



UNIVERSIDAD DE GRANADA

Escuela Técnica Superior de Ingeniería Informática y Telecomunicaciones

Práctica 2

Programación Web

Mario Illán Valero

Índice

1. Permitir que el público general se dé de alta. (Registrarse)	3
2. Iniciar Sesión	3
3. Gestión Obras Administrador	3
4. Visualización colección.php	3
5. Permitir que los usuarios de público en general puedan realizar comentarios (experiencias.php)	3
6. Validar formularios en JavaScript	3
7. Ventana Emergente Obra	3
8. Aspecto Innovadores	3

1. Permitir que el público general se dé de alta. (Registrarse)

El objetivo es que cualquier persona sea capaz de registrarse en el sistema. Para ello, he creado un formulario de registro para llenar con datos (nombre, apellidos, usuario, contraseña, correo, fecha de nacimiento y número de teléfono).

Los archivos involucrados en esta parte son:

- altausuarios.php
- registro.php
- validacion_registro.php

1.1 altausuarios.php

Es donde se ubica el formulario para ingresar la información (nombre, apellidos, usuario, contraseña, correo, fecha de nacimiento y número de teléfono).

Datos de Registro

Nombre

Apellidos

Usuario

Correo Electrónico

Nueva Contraseña

dd/mm/aaaa

Teléfono

Registrarse

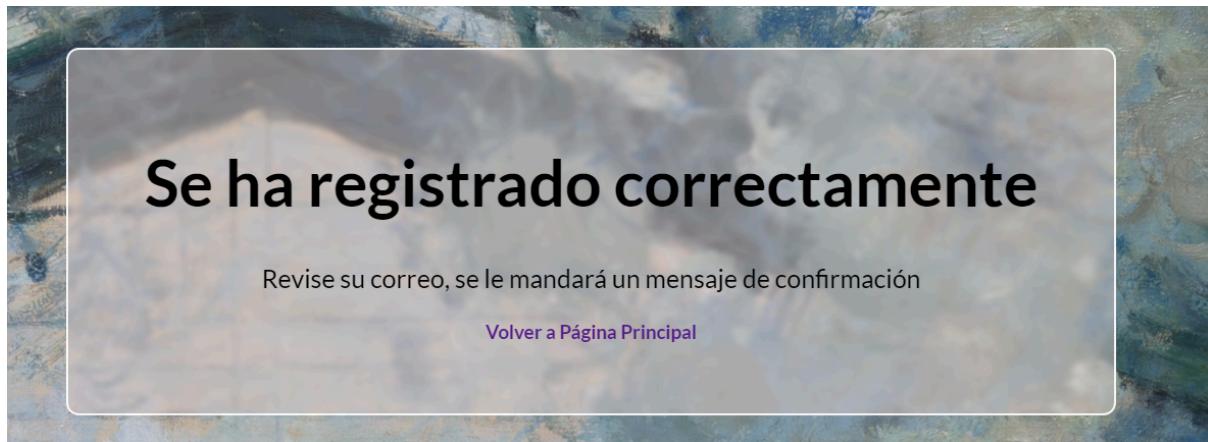
1.2 registro.php

El archivo registro.php se encarga de almacenar los datos en la base de datos. Se verifica que no haya duplicados en la base de datos para los campos de usuario, correo y teléfono. También se verifica que la fecha de nacimiento introducida sea inferior a la fecha actual. Después de este proceso se crea una nueva columna en la tabla usuarios con la información del usuario que ya podrá iniciar sesión con su nuevo usuario y contraseña.

MariaDB [dbmarioillan_pw2324]> select * from usuarios;					
id nombre	apellidos	usuario	correo	contrasenia	fecha_nacimiento telefono
1 Mario	Illán Valero	marioillan	marioillan18@gmail.com	\$2y\$10\$gCUDk/mYtg1UZVmSKPMkY.OHkvnu.R2x656GPtXjy.KqGPUTar/CS	2001-07-30 639468882
9 Antonio Jesús	García García	antoniojgg	antonio@gmail.com	\$2y\$10\$1w2aS60FrwMa0zcx/cxr0nDuQ/w61nDaeGBScpZCrErA7aR/HOMC	2015-01-06 634213456
2 rows in set (0,000 sec)					

1.3 validacion_registro.php

El archivo validacion_registro.php se encarga de mostrar un mensaje de confirmación del proceso de registro, una vez que éste se haya registrado correctamente. Te permite redirigir a index.php.



