	<p style="text-align: center;"><b>UNIVERSIDAD DON BOSCO</b>  <b>FACULTAD DE INGENIERÍA</b>  <b>ESCUELA DE COMPUTACIÓN</b></p>
<p style="text-align: center;"><b>CICLO I-2021</b></p>	<p style="text-align: center;"><b>Lenguajes Interpretados en el Servidor</b>  <b>Guía de práctica No. 11</b>  <b>Protocolo HTTP, Autenticación de usuarios, cookies y sesiones</b></p>

## I. OBJETIVOS

- Comprender el proceso de autenticar a un usuario para darle o negarle acceso a una página web.
- Emplear la técnica de redirección para redirigir a los usuarios a una página distinta a la solicitada si no se autentican correctamente.
- Emplear variables de sesión para proteger las páginas web de un sitio.
- Hacer uso de funciones de PHP para manejo de sesiones y de la matriz superglobal \$\_SESSION.

## II. INTRODUCCIÓN TEÓRICA

### Preservar el estado en una aplicación web

En una aplicación web, se cuenta con una serie de páginas web o secuencias de comando (scripts) escritos en algún lenguaje del lado del servidor que en mayor o menor medida, podrían estar relacionadas. La característica de estas secuencias de comando es que cada vez que se ejecutan comienzan con un “borrón y cuenta nueva”, esto se debe al modelo de petición-respuesta del protocolo HTTP, caracterizado también por ser un protocolo sin estado (stateless), lo que significa que nunca se almacena información acerca de preferencias del usuario, número de accesos, recursos visitados, etc.

PHP proporciona tres mecanismos para almacenar información entre distintas peticiones HTTP dentro de una aplicación web:

1. Paso de información común en controles ocultos y cadenas de consulta (query strings).
2. Utilización de cookies.
3. Trabajo con sesiones

### Cookies en PHP

Básicamente, las cookies son archivos de texto ASCII que almacenan información siguiendo una estructura básica de pares, de la forma: identificador = valor. El tamaño de un archivo de cookie es relativamente pequeño, no superando los 4 Kb. El modo en que se almacenan, el directorio en el que se guardan y el resto de características propias de una operación de lectura/escritura sobre disco dependen en gran manera del sistema operativo y del navegador que se utilice para navegar en la aplicación web.

La posibilidad de hacer uso de cookies depende de que el navegador cuente con la característica de usar cookies y que además, esa característica esté habilitada. La estructura básica de una cookie se muestra en la siguiente tabla:

Elemento	Significado
Nombre	Indica el nombre que se le ha dado a la cookie.
Valor	Indica el valor de la cookie; es decir, el contenido que tiene.
Caducidad	Indica cuál es el tiempo de validez de la cookie.
Dominio	Indica el rango de dominios en los cuales es válida la cookie.
Ruta	Contiene el directorio a partir del cual la cookie tiene validez.
Seguro	Indica que la cookie será transmitida únicamente por un canal seguro SSL.

### Utilización de cookies en PHP

PHP dispone de una sola función para el manejo básico de cookies. Su sintaxis se muestra a continuación: `bool setcookie(string $name [, string $value [, int $expire [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]]);`

### Creación de cookies

Como se puede observar en la sintaxis de la función `setcookie()`, cada uno de los argumentos coincide con los elementos que componen la estructura básica de una cookie. También puede observar que el único argumento obligatorio es el nombre de la cookie, el resto son opcionales. Sin embargo, en el caso de la creación, también será necesario que se le asigne un valor inicial.

### Eliminación de cookies

Para borrar una cookie se utiliza la misma función que para crearla: `setcookie()`, enviando como único parámetro en la llamada el nombre de la cookie. `setcookie("nombre_cookie");`

### Consultar contenidos de una cookie

Para poder acceder a los contenidos de una cookie se utiliza la matriz superglobal asociativa `$_COOKIE`. Esta matriz contiene todas las variables pasadas a través de cookies, tal y como sucede con las matrices superglobales `$_POST` y `$_GET`. En versiones anteriores a la versión 4.2.0 de PHP se utiliza la matriz superglobal `$HTTP_COOKIE_VARS`.

