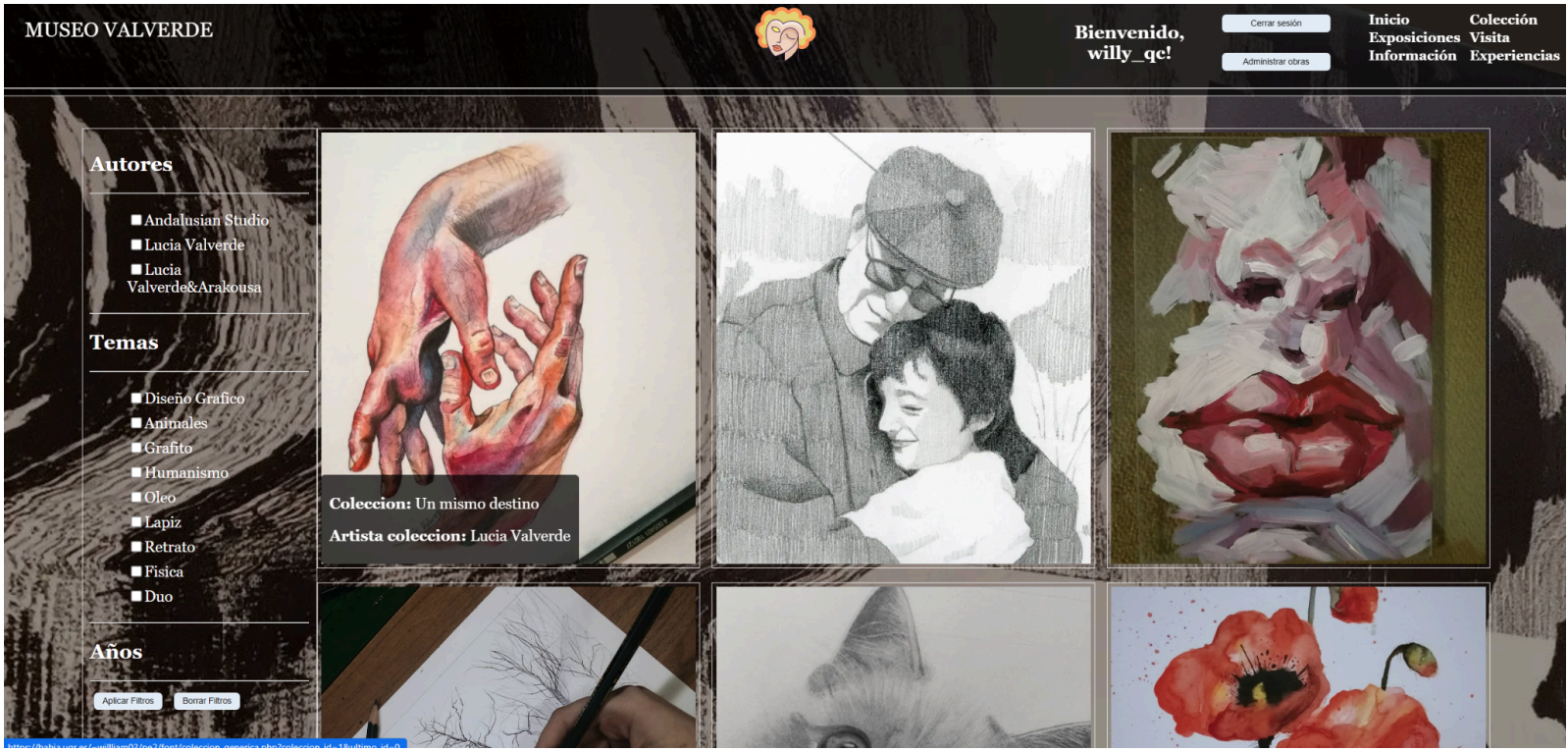
2023/2024

Memoria Práctica 2

Programación Web 3ºA2

William Patrick Quinn Cortes



Índice

Introducción....3
PW-PII....4
Paso 0: Base de datos....4
Paso 1: Dar de Alta Usuarios....5
Paso 2: Sesión iniciada....6
Paso 3: Mensaje Bienvenida....6
Paso 4: Administrar Obras....7
Paso 5: Colecciones Dinámicas....8
Menú....8
Colecciones.php....8
Coleccion_generica.php....9
Paso 6: Comentarios....10
Paso 7: Validación con JavaScript....11
Paso 8: PopUP....12
Mejoras Visuales....13
Botones Principales....13
Página Principal....14
Registro de Usuario....15
Colección....16
Experiencias....17
Administrar Obras....18
Elementos Extras....19
Borrar Comentarios....19
Menú administrador....20
Carrusel Obras....21
Alertas Modificadas....22

Introducción

Para esta práctica usaré otro enfoque para explicar la memoria ya que la anterior se me quedó con poco contenido. Iré explicando punto por punto los contenidos que venían en el guión de la práctica 2 con sus implementaciones y archivos nuevos añadidos y luego posteriormente tendré explicare en un apartado las innovaciones y mejoras respecto a la anterior práctica.

Primeramente todos los archivos los he transformado a .php para que se guarde el inicio de sesión a lo largo de todo el menú.

También presentó archivos javascript para validar el contenido, hay trozos donde lo compruebo directamente con javascript en el php ya que el contenido a validar era ínfimo.

Por último hay que añadir que el archivo colección la he cambiado un poco, es decir, tengo una colección donde hay 9 colecciones a elegir (coleccion.php) y posteriormente dentro de cada una las distintas obras (coleccion_especifica.php).

PW-PII

Paso 0: Base de datos