2. Iniciar Sesión

El objetivo es que los usuarios registrados y el administrador del sistema puedan iniciar sesión y tenerla activa hasta que cedican cerrarla mediante el cierre de sesión implementado.

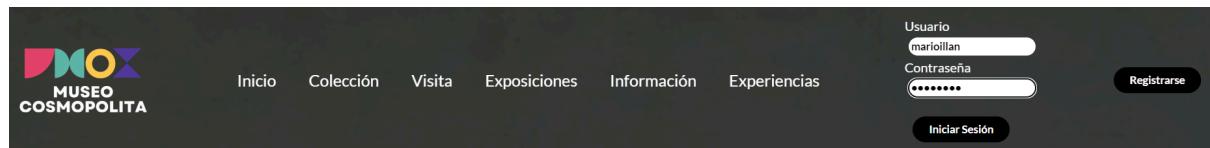
Los archivos involucrados en esta parte son:

- login.php
- logout.php
- Headers de los php de la Web

2.1 Headers de los phps

En los headers de los php de la Web se incluye un formulario para que los usuarios y el administrador ingresen su usuario y contraseña. Este formulario envía los datos a login.php para su validación.

La sesión permanecerá activa en todas las páginas (session_start();) hasta que el usuario decida cerrar la sesión.



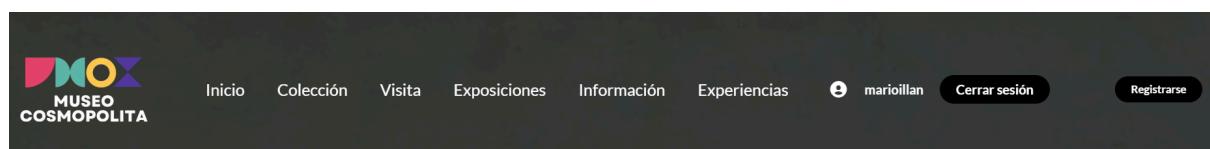
2.2 login.php

El archivo login.php se encarga de procesar los datos enviados del formulario (usuario y contraseña). Valida comparándolas con las almacenadas en la base de datos y si las credenciales son correctas se inicia sesión y redirige a los usuarios a index.php dependiendo de si es un usuario o un administrador.

Los administradores iniciaran sesión con usuario: admin y contraseña:admin y los usuarios con sus datos que introdujeron en el formulario de registro en la tabla usuarios.

2.3 logout.php

El archivo logout.php se encarga de cerrar sesión del usuario que tiene la sesión ya iniciada. El archivo simplemente destruye la sesión del usuario y lo redirige a index.php



3. Gestión Obras Administrador

Permite que el Administrador de la Web gestione las obras presentes (creando, modificando y eliminando las obras).

Los archivos involucrados en esta parte son:

- colección.php
- formularioObras.php

Estos archivos permiten al administrador interactuar con la base de datos para gestionar las obras, asegurando que las imágenes de estos se encuentren ya en el directorio del servidor.

3.1 colección.php

En este archivo si la sesión que está activa es de admin se podrá visualizar un enlace que te lleva a formularioObras.php en donde podrá crear, modificar o eliminar obras.

3.2 formularioObras.php

Este archivo contiene el formulario que permite al Administrador crear, modificar y eliminar obras. El formulario según lo que quieras hacer te pide una serie de datos y realiza las acciones correspondientes en la base de datos.

Para crear:

The screenshot shows a web form titled "Gestión Obras" (Work Management) with a sub-section "Añadir Obra" (Add Work). A descriptive message "Introduzca los datos de la obra que quiera añadir." (Enter the data of the work you want to add.) is displayed above the form fields. The form consists of five input fields labeled "Título:", "Autor:", "Siglo:", "Tema:", and "Imagen:". Below the input fields is a black "Añadir" (Add) button.

