



# UNIVERSIDAD DE GRANADA

PW

PRACTICAS PROGRAMACIÓN WEB

## Práctica 2

---

**Autor:** Patricia Villalba Crucelaegui

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Curso 2021 - 2022

---

**ETSIIT**  
Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicación

---



# Índice

<b>1. Objetivos alcanzados</b>	<b>2</b>
<b>2. Organización de los ficheros de la práctica</b>	<b>2</b>
<b>3. Desarrollo de la práctica</b>	<b>3</b>
3.1. MySQL . . . . .	3
3.2. PHP . . . . .	4
3.3. JavaScript . . . . .	5
3.3.1. Formularios . . . . .	6
3.3.2. Usuarios . . . . .	6
3.4. Innovación . . . . .	7
3.5. Creación exposiciones . . . . .	7

## 1. Objetivos alcanzados

Para la realización de esta práctica se pedía mejorar nuestro sitio web en base a los siguientes puntos:

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.
2. Que el usuario activo pueda modificar sus datos personales.
3. Modificar el documento principal para permitir identificarse al usuario y abrir así una sesión, manteniéndola activa hasta que el usuario, de alguna forma, cierre la misma.
4. Una vez identificado, el usuario activo se quedará en el index.php pero aparecerá su nombre de usuario y su tipo en la esquina superior derecha, así como un enlace o botón para cerrar sesión.
5. Permitir que usuarios comisarios puedan crear exposiciones (asignándolas a un tipo de exposición), editarlas (sólo datos básicos, no piezas) y borrarlas (borrando todas las piezas asignadas), así como añadir piezas a las exposiciones. La edición y el borrado de piezas es voluntario.
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 mostrarán por fecha de exposición.
7. Permitir que los usuarios de público en general puedan realizar comentarios en las exposiciones y que los existentes se muestren al entrar en el documento de una exposición.
8. Validar la corrección de todos los formularios con JavaScript. En relación a los formularios, deben existir campos obligatorios, los comentarios a las entradas deben tener una longitud máxima, deben existir en los formularios campos que se correspondan con valores numéricos, números de teléfonos y direcciones de correo electrónico o URLs (todos validados según su formato correspondiente y en JavaScript).
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.

La práctica cumple con una gran parte de los puntos: se permite al usuario y al comisario que se den de alta mediante un formulario, el usuario activo puede modificar sus datos, el usuario puede iniciar sesión y esta mantendrá activa hasta que el usuario decida cerrarla con el botón de “Cerrar sesión”.

Los usuarios comisarios pueden crear exposiciones, borrarlas y modificarlas. **Primera cosa que falta:** no se pueden añadir piezas y con lo que esto supone, la eliminación y modificación de las mismas.

**Segunda cosa que falta:** los usuarios no podrán dejar comentarios.

Todos los formularios están validados con JavaScript. Encontramos campos de texto y direcciones de correo electrónico, pero **no** encontramos campos numéricos.

También se cumple con el último punto de la práctica: al pasar el ratón sobre la imagen de la exposición se nos abre una ventana emergente con el título y tipo de exposición.

## 2. Organización de los ficheros de la práctica

Esta práctica está formada por diversos ficheros .php y un fichero .css que proporciona el estilo al sitio web. Además de las distintas imágenes usadas (entre las que encontramos diferentes extensiones). La estructura de nuestro sitio web es la siguiente:

- Nuestro directorio principal denominado **pe2** que albergará los diferentes directorios
  - **Directorio css:** que contiene el fichero .css
  - **Directorio exposiciones:** que contiene los diferentes ficheros .php utilizados para la creación del sitio web

- **Directorio imágenes:** subdivido en cuatro directorios: expo1, expo2, expo3 y expo4, que contendrán las diferentes imágenes de las piezas de las exposiciones
- El fichero **index.php** que es nuestro fichero principal
- El informe **como\_se\_hizo.pdf** que recoge una pequeña memoria sobre el desarrollo de la práctica.

La práctica sigue la misma organización que en la práctica anterior. Los ficheros siguen el mismo patrón: cabecera, barra nav, main, aside y footer.

