
Realicé esta práctica en Ubuntu, ya que en Windows me dió problemas.

Primero actualizar el sistema con:

sudo apt-get update

sudo apt-get upgrade (este tarda un rato).

En caso de que tengas el xampp instalado es preferible desinstalarlo para evitar errores.

Por si acaso le damos permisos a la carpeta del xampp para poder modificarla

sudo chmod 777 /opt/lampp -R

cd /opt/lampp

sudo ./uninstall

Borramos también la carpeta del Xampp.

sudo rm -r /opt/lampp

Ahora ejecutamos el archivo `instalar_php_apache.sh`:

desde consola, primero vamos a la carpeta donde se encuentra el archivo, en mi caso esta en descargas en la carpeta ejercicio, **cd Descargas/ejercicio**, lo siguiente es instalarlo,

sudo sh instalar_php_apache.sh

```
zell@zell-VirtualBox:~$ cd Descargas/ejercicio
zell@zell-VirtualBox:~/Descargas/ejercicio$ sudo sh instalar_php_apache.sh
[sudo] contraseña para zell:
```

desde la carpeta botón derecho **run as a program**, a mí no me da esta opción así que lo hice desde la consola.

Cuando acabe, volvemos al