El formulario recoge los datos de la nueva obra y envía una solicitud POST con la acción “añadir” y el servidor inserta los datos en la base de datos rellenando una nueva columna en la tabla obras.

Para modificar:

Editar Obra

Introduzca los datos de la obra que quiera editar.

Identificador de la obra:

Título:

Autor:

Siglo:

Tema:

Imagen:

Editar

El formulario recoge el identificador de la obra y los nuevos datos, envía una solicitud POST con la acción “editar” y el servidor actualiza los datos de la obra correspondiente en la base de datos.

Para eliminar:

Eliminar Obra

Introduzca el ID de la obra que quiera borrar.

Identificador de la obra:

Eliminar

El formulario recoge el identificador de la obra que quiera eliminar y envía una solicitud POST con la acción “eliminar” y el servidor elimina la obra de la base de datos.

4. Visualización colección.php

Para la implementación de la visualización dinámica de las colecciones del museo he desarrollado un sistema que permite mostrar las obras siguiendo un filtro (autor, época y temática) gestionando la páginación haciendo que haya 9 obras como máximo en cada página listandolas en filas de 3 como pide la práctica.

El archivo involucrado es:

- colección.php

4.1 colección.php

Este archivo muestra dinámicamente las obras según los filtros seleccionados del menú de la izquierda que se genera de manera dinámica para incluir los valores en los campos: época, autor y temática. La paginación se gestiona para mostrar 9 obras por página.

Gestión Obras

Colecciones

Temas

Grecia Antigua
Katsukawa
Corrientes Africanas
Europa Moderna
Américas
India

Autor

Artista Desconocido
Katsukawa Shunsho
Georges Seurat
Thomas Cole

Época

400-370 BCE
Japón, 1726-1792
Mitad de siglo 19
Siglo XVI

Todas las obras

Fish Plate

Temática: Grecia Antigua
Autor: Artista Desconocido
Época: 400-370 BCE

"Silkworm Cultivation (Kaiko yashinai gusa)"

Temática: Katsukawa
Autor: Katsukawa Shunsho
Época: Japón, 1726-1792

Male Figure (Nkisi Nkondi)

Temática: Corrientes Africanas
Autor: Artista Desconocido

1 2 3

Obtención de filtros dinámicos

La función getFiltro obtiene los valores únicos de los campos época, temática y autor desde la base de datos.

Los filtros se generan dinámicamente en el menú de la izquierda junto con un enlace “Todas las obras” que te redirige a colección.php

Paginación

Se calcula el número total de páginas basado en el número total de obras y el límite de obras por página.

Las obras se visualizan según el filtro aplicado y el número de página actual.

Mostrar Obras

Las obras quedan puestas en un bloque principal con su título, imagen, temática, autor y época. Estos valores estarán llenos en la tabla obras en la base de datos, Las imágenes de cada obra se encuentran en un directorio del servidor.

Si puede acceder a la obra pinchando sobre la imagen redirigiendo a detalle_obra.php

5. Permitir que los usuarios de público en general puedan realizar comentarios (experiencias.php)

Implementé la funcionalidad para que los usuarios registrados en el sistema puedan realizar valoraciones y visualizar los otros comentarios de los usuarios.