## 3. Desarrollo de la práctica

### 3.1. MySQL

Lo primero que hice al empezar con la práctica fue crear mis tablas para el desarrollo de esta. Me conecte a MySQL a través de bahía y crear las siguientes tablas:

- Tabla para el público

```
CREATE TABLE publico(
    nombre VARCHAR(20) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    id_publico VARCHAR(20) NOT NULL,
    email VARCHAR(50) NOT NULL,
    pass VARCHAR(50) NOT NULL,
    PRIMARY KEY (email)
);
```

- Tabla para los comisarios

```
CREATE TABLE comisario(
    nombre VARCHAR(20) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    id_comisario VARCHAR(20) NOT NULL,
    email VARCHAR(50) NOT NULL,
    pass VARCHAR(50) NOT NULL,
    PRIMARY KEY (email)
);
```

- Tabla para las exposiciones

```
CREATE TABLE exposiciones(
    nombre VARCHAR(20) NOT NULL,
    autor VARCHAR(50) NOT NULL,
    tipo VARCHAR(20) NOT NULL,
    PRIMARY KEY (nombre)
);
```

He creado un fichero llamado *configuracion.inc* donde se encuentran las credenciales necesarias para conectarse con la base de datos en la cual estamos trabajando.

```
≡ configuracion.inc
1  <?php
2      define("DB_DSN", "mysql:host=localhost; dbname=db77392301_pw2122");
3      define("DB_USUARIO", "pw77392301");
4      define("DB_CONTRASENIA", "77392301");
5  ?>
```

Para la conexión y desconexión de la base de datos, utilizo el fichero *datosObject.class.inc* el cual contiene funciones para realizar estas operaciones.

```
<?php
require_once('configuracion.inc');

abstract class DataObject {
    protected $datos = array();

    public function __construct( $datos ) {
        foreach ( $datos as $clave => $valor )
            if ( array_key_exists( $clave, $this->datos ) ) $this->datos[$clave] = $valor;
    }

    public function devolverValor( $campo ) {
        if ( array_key_exists( $campo, $this->datos ) ) {
            return $this->datos[$campo];
        } else die( "Campo no encontrado" );
    }

    protected static function conectar() {
        try {
            $conexion = new PDO( DB_DSN, DB_USUARIO, DB_CONTRASENIA );
            // Se permite a PHP que mantenga la conexión MySQL abierta para
            // que se emplee en otras partes de la aplicación.
            $conexion->setAttribute( PDO::ATTR_PERSISTENT, true );
            $conexion->setAttribute(PDO::ATTR_ERRMODE,
                PDO::ERRMODE_EXCEPTION );
            $conexion->setAttribute(PDO::ATTR_EMULATE_PREPARES, 1);
        } catch ( PDOException $e ) {
            die( "Conexión fallida: " . $e->getMessage() );
        }
        return $conexion;
    }

    protected static function desconectar( $conexion ) {
        $conexion = "";
    }
}

?>
```

Además, hemos usado varias sentencias SQL para la creación, modificación, eliminación así como la validación del login del usuario. Estas sentencias se encuentran dentro de los métodos usados en las distintas clases de los usuarios, así como la clase de exposiciones.

### 3.2. PHP

Vamos a ver los diferentes fragmentos de PHP escritos a lo largo del código.

Para empezar, en todos los ficheros vamos a encontrar el siguiente fragmento de código el cual se encarga de mantener la sesión abierta:

```
<?php
|     session_start();
?>
```

Así pues, en la cabecera, he puesto el siguiente fragmento de código que realiza lo siguiente: si hay una sesión iniciada, se muestra el correo del usuario y el tipo de usuario que es (si es público o comisario). Permite también el inicio de sesión de los distintos usuarios. Muestra también un enlace para ambos para cerrar sesión (cuando la sesión esté activa), así como para modificar sus datos. Si además el usuario es un comisario, muestra también un enlace para la gestión de las exposiciones.

```

<section class="formulario">
<?php
    if(!empty($_SESSION['Publico'])){
        echo '<h1 id="u"> '.$_SESSION['Publico'].' (identificado como publico) </h1>';
        <a class="enlace" href="exposiciones/procesar_registro.php" id="u2"> Cerrar Sesión </a>
        <a class="enlace" href="exposiciones/modificar_publico.php" id="u3"> Editar perfil </a>;

    }
    else if(!empty($_SESSION['Comisario'])){
        echo '<h1 id="u"> '.$_SESSION['Comisario'].' (identificado como comisario) </h1>';
        echo '<a class="enlace" href="exposiciones/procesar_registro.php" id="u2"> Cerrar Sesión </a>';
        echo '<a class="enlace" href="exposiciones/gestion_exposiciones.php" id="u3"> Gestion exposiciones </a>';
        echo '<a class="enlace" href="exposiciones/modificar_comisario.php" id="u3"> Editar perfil </a>';
    }else{
        echo '<section class="iniciarsesion">
        <form action="exposiciones/comprobar_registro.php" method="post">

            <label for="email" name="email"><b>Email</b></label>
            <input type="email" name="email">

            <br>
            <br>

            <label for="pass" name="pass"><b>Contraseña</b></label>
            <input type="password" name="pass">

            <br><br>

            <input type="submit" id="button1" value="Iniciar sesion" />

        </form>
        </section>';
    }
}
?>

