



DESARROLLO WEB ENTORNO SERVIDOR [DWES]

**TAREA 4
WENCESLAO CRIADO BÁEZ**

ÍNDICE

1. Descripción de la tarea.

2. Inicio.php

3. Sesion.php

4. Aspectos de mejora.

1. Descripción de la tarea.

Crear dos páginas PHP con la siguiente información:

inicio.php

Página de recogida de datos con los siguientes requisitos:

- Esta página contendrá un formulario en el que se mostrará el color favorito del usuario. Este dato se hará persistente en la página mediante una cookie y un check que indique que se quiere guardar.
- Además del campo anterior nuestro formulario tendrá campos para almacenar los datos del usuario: Nombre, apellidos y población. Estos datos se podrán hacer persistentes en una sesión con un check que así lo indique.
- El método a usar por el formulario será POST y el atributo action será inicio.php.
- En caso de que exista una sesión o la cookie contenga información, esta página mostrará los campos del formulario con los contenidos de la cookie y de la sesión.
- Mostrará además un enlace a la página sesion.php

sesion.php

Página que mostrará información sobre la sesión que contendrá la siguiente información:

- Un formulario para cerrar la sesión con un simple botón y con destino la página sesion.php.
- El formulario tendrá un campo oculto llamado apagar y valor 1
- Cuando la página sesion.php reciba por POST el valor apagar = 1 deberá destruir la sesión.
- Si la sesión está activa, la página, mostrará todos los datos almacenados en la sesión

2. Inicio.php

Para la creación de mi página inicio.php, voy a confeccionar en primer lugar la estructura html que contendrá el formulario.

Para ello voy a usar unos div que contendrán tanto el formulario, como los elementos con los que vamos a trabajar, además e indicar el método y el action del mismo.

```
<!-- Div que contiene tod-o mi formulario -->
<div class="contenedor">
  <!-- Div Formulario -->
  <div id="f_datos" class="formulario">
    <form method="POST" action="inicio.php">
```

Destacar que en cuanto al valor de los input, voy a hacer uso de variables php, que por supuesto tendré que inicializar para que su estado no sea Undefined.

A todos los elementos les doy un name para poder identificarlos e interactuar con ellos.

```
<!-- En función del input con php muestro un valor por defecto que es "" o el que haya guardado el usua
<label>Nombre:</label>
<input type="text" placeholder="Nombre" name="nNombre" value="<?php echo $Nombre ?>" />
</div>
<div>
  <label>Apellidos:</label>
  <input type="text" placeholder="Apellidos" name="nApellidos" value="<?php echo $Apellidos?>" />
</div>
```

Por último voy a usar unos checkbox para el control de lo que el usuario ha seleccionado, un submit para controlar el envío de información y por último un enlace a la página sesion.php

```
<!-- Div que contiene los check y el enlace a los datos de sesión -->
<div class="inferior">
  <div class="check"><input type="checkbox" name="chColor"/><span>Guardar Color en Cookie</span></div>
  <div class="check"><input type="checkbox" name="cSesion"/><span>Guardar todo en sesión</span></div>
  <!-- Input para controlar el envío de datos -->
  <input type="submit" name="iControl" value="Guardar" />
  <!-- Enlace a la página que contiene los datos de Sesión -->
  <a href="sesion.php" rel="register" class="linkform">Ir a Datos de Sesión</a>
</div>
```

Esta sería la estructura básica de la página, a continuación voy a detallar las funcionalidades que vamos a conseguir a través de PHP.

Voy a dividir en dos partes la explicación:

- Sesión : Código referente a la sesión y sus funciones.
- Cookie : Código referente a la Cookie y sus funciones.

Sesión

Imprescindible cuando se trabaja con sesiones, iniciar la sesión al principio del documento.

```
<!-- Creación de una sesión -->
```

```
<?php
```

```
session_start();
```

```
// Comprobación de que la variable $_SESSION esté seteada
```

Una vez cumplido tal requisito, empiezo con la comprobación de las variables de sesión.

