

# ¿COMO AFRONTAR UN EJERCICIO EN PHP?

## Enunciado :

### Blog con Gestión de Usuarios, Artículos y Comentarios

Desarrolla una aplicación web para un blog que permita a los usuarios iniciar sesión, ver artículos y comentarios, crear contenido y gestionar la visibilidad de sus propios artículos. La aplicación debe incluir las siguientes funcionalidades:

### Funcionalidades Básicas

#### *Login de Usuarios:*

Al acceder a la página, los usuarios que no hayan iniciado sesión serán identificados como "invitado".

Los usuarios registrados pueden iniciar sesión proporcionando un nombre de usuario y contraseña.

Un usuario invitado solo tiene acceso a la visualización de los artículos y comentarios públicos.

Una vez logueado, el usuario puede crear artículos y comentar en los existentes.

#### *Gestión de Artículos:*

Los artículos deben mostrar el título, el contenido, el autor, la fecha de publicación y, si aplica, una imagen asociada.

Todos los usuarios, logueados o no, pueden ver los artículos públicos (visibles) junto con sus comentarios.

Los usuarios logueados pueden crear nuevos artículos, que por defecto son visibles.

Cada usuario puede hacer visible o no sus propios artículos mediante un botón que alterna la visibilidad.

#### *Gestión de Comentarios:*

Cada artículo puede tener múltiples comentarios.

Los comentarios deben mostrar el nombre del autor, la fecha del comentario y el contenido del comentario.

Los usuarios logueados pueden añadir comentarios a cualquier artículo visible.

#### *Visualización de Artículos en Detalle:*

Al hacer clic en el título de un artículo, este debe expandirse para mostrar solo el artículo seleccionado en detalle.

## 1. ENTENDER LOS REQUISITOS DEL EJERCICIO

Antes de escribir cualquier línea de código, es importante entender bien lo que se pide. En este caso se necesita:

### a. Estructura de la base de datos:

Según el enunciado, se identifican tres tablas: user, article y coment.

Antes de comenzar el código, se debe tener la estructura de la base de datos planteada y creada.

### b. Login de usuarios:

Si el usuario no está logueado, será un usuario “invitado” con permisos limitados.

Si el usuario se loguea correctamente, tendrá acceso a funcionalidades adicionales.

### c. Gestión de artículos:

Todos los usuarios pueden ver todos los artículos visibles.

Los usuarios logueados pueden crear y administrar sus propios artículos.

### d. Gestión de comentarios:

Todos los usuarios pueden ver los comentarios de cada artículo.

Los usuarios logueados pueden añadir comentarios a los artículos.

## 2. PLANTEAR LA ESTRUCTURA DEL CÓDIGO

El código estará dividido en dos partes principales:

- **Parte superior (PHP):** Contendrá la lógica, operaciones con la base de datos y la gestión de sesión del usuario.
- **Parte inferior (HTML Y PHP):** Mostrará la interfaz gráfica, incluyendo el formulario de login, los artículos y la funcionalidad para que los usuarios interactúen con el contenido.

## 3. IMPLEMENTAR LA GESTIÓN DE SESIONES

### a. Iniciar la sesión y establecer la conexión a la base de datos.

Este paso debe realizarse siempre, y debe ubicarse al inicio del archivo. Es fundamental para cualquier interacción que sigue en el código.

```
<?php
session_start();
$db = new mysqli("localhost", "root", "", "blog");
?>
```

### b. Abrir el documento HTML e incluir la cabecera básica del sitio.

Para hacer la división de la parte lógica y gráfica, después de establecer el inicio del bloque PHP, se puede iniciar el bloque gráfico HTML.