Primero mostraré las tablas que tengo en mi base de datos ya que desde ellas se guarda todo el contenido y se obtiene.

Comentarios (`id` int(11) NOT NULL AUTO_INCREMENT, `usuario` varchar(50) DEFAULT NULL, `comentario` text NOT NULL, PRIMARY KEY (`id`), KEY `usuario` (`usuario`));	**Colecciones** (`id` int(11) NOT NULL AUTO_INCREMENT, `titulo` varchar(255) NOT NULL, `ruta` varchar(255) NOT NULL, `fecha` date NOT NULL, `autor` varchar(255) NOT NULL, PRIMARY KEY (`id`));
Usuarios (`id` int(11) NOT NULL AUTO_INCREMENT, `nombre` varchar(50) NOT NULL, `usuario` varchar(50) NOT NULL, `contrasena` varchar(255) NOT NULL, `email` varchar(255) NOT NULL, `telefono` varchar(15) DEFAULT NULL, PRIMARY KEY (`id`), UNIQUE KEY `usuario` (`usuario`), UNIQUE KEY `email` (`email`));	**Obras** (`id` int(11) NOT NULL AUTO_INCREMENT, `titulo` varchar(255) NOT NULL, `autor` varchar(255) NOT NULL, `fecha` date NOT NULL, `descripcion` text DEFAULT NULL, `ruta` varchar(255) NOT NULL, `coleccion_id` int(11) DEFAULT NULL, `anio` varchar(50) DEFAULT NULL, `tema` varchar(255) DEFAULT NULL, PRIMARY KEY (`id`));

Paso 1: Dar de Alta Usuarios

Para este paso reorganice mi pagina de dar de alta y la puse más llamativa. También tengo tres archivos para esto:

1. **altausuario.php**: Es donde se implementa el diseño html.
2. **registro.php**: Formulario el cual lleva el tema de base de datos.
3. **altausuario_validation.js**: Archivo javascript para confirmar que los campos se escriben bien.

Dentro de este apartado los usuarios se pueden registrar rellenando todos los campos y además que su nombre de usuario sea diferente en caso de que ya exista uno con el mismo nombre. También la fecha se debe escribir conforme al formato *AÑO-MES-DIA* y el número de teléfono contenga solo 9 números sin separaciones, además de que el email sea un correo válido.

