción y hacer una petición** para hacer alguna operación de la base de datos, tal como añadir o recuperar datos.
 4. **Espere a que se complete la operación** de escucha de la clase correcta de eventos DOM.
 5. **Hacer algo con el resultado** (El cual puede ser encontrado en el objeto de la petición).



Almacenamiento en HTML5. IndexedDB

Abrir/Crear la BD

- Lo primero de todo es abrir o crear la BD, lo que conseguimos con este método:

```
var solicitud = window.indexedDB.open("BDMontecastelo");
```



Almacenamiento en HTML5. IndexedDB

Insertar

- Este método nos permite insertar datos:

```
function agregarObjeto(){
    var dni = document.getElementById("dni").value;
    var nombre = document.getElementById("nombre").value;
    var apellidos = document.getElementById("apellidos").value;

    var transaccion = bd.transaction(["alumnos"], "readwrite");

    var almacen = transaccion.objectStore("alumnos");

    var agregar = almacen.add({DNI: dni, NOMBRE: nombre, APELLIDOS: apellidos});
}
```




Almacenamiento en HTML5. IndexedDB

Lectura de un registro

- En los siguientes métodos vemos la forma más sencilla de leer registros:

```
function mostrar() {  
  
    zonadatos.innerHTML = "";  
  
    var transaccion = bd.transaction(["alumnos"], "readonly");  
  
    var almacen = transaccion.objectStore("alumnos");  
  
    var cursor = almacen.openCursor();  
  
    cursor.addEventListener("success", mostrarDatos, false);  
  
}
```



Almacenamiento en HTML5. IndexedDB

Lectura de un registro

- En los siguientes métodos vemos la forma más sencilla de leer registros:

```
function mostrarDatos(e) {  
  
    var cursor = e.target.result;  
  
    if (cursor) {  
  
        zonadatos.innerHTML += "<div>" + cursor.value.dni + " - "  
        + cursor.value.nombre + " - " + cursor.value.apellidos + "</div>";  
  
        cursor.continue();  
  
    }  
  
}
```



Almacenamiento en HTML5. IndexedDB

Modificación de datos

- Para actualizar un repositorio de datos:

```
function actualizarObjeto(){
    var dni = document.getElementById("dni").value;
    var nombre = document.getElementById("nombre").value;
    var apellidos = document.getElementById("apellidos").value;

    var transaccion = bd.transaction(["alumnos"], "readwrite");

    var almacen = transaccion.objectStore("alumnos");

    var actualizar = almacen.put({dni: dni, nombre: nombre, apellidos: apellidos});

    actualizar.addEventListener("success", mostrar, false);
}
```



Almacenamiento en HTML5. IndexedDB

Eliminación de un registro

- Para eliminar registros o documentos lo más sencillo es hacerlo a través del ID.

```
function borrarObjeto(){  
    var dni = document.getElementById("dni").value;  
  
    var transaccion = bd.transaction(["alumnos"], "readwrite");  
  
    var almacen = transaccion.objectStore("alumnos");  
  
    var actualizar = almacen.delete(dni);  
  
    actualizar.addEventListener("success", mostrar, false);  
  
}
```



Almacenamiento en HTML5. IndexedDB

Eliminación de todos los registros

- La eliminación completa de registros de una tabla no tiene mucho misterio, ya que se hace sin condiciones.

```
function vaciarBD() {  
    var transaccion = bd.transaction(["alumnos"], "readwrite");  
  
    var almacen = transaccion.objectStore("alumnos");  
  
    var borrar = almacen.clear();  
  
    borrar.addEventListener("success", mostrar, false);  
  
}
```



Almacenamiento en HTML5. IndexedDB

Eliminación de la BD

- Tenemos la opción de eliminar absolutamente toda la BD con un solo comando:

```
function eliminarBD() {  
    window.indexedDB.deleteDatabase("BDMontecastelo");  
}
```



Almacenamiento en HTML5. IndexedDB

Ejemplo completo

- Lógicamente, todos estos métodos deben ser llamados desde el HTML, con una estructura HTML correspondiente.
- Para el ejemplo, **hemos definido un método iniciar() que se llama cuando arranca la aplicación y que define todos los eventos que se deben activar.**



