**Universidad de Guayaquil**

**Facultad de Ciencias Matemáticas y Físicas**

**Carrera de Software**

**Asignatura:** Desarrollo de Aplicaciones Web.

**Paralelo**: SOF-S-MA-6-1.

**Integrantes del grupo 2:**

* Beltran Santistevan Carlos Steven.
* Reyes Toala Nefi.
* Cantos James.
* Rendon Ivan.
* Gonzales Espinoza.

**Nota: Todos trabajaron en la realización de la tarea.**

**TAREA 1 – SEGUNDO PARCIAL**

**Tema: Técnicas de mantenimiento de estado: Cookies y sesiones.**

Mientras un usuario visita nuestro sitio o aplicación web él puede visitar varias de sus páginas, las técnicas de mantenimiento de estado nos ayudan a poder mantener información que el usuario genera mientras navega y que podremos acceder en cualquiera de esas páginas.

Los portales web reconocen las acciones que el usuario ha realizado anteriormente: si se ha registrado, qué elementos ha visitado, las compras añadidas a la cesta, etc. Y según está información modifican su contenido o funcionalidad. Para conseguir esto, es necesario almacenar esta información temporalmente utilizando ya sean cookies o sesiones.

Para poder revisar estos temas se propone el desarrollo de los siguientes enunciados:

**Parte 1: Cookies**

1. **¿Qué son y dónde se almacenan las cookies?**

Una cookie se usa a menudo para identificar a un usuario. Una cookie es un pequeño archivo que el servidor incrusta en la computadora del usuario. Cada vez que la misma computadora solicita una página con un navegador, también enviará la cookie. Con PHP, puede crear y recuperar valores de cookies. Una cookie es un archivo creado por un sitio de Internet para almacenar información en el equipo, como por ejemplo, tus preferencias cuando visitas ese sitio. Cuando visites un sitio que usa cookies, el sitio puede pedirle a Firefox que guarde una o más cookies en el disco duro. En algunos navegadores, cada cookie es un pequeño archivo, pero en Firefox todas las cookies se guardan en un solo archivo ubicado en la carpeta perfil.

1. **Indique cuáles y para qué sirven los parámetros que puede recibir el método setcookie(….)**

Se crea una cookie con la setcookie( ) función. setcookie (name, value, expire, path, domain, secure, httponly); Solo se requiere el parámetro de nombre. Todos los demás parámetros son opcionales. Con los parámetros se puede especificar una fecha de vencimiento o duración, después de lo cual ya no se envía la cookie. Además, se pueden establecer restricciones a un dominio y ruta específicos, lo que limita el lugar donde se envía la cookie.

1. **Escribe la línea de código php para crear una cookie que dure 3 horas y que sea válida para cualquier directorio.**

<?php

// (43200 es igual a 12 horas) Expira en 12 horas

$cookie\_name = "user";

$cookie\_value = "Grupo 2 ";

setcookie ($cookie\_name, $cookie\_value, time( ) + (43200), "/");

?>

1. **Escribe la línea de código php para crear una cookie que dure 3 días y que sea válida solo para el directorio /productos/nuevos**

<?php

// (86400 \* 3 es igual a 3 días) Expira en 3 días

$cookie\_name = "user";

$cookie\_value = "Adriana Zambrano";

setcookie ($cookie\_name, $cookie\_value, time( ) + (86400 \* 3), "/productos/nuevos ");

?>

1. **Escriba una aplicación de ejemplo de uso de cookies que conste de un formulario en el cual el usuario ingrese o seleccione alguna preferencia, luego se almacene sus preferencias en cookies y en otra página de su aplicación se muestren.**

**Archivo donde se guardan las cookies**

from\_form1.php (no se guardan en el archivo, el código de persistencia está escrito en el)

**Archivo donde se recupera las cookies**

Gonzalez\_Espinoza\_Christian\_Eduardo\_Form\_2.php y se usan en el mismo archivo

**Código (from\_form1.php)**

(Aquí llega la petición desde Gonzalez\_Espinoza\_Christian\_Eduardo\_Form.html)