Ejemplo: En el siguiente ejemplo se muestra cómo crear un seguimiento del número de accesos a una página de un sitio web.

Primer script:

```
<?php
$accesos = 1;
```

```

if(isset($_COOKIE['num_accesos'])){
    $accesos = $_COOKIE['num_accesos'] + 1;
}
setcookie("num_accesos", $accesos, time() + 3600);
?>
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8" />
<title>Control de accesos al sitio web</title>
</head>
<body>
<div id="info">
<h2>Trabajando con cookies</h2>
<h3>Contador de accesos</h3>
<?php
if($accesos > 1):
echo "Has ingresado a esta página <em>" . $accesos . "</em> veces";
else:
    echo "Es la primera vez que accedes a esta página";
endif;
?>
<br /><br />
<a href="<?php echo $_SERVER['PHP_SELF']; ?>">Actualizar</a>
<a href="eliminarcookies.php">Eliminar</a>
</div>
</body>
</html>

```

Segundo script:

```

<?php
//Eliminar la cookie
setcookie("num_accesos");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>Eliminar cookies</title>
</head>
<body>
<div id="info">
<h2>Trabajando con cookies</h2>
<h3>Contador de accesos borrado</h3><br /><br />
<pre> <a href="accesoscookies.php">Volver</a>

```

```
</pre> </div>
</body>
</html>
```

### Autenticación de usuarios

La autenticación de usuarios es uno de los retos importantes a nivel de páginas web ya que este proceso permitirá tener sitios web mucho más seguros, sobre todo si estos manejan información confidencial. La autenticación o autentificación consiste principalmente en verificar la identidad del usuario que pretende ingresar a un sitio web; es decir, comprobar que es quien dice ser.

Se tienen varias estrategias de autenticación de usuarios, entre ellas:

- a) Autenticación HTTP con Apache Web Server
- b) Autenticación HTTP con PHP
- c) Autenticación con variables de sesión de PHP

### Autenticación HTTP con Apache Web Server

El servidor Apache incorpora una gran cantidad de módulos de autenticación diferentes que pueden ser utilizados para controlar el acceso a los archivos y páginas web alojadas en dicho servidor. El módulo más sencillo de utilizar es el módulo `mod_auth` que opera comparando pares de nombre y contraseña con líneas de un archivo de texto incluido en el servidor, donde aparecen los usuarios y contraseñas de los usuarios autorizados a navegar en una carpeta del servidor.

Las directivas necesarias para configurar el acceso con autenticación deben ser introducidas en el archivo de configuración principal del servidor (para nuestro caso Apache, en el archivo `httpd.conf`) o en archivos de configuración almacenados en carpetas o directorios específicos (`.htaccess`).

En esta práctica de laboratorio abordaremos el segundo método. Para esto se requerirá que en el archivo de configuración del servidor web (`httpd.conf`) se edite dentro de la sección `<Directory /> ... </Directory>` de dicho archivo la línea donde aparece la directiva `AllowOverride`, la cual es utilizada para indicar cuáles directivas, en caso de existir, pueden ser colocadas en los archivos de configuración por directorios o carpetas. Localice la sección siguiente dentro del archivo de configuración de su servidor `httpd.conf`:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    Satisfy all
</Directory>
```

La línea destacada en negrita y color diferente debe ser cambiada por la siguiente: `AllowOverride AuthConfig`, de modo que se archivo de configuración debería lucir de la siguiente forma:

```
<Directory />
```

```

        Options FollowSymLinks
#       AllowOverride None
        AllowOverride AuthConfig
        Order deny,allow
        Deny from all
        Satisfy all
</Directory>

```

### Autenticación HTTP con PHP

Este método consiste en comprobar las credenciales del usuario dentro del mismo código PHP. En otras palabras no será necesario modificar ningún archivo de configuración dentro del servidor web. Sin embargo, será necesario enviar ciertos encabezados especiales al servidor web. Estos encabezados son: `header("WWW-Authenticate:Basic realm=\"Protected Area\"");` `header("HTTP/1.0 401 Unauthorized");`

El código 401 del protocolo HTTP significa no autorizado, ante lo cual el navegador debería presentar un diálogo de autenticación. Algunos navegadores lo mostrarán un número infinito de veces hasta que se introduzcan los datos correctos, otros, como el Internet Explorer solo lo harán 3 veces, después de lo cual lanzará el mensaje de usuario no autorizado.

Para realizar este tipo de autenticación es preciso utilizar la matriz superglobal `$_SERVER` para examinar el valor de sus índices `PHP_AUTH_USER` y `PHP_AUTH_PW` que contienen el usuario y la contraseña que ha introducido un usuario en la ventana modal del explorador.

Si se desea un script que tome en cuenta si se está ejecutando PHP en un servidor web Apache o Internet Information Server, deberá ejecutar un código que conllevar mayor complejidad, sobre todo por el uso de otra variable de servidor como `$_SERVER['HTTP_AUTHORIZATION']`, si es que está disponible. Esta variable contiene una cadena que inicia con Basic y luego de un espacio en blanco, una cadena codificada en base 64, con el nombre de usuario y la contraseña ingresada por este. El código puede ser como el mostrado a continuación:

```

<?php
if(isset($_SERVER['PHP_AUTH_USER'])){
    $user = $_SERVER['PHP_AUTH_USER'];
    $pass = $_SERVER['PHP_AUTH_PW'];
}
elseif(isset($_SERVER['HTTP_AUTHORIZATION'])){
    if(substr($_SERVER['HTTP_AUTHORIZATION'], 0, 5) == "Basic"){
        $userpass = split(":", base64_decode(substr($_SERVER['HTTP_AUTHORIZATION'], 6)));
        $user = $userpass[0];
        $pass = $userpass[1];
    }
}
if(isset($user)){

```

```

echo "Username / Password: ";
        echo htmlspecialchars($user) . " " . htmlspecialchars($pass);
    }
    else{
header("WWW-Authenticate:Basic realm=\"Protected Area\"");
header("HTTP/1.0 401 Unauthorized");
    }
    ?>

```

### **Autenticación con sesiones**

Una definición simple de lo que es una sesión sería el tiempo que un usuario permanece conectado a un sitio web. Una definición más técnica y relacionada a la programación sería: “una sesión es un bloque de información que almacena todo tipo de variables y valores relacionados con los usuarios y sus visitas a un sitio web en particular”. Lo que una sesión posibilita es la capacidad de guardar variables (incluyendo matrices y clases) entre ejecuciones de scripts para poder acceder a estas desde cualquier página donde se inicie sesión.

### **Iniciar la sesión**

Existen dos formas de iniciar sesiones en PHP: La primera es utilizando la función `session_start()`, con la cual se puede crear un identificador de sesión nuevo, o retomar un id de sesión creado previamente. Se requiere que utilice la llamada a esta función al inicio de todas las secuencias de comando que vayan a utilizar sesiones.

La segunda forma, consiste en configurar el archivo `php.ini` para que inicie sesiones de forma automática cuando alguien visite el sitio. Para ello se debe asignar el valor de uno a la variable de configuración `session.auto_start`, cuyo valor por defecto es cero. Este método tiene el inconveniente que no permite utilizar objetos como variables de sesión.

### **Registrar las variables de sesión**

Debe utilizarse la matriz superglobal `$_SESSION` para crear una variable de sesión. Una variable de sesión se crea cuando se coloca como índice de la matriz superglobal `$_SESSION` el nombre de alguna variable, pero sin el símbolo de dólar. Esto es similar a como se crean variables pasadas por la URL con la matriz superglobal `$_GET` o campos de formulario con el método POST utilizando la matriz superglobal `$_POST`.

Un ejemplo de registro de una variable de sesión es el siguiente:

```

$_SESSION['estado'] = 1;
$_SESSION['page'] = $pagina;
$_SESSION['nameusr']=$row['nom'] . " " . $row['ape'];
$_SESSION['perfil'] = $row->perfil;
$_SESSION['photo'] = $preferencias['img'];

```

### **Utilizar variables de sesión**

Es preciso para poder acceder a los valores almacenados en variables de sesión iniciar el script con una llamada a la función `session_start()`. Si se utiliza un objeto como variable de sesión, es preciso incluir la definición de la clase antes de invocar a la función `session_start()` para volver a cargar las variables de sesión. De esta forma PHP sabe cómo reconstruir el objeto de sesión. Debe prestar atención a verificar si las variables de sesión están registradas o no. Puede utilizar para ello las funciones `isset()` o `empty()`. Por ejemplo: `isset($_SESSION['name']);`

#### **Anular las variables de sesión**

Si ya no se va a utilizar una variable de sesión es aconsejable anular su registro. Esto ayudará a liberar recursos del servidor. Para anular el registro de una variable de sesión puede utilizar la función `unset`.

Ejemplo: `unset($_SESSION['page']);`

**NOTA IMPORTANTE:** El uso de las funciones `session_unregister()` y `session_unset()` ya no es necesario ni recomendable.

Para anular el registro de todas las variables utilice: `$_SESSION = array();`

#### **Eliminar la sesión**

Cuando la sesión se termine debe anular el registro de todas las variables de sesión y a continuación invocar la función `session_destroy()`. La llamada a esta función borra el id de sesión.

**IMPORTANTE:** A pesar de que es una práctica recomendada anular el registro de variables de sesión y destruir la sesión, cuando se cierre la ventana del navegador las sesiones serán destruidas automáticamente por PHP cuando se acabe el script o cuando se cierre la ventana del navegador.

### **III. MATERIALES Y EQUIPO**

Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Material	Cantidad
1	Guía de práctica #11: Protocolo HTTP, autenticación de usuarios cookies y sesiones	1
2	Computadora con WampServer instalado y funcionando correctamente	1
3	Editor PHP sublime Texto Eclipse PHP	1
4	Memoria USB o disco flexible	1

#### IV. PROCEDIMIENTO

Digite los scripts que se enumeran a continuación y guárdelos en el formato apropiado como páginas .html o como scripts .php según se indique.

##### Ejercicio #1:

El siguiente ejercicio implementa la autenticación básica con PHP.

Script 1: autenticacionbasica.php

```
<?php
//Para trabajar con sesiones
session_start();
//Si el usuario está usando IIS, se necesita establecer
//las variables globales $PHP_AUTH_USER y $PHP_AUTH_PW
if(substr($_SERVER['SERVER_SOFTWARE'], 0, 9) == 'Microsoft' &&
    !isset($_SERVER['PHP_AUTH_USER']) &&
    !isset($_SERVER['PHP_AUTH_PW']) &&
    substr($_SERVER['HTTP_AUTHORIZATION'], 0, 6) == 'Basic '){
    list($_SERVER['PHP_AUTH_USER'], $_SERVER['PHP_AUTH_PW']) =
    explode(':', base64_decode(substr($_SERVER['HTTP_AUTHORIZATION'],
    6)));
    }
    //Si no se está usando un archivo con usuario y contraseña en el
    servidor
    //utilizar estos datos de usuario y contraseña
    if($_SERVER['PHP_AUTH_USER'] != 'lis' || $_SERVER['PHP_AUTH_PW']
    != 'ingenieria'){
        //Si el visitante no ha introducido sus datos o si los
        //datos proporcionados no son correctos redirigirlo a
        //la ventana de autenticación básica HTTP
        header('WWW-Authenticate: Basic realm="Realm-Name"');
        if(substr($_SERVER['SERVER_SOFTWARE'],0,9) == 'Microsoft'){
            header('Status: 401 Unauthorized');
        }
        else{
            header('HTTP/1.0 401 Unauthorized');
        }
        $msgden = "<h2
    style=\"font-family:Impact;font-size:15pt;color:Red;\">";
```