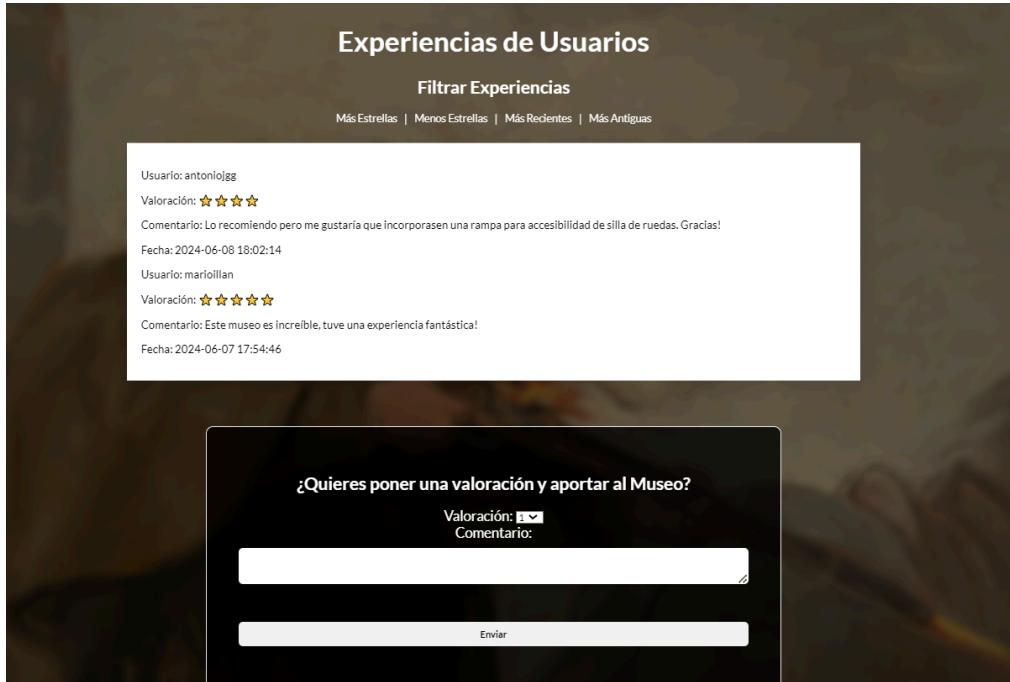
Los archivos involucrados son:

- experiencias.php
- procesar_experiencia.php

5.1 experiencias.php

En este archivo se muestran todas las valoraciones existentes y proporciona un formulario para que los usuarios registrados y con la sesión iniciada puedan añadir comentarios nuevos. Además he implementado un filtro para mejorar la visualización de estos comentarios pero eso lo explicaré mejor en los aspectos innovadores.

Si no hay ninguna sesión de usuario iniciada salta un mensaje en vez del formulario pidiendo que inicie sesión o que se registre pinchando un enlace en el que se redirigirá a altausuarios.php



Experiencias de Usuarios

Filtrar Experiencias

Más Estrellas | Menos Estrellas | Más Recientes | Más Antiguas

Usuario: antoniojgg
Valoración: ★ ★ ★
Comentario: Lo recomiendo pero me gustaría que incorporasen una rampa para accesibilidad de silla de ruedas. Gracias!
Fecha: 2024-06-08 18:02:14

Usuario: marlolilan
Valoración: ★ ★ ★ ★
Comentario: Este museo es increíble, tuve una experiencia fantástica!
Fecha: 2024-06-07 17:54:46

Por favor, inicia sesión para enviar tu experiencia. Si no estás registrado, pinche aquí.

5.2 procesar_experiencias.php

Este archivo procesa el formulario de valoraciones y almacena los datos en la base de datos si es correcto se redirige a experiencias.php. Se verifica que el usuario está registrado en el sistema

Visualización de valoraciones

Se obtiene la lista de comentarios existentes desde la base de datos.

Formulario de valoraciones

Se muestra un formulario para que los usuarios registrados puedan añadir nuevos comentarios.

Si el usuario no está registrado, se le solicita que inicie sesión o se registre.