Paso 2: Sesión iniciada

Para este apartado el usuario puede iniciar sesión desde cualquier archivo disponible en el museo, para ello dentro de cada archivo mantenga el atributo de *session_start()*, también para este apartado se mantiene 3 archivos:

1. **login.php**: Mantiene la logística para iniciar sesión y notificarlo.
2. **logout.php**: Mantiene la logística para cerrar sesión tras darle al botón.
3. **iniciarsesion_validation.js**: Mantiene la comprobación de que los campos estén completos.

Tras iniciar sesion, si hay algún error se notificará al usuario mediante un tipo de alerta y en caso de iniciar sesion de manera exitosa se redirige a la pagina principal y ya le saldrá un mensaje de bienvenido usuario.

Paso 3: Mensaje Bienvenida

Como he mencionado anteriormente se le muestra un mensaje de bienvenida y además un botón de cerrar sesión. Para saber si mostrar la parte de iniciar sesión o la sesión iniciada tengo un *if* con php para manejar dicha lógica.

Paso 4: Administrar Obras

Para administrar obras, si nos vamos a algún apartado de colección, siendo administrador, esto se comprueba si el nombre de usuario es “admin” (solo puede haber 1), nos aparecerá un botón de administrar obra, este nos redirige a la página correspondiente. Para esta parte se basa en los siguientes archivos:

1. **administrador_obras.php**: Donde se almacenan los 3 menús y se despliegan las respectivas alertas
2. **delete_obra.php**: Mantiene la logística para el borrado de obras en la base de datos
3. **add_obra.php**: Mantiene la logística para la inserción de obras en la base de datos
4. **modify_obra.php**: Mantiene la logística para el modificado de obras en la base de datos
5. **administrador_obras_validation.js**: Contiene las comprobaciones para *add_obra.ph*p y *modify_obra.php*, para añadir una obra todos los campos deben de estar completos, para modificar obra la única restricción es que la fecha se escriba conforme un estilo.

Paso 5: Colecciones Dinámicas

En este apartado hablaré de 3 cosas, el menú, colecciones.php y coleccion_generica.php.

Menú

Para realizar el menú y que los filtros funcionen he usado checkbox. Se puede seleccionar un filtro de cada apartado, Autores, Temas, Años, si se selecciona más de un apartado solo se estima el primero seleccionado. Tras ser seleccionado lo que hago es añadir dicho filtro si no está vacío a la url, así al cargar el archivo se muestran las obras según el filtro. También está el botón de limpiar filtro que elimina todos los autores etc y devuelve al usuario a coleccion_generica.php pero con todas las obras del museo.

Colecciones.php

En este archivo se encuentra tanto el menú como las 9 colecciones que posee el museo, este archivo sigue manteniendo los pop-ups pedidos en el guión y tras pinchar en la imagen de una colección le mostrará las obras que pertenecen a ella.

Coleccion_generica.php

Este es el archivo más grande, tras seleccionar una colección le muestra las obras, el menú y los pop-ups. Para llevar a cabo esto tras seleccionar una colección se envía en el url el id de la colección para así saber qué obras mostrar. También para cuando hay más obras de 9 lo que hago es que siempre guardo el último id de la obra que se está mostrando, al darle al botón de siguiente página se muestra las 9 siguientes obras, o menos, que siguen a esa. También mantengo durante todo el movimiento de páginas, el id de colección o si no hay colección y es mediante filtros, los filtros correspondientes. Si no hay más obras disponibles se muestra un mensaje identificativo. Por último, tras seleccionar una obra en colección genérica, te enviará a *detalle_obra.php* donde contiene toda la información relativa a la obra.

Paso 6: Comentarios

Para este apartado he rediseñado un poco la página de comentarios poniéndola más bonita, luego he añadido el apartado para que los usuarios registrados puedan escribir sus comentarios. En dicho comentario aparece el texto y el usuario que lo ha realizado. Se emplean los siguientes archivos:

1. **experiencias.php**: Contiene tanto el html como la logística para obtener todos los comentarios de la base de datos y mostrarlos.
2. **registrarcomentario.php**: Alberga las operaciones para registrar un comentario en la base de datos.
3. **delete_comentario.php**: Contiene las operaciones para borrar un comentario en la base de datos.

