

Implementación de una aplicación web

Free Images - Aplicación para ver y compartir fotos en línea



Ibai Lacasa
Daniel Ruskov
SAR 2019/20 C01

Índice

Índice	1
Introducción	2
Estructura principal - descripción	3
Conclusiones y posibles mejoras	5

Introducción

En esta práctica hemos implementado una aplicación web en la cual podemos ver imágenes en línea. Tenemos la opción de ver imágenes subidas por diferentes usuarios, o subir nosotros una imagen. Cada imagen tiene título y descripción, siendo las dos necesarias a la hora de publicar una imagen. También podemos buscar imágenes mediante un filtro de título y/o descripción. Nuestra aplicación permite la existencia de diferentes fotos con el mismo título y/o descripción.

El sitio web se estructura en tres páginas: Inicio, Insertar imagen y Ver imágenes. En el inicio se encuentra una portada representativa de la aplicación web. En Insertar imagen podemos encontrar un formulario mediante el cual tendremos la posibilidad de agregar nuestra imagen a la aplicación. En la página de Ver imágenes se puede realizar la búsqueda anteriormente mencionada - mediante el título y/o descripción. Las tres páginas Tienen el mismo título, descripción de la aplicación y un menú para poder moverse entre las tres partes que componen esta aplicación.

Cabe mencionar que en esta aplicación no existe la oportunidad de registrarse y/o iniciar sesión, ya que no se hace uso de usuarios para poder navegar en ella.



Estructura principal - descripción

El conjunto de archivos que componen esta aplicación está repartido en diferentes carpetas con nombre representativo del tipo de archivos que contiene cada una. A continuación explicaremos que contiene cada carpeta y que función cumple cada archivo de la página web.

❖ **/html** - Contiene los ficheros base html que componen la aplicación. Entre ellos esta nos siguientes elementos:

➤ **/html/index.html** - En este fichero encontramos la portada de la aplicación. En ella está el logo, menú de navegación, y dos tablas con los nombres de los autores y una imagen seleccionada por cada uno.

❖ **/img** - Contiene las imágenes base de la portada index.html

➤ **/img/logo** - Es la imagen logotipo de nuestra aplicación.

➤ **/img/Rana.jpg** y **/img/smile.png** - Son las fotos seleccionadas por los autores para las tablas de la portada.

❖ **/css**

➤ **/css/style.css** - Contiene el estilo de todas las páginas que componen la aplicación web. En ella se hace referencia a los diferentes elementos mediante las etiquetas, clases o identificadores de componentes de html.

❖ **/php**

➤ **/php/InsertarImagen.php** - En este archivo lo que hemos definido es un formulario sin uso de ajax el cual contiene dos campos (Nombre y descripción) además de un campo para añadir una imagen y los botones de insertar y limpiar.

En caso de pulsar el botón insertar, este mediante el action del formulario ejecuta el php "Insertar.php". Este revisa que en el formulario se hayan rellenado todos los campos, aportando feedback para los casos en los que algún parámetro falta y aportando otro tipo de feedback en caso de que todos los parámetro estuvieran llenos y se insertarán los datos en el fichero "Images.xml".

➤ **/php/Insertar.php** - Este archivo es el que se ocupa de insertar en el fichero xml como ya hemos explicado antes, pero lo que tiene interesante es cómo hace para insertar las imágenes en el xml.

Para poder hacer esto lo que hacemos es transformar en bits la imagen y añadirla de esta forma al fichero xml, que luego para poder visualizarlas las hemos de dejar de la forma en la que se encontraban originalmente.

Para codificarlo lo que usamos es "base64.encode(...)" y para decodificarlo

lo que usamos dentro de las propiedades de la etiqueta img es (src="data:image/*;base64, 'Aquí va la imagen' ").

- **/php/VerImagen.php** - En este archivo lo que hacemos es añadir un formulario (Sin action ni method) el cual utilizamos para realizar el filtro de imágenes.
Este contiene dos botones, de los cuales "Ver todas las imágenes" nos muestra todas las imágenes que tiene el fichero xml sin filtrar y el botón "Ver imágenes filtradas" en cambio nos muestra todas las imágenes la cuales tengan el mismo título y descripción especificados por el usuario en el formulario. Para que esto tenga sentido, los nombres y descripciones se muestran con una select.
- **/php/Mostralmagenes.php** - Este php lo llamamos desde la función "Mostrar" del fichero javascript "VerTodas.js". Este es el cual se encarga de sacar todas las imágenes del fichero xml.
- **/php/MostralmagenesFiltradas.php** - Este php lo llamamos desde la función "MostrarFiltradas" del fichero javascript "VerFiltradas.js". Este es el cual se encarga de sacar todas las imágenes del fichero xml pero mediante un filtro, el cual no permite que el nombre y descripción del formulario de "VerImagenen" estén vacíos y también avisa de si no existen imágenes con el nombre y descripción seleccionados (se puede filtrar solo por nombre y solo por descripción).

◆ /xml

- **/xml/Images.xml** - El fichero utilizado para guardar todas las imágenes que se insertan y para visualizar de distintas maneras dichas imágenes.

En este hemos definido una estructura del estilo a:

```
<assessmentItems>
  <assessmentItem name="..." descripcion="...">
    <itemBody>
      <imagen>...</imagen>
    </itemBody>
  </assessmentItem>
</assessmentItems>
```

En esta estructura como bien se puede intuir guardamos los nombres y descripciones de cada imagen en los atributos de cada "<assessmentItem>" y dentro de cada "<itemBody>" guardamos las imágenes en las etiquetas "<imagen>".

◆ /js

- **/js/jquery-3.4.1.min.js** - Este script es para poder utilizar las librerías de jquery y lo hemos añadido ya que lo utilizamos en "ShowImagenInForm.js".

- **/js/ShowImagenInForm.js** - Este script es un script muy sencillo el cual vale para a la hora de añadir una imagen mediante el selector en el fichero "InsertarImagen.php" esta se muestre al final del formulario. Mediante esto el usuario puede visualizar la imagen que ha seleccionado antes de insertarla.
- **/js/VerFiltradas.js** - Este script junto con el siguiente tiene comportamiento ajax. Este en específico es utilizado para hacerle llegar al php "MostrarmagenesFiltradas.php" los datos que este necesita para poder mandarle de vuelta a "VerFiltradas.js" las imágenes ya filtradas y desde este mostrarlas en el php "VerImagen.php".
- **/js/VerTodas.js** - Este tiene el mismo funcionamiento que el anterior solo que en vez de llamar a "MostrarmagenesFiltradas.php" llama a "Mostrarmagenes.php" y este devuelve al script sin filtrar todas las imágenes del fichero xml.

Conclusiones y posibles mejoras

Como posible mejora se nos ocurrió el añadir un login y registro para además de lo que ya guardamos se guardará el nombre/correo de la persona que añadió la imagen (Esto también permitiría mostrar al usuario solamente las imágenes que este ha subido).

No hemos realizado esa implementación por falta de tiempo.