<?php

//Mis funciones

include 'assets/php/CGE\_Funciones.php';

if(!empty($\_POST['nombre']) && !empty($\_POST['apellido'])){

//Variables por metodo POST

$nombre = $\_POST['nombre'];

$apellido = $\_POST['apellido'];

setcookie("nombre", $nombre ,time()+tiempo(0,0,5,0), "/");

setcookie("apellido",$apellido ,time()+tiempo(0,0,5,0), "/");

header("Location: .$..\..\Gonzalez\_Espinoza\_Christian\_Eduardo\_Form\_2.php");

die();

}else{

echo "Hola que hace, me quiere hackear o que hace";

}

?>

**Código (Gonzalez\_Espinoza\_Christian\_Eduardo\_Form\_2.php)**

**Desde línea 4**

<?php

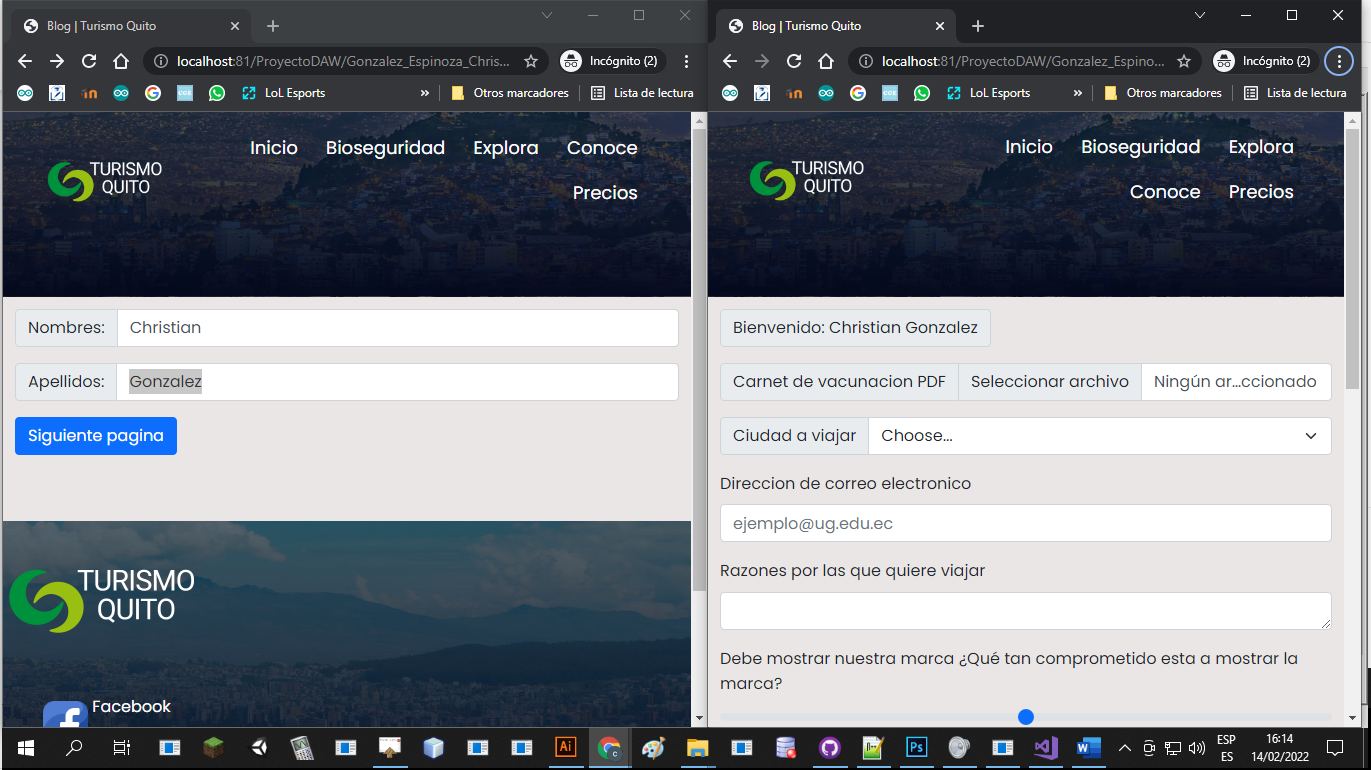
$nombre = $\_COOKIE["nombre"] ;

$apellido = $\_COOKIE["apellido"] ;

?>

**En línea 69**

<span class="input-group-text">Bienvenido: <?php echo $nombre." ".$apellido;?></span>

****

**Parte 2: Sesiones**

1. **Qué son y dónde se almacenan las variables de sesión?**

Las variables de sesión son variables de enganche globales para toda la sesión de inicio. Esto significa que puede establecer la variable de sesión en cualquier tipo de enganche y, posteriormente, leerla de nuevo en cualquier tipo de enganche. El valor persiste para toda la sesión.

Las variables de sesión se almacenan en memoria del servidor y de disco como variables aplicación son. ASP.NET documentación : ASP.NET estado de sesión es compatible con varios opciones de almacenamiento para las variables de sesión. Cada opción se identifica como una Tipo de estado de sesión de modo.

1. **Indique para qué sirven cada uno de los siguientes métodos**
   1. **session\_start()**

Este método permite crear una sesión o reanudar la que se está usando actualmente en un tipo de identificador de sesión pasado por medio de una petición GET o POST o también mediante una cookie. Cuando este método en mención es llamado o invocado se autoinicia una sesión, al realizar esta acción PHP llamará a los gestores de almacenamiento de sesiones open y read. La llamada de retorno read recuperará cualquier información de sesión existente, se almacenará en un formato serializado especial; y será deserializada y usada para rellenar automáticamente la variable superGlobal $SESSION cuando la llamada de retorno read devuelva la información de sesión guardada a la gestión de sesiones de PHP.

* 1. **session\_unset()**

Esta función es de tipo void la cual libera todas las variables de sesión actualmente registradas. Se hace una advertencia al usar esta función porque al usarla deshabilitará el registro de variables de sesión a través del array superGlobal $\_SESSION.

* 1. **sesion\_destroy()**

Esta función de tipo booleana destruye la información asociada con la sesión actual. No destruye ninguna de las variables globales asociadas con la sesión, ni destruye la cookie de sesión. Para volver a utilizar las variables de sesión se debe llamar a session\_start(). Si queremos destruir la sesión de forma completa, como desconectar al usuario, el id de la sesión también debe ser destruido. Si utilizamos una cookie para propagar el id de la sesión entonces, la cookie de sesión se debe eliminar. Lo más idóneo sería utilizar setcookie().

**A diferencia de las cookies, en las sesiones podemos almacenar no solo cadenas de caracteres sino todo tipo de datos. En el ejercicio a continuación se solicita almacenar en sesión un arreglo de objetos.**

1. **Crear una aplicación web que simule el uso de un carrito de compras.  
   Para lograrlo, cree una primera página web donde el usuario pueda seleccionar uno de entre varios productos o servicios y pueda también indicar la cantidad requerida de ese producto. Con estos datos cree un objeto que contenga al menos el nombre, precio y cantidad.**

**Almacene el objeto en un arreglo de objetos en la sesión. Este arreglo de objetos simula el carrito de compras del usuario que está en ese momento utilizando la aplicación.**

**Cada vez que el usuario seleccione un nuevo producto se agregará al arreglo de objetos que está en la sesión el nuevo producto seleccionado.**

**Agregue una página a su aplicación donde se pueda visualizar todos los productos que se han ido almacenando en el carrito de compras (arreglo de objetos de la sesión).**

**Adicionalmente la sesión deberá expirar en 5 minutos.**

**Enlace de Ejemplo arreglos de objetos:** [**https://www.youtube.com/watch?v=k7b9Oq5ZO\_I**](https://www.youtube.com/watch?v=k7b9Oq5ZO_I)