He añadido nueva funcionalidad exclusiva del administrador y también la validación con javascript se hace dentro de *experiencias.php.*

Paso 7: Validación con JavaScript

Como he ido comentando anteriormente he usado 3 archivos Javascript para validar el contenido **administrador_obras_validation.js, iniciarsesion_validation.js, altausuario_validation.js**, también hay pequeñas partes de código para validar el contenido que al ser simplemente poner un atributo en solo lectura por ejemplo, la implementó en el propio *php.* El enfoque que he seguido para validar ha sido comprobar que los campos no estén vacíos, en caso de estar vacíos mostrar una alerta o un mensaje debajo del campo vacío, tras ser rellenado se borra el mensaje de error.

También otros apartados como que la fecha o el teléfono tenga un formato concreto o incluso el límite de palabras para algunos apartados como el de comentarios.

Paso 8: PopUP

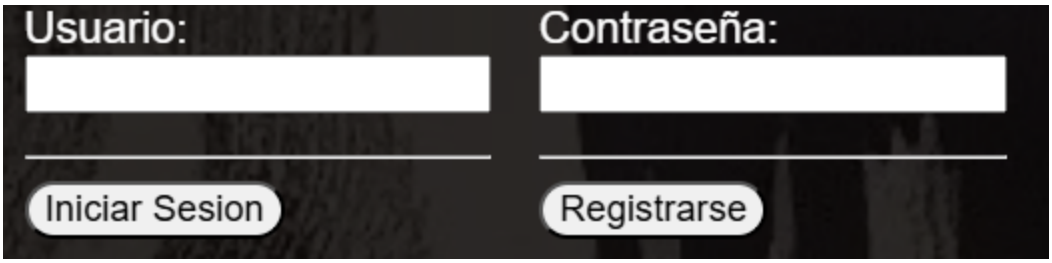
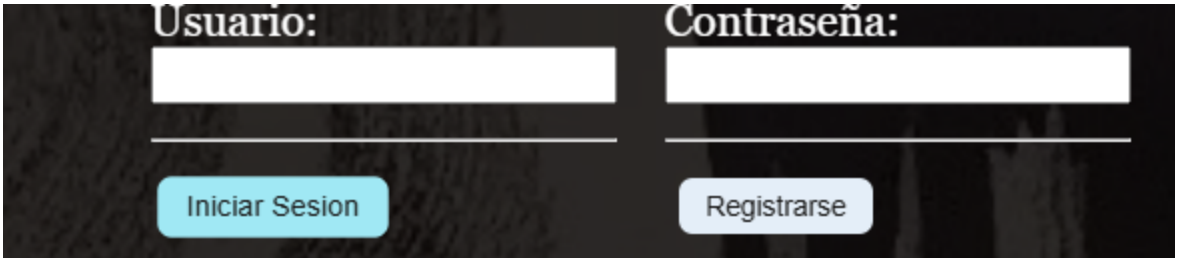
En este apartado para mostrar el pop-up está dividido según si estamos en *coleccion.php* o *coleccion_generica.php*. Si nos encontramos en colección general, en dicho pop-up se mostrará el nombre de la colección y la artista que lo ha realizado, si ya nos adentramos dentro de la colección en dicho pop-up se muestra el título, tema y autor. Su diseño es un cuadrado perfilado negro translucido, posicionado en la esquina inferior izquierda, ademas de con una animacion para su despliegue.

Para realizar este pop-up con javascript me he apoyado en funciones para crear la ventana emergente y que se muestre al pasar el ratón por la imagen. También hubo problemas como que la información de la imagen la tuve que guardar en un data-obra, es decir al generar las imágenes con php se guarda en cada una su información ***< data-obra='<?php echo json_encode($coleccion) >*** luego esto es recogido por mi función javascript para crear con innerHTML dicha ventana.

Mejoras Visuales

En esta sección iré mostrando con comparativa de imágenes las distintas mejoras al proyecto.