Almacenamiento en HTML5. IndexedDB

```
var bd;
window.addEventListener("load", iniciar, false);

function iniciar(){

    zonadatos = document.getElementById("zonadatos");
    botonGrabar = document.getElementById("grabar");
    botonActualizar = document.getElementById("actualizar");
    botonBorrar = document.getElementById("borrar");
    botonVaciarBD = document.getElementById("vaciar");
    botonEliminarBD = document.getElementById("eliminar");

    botonGrabar.addEventListener("click", agregarObjeto, false);
    botonActualizar.addEventListener("click", actualizarObjeto, false);
    botonBorrar.addEventListener("click", borrarObjeto, false);
    botonVaciarBD.addEventListener("click", vaciarBD, false);
    botonEliminarBD.addEventListener("click", eliminarBD, false);

    var solicitud = window.indexedDB.open("BDMontecastelo");

    solicitud.onsuccess = function (e) {
        bd = e.target.result;
    };
    solicitud.onupgradeneeded = function (e){
        bd = e.target.result;
        bd.createObjectStore("alumnos", {keyPath: "dni"});
    };
}
```




Actividad 7

Utilizando los métodos definidos previamente, crea tu propia aplicación HTML5 donde tengas una biblioteca, donde almacenas libros.

Cada libro tiene:

- *ISBN*
- *Título*
- *Año*

Crea las páginas necesarias para poder hacer todas las operaciones (Inserción, eliminación, actualización, borrado completo...)



Geolocalización

- HTML5 ofrece la posibilidad de que una página web pueda extraer la localización del usuario.
- La geolocalización de un usuario es un dato ciertamente privado y no se puede tomar sin más.
- El usuario debe dar su permiso para obtener la localización



Geolocalización

- Esto ha abierto las puertas a una serie de aplicaciones que giran en torno a la proximidad entre usuarios como Wallapop o Tinder.
- En caso de error, los posibles códigos que pueden obtenerse son:
 - **PERMISSION DENIED = 1**
 - **POSITION UNAVAILABLE = 2**
 - **TIMEOUT = 3**



Geolocalización. Ejemplo completo

- A continuación, vamos a ver un sencillo ejemplo con una página que no tiene más que un botón y un bloque div para mostrar el resultado.
- Debemos definir unas funciones en JavaScript que nos ayuden a localizar la posición.



Geolocalización. Ejemplo completo

- **Geo_consulta:** Nos ayudará a obtener la posición del usuario.

```
function geoConsulta(position) {  
    x.innerHTML = "Latitud: " + position.coords.latitude +  
        "<br>Longitud: " + position.coords.longitude;  
}
```



Geolocalización. Ejemplo completo

- **Manejo_errores:** Controla si el usuario permite el acceso a la ubicación o no.

```
function manejo_errores(error) {  
  switch (error.code) {  
    case error.PERMISSION_DENIED:  
      alert("El usuario no compartió su ubicación geográfica");  
      break;  
    case error.POSITION_UNAVAILABLE:  
      alert("No se pudo detectar la posición geográfica actual");  
      break;  
    case error.TIMEOUT:  
      alert("Se ha agotado el tiempo de espera al consultar posición geográfica");  
      break;  
    default:  
      alert("Error desconocido");  
      break;  
  }  
}
```



Geolocalización. Ejemplo completo

- **Iniciar_geolocalizacion:** Utilizando las dos funciones anteriores, obtenemos la posición actual

```
function getLocation() {  
    if (navigator.geolocation) {  
        navigator.geolocation.getCurrentPosition(geoConsulta, manejo_errores);  
    } else {  
        x.innerHTML = "La geolocalización no está soportada en este navegador.";  
    }  
}
```



Geolocalización. Ejemplo completo

- En el HTML tendremos el siguiente elemento que nos permitirá mostrar la latitud y longitud obtenida:

```
<p id="Posicion"></p>
```

- En JavaScript lo referenciaremos y le añadiremos la latitud y longitud:

```
var x = document.getElementById("Posicion");
```




Actividad 8

Utilizando los métodos definidos previamente, júntalos en una misma aplicación y ponla en funcionamiento.

- *¿Solicita los permisos? Si no se los concedes, ¿qué sucede?*
 - *¿Obtiene la posición correctamente?*

Tema 5: Implantación de contenido multimedia



Ciclo Superior DAW

Asignatura: Diseño de interfaces

Curso 20/21