Puede verse de la siguiente forma:

```

<!-- PARTE LÓGICA -->

<?php
session_start();
$db = new mysqli("localhost", "root", "", "blogAna");

?>

<!-- PARTE VISUAL -->

<!DOCTYPE html>

<html lang='es'>

<head>
    <meta charset='UTF-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1.0'>

    <!-- ESTILOS -->

    <style> </style>

    <title>Blog Ana</title>
</head>

<body>

</body>

</html>

```

### c. Inicializar la sesión de invitado (valores por defecto)

El lugar idóneo para establecer los valores de inicio de sesión es justo después de iniciarla “**session\_start()**”

En este bloque se garantiza que, si no hay un usuario logueado, se establecen los valores de invitado por defecto. De esta forma, siempre hay un estado conocido para \$\_SESSION.

```

// valores para la sesión
if (!isset($_SESSION['login']) || !isset($_SESSION['id_user'])) {
    $_SESSION['login'] = false;
    $_SESSION['userName'] = "invitado";
    $_SESSION['id_user'] = 0; // Usuario invitado por defecto
}

```

#### d. Crear el formulario login/logout

Aún en la parte de la lógica de PHP, inmediatamente después del paso anterior, se verifica si el usuario ha enviado los datos de inicio de sesión y actualiza los datos de la sesión en consecuencia.

```
// formulario login/logout
if (isset($_POST['username']) && isset($_POST['password'])) {
    $q = "SELECT * FROM user WHERE username = '" . $_POST['username'] . "'";
    $result = $db->query($q);
    if ($row = $result->fetch_assoc()) {
        if ($row['password'] == $_POST['password']) {
            $_SESSION['login'] = true;
            $_SESSION['username'] = $row['username'];
            $_SESSION['id_user'] = $row['id'];
            $info = "Sesión iniciada correctamente";
        } else {
            $info = "Contraseña incorrecta";
        }
    } else {
        $info = "Usuario no encontrado";
    }
}
```

#### e. Mostrar el formulario al usuario.

En este punto, se establecerá qué verá el usuario en cada momento. Si no está logueado verá el formulario de login y si lo está, se mostrará el botón “logout”.

Esta parte, como es una visualización para el usuario, debe ir ubicada en la parte gráfica de HTML.

```
<body>

<!-- Mostrar formulario login/logout -->
<?php if ($_SESSION['login']) { ?>
    <a href="blog.php?logout=1">Logout</a><br><br>
<?php } else { ?>
    <form method="post" action="blog.php">
        <legend>INICIA SESIÓN</legend>
        <input type="text" name="username" placeholder="Nombre de usuario" required />
        <input type="password" name="password" placeholder="Contraseña" required />
        <button type="submit">Enviar</button>
    </form>
<?php } ?>

</body>
```

#### f. Cerrar sesión

Aquí se define qué ocurre cuando el usuario hace click en el botón de logout. En este caso, se restablecen los valores de sesión a los de invitado.

Como se trabaja en valores de \$\_SESSION, este bloque debe ir en la parte lógica del código, es decir, arriba.

```
// Cierre de sesión
if (isset($_GET['logout'])) {
    $_SESSION['login'] = false;
    $_SESSION['username'] = "invitado";
    $_SESSION['id_user'] = 0;
    header("Location: blog.php");
    exit();
}
```

## 4. CREAR ARTÍCULOS

### a. Consultar los artículos visibles (lógica PHP)

Obtener los artículos que deben ser visibles en la página para el usuario actual (invitado o logueado). Se realiza una consulta a base de datos, que se ejecutará una vez por carga de la página.

```
// Consultar artículos
$query_articles = "
    SELECT article.id, title, text, date, img, username AS author, visible
    FROM article
    JOIN user ON article.id_user = user.id
    WHERE article.visible = TRUE OR article.id_user = " . $_SESSION['id_user'];
$articles_result = $db->query($query_articles);
```

Esta consulta dependerá de que es lo que se quiere mostrar y de como se ha estructurado la base de datos.