Botones Principales

Museo V.1

Museo V.2


Página Principal

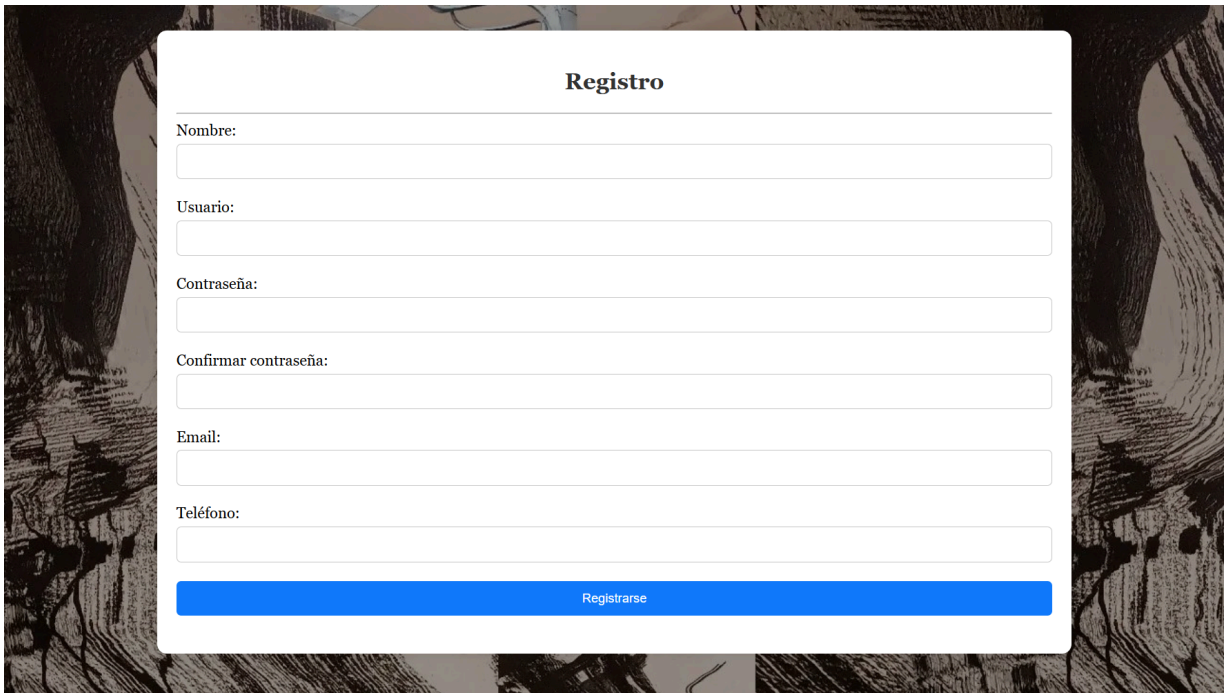
Museo V.1

Museo V.2

Registro de Usuario

Museo V.1

Museo V.2



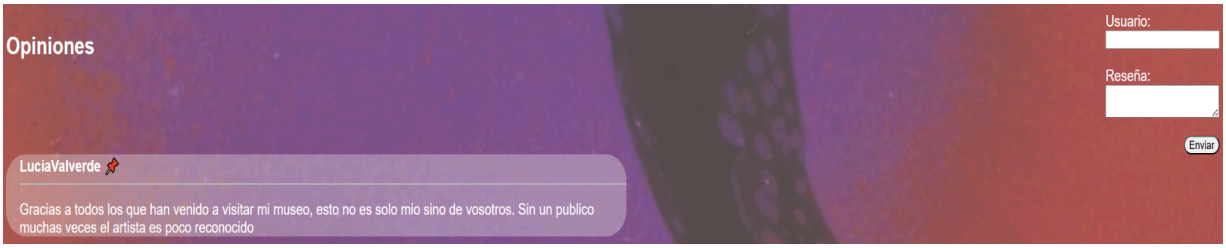
Colección

Museo V.1

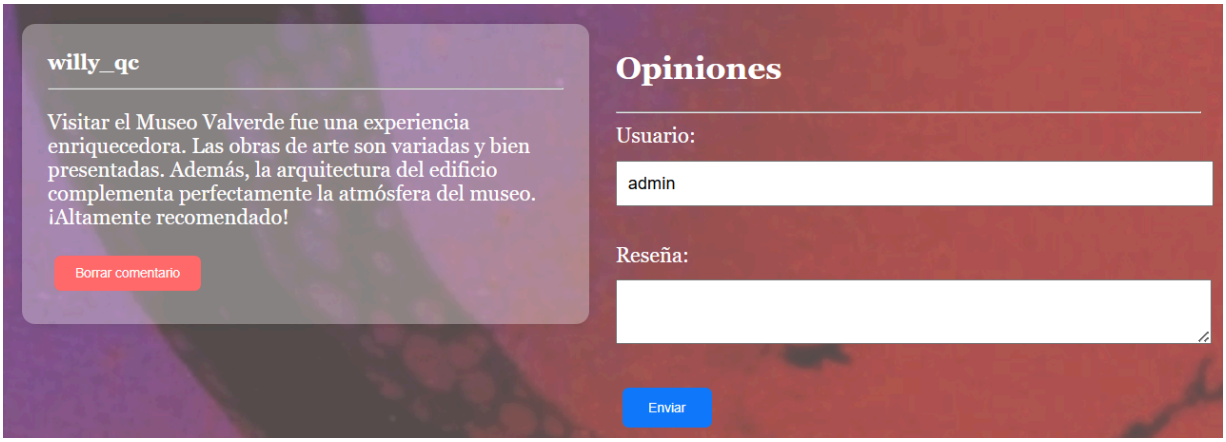
Museo V.2

Experiencias

Museo V.1



Museo V.1



