



**CICLO: D.A.W.
MÓDULO DE DESARROLLO WEB
ENTORNO SERVIDOR**

Tarea N° 04

**Alumno:
Armando Herrero Silva
76442135W**

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

© 2022 Fomento Ocupacional FOC SL todos los derechos reservados.

Contenido

1.	RA04_a.	2
2.	RA04_b.	3
3.	RA04_c.	5
4.	RA04_d.	7
5.	RA04_e.	9
6.	RA04_f.	10

1. RA04_a.

Ejercicio de Sesiones: Contador de Visitas por Sesión

- **Objetivo:** Crear una aplicación PHP que cuente las visitas del usuario durante una misma sesión.
- **Instrucciones:**
 1. Cuando el usuario acceda a la página, verifica si existe una sesión iniciada y, de ser así, muestra un contador de visitas.
 2. Si es la primera vez que el usuario accede, inicializa la sesión y establece el contador en 1.
 3. Cada vez que el usuario recargue la página, aumenta el contador en 1.
 4. Muestra el número de visitas actuales dentro de la misma sesión.

Lo primero que debemos hacer es iniciar una sesión mediante la sentencia `session_start()`.

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

Luego deberemos crear el contador con una sentencia `if` para que en caso de que no se haya iniciado nunca una sesión, el contador de visitas inicie en 1.

```
if (isset($_SESSION['contador_visitas'])) {  
    $_SESSION['contador_visitas']++;  
} else {  
    $_SESSION['contador_visitas'] = 1;  
}
```

Por último, debemos mostrar el mensaje mediante el cual mostraremos el número total de visitas de la sesión actual.

```
echo "<h1>Bienvenido</h1>";  
echo "<p>Esta página ha sido visitada un total de " . $_SESSION['contador_visitas'] . " veces.</p>";  
?>
```

2. RA04_b.

Ejercicio de Cookies: Recordar Preferencias del Usuario

- **Objetivo:** Crear una página PHP que recuerde el tema de color preferido del usuario utilizando

cookies.

- **Instrucciones:**

1. Ofrece al usuario dos opciones de tema: claro y oscuro.
2. Cuando el usuario seleccione un tema, guarda esta preferencia en una cookie con una duración de una semana.
3. La próxima vez que el usuario acceda a la página, verifica si existe la cookie del tema y aplica automáticamente el tema guardado.

Primero vamos a crear los dos estilos de la web, el oscuro y el claro.

```
<style>
  body.claro {
    background-color: white;
    color: black;
  }

  body.oscuro {
    background-color: black;
    color: white;
  }
</style>
```

Para que esto se aplique según esté establecida nuestra cookie, declaramos lo siguiente en la etiqueta del body:

```
<body class="php echo isset($_COOKIE['oscuro']) ? 'oscuro' : 'claro'; ?">
```

De esta forma si la cookie está establecida en oscuro, se aplica el tema oscuro y si no es así aplica el claro.

Procedemos a crear un botón con el que podamos alternar entre los dos temas.

```
<form method="POST">
  <?php
    if (isset($_COOKIE["oscuro"])) {
      echo "<button type='submit'>Modo claro</button>";
    } else {
      echo "<button type='submit'>Modo oscuro</button>";
    }
  ?>
</form>
```

De esta forma si pulsamos el botón cuando la cookie este establecida en oscuro, el botón mostrara el mensaje “Modo claro” y viceversa.

Para que esto funcione lo primero que debemos hacer es iniciar una sesión y luego declarar las variables oscuro y claro. Si se ha enviado información al servidor, se comprobará si la cookie está establecida como “oscuro”. Si es así, la cookie “oscuro” se eliminará y se establecerá la cookie claro, guardándose durante una semana. En caso contrario, se hará totalmente lo contrario.

Por último, se refrescará la página al momento y se saldrá de la condición.

```
session_start();

$claro = "modo claro";
$oscuro = "modo oscuro";

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    if (isset($_COOKIE["oscuro"])) {
        setcookie("oscuro", "", time() - 1);
        setcookie("claro", $claro, time() + 604800);
    } else {
        setcookie("claro", "", time() - 1);
        setcookie("oscuro", $oscuro, time() + 604800);
    }

    header("Refresh:0");
    exit;
}

if (isset($_COOKIE["oscuro"])) {
    echo "Estás en modo oscuro.<br>";
} else {
    echo "Estás en modo claro.<br>";
}
```

3. RA04_c.

Se han utilizado mecanismos para almacenar información en el cliente web y para recuperar su contenido.