Lo he estructurado de la siguiente forma:

1. **Comprobar si está seteada la variable de sesión.**
2. **Comprobar si está seteado el POST.**
 - 2.1. **Comprobar si está marcado el check de sesión.**
 - 2.2. **Comprobar si todos los campos están completos.**

Que detallo a continuación

1. Si la variable \$_SESSION usuario está seteada, quiere decir que hay una sesión, por tanto puedo hacer uso de las variables que se han creado y mostrar al usuario la información que necesite. En nuestro caso podemos crear el estado de Persistencia.

```
if(isset($_SESSION["usuario"]))  
{  
    // Variable para mostrar el estado de la sesión  
    $info_sesion = "Sesión Activa";  
    $Nombre = $_SESSION['usuario'];  
    $Apellidos = $_SESSION['apellidos'];  
    $Pob = $_SESSION['poblacion'];  
}
```

En caso de que no esté seteada la variable de sesión, quiere decir que no hay sesión, por lo que tendré que ir al punto 2 que detallo a continuación.

2. Si la variable de sesión no está seteada, entonces empiezo por comprobar si ha habido un POST por parte del usuario, en caso de que la variable \$_POST iControl esté seteada, paso al 2.1 , de lo contrario dejaría mis variables con un valor vacío (""), ya que el usuario no habría hecho SUBMIT.

```
// Variable iControl que permite comprobar si se ha pulsado el submit independientemente de que s  
if (isset($_POST['iControl'])) {  
    // Comprobación de que le check de Sesión está marcado  
    if (isset($_POST['cSesion'])) {
```

2.1 Compruebo si el check de sesión está seteado, en caso de que el usuario haya marcado la opción de “Guardar la Sesión”, pasaría al 2.2, de lo contrario, quiere decir que el usuario ha hecho SUBMIT pero no ha marcado el check de “Guardar Sesión”, aquí hay que volver a inicializar las variables.

```
// No se ha marcado el check de guardar sesión
else
{
    $info_sesion = "Tienes que marcar el check para guardar Sesión";
    $Nombre = "";
    $Apellidos = "";
    $Pob = "";
    $colorCookie = "";
}
```

2.2 Compruebo que todos los campos están completos, de cumplirse esta condición se crearía la sesión y comenzaría a crear las variables de sesión con el contenido que el usuario ha introducido. El contenido de estas variables son las que vamos a mostrar en el value de los inputs. En el caso de que no se cumpla la condición de que todo los campos estén rellenos, al usuario se le mostrará la información del “error” y se volverán a inicializar las variables.

```
// Comprobación de que el check de Sesión está marcado
if (isset($_POST['cSesion'])) {
    // Comprobación de que los campos del formulario no están en blanco
    if (($POST['nNombre']) != "" && ($POST['nApellidos']) != "" && ($POST['nPob']) != "") {
        // Asignación del contenido a las variables en caso de que esté todo o seteado y relleno
        $id = session_id();
        $_SESSION['usuario'] = $_POST['nNombre'];
        $_SESSION['apellidos'] = $_POST['nApellidos'];
        $_SESSION['poblacion'] = $_POST['nPob'];
        $_SESSION['color'] = $_POST['nColor'];
        $info_sesion = " Se han guardado datos de sesión";
        $Nombre = $_SESSION['usuario'];
        $Apellidos = $_SESSION['apellidos'];
        $Pob = $_SESSION['poblacion'];
    }
}
```

Como aclaración quiero indicar que no hay persistencia en la variable de Color, puesto que mi intención era que los datos del usuario fueran persistentes con la sesión y el color con la cookie.

