

Práctica Evaluable II :

Linqi Zhu

24 de junio de 2022



**UNIVERSIDAD
DE GRANADA**



1. Introducción

Esta práctica esta realizó con siguientes lenguajes:

HTML, CSS, PHP y JavaScript y con la base de dato MySQL.

Todo los archivos están alojado en el directorio pe2 del servidor bahia.

En *configuracion.php* y *conexión.php* están los código para conectar al base de dato del servidor bahia que proporciona profesor.

2. Base de dato MySQL

Los datos de trabajo esta guardado en database db06300759_pw2122,

Que contiene siguientes tablas:

```
[MariaDB [db06300759_pw2122]> show tables;
+-----+
| Tables_in_db06300759_pw2122 |
+-----+
| comentarios                  |
| comisarios                   |
| exposiciones                 |
| piezas                      |
| usuarios                    |
+-----+
```

Información de cada tablas:

```
[MariaDB [db06300759_pw2122]> desc comentarios;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| usuario    | varchar(40)   | NO   |     | NULL    |       |
| comentario | varchar(400)  | NO   |     | NULL    |       |
| nombre_exp | varchar(60)   | NO   |     | NULL    |       |
+-----+
```

```
[MariaDB [db06300759_pw2122]> desc exposiciones;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| titulo     | varchar(60)   | NO   | PRI | NULL    |       |
| tipo       | varchar(30)   | NO   |     | NULL    |       |
| fechaalta  | date          | NO   |     | NULL    |       |
| fechabaja  | date          | NO   |     | NULL    |       |
| urlImagen  | varchar(50)   | YES  |     | NULL    |       |
| urlExp     | varchar(50)   | NO   |     | NULL    |       |
| descripcion | varchar(400)  | NO   |     | NULL    |       |
+-----+
```

```
[MariaDB [db06300759_pw2122]> desc comisarios;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| dni        | varchar(12)   | NO   |     | NULL    |       |
| usuario    | varchar(10)   | NO   |     | NULL    |       |
| password   | varchar(40)   | NO   |     | NULL    |       |
| email      | varchar(100)  | NO   |     | NULL    |       |
| telefono   | varchar(10)   | NO   |     | NULL    |       |
| tipo       | varchar(10)   | NO   |     | NULL    |       |
+-----+
```

```
[MariaDB [db06300759_pw2122]> desc piezas;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| nombre     | varchar(40)   | NO   |     | NULL    |       |
| autor      | varchar(40)   | NO   |     | NULL    |       |
| urlImagen  | varchar(50)   | NO   |     | NULL    |       |
| descripcion | varchar(500)  | NO   |     | NULL    |       |
| nombre_exp | varchar(60)   | NO   |     | NULL    |       |
+-----+
```

```
[MariaDB [db06300759_pw2122]> desc usuarios;
+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+
| usuario    | varchar(10)   | NO   |     | NULL    |       |
| password   | varchar(50)   | NO   |     | NULL    |       |
| email      | varchar(100)  | NO   |     | NULL    |       |
| tipo       | varchar(10)   | NO   |     | NULL    |       |
+-----+
```

3. Pagina principal: index.php

La pagina principal esta estructura en Header, Main y Footer

Header: site-logo, site-name, site-login, site-menu

Main: aside de noticias, section de exposiciones

Footer: contacto y como_se_hizo.pdf

(En archivo index.css están los formato)

4. Implementación de los ejercicios pedidos

1. Permitir que los comisarios y el público general se den de alta en la aplicación de forma autónoma (ellos mismos), con el correspondiente formulario de alta.

Los archivos : altaUsu.php, altaCom.php, RegistroCom.php y

RegistroUsu.php

La idea es lo siguiente :

En *Registro*.php* están los formularios para rellenar y sera comprobado por las funciones de *R_Usuario.js*(implantado en javascript como pide el ejercicio 8).