Administrar Obras

Museo V.1
No existe
Museo V.2

Elementos Extras

Borrar Comentarios

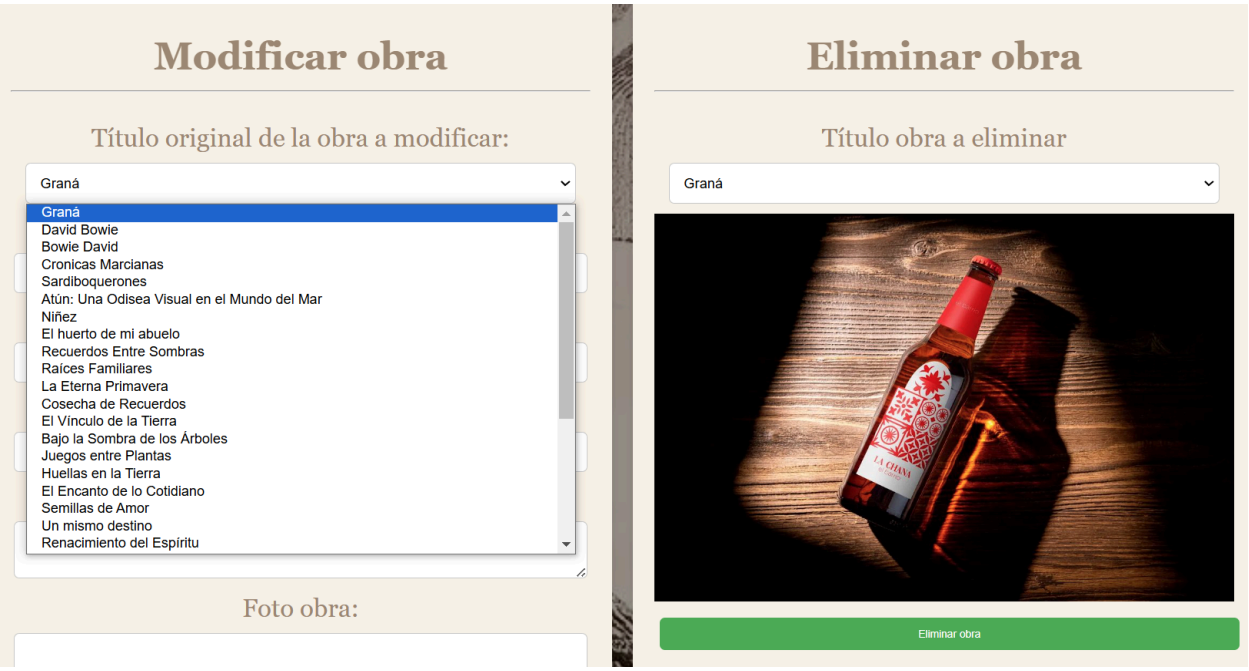
En esta parte he incluido una sección en la que si nos situamos en experiencias e iniciando sesión como administrador, debajo de cada comentario aparece un botón para borrar dicho comentario de la base de datos.