Cookie

En cuanto al apartado de Cookie las comprobaciones se van a efectuar de la siguiente manera:

1. Comprobar si la variable Cookie está seteada
2. Comprobar si el Post está seteado
 - 2.1. Comprobar si está marcado el check de “Guardar Color en Cookie”
 - 2.2. Comprobar si hay un datos para guardar

A continuación lo detallo:

1. Compruebo si la variable \$_COOKIE está seteada, de cumplirse la condición, quiere decir que hay una cookie existente, al usuario muestro la información correspondiente. En caso contrario, pasaríamos al punto 2.

```
// Comprobar que la variable $_COOKIE está seteada
if (isset($_COOKIE['miCookie']))
{
    // Asignar a la variable $colorCookie el contenido guardado en la Cookie "miCookie" y asignar valor
    $colorCookie=$_COOKIE['miCookie'];
    $info= "Hay una cookie guardada con el color ".$_COOKIE['miCookie'];
}
```

2. Comprobar si la variable \$_POST iControl está seteada, en caso de que se cumpla la condición, pasaríamos al punto 2.1, de lo contrario inicializaríamos nuestras variables.

```
    $colorCookie = "";
}
}
// En caso de que no esté seteada la variable COOKIE pongo a "" todas las variables
else
{
    $info = "";
    $Nombre = "";
    $Apellidos = "";
    $Pob = "";
    $colorCookie = "";
}
```

- 2.1 Comprobar si está marcado el check “cSesion”, en caso de que se cumpla la condición, pasaríamos al punto 2.2, de lo contrario quiere decir que ha habido un POST por parte del usuario pero que no ha marcado el check para guardar los datos en la cookie.

```
// En caso de que no se haya marcado el check de Guardar Cookie y pongo a "" todas las variables
else {
    $info = "Tienes que marcar el check para guardar Cookie";
    $Nombre = "";
    $Apellidos = "";
    $Pob = "";
    $colorCookie = "";
}
```

- 2.2 Comprobar si el usuario ha introducido algún valor en el campo del input de color, en caso de que se cumpla la condición se crearía la cookie, de no cumplirse la condición, se le indicaría al usuario que tiene que introducir un color y se inicializarían las variables

```
// Comprobar con iControl que se ha pulsado el SUBMIT
if (isset($_POST['iControl'])) {
    // Comprobar si está marcado el check de guardar color en COOKIE
    if (isset($_POST['chColor'])) {
        // Comprobar si ha escrito algo en el input de COLOR
        if ($_POST['nColor'] != "") {
            // En caso de que todo esté correcto, creamos la cookie y guardamos el valor en las variables
            $colorCookie = $_POST['nColor'];
            setcookie("miCookie", $colorCookie, time() + 3600);
            $info = "Se ha guardado el color " . $colorCookie . " en la Cookie";
        }
    }
}
```

3. Sesion.php

Para la realización de esta página me voy a servir de la misma estructura anterior, voy a utilizar unos div para confeccionar mi formulario y voy a hacer uso de las variables PHP para mostrar al usuario la información correspondiente. Además voy a contar con input de tipo oculto y con un valor dado.

```
<div class="formulario">
    <form action="" method="POST">
        <h3>Datos de Sesión</h3>
        <!-- En función del elemento muestro información al usuario -->
        <label>Info: <span id="lInfo"><?php echo $info ?></span></label>
        <label>Nombre: <span><?php echo $nombre ?></span></label>
        <label>Apellido: <span><?php echo $apellidos ?></span></label>
        <label>Población: <span><?php echo $pob ?></span></label>
        <label>Color: <span><?php echo $color ?></span></label>
        <label>ID: <span><?php echo $id ?></span></label>
        <!-- Div que contiene el Submit y un input oculto con su valor -->
        <div class="inferior">
            <input type="submit" value="Cerrar Sesión" name="cSesion">
            <input type="hidden" value="0" name="s0" >
        </div>
    </form>
</div>
```

Este apartado podría estructurarlo en dos partes

- Sesión con código PHP
- Script para controlar propiedad de un elemento

Sesión

En primer lugar hay que iniciar la sesión al igual que la página anterior.

```
<!-- Inicio de sesión -->
<?php
session_start();
```

Una vez cumplido con el requisito voy a realizar las siguientes comprobaciones:

1. **Comprobar si está seteada la variable de sesión usuario.**
2. **Comprobar si está seteado el Post del input oculto y que el valor sea igual a 0.**