- **Crear una página "sesion.php", que tiene los siguientes elementos:**
 - **Campo telefono**
 - **Campo email**
 - **Botón Grabar: Graba los datos en una sesión (los dos campos son requeridos).**
 - **Botón Borrar: Borra todos los datos de la sesión.**
 - **Si la sesión no está establecida (no hay datos en las variables de sesión), se redirige a login.php.**
- **En la página "sesion.php", si ya hay una sesión establecida, los datos de la sesión se cargan en estos campos, si no, los campos se muestran vacíos.**
- **A la página "sesion.php", añadir un desplegable "Horario", con los valores "Mañana", "Tarde" y "Noche". La selección del usuario se guarda en una cookie llamada "horario", que se carga en la página.**
- **Añadir un botón "Grabar horario" para grabar la cookie.**
- **Al pulsar el botón "Borrar", también se borra la cookie.**
- **Si la cookie está establecida, se selecciona el valor en el desplegable.**

Hay que tener en cuenta en donde vamos a colocar nuestro fragmento php dentro de nuestro código de la página. Lo suyo sería ponerlo al principio del código debido a que, si primero ponemos el formulario, lo primero en ejecutarse en el programa será esto y no nuestro php, por lo que no se podrán manejar los datos de las sesiones correctamente.

Después de iniciar la sesión, inicializamos las variables donde vamos a guardar los datos que se nos piden. En caso de que no estén inicializadas, las inicializamos con un espacio en blanco (' ').

```
1 <?php
2 session_start();
3
4 $telefono = $_SESSION['telefono'] ?? ' ';
5 $email = $_SESSION['email'] ?? ' ';
6 $horarioSeleccionado = $_COOKIE['horario'] ?? ' ';
7
```

A continuación, pondremos la lógica de los botones de nuestro formulario. Cuando estos envíen la información correspondiente al servidor, dicha información se guardará en sus correspondientes variables anteriormente creadas.

```

✓ if ($_SERVER['REQUEST_METHOD'] === 'POST') {
✓     if (isset($_POST['grabar']) && !empty($_POST['telefono']) && !empty($_POST['email'])) {
        $_SESSION['telefono'] = $_POST['telefono'];
        $_SESSION['email'] = $_POST['email'];
✓     } elseif (isset($_POST['grabar'])) {
        echo "<p style='color:red;'>Ambos campos son obligatorios.</p>";
        }

✓     if (isset($_POST['grabar_horario']) && isset($_POST['horario'])) {
        setcookie('horario', $_POST['horario'], time() + 3600);
        $horarioSeleccionado = $_POST['horario'];
        }

✓     if (isset($_POST['borrar'])) {
        session_unset();
        session_destroy();
        setcookie('horario', '', time() - 3600);
        header("Location: login.php");
        exit();
        }
    }
}
?>

```

Por último para que todo esto funcione correctamente debemos crear un formulario con todo lo que nos pide el enunciado.

```

<body>
    <form method="post" action="sesion.php">
        <label for="telefono">Teléfono:</label><br>
        <input type="text" id="telefono" name="telefono" value="<?php echo htmlspecialchars($telefono); ?>" required<br><br>

        <label for="email">Email:</label><br>
        <input type="email" id="email" name="email" value="<?php echo htmlspecialchars($email); ?>" required<br><br>

        <label for="horario">Horario:</label><br>
        <select id="horario" name="horario">
            <option value="Mañana" <?php echo ($horarioSeleccionado === 'Mañana') ? 'selected' : '' ?>>Mañana</option>
            <option value="Tarde" <?php echo ($horarioSeleccionado === 'Tarde') ? 'selected' : '' ?>>Tarde</option>
            <option value="Noche" <?php echo ($horarioSeleccionado === 'Noche') ? 'selected' : '' ?>>Noche</option>
        </select><br><br>

        <button type="submit" name="grabar">Grabar</button>
        <button type="submit" name="grabar_horario">Grabar horario</button>
        <button type="submit" name="borrar">Borrar</button>
    </form>
</body>

```


4. RA04_d.

Objetivo: Crear un sistema de autenticación en PHP que permita acceder a diferentes páginas dependiendo del rol del usuario.

Instrucciones:

- Crea una página `login.php` con un formulario de inicio de sesión que tenga tres campos: `usuario`, `contraseña` y `rol`.
- Usa dos usuarios de prueba:
- Usuario 1: Nombre de usuario `admin`, contraseña `admin123`, rol `administrador`.
- Usuario 2: Nombre de usuario `user`, contraseña `user123`, rol `usuario`.
- Cuando el usuario se autentique correctamente:
- Si el rol es `administrador`, redirígelo a `admin.php`.
- Si el rol es `usuario`, redirígelo a `user.php`.
- Si las credenciales son incorrectas, muestra un mensaje “Credenciales incorrectas” y vuelve a cargar el formulario.
- Ninguna de las páginas (`admin.php` o `user.php`) debe ser accesible sin una sesión activa.