Menú administrador

En la parte de Colección tras ser "*admin*" puedes administrar dichas obras, pues en este apartado además de lo que pedía he añadido alguna funcionalidad con JavaScript y PHP.

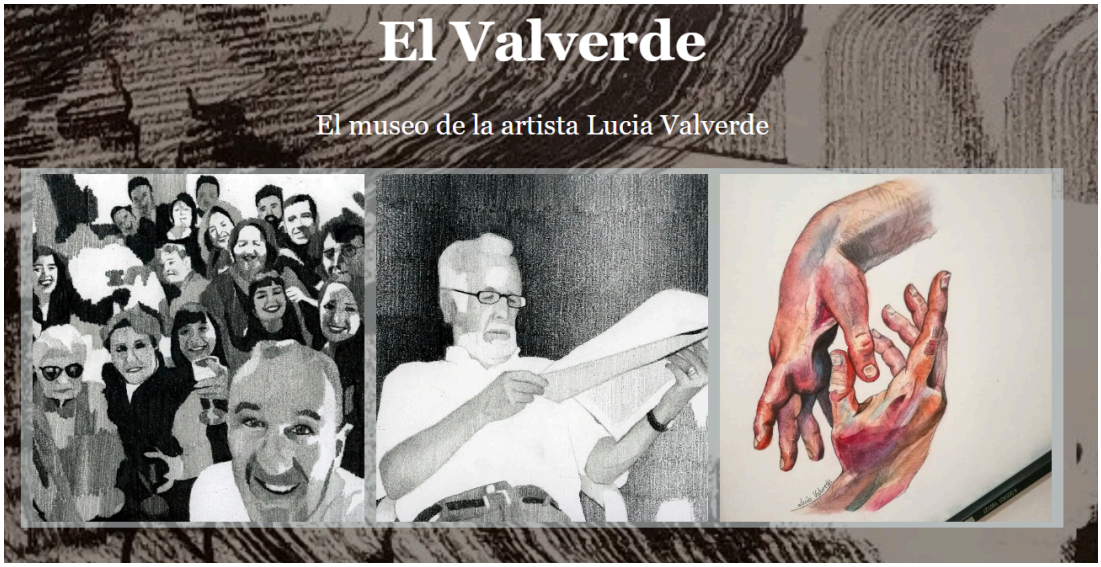
Con php hago que ciertos menús sean *Select* y recojo dicho contenido de la base de datos. Por ejemplo en modificar obra, para saber que obra modificar muestro todos los títulos de las obras y solo se puede seleccionar uno, así controlo que la obra ya exista.

Luego otro punto seria que por ejemplo en borrar obra, tras seleccionar el título con JavaScript he hecho que te aparezca abajo la imagen de la obra a borrar.



Carrusel Obras

En la página inicial del museo he añadido una sección donde cada 3 segundos se cambian 3 obras que se van mostrando, simulando así como un carrusel. Para ello seleccionó aleatoriamente las imágenes de las obras en mi base de datos con JavaScript y las voy mostrando sucesivamente, mantengo controles para que siempre se muestran 3 imágenes.



3 segundos después →

Alertas Modificadas

Un último punto que añadí fue la biblioteca de sweetalert, esta la utilizo para mostrar ciertas alertas en caso de realizar algo correctamente o erróneamente, aquí muestro algunos ejemplos.