### b. Formulario para crear artículos (si el usuario está logueado)

Después de establecer el formulario para login/logout en la parte gráfica, se comprueba si realmente existe un usuario logueado para mostrar el formulario que permite crear un nuevo artículo. Este bloque debe ir en la parte gráfica.

```
<!-- Mostrar formulario para crear artículos -->
<?php if ($_SESSION['login'] && isset($_POST['art'])) { ?>
    <form method='post' action='blogAna.php' enctype='multipart/form-data'>
        <legend>CREAR ARTÍCULO</legend>
        <input type='text' name='title' required placeholder="Titulo"/>
        <input type='text' name='content' required placeholder="Contenido"/>
        <input type='file' name='img' />
        <button type='submit'>Enviar</button>
    </form>
<?php } else if ($_SESSION['login']) { ?>
    <form method='post' action='blogAna.php'>
        <button type='submit' name='art' value='art'>Crear artículo</button>
    </form>
<?php } ?>
```



### c. Procesar formulario para insertar el artículo

Para manejar la inserción del artículo en la base de datos, se debe comprobar que el usuario esté logueado.

```
// Procesar y guardar nuevo artículo en la base de datos
if ($_SESSION['login'] && isset($_POST['submit_article'])) {
    $title = $_POST['title'];
    $text = $_POST['text'];
    $date = date('Y-m-d'); // Fecha actual
    $id_user = $_SESSION['id_user'];
    $img = '';

    // Si se ha subido una imagen, procesa el archivo
    if (!empty($_FILES['img']['name'])) {
        $img = $_FILES['img']['name'];
        $target = "../img/" . basename($img);
        // Mueve el archivo subido a la carpeta de destino
        if (!move_uploaded_file($_FILES['img']['tmp_name'], $target)) {
            echo "Error al subir la imagen.";
        }
    }

    // Inserta el artículo en la base de datos
    $insert_article = "INSERT INTO article (title, text, date, id_user, img, visible)
    VALUES ('$title', '$text', '$date', '$id_user', '$img', 1)";
    if ($db->query($insert_article) === TRUE) {
        echo "Artículo subido con éxito.";
    } else {
        echo "Error al subir el artículo: " . $db->error;
    }
    header("Location: blogAna.php");
}
```

### d. Mostrar artículos al usuario

Para mostrar los artículos, se debe iterar sobre la base de datos para recorrer todos los artículos que tiene y mostrarlos con sus respectivos comentarios (que se crearán más adelante). Este bloque debe ir justo detrás del anterior.

```
<!-- Mostrar artículos -->
<h2>Artículos del Blog</h2>
<?php
if ($articles_result->num_rows) {
    while ($article = $articles_result->fetch_assoc()) {
        echo "<div class='article'>";
        echo "<h3>" . $article['title'] . "</h3>";
        echo "<p>" . $article['text'] . "</p>";
        echo "<p><strong>Autor:</strong> " . $article['author'] . "</p>";
        echo "<p><strong>Fecha:</strong> " . $article['date'] . "</p>";
        if ($article['img']) {
            echo "<img src='../img/" . $article['img'] . "' width='600px' />";
        }
        // Sección para comentarios aquí
        echo "</div>";
    }
} else {
    echo "<p>No hay artículos disponibles.</p>";
}
?>
```

## 5. CREAR Y AÑADIR COMENTARIOS

### a. Consulta de comentarios para el artículo concreto (lógica PHP)

Después de cargar el artículo, se realiza una consulta a la base de datos para obtener todos los comentarios asociados con el artículo actual.

```
// Consultar comentarios
$article_id = $article['id_article'];
$query_comments = "SELECT coment, comentDate, username AS author
                    FROM coment
                    JOIN user ON coment.id_author = user.id
                    WHERE id_article = $id_article";
$comments_result = $db->query($query_comments);
```

### b. Agregar comentarios a la base de datos

Si el usuario ha enviado un comentario (se evalúa cuando el formulario de comentario es enviado), se procesa la entrada y se inserta en la base de datos.