```

        $msgden .= "No tienes acceso a este sitio</h2>";
        echo $msgden;
    }
    else{
        //Si estamos acá es porque el usuario introdujo los datos
correctos
        $_SESSION['user'] = $_SERVER['PHP_AUTH_USER'];
        $_SESSION['pass'] = $_SERVER['PHP_AUTH_PW'];
        header("Location: material.php");
    }
?>

```

Script 2: material.php

```

<?php
    session_start();
    if(!isset($_SESSION['user']) || !isset($_SESSION['pass'])){
        header("Location: autenticacionbasica.php");
    }
    else{
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.
min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwi
GgFAW/dAiS6JXm" crossorigin="anonymous">
    <title>Descarga de materiales de la materia LIS</title>

</head>
<body class="container">
<header>
<div class="row">
    <div class="col-9">

```

```

    <h1>Lenguajes Interpretados en el Servidor</h1>
  </div>
  <div class="col-3 float-right">
    <a href="logout.php" class="btn btn-primary btn-lg "
role="button" aria-disabled="true">Cerrar sesión</a>
  </div>
</div>
</div>
</header>
<section>
<article>
  <ul class="list-group">
    <li class="list-group-item active">Clases</li>
    <li class="list-group-item ">
      <a
href="http://www.mediafire.com/download/ccyra1rm614t05a/Clase+01+--+P
rogramaci%C3%B3n+web+del+lado+del+servidor+--+2014.pdf"
target="_blank">
        Clase 01: Programación web del lado del servidor
      </a>
    </li>
    <li class="list-group-item ">
      <a
href="http://www.mediafire.com/download/4le9g50t1d3wng1/Clase+02+--+I
ntroducci%C3%B3n+a+la+programaci%C3%B3n+y+sintaxis+de+PHP+--+2014.pdf
" target="_blank">
        Clase 02: Introducción a la programación y sintaxis
de PHP
      </a>
    </li>
    <li class="list-group-item ">
      <a
href="http://www.mediafire.com/download/9zg2du274b6d3fu/Clase+03+--+E
structuras+de+control+sentencias+condicionales+y+repetitivas+--+2014.
pdf" target="_blank">
        Clase 03: Estructuras de control - Sentencias
condicionales y repetitivas

```

```

        </a>
    </li>
</ul>
    <ul class="list-group">
    <li class="list-group-item active">Guías de práctica</li>
        <li class="list-group-item ">
            <a
href="http://www.udb.edu.sv/udb/archivo/guia/informatica-tecnologico
/lenguajes-interpretados-en-el-servidor/2014/i/guia-1.pdf"
target="_blank">
                Guía 01: Introducción a la Programación web con PHP
            </a>
        </li>
        <li class="list-group-item ">
            <a
href="http://www.udb.edu.sv/udb/archivo/guia/informatica-tecnologico
/lenguajes-interpretados-en-el-servidor/2014/i/guia-2.pdf"
target="_blank">
                Guía 02: Estructuras de Control - Sentencias
condicionales y repetitivas
            </a>
        </li>
        <li class="list-group-item ">
            <a
href="http://www.udb.edu.sv/udb/archivo/guia/informatica-tecnologico
/lenguajes-interpretados-en-el-servidor/2014/i/guia-3.pdf"
target="_blank">
                Guía 03: Matrices y funciones en PHP
            </a>
        </li>
    </ul>

    <ul class="list-group">
    <li class="list-group-item active">Sitios web</li>
        <li class="list-group-item ">
            <a href="http://www.php.net/manual/es" target="_blank">
                Sitio web oficial de PHP
            </a>
        </li>
    </ul>

```