Una vez introducido los datos en el formulario pasa a archivo *alta*.php*

(con atributo *action="alta*.php"* de formulario) donde se conecta al base de dato de servidor, con `$_POST` de php procesamos los datos del formulario html. Con consulta select comprobamos si están ya en BD, si no esta hace un INSERT, que inserta los dato en la tabla, y desconectamos del BD.

Una vez termina si no sale ninguna error ya están dado de alta.

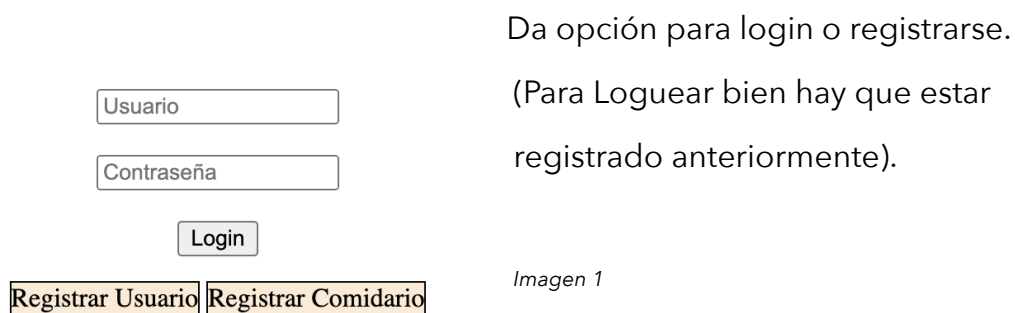
2. Que el usuario activo pueda modificar sus datos personales

3. Abrir un sesión, manteniéndola activa hasta que el usuario, de alguna forma, cierre la misma.

4. Una vez identificado, el usuario activo quedará en index.php pero pacerá su nombre y tipo en la esquina superior derecha, y un botón para cerrar sesión.

Los archivos: login.php, formularioLogin.php, logout.php, editarCom.php, editarUsu.php,, procesarEdiUsu.php, procesarEdiCom.php y index.php

Una vez entrado al pagina principal (index.php) en la parte izquierda aparecerá un formulario para login (implementado en archivo formularioLogin.php)



Da opción para login o registrarse.
(Para Loguear bien hay que estar registrado anteriormente).

Imagen 1

Formulario de login y registro. Incluye campos para Usuario y Contraseña, un botón Login, y dos botones para registrar: Registrar Usuario y Registrar Comidario.

Una vez Registrado y introduce dato en el formulario de *formularioLogin.php*, los datos pasan a archivo login.php, y en login.php hace dos consultas a base de dato uno con tabla usuario y otro con tablas comisario para encontrar que tipo de usuario es, una vez encontrado y guardamos los datos de usuario que se inicia en `$_SESSION` que sera utilizado después.

Una vez logueado cambiara el formulario de Index.php () en siguiente:

Comisario : zhulinqi

[Editar](#) [Cerrar Sesion](#)

Imagen 2

Usuario : zhulinqi

[Editar](#) [Cerrar Sesion](#)

Imagen 3

(Las dos caso, para ello en index.php en section site-login, un if de si usuario están iniciado de sesión aparece el nombre y el tipo de usuario que están guardado en `$_SESSION['usuario']`, `$_SESSION['tipo']`)

También aparece dos botones uno para cerrar sección que es un enlace al archivo `logout.php`, que cierra la sesión y la pagina `index.php` volver al estado de la *imagen 1*.

Y otro que es el enlace a los archivos `editar*.php` para hacer modificaciones de sus datos.

Tienes el mismo proceso que para registrar solo que esta vez las consultas de BD (realizado en `procesarEdi*.php`) es un UPDATE, y en el formulario de `editar*.php` ya esta relleno gracias `$_SESSION['usuario']`, `$_SESSION['email']`, `$_SESSION['password']` etc. , después de la consulta de SQL en BD también se actualiza los valores de `$_SESSION`.