Tras iniciar la sesión definimos los usuarios que nos piden con sus contraseñas y sus roles.

```
1 <?php
2 session_start();
3
4 $usuarios = [
5     'admin' => ['password' => 'admin123', 'rol' => 'administrador'],
6     'user' => ['password' => 'user123', 'rol' => 'usuario']
7 ];
8
```

A continuación, como en el ejercicio anterior, creamos unas variables para guardar la información de usuario y contraseña tras ser introducidos por el formulario y si no están inicializados, los inicializamos en blanco ('').

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $usuario = $_POST['usuario'] ?? '';
    $password = $_POST['password'] ?? '';
```

Luego comprobamos si el usuario y contraseña introducidos pertenecen a un rol u otro o si no corresponde a nada de lo que tengamos registrado.

```
    if (isset($usuarios[$usuario]) && $usuarios[$usuario]['password'] === $password) {
        $_SESSION['usuario'] = $usuario;
        $_SESSION['rol'] = $usuarios[$usuario]['rol'];

        if ($_SESSION['rol'] === 'administrador') {
            header("Location: admin.php");
            exit();
        } elseif ($_SESSION['rol'] === 'usuario') {
            header("Location: user.php");
            exit();
        }
    } else {
        $error = "Credenciales incorrectas";
    }
}
?>
```

Por último, creamos el formulario.

```
<body>
    <h2>Inicio de Sesión</h2>
    <?php if (!empty($error)): ?>
        <p style="color:red;"><?php echo $error; ?></p>
    <?php endif; ?>
    <form method="post" action="login.php">
        <label for="usuario">Usuario:</label><br>
        <input type="text" id="usuario" name="usuario" required><br><br>

        <label for="password">Contraseña:</label><br>
        <input type="password" id="password" name="password" required><br><br>

        <label for="rol">Rol:</label><br>
        <select id="rol" name="rol" required>
            <option value="administrador">Administrador</option>
            <option value="usuario">Usuario</option>
        </select><br><br>

        <button type="submit">Iniciar Sesión</button>
    </form>
</body>
```

5. RA04_e.

Se han escrito aplicaciones que integren mecanismos de autenticación de usuarios.

- *Escribir una página llamada "login.php", consistente en dos campos: usuario y contraseña.*
 - *El usuario es "foc", y la contraseña "Fdwes!22".*
 - *Si usuario y contraseña son correctos, se muestra la página web "sesion.php" (ver siguientes apartados).*
 - *Si usuario y contraseña no son correctos, se da el mensaje "Credenciales incorrectas", y se vuelve a cargar el formulario.*
 - *Ninguna página distinta de login.php se carga si no está la sesión establecida (no hay datos en las variables de sesión).*

Este ejercicio es muy similar al que acabamos de hacer.

```
<?php
session_start();

$usuario_correcto = 'foc';
$password_correcto = 'Fdwes!22';
```

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $usuario = $_POST['usuario'] ?? '';
    $password = $_POST['password'] ?? '';

    if ($usuario === $usuario_correcto && $password === $password_correcto) {
        $_SESSION['usuario'] = $usuario;
        header("Location: sesion.php");
        exit();
    } else {
        $error = "Credenciales incorrectas";
    }
}
?>
```

```

<body>
  <h2>Inicio de Sesión</h2>
  <?php if (!empty($error)): ?>
  |   <p style="color:red;"><?php echo $error; ?></p>
  <?php endif; ?>
  <form method="post" action="login.php">
    <label for="usuario">Usuario:</label><br>
    <input type="text" id="usuario" name="usuario" required><br><br>

    <label for="password">Contraseña:</label><br>
    <input type="password" id="password" name="password" required><br><br>

    <button type="submit">Iniciar Sesión</button>
  </form>
</body>
</html>

```

6. RA04_f.

Explica los mecanismos de seguridad que existen con las sesiones, pon ejemplos de ellos, cuales son más seguros.

Los mecanismos de seguridad que se usan con las sesiones en php es la encriptación de la información mediante una serie de mecanismos de encriptado:

sha1:

- **Función:** sha1(string \$str, bool \$raw_output = false)
- **Descripción:** Calcula y devuelve el hash SHA-1 de una cadena.

md5:

- **Función:** md5(string \$str, bool \$raw_output = false)
- **Descripción:** Calcula y devuelve el hash MD5 de una cadena, resultando en un valor hexadecimal de 32 caracteres.

hash:

- **Función:** hash(string \$algo, string \$dato, bool \$raw_output = false)
- **Descripción:** Genera un valor cifrado del dato en base al algoritmo especificado.

crypt:

- **Función:** crypt(string \$str, string \$salt)
- **Descripción:** Genera un valor cifrado usando un salt específico.

PHP recomienda que de todas estas funciones se use la función de crypt.