Esta parte se coloca al inicio del bloque PHP, después de gestionar el login/logout. Esto se debe a que primero es necesario asegurar que el usuario esté identificado y si el usuario está logueado y ha enviado un comentario, se procesa la información y se almacena en la base de datos.

```
// Agregar comentarios a la base de datos
if ($_SESSION['login'] && isset($_POST['coment']) && isset($_POST['article_id'])) {
    $coment = $_POST['coment'];
    $id_user = $_SESSION['id_user'];
    $article_id = $_POST['article_id'];

    $insert_comment = "INSERT INTO coment (id_author, comentDate, coment, article_id)
                      VALUES ($id_user, NOW(), '$coment', $article_id)";
    $db->query($insert_comment);
    header("Location: blog.php");
    exit();
}
```

### c. Mostrar los comentarios

Se utiliza un bucle para recorrer todos los artículos y con PHP se verifica si hay comentarios para mostrar.

```
<!-- Mostrar comentarios -->
<?php
if ($comments_result->num_rows > 0) {
    while ($comment = $comments_result->fetch_assoc()) {
        $date = new DateTime($comment['comentDate']);
        echo "<div class='comment'>";
        echo "<p><strong>" . $comment['author'] . " :</strong> " . $comment['coment'] .
            " <em>(" . $date->format('d-m-Y H:i:s') . ")</em></p>";
        echo "</div>";
    }
} else {
    echo "<p>No hay comentarios.</p>";
}
?>
```

#### d. Formulario para crear comentarios

Este formulario solo se mostrará si el usuario está logueado.

Antes se reservó un espacio en el bucle para recorrer los artículos, donde se especificaba que pertenecía a la sección de comentarios. Ese es el lugar donde se ubicará el formulario para añadir un comentario para que pertenezca al artículo actual.

```

if ($article['img']) {
    echo "<img src='../img/' . $article['img'] . '" width='600px' />";
}

// Sección para comentarios aquí
if ($_SESSION['login']) { ?>
    <form method="post" action="blog.php">
        <input type="hidden" name="article_id" value="<?php echo $article['id']; ?>" />
        <textarea name="coment" placeholder="Escribe un comentario... " required></textarea>
        <button type="submit">Enviar</button>
    </form>
<?php }

echo "</div>";
}
se {
echo "<p>No hay artículos disponibles.</p>";

```

## 6. HACER VISIBLE O NO LOS ARTICULOS DEL USUARIO

### a. Agregar botón para cambiar la visibilidad de los artículos

Colocar un botón “Mostrar/Ocultar” debajo de cada artículo SOLO si el usuario logueado es el autor del artículo.

```

<!-- Mostrar cambio de visibilidad -->
<?php if ($article['author'] == $_SESSION['userName']) { ?>
    <form method="post" action="blogAna.php">
        <input type="hidden" name="article_id" value="<?php echo $article['id_article']; ?>" />
        <button type="submit" name="visibility">
            <?php echo $article['visible'] ? "Ocultar" : "Mostrar"; ?>
        </button>
    </form>
<?php } ?>

```

### b. Procesar la solicitud del cambio

Establecer la lógica que recibirá el “article\_id” y alternará el valor de “visible” para este artículo.

Integrar este bloque después de la lógica de login/logout y antes de la consulta de artículos, asegurará que, al cambiar la visibilidad, la página se actualice con el nuevo estado del artículo.



```
// Procesar solicitud visibilidad
if (isset($_POST['visibility']) && isset($_POST['article_id'])) {
    $article_id = $_POST['article_id'];

    // Consulta para obtener el estado actual de visibilidad del artículo
    $q = "SELECT visible FROM article WHERE id = $article_id";
    $result = $db->query($q);
    if ($row = $result->fetch_assoc()) {

        // Cambiar la visibilidad: si es 1, cambiar a 0; si es 0, cambiar a 1
        $new_visibility = $row['visible'] ? 0 : 1;
        $update_visibility = "UPDATE article SET visible = $new_visibility WHERE id = $article_id";
        $db->query($update_visibility);
    }

    // Redirigir para evitar el reenvío del formulario
    header("Location: blogAna.php");
}
```