5. Permitir que usuario comisarios puedan crear exposiciones, editarlas y borrarlas, así como añadir piezas a las exposiciones y el borrado de las pieza.

6. Las exposiciones y piezas mostradas en la aplicación web son las dadas de alta por los comisarios y almacenadas en la base de datos. Las primeras siempre se postran por fecha de exposición.

Los archivos: `editarPiezas.php`, `editarExpo.php`, `mostrarExpo.php`, `mostrarPiezas.php`, `procesarBorradoExpo.php`, `procesarBorradoPieza.php`, `procesarEditarExpo.php`, `procesarSubirExpo.php`, `procesarSubirPieza.php`, `index.php` y `Enamórate de la Ciudad Prohibida.php`.

Una vez iniciado como comisario aparecerá en la pagina principal un botón para Editar Exposición, que es un enlace a *editarExpo.php*.

En el archivo *editarExpo.php* hay tres section uno para subir, uno para borrar y otro para editar:

- El para subir es un formulario para subir nuevo exposición al servidor, con los datos necesario para la tabla de exposiciones de BD, una vez relleno el formulario pasa los datos con `$_POST` a `procesarSubirExpo.php` y hace una consulta INSERT para insertar nuevo exposición, antes de eso comprueba si están o no.

- Para borrar, marcamos el exposición que queremos borrar y pasa al archivo procesarBorradoExpo.php que con la consulta DELETE borra el exposición seleccionado en el formulario.

(Aparece todo los nombre de exposición para marcar)

- Para editar, para cada Exposición se genera un formulario, con los dato ya relleno, realizar cambio en casilla que queramos y pasa a los datos con \$_POST s procesarEditarExpo.php que realiza una consulta UPDATE para realizar cambio.

(Aparece formulario de todos los exposición que están registrado en BD)

Si entra a un exposición (Ej.: Enamórate de la Ciudad Prohibida.php, una exposición creado para hacer prueba, no he conseguido creación automática de pagina en php de exposición una vez dado de alta) aparece un botón de Editar Piezas, una enlace a editarPiezas.php.

Tiene opción para subir pieza y borrar pieza (no me da tiempo para hacer otro de editar).

Tiene las misma estructura de código para editar y borrar exposición solo que con piezas.

Para mostrar las exposiciones y las piezas (en archivos mostrar*.php) hacemos un consulta de SELECT, guardar todo los datos en una variable que devolver las funciones mostrar*(), sacando las filas de los datos con \$fila para mostrar los datos.

La mostrar según fecha no esta implementado, no lo conseguidos.

7. Permitir los usuario de publico en general puedan realizar comentarios en las exposiciones y mostracion de ello.

Los archivos: subirComentario.php, verComentario.php, mostrarComentario.php, procesarSubirComen.php y Enamórate de la Ciudad Prohibida.php.

Si esta iniciado de sesión como usuario de publico en generar dentro de la página de exposición (Enamórate de la Ciudad Prohibida.php) aparece

botón de Ver comentario, un enlace a verComentario.php donde puede ver todo los comentario que hay en BD, con mostrarComentario.php donde hacemos la consulta de SELECT de tabla comentarios. (Tiene la misma estructura de mostrar exposición y pieza). Y también un botón para subir nuevo comentario. (subirComentario.php y procesarSubirComen.php suben los comentario en BD, tienes las misma estructura que para subir exposición y pieza).

8. Validar la corrección de todos los formularios con JavaScript

El archivo: R_Usuario.js

Están los funciones para la comprobación de de formularios.

9. En el documento index.php, al pasar el ratón por la imagen asociada cada exposición, aparecerá una ventana emergente con el título de la exposición y el tipo de exposición a la que pertenece.

Con onmouseover = `function ()` (en JavaScript) mostrar una aviso de emergencia con los datos pedidos en el ejercicio.