6. Validar formularios en JavaScript

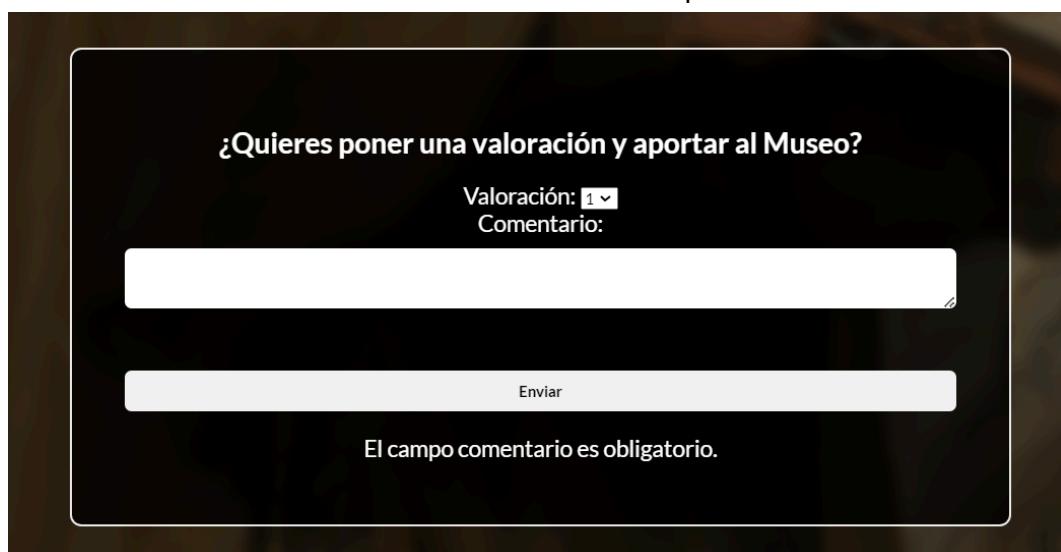
Para asegurarnos la validación de los diferentes formularios que tenemos en el sistema he utilizado como pedía la práctica JavaScript.

Los archivos involucrados son:

- validar_experiencias.php
- validar_login.php
- validar_registro.php

6.1 validar_experiencias.php

Este script se encarga de validar el formulario que hay en experiencias asegurándose que el campo de comentario no se encuentre vacío a la hora de procesar los datos del formulario.



6.2 validar_login.php

Este script valida el formulario de loguearse en el sistema asegurándose de que los campos usuario y contraseña se introduzcan obligatoriamente.

The screenshot shows a dark-themed login interface. At the top, the word "Usuario" is displayed above two input fields. Below these fields is a large, prominent "Iniciar Sesión" button. In the center of the page, there are two lines of text: "Introduzca el usuario" and "Introduzca la contraseña".

6.3 validar_registro.php

Este script valida el formulario de registro asegurandose que todos los campos estén rellenos antes de enviar éste.

The screenshot displays a registration form titled "Datos de Registro". Above the form, five validation messages are listed: "El campo nombre es obligatorio.", "El campo apellidos es obligatorio.", "El campo usuario es obligatorio.", "El campo correo es obligatorio.", and "El campo fecha de nacimiento es obligatorio.". The form itself contains seven input fields arranged in two columns: "Nombre" (with a person icon), "Apellidos" (with a person icon), "Usuario" (with an envelope icon), "Correo Electrónico" (with a mail icon), "Nueva Contraseña" (with a lock icon), "dd/mm/aaaa" (with a calendar icon), and "Teléfono" (with a phone icon). At the bottom of the form is a large "Registrarse" button.

Para que funcione correctamente he tenido que quitar el atributo required.

7. Ventana Emergente Obra

Funcionalidad de ventana emergente con JavaScript que salta cuando se posiciona el cursor sobre las imágenes de las obras listadas en colección.php. Se muestra una ventana con el título y la temática de la obra.

Los archivos involucrados son:

- detalleObra.js
- colección.php

7.1 detalleObra.js

Este script se encarga de crear la ventana emergente y manejar el evento de pasar el cursor por la imagen para mostrar la información.

Selecciona todas las imágenes de las obras de la página.