```

        </a>
    </li>
    <li class="list-group-item ">
        <a
href="http://www.manualdephp.com/section/manual-de-php/"
target="_blank">
            Manual de PHP
        </a>
    </li>
</ul>
</article>
</section>
</body>
</html>
<?php
    }
?>

```

Script 2: logout.php

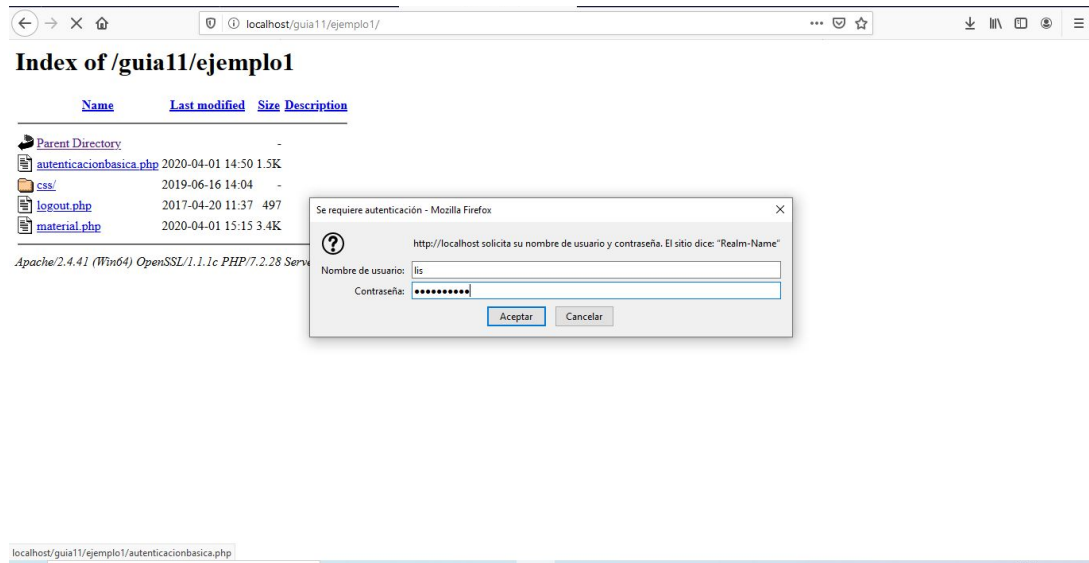
```

<?php
    session_start();
    unset($_SESSION['user']);
    unset($_SESSION['pass']);
    $_SESSION = array();
    session_destroy();
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Fin de la sesión</title>
</head>
<body>
<header>
    <h1>Has salido del sistema</h1>

```

```
</header>
<section>
<article>
  <p>
    Para reingresar haz clic en <a
href="autenticacionbasica.php">
    este enlace</a>.
  </p>
</article>
</section>
</body>
</html>
```

Después de iniciar sesión en una ventana emergente del navegador:



← → ↻ ⓘ localhost/guia11/ejemplo1/material.php ☆ ⓘ

## Lenguajes Interpretados en el Servidor

Cerrar sesión

<b>Clases</b>
Clase 01: Programación web del lado del servidor
Clase 02: Introducción a la programación y sintaxis de PHP
Clase 03: Estructuras de control - Sentencias condicionales y repetitivas
<b>Guías de práctica</b>
Guía 01: Introducción a la Programación web con PHP
Guía 02: Estructuras de Control - Sentencias condicionales y repetitivas
Guía 03: Matrices y funciones en PHP
<b>Sitios web</b>
Sitio web oficial de PHP
Manual de PHP

← → ↻ ⓘ localhost/guia11/ejemplo1/logout.php ... ⓘ ☆ ⬇ ⌨ 🔍 ☰

### Has salido del sistema

Para reingresar haz clic en [este enlace](#).

### Ejercicio #2:

Implementación de un carrito de compra básico usando cadenas de consulta (query string) Script 1: funciones.php

```
<?php
function generarCarrito(){
    //Se utilizará una matriz para manejar el carrito
    $carrito = array();
    //Los artículos y sus cantidades se enviarán con el
    //método GET, ya sea en la cadena de consulta o a
    //través de campos ocultos de formulario
    foreach($_GET as $ref => $unidades){
        if(preg_match("/^ref/", $ref)) //Expresión regular
            $carrito[$ref] = $unidades;
    }
}
```