```

### 3.3. JavaScript

Para la validación de formularios he usado el lenguaje JavaScript, tal y como se pedía en la práctica. Esto está complementado con una clase creada en PHP que se llama *procesarformulario.php*, la cual se encarga de comprobar si los campos son correctos o no. A continuación dejo algunos ejemplos de como lo he empleado en la validación de los formularios:

```

<script>
function validarFormulario(){
    let required=["nombre", "apellidos", "id_publico","email","pass"];

    for(campo of required){
        if(document.forms["modificarperfil"][campo].value==""){
            alert("El campo "+campo+" es obligatorio");
            return false;
        }
    }

    if(document.forms["modificarperfil"]["pass"].value==""){
        alert("Introduce la nueva contraseña (Puede ser igual a la actual)");
        return false;
    }

    let correoElectronico=document.forms["modificarperfil"]["email"].value;
    if(correoElectronico.indexOf(' ')>-1){
        alert("No se admiten espacios en blanco");
        return false;
    }else if(correoElectronico.indexOf('@')<1){
        alert("Formato incorrecto. Pruebe este formato: <nombre>@<dominio>.<extension>");
        return false;
    }
}
</script>

```

```

ction class="registro">
<h1 class="rh1">Formulario para modificar el perfil del público</h1>

<form name="modificarperfil" action="../../../exposiciones/procesar_modificacion.php" onsubmit="return validarFormulario()" method="post">

    <label for="nombre" name="nombre">Nombre</label>
    <input type="text" name="nombre" placeholder="Introduzca su nombre" value="<?php echo $nombre ?>"/>

    <br>

    <label for="apellidos" name="apellidos">Apellidos</label>
    <input type="text" name="apellidos" placeholder="Introduzca sus apellidos" value="<?php echo $apellidos ?>"/>

    <br>

    <label for="id_publico" name="id_publico">Nombre de usuario</label>
    <input type="text" name="id_publico" placeholder="Introduzca su nombre de usuario" value="<?php echo $id_publico ?>"/>

    <br>

    <label for="email" name="email">Email</label>
    <input type="email" name="email" placeholder="ejemplo@email.com" value="<?php echo $_SESSION['Publico'] ?>"/>

    <br>

    <label for="pass" name="pass">Contraseña</label>
    <input type="password" name="pass" placeholder="Introduzca su contraseña" required>

    <br>

    <input class="center" type="submit" value="Enviar"/>

</form>
ction>

```

También lo he usado para que me muestre ventanas emergentes con los distintos casos de éxito o fracaso a la hora de registrar un nuevo público, por ejemplo.

```

<?php
require_once ('comisario.class.php');

if ($_SERVER['REQUEST_METHOD'] === 'POST')
{
    $nombre = $_POST['nombre'];
    $apellidos = $_POST['apellidos'];
    $id_comisario = $_POST['id_comisario'];
    $email = $_POST['email'];
    $pass = $_POST['pass'];

    if (Comisario::insertarComisario($nombre, $apellidos, $id_comisario, $email, $pass))
    {
        echo"
        <script language='javascript'>
            alert('Comisario registrado correctamente')
            window.location = '../index.php'
        </script>";
        exit();
    }

    else
    {
        echo"
        <script language='javascript'>
            alert('Registro malo')
            window.location = '../exposiciones/altacomisario.php'
        </script>";
    }
}
?>

```

### 3.3.1. Formularios

La práctica contiene un formulario para iniciar sesión, un formulario para darse de alta como nuevo publico, un formulario para darse de alta como nuevo comisario, un formulario para modificar los datos de un perfil (tanto comisario como publico), un formulario para añadir una exposición y un formulario para modificar los datos de la exposición.

Cada formulario tendrá su correspondiente clase que lo procesará. Para esto usamos el lenguaje PHP.

### 3.3.2. Usuarios

Tenemos el usuario publico y el usuario comisario. El usuario comisario será el único que tenga permisos para la creación y administración de nuevas exposiciones.

Cada uno tendrá su propia clase denominada *publico.class.php* o en otro caso *comisario.class.php*. En estas clases encontraremos todas las funciones relativas a la inserción, modificación, eliminación, etc de los usuarios.

De igual manera, aunque no sea un usuario, pero para las exposiciones tendremos también su correspondiente clase *exposiciones.class.php* con sus correspondientes funciones.