A continuación lo detallo

1. Compruebo si la variable \$_SESSION usuario está seteada, en tal caso quiere decir que hay una sesión activa, en tal caso recupero de las variables de sesión la información que quiero mostrar al usuario. Además establezco un valor a una variable \$css, que posteriormente detallaré el porqué. En caso de no cumplirse la condición quiere decir que no hay sesión activa, por tanto las variables de los inputs se inicializarían a "".

```
// Comprobación de que la variable $_SESSION está seteada
if(isset($_SESSION['usuario']))
{
    // Obtener de las variables de sesión los datos a mostrar
    $id = session_id();
    $nombre = $_SESSION['usuario'];
    $color = $_SESSION['color'];
    $apellidos = $_SESSION['apellidos'];
    $pob = $_SESSION['poblacion'];
    $info= "Sesión Activa";
    // Variable para indicar a través de la propiedad style del elemento el tipo de
    $css=1;
    // En caso de que no esté seteado comprobar que el submit oculto no lo esté y que
```

2. Con la comprobación del input oculto y además de su valor conseguimos que se cumpla siempre la condición al hacer el SUBMIT, en tal caso la sesión y la cookie quedarían destruidas y las variables se inicializarían.

```
if(isset($_POST['s0']) && $_POST['s0'] == 0){
    // Se destruye la sesión y la Cookie , se inicializan variables
    session_destroy();
    setcookie('miCookie',null, -1);
    $info= "Se ha destruido la sesión y Cookie";
    $id = "";
    $nombre = "";
    $apellidos = "";
    $pob = "";
    $color = "";
    // Cambio el valor de la variable a 0 para que muestre el fondo
    $css=0;
}
```

Script

He querido incluir en mi ejercicio una función JS, que se ejecute la cargarse el body de mi página. Le paso como parámetro el id de un elemento, con esta función consigo gracias a la variable \$css cambiar el fondo del elemento en función de si la sesión está activa o no.

```
<body onload="comprobarActivo('lInfo')">
```

En cuanto al intercambio de variables entre PHP y JS, simplemente he tenido que declarar la variable JS primero y el contenido asignarlo a través de <?php echo \$nombrevariable ?>

```
<script>
function comprobarActivo(info) {
    // Guardo en una variable JS el contenido de la variable PHP
    var activo = <?php echo $css ?>;
    // Acceso al elemento
```

Una vez tengo el valor y el id del elemento, hago la comprobación y accedo a su propiedad style.backgroundColor y cambio el color

```
var lInfo = document.getElementById(info);
// Comprobación del valor de la variable y acceder a la propiedad backgroundColor y
if ( activo == 1){
    lInfo.style.backgroundColor = "green";
}
else {
    lInfo.style.backgroundColor = "red";
}
```

4. Aspectos de mejora.

Para concluir la tarea, he querido dotarle de un aspecto visual limpio y claro, para ello me he ayudado de un documento css que he enlazado a mis dos páginas.

Me he servido de algunas clases que he creado con el fin de reutilizar los atributos en ambos documentos.

El resultado final es el siguiente

The image displays three screenshots of a web application interface, illustrating the state of user data and session management.

Screenshot 1: Datos del usuario

This form is used to input user information. It includes the following fields and controls:

- Nombre:** Input field with placeholder text "Nombre".
- Apellidos:** Input field with placeholder text "Apellidos".
- Provincia:** Input field with placeholder text "Población".
- Color:** Input field with placeholder text "Color".
- ☐ Guardar Color en Cookie
- ☐ Guardar Datos Usuario en Sesión
- [Ir a Datos de Sesión](#)
- Guardar** button

Screenshot 2: Datos de Sesión

This form displays session information when no session is active:

- Info:** No hay sesión Activa
- Nombre:**
- Apellido:**
- Población:**
- Color:**
- ID:**
- Cerrar Sesión** button

Screenshot 3: Datos de Sesión

This form displays session information when a session is active:

- Info:** Sesión Activa
- Nombre:** Wence
- Apellido:** Criado
- Población:** Granada
- Color:** Azul
- ID:** ffu5nlvrg7ulljrjqkt40pr46
- Cerrar Sesión** button