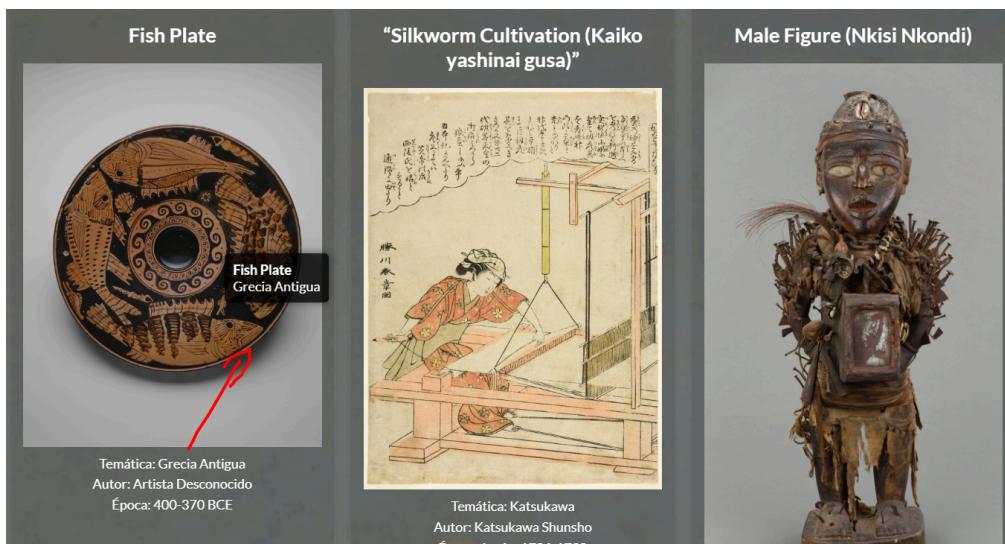
Crea un div y lo añade al cuerpo de colección.php

Añade eventos para cada imagen:

- onmouseover: Esto hace que se muestre con el título y la temática
- onmousemove: Mueve la ventana emergente mientras junto con el ratón mientras que esta se encuentre encima de la imagen de la obra.
- onmouseout: Oculta la venta emergente cuando el ratón se sale de la imagen.

7.2 colección.php

Se añaden los atributos título y temática a las imágenes para que el script pueda acceder a ellos y muestrarlos en la ventana emergente.

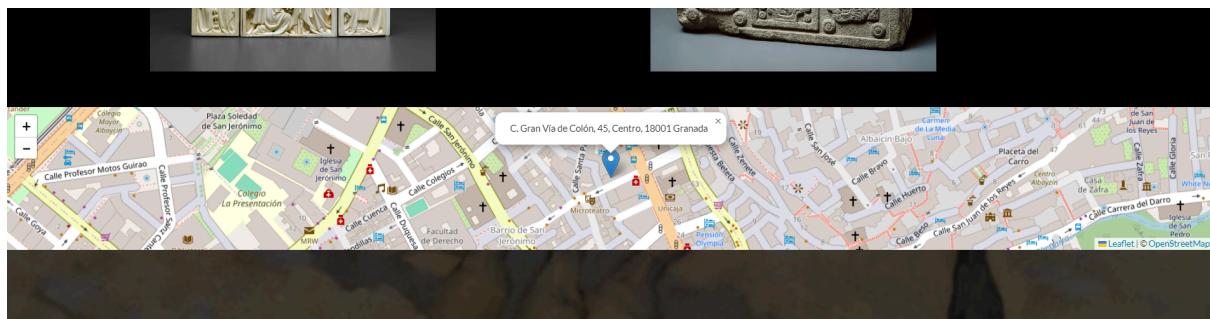


8. Aspecto Innovadores

Los aspectos innovadores que he implementado son dos:

- Visualización de la dirección del museo con el movimiento del mapa para una mejor orientación.
- Filtrado de experiencias mejorando la visualización de estas.

8.1 Visualización del Mapa



Se incluye la biblioteca Leaflet para la visualización de mapas.

Se inicializa un mapa centrado en la ubicación que queramos, en mi caso la del museo ([37.1789, -3.5998]), con un marcador y un popup con la dirección del museo.

8.2 Filtro Experiencias

Usuario	Valoración	Comentario	Fecha
marioillan	★★★★★	Este museo es increíble, tuve una experiencia fantástica!	2024-06-07 17:54:46
antoniojgg	★★★★★	Lo recomiendo pero me gustaría que incorporasen una rampa para accesibilidad de silla de ruedas. Gracias!	2024-06-08 18:02:14

El archivo filtroexperiencias.php se encarga de la lógica del filtrado de las experiencias. Dependiendo del filtro seleccionado se realiza una consulta SQL específica para ordenar y filtrar las experiencias.

Los filtros son más estrellas, menos estrellas, más recientes y más antiguas.