### 3.4. Innovación

Como innovación en el punto 9, he usado un evento que se denomina *onMouseOver*, este evento lo que hace es que al pasar el ratón ya sea por una zona de texto o una imagen, cambie su contenido". En este caso, se pedía una ventana emergente al pasar el ratón por encima de la exposición, para ello, con JavaScript, he creado una función la cual lanza una alerta y esta función será la que usará con *onMouseOver*. No lo he hecho de la manera más eficiente posible, pero cumple su papel.

A continuación dejo una imagen:

```
<script language="javascript">
    function mialerta() {
        alert("Titulo exposicion: " + document.getElementById("titulo").innerHTML + " y tipo de exposicion: " + document.getElementById("tipo").innerHTML );
    }

    function mialerta2() {
        alert("Titulo exposicion: " + document.getElementById("titulo2").innerHTML + " y tipo de exposicion: " + document.getElementById("tipo2").innerHTML );
    }

    function mialerta3() {
        alert("Titulo exposicion: " + document.getElementById("titulo3").innerHTML + " y tipo de exposicion: " + document.getElementById("tipo3").innerHTML );
    }

    function mialerta4() {
        alert("Titulo exposicion: " + document.getElementById("titulo4").innerHTML + " y tipo de exposicion: " + document.getElementById("tipo4").innerHTML );
    }
</script>

<section class="articulo">
    <a href="exposiciones/expo1.php" onMouseOver="mialerta()"> <img src = "imagenes/expo1/expo1-portada.jpeg" alt = "Foto exposicion 1"/> </a>
    <p id="titulo">Arte y mito.</p>
    <p><i>Autor:</i> Varios</p>
    <p>Óleo sobre lienzo</p>
    <p>1636-1638</p>
    <p id="tipo">Pintura</p>
</section >

<section class="articulo">
    <a href="exposiciones/expo2.php" onMouseOver="mialerta2()"> <img src = "imagenes/expo2/expo2-portada.jpeg" alt = "Foto exposicion 2"/> </a>
    <p id="titulo2">Goya y su pintuta.</p>
    <p><i>Autor:</i> Goya</p>
    <p>Óleo sobre lienzo</p>
    <p>1781</p>
    <p id="tipo2">Pintura</p>
</section >
```

### 3.5. Creación exposiciones

Para la creación de exposiciones he conseguido que se me creen y pueda modificarlas y borrarlas, todo en la base de datos. Lo que no he conseguido es mostrarlas en el index cada vez que creaba una. Si que conseguí mostrarlas, pero no era una solución que me gustase y creo que no era la correcta, así que he preferido no añadirla a la práctica.

Es por eso por lo que la página cuenta con las exposiciones estáticas que se encontraban en la pe1.

A continuación dejo un ejemplo del funcionamiento de la creación, modificación y borrado.





**comisario@correo.es**  
 (identificado como comisario)

[Cerrar Sesión](#)
[Gestion exposiciones](#)
[Editar perfil](#)

[Inicio](#)
[Fotografía](#)
[Pintura](#)
[Escultura](#)

## Alta exposicion

**Nombre de la exposicion:**

**Autor de la exposicion:**

**Tipo de exposicion:**

[Enviar](#)



Figura 1: Añadir exposicion

## Gestion de las exposiciones

**Añadir exposicion**

[Añadir exposicion](#)

**Eliminar exposicion:** 

[Eliminar](#)

**Modificar exposicion**

**Nombre de la exposicion que se quiere modificar:**

**Autor de la exposicion que se quiere modificar:**

**Tipo de exposicion que se quiere modificar:**

[Enviar](#)

Figura 2: Modificar exposicion

[Inicio](#)
[Fotografía](#)
[Pintura](#)
[Escultura](#)

## Gestion de las exposiciones

**Añadir exposicion**

[Añadir exposicion](#)

**Eliminar exposicion:** 

[Eliminar](#)

**Modificar exposicion**

**Nombre de la exposicion que se quiere modificar:**

**Autor de la exposicion que se quiere modificar:**

**Tipo de exposicion que se quiere modificar:**

Figura 3: Eliminar exposicion

```

MariaDB [db77392301_pw2122]> select * from exposiciones;
+-----+-----+-----+
| nombre | autor      | tipo    |
+-----+-----+-----+
| Prueba | Comisario  | Pintura |
+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [db77392301_pw2122]> select * from exposiciones;
+-----+-----+-----+
| nombre | autor      | tipo    |
+-----+-----+-----+
| Prueba | Comisario nuevo | Escultura |
+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [db77392301_pw2122]> select * from exposiciones;
Empty set (0.000 sec)

MariaDB [db77392301_pw2122]>

```

Figura 4: MySQL