```

    }
    return $carrito;
}

function mostrarCarrito($carrito){
    //Generación de la cabecera de la tabla
    echo "<table class='table ' border=\"1\" align=\"center\">\n";
    echo "<tr class='bg-info'>\n<th>\nReferencia</th>\n";
    echo "<th>\nUnidades</th>\n</tr>\n";
    //Mostramos el carrito
    $totalUnidades = 0;
    if(empty($carrito)){
        echo "<tr>\n<td align=\"center\" colspan=\"2\">\n";
        echo "El carrito está vacío\n</td>\n</tr>\n";
    }
    else{
        foreach($carrito as $ref => $unidades){
            echo "<tr >\n<td>\n$ref</td>\n";
            echo "<td align=\"center\">$unidades</td>\n</tr>\n";
            $totalUnidades += $unidades;
        }
    }
    //Cerrar la tabla
    echo "<tr class='bg-light'><td align=\"center\"
colspan=\"2\">\n";
    echo "Número de unidades: ";
    echo $totalUnidades . "</td>\n</tr>\n";
    echo "</table>\n";
    return true;
}

//Método que muestra los artículos disponibles en la tienda
function estantes($carrito){
    //Generación del query string que contiene las referencias
    //de los productos y las cantidades a llevar de cada uno
    $querystring = "";
    foreach($carrito as $ref => $unidades){

```

```

        $querystring .= "$ref=$unidades&";
    }
    echo"
    <div class='row'>
    <div class='col'>
        <div class='card'>
            <div class='card-header bg-info'>ref1</div>
            <div class='card-body'>Descripcion: Artículo 1<br>Precio:5
&euro;</div>
            <div class='card-footer'><a
href='./compra.php?{$querystring}articulo=ref1' title='Añadir al
carrito'>Comprar</a></div>
        </div>
    </div>
    <div class='col'>
        <div class='card'>
            <div class='card-header bg-info'>ref2</div>
            <div class='card-body'>Descripcion: Artículo
2<br>Precio:3 &euro;</div>
            <div class='card-footer'><a
href='./compra.php?{$querystring}articulo=ref2' title='Añadir al
carrito'>Comprar</a></div>
        </div>
    </div>
    <div class='col'>
        <div class='card'>
            <div class='card-header bg-info'>ref3</div>
            <div class='card-body'>Descripcion: Artículo 3<br>Precio:2
&euro;</div>
            <div class='card-footer'><a
href='./compra.php?{$querystring}articulo=ref3' title='Añadir al
carrito'>Comprar</a></div>
        </div>
    </div>";
    return true;
}

```



```
?>
```

Script 2: tienda.php

```
<?php
    require_once("funciones.php");
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.
min.css"
integrity="sha384-Gn5384xqQ1aowXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwi
GgFAW/dAiS6JXm" crossorigin="anonymous">

    <title>PHP [hidden:urls]: tienda.php</title>
</head>
<body class="container">
<header>
    <h1 align="center">Carrito de compra</h1>
</header>
<section>
<article>
<?php
    //Recuperando objetos pertenecientes al carrito
    $carrito = generarCarrito();
    if(isset($_GET['articulo']))
        //Incrementar el número de unidades del artículo elegido
        if(empty($carrito[$_GET['articulo']])):
            $carrito[$_GET['articulo']] = 1;
        else:
            $carrito[$_GET['articulo']]++;
        endif;
    //Mostrar los artículos disponibles
```

```

    estantes($carrito);
    echo "<hr />\n";
    //Mostrar el contenido del carrito
    mostrarCarrito($carrito);
?>

<hr />
<form action="./caja.php">
    <div align="center">
        <?php
            $hidden = "";
            //Generar los controles ocultos con las variables
            //comunes de la aplicación
            foreach($carrito as $ref => $unidades){
                $hidden .= "<input type=\"hidden\" name=\"$ref\"
value=\"$unidades\" />\n";
            }
            echo $hidden;
        ?>
        <button type="submit">Caja</button>
    </div>
</form>
</article>
</section>
</body>
</html>

```

Script 3: caja.php

```

<?php
    //Incluir librería con las funciones auxiliares
    require_once("funciones.php");
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />

```

```

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.
min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwi
GgFAW/dAiS6JXm" crossorigin="anonymous">

    <title>PHP [hidden y urls]: caja.php</title>
</head>
<body class="container">
<section>
<article>
<?php
    echo "<h1 align=\"center\">Contenido del carrito</h1>\n";
    //Recuperar los objetos pertenecientes al carrito
    $carrito = generarCarrito();
    //Mostrar el contenido
    mostrarCarrito($carrito);
?>
<p align="center">
    Pulsa <a href="./tienda.php">aquí</a> para continuar.
</p>
</article>
</section>
</body>
</html>

```

Script 4: compra.php

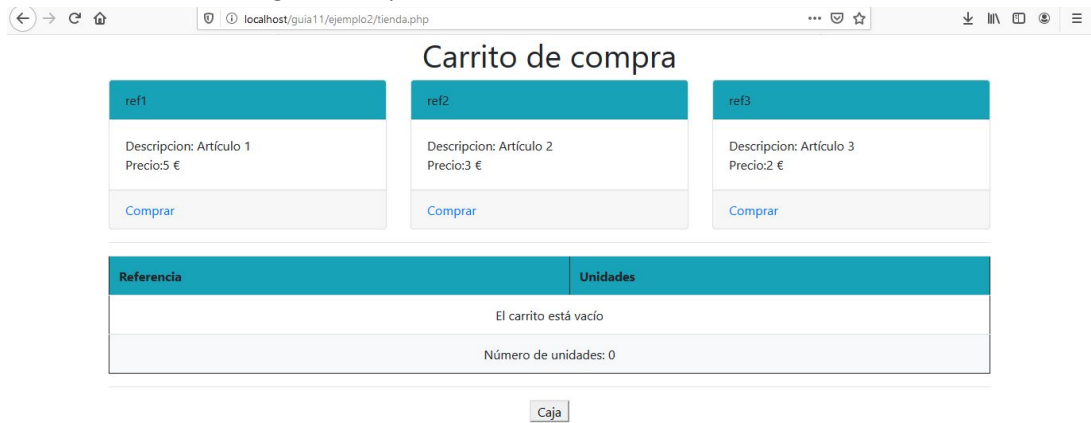
```

<?php
    //Comprobar que hay suficiente producto en stock
    //para satisfacer los productos y cantidades guardados
    //en $carrito. Esto lo implementará ud.
    $tienda = dirname("http://" . $_SERVER['SERVER_NAME'] .
        $_SERVER['SCRIPT_NAME']) . '/tienda.php?' .
        $_SERVER['QUERY_STRING'];
    header("Location: $tienda");

```



Al visualizar en el navegador la aplicación debería verse así:



## V. DISCUSIÓN DE RESULTADOS

1. Determine un mecanismo para hacer que el ejercicio del carrito considere el precio de cada artículo para obtener además de la cantidad de artículos por producto el total a pagar por esa cantidad de artículos, así como el total de todos los productos seleccionados. Puede utilizar bases de datos, sesiones o cookies o cualquier combinación para producir el resultado deseado. En otras palabras la tabla de resultados deberá lucir como esta:

Referencia	Unidad	Precio
ref1	1	5 €
ref2	2	6 €
ref3	4	8 €
Total	7	19 €

## VI. BIBLIOGRAFÍA

- Gutiérrez, Abraham / Bravo, Ginés. PHP 5 a través de ejemplos. Editorial Alfaomega RAMA. 1ra edición. México. Junio 2005.
- Gil Rubio, Francisco Javier/Villaverde, Santiago Alonso/Tejedor Cerbel, Jorge A. Creación de sitios web con PHP 5. Editorial McGraw-Hill. 1ra edición. Madrid, España, 2006.
- John Coggeshall. La Biblia de PHP 5. 1ra Edición. Editorial Anaya Multimedia. Madrid España.
- <http://www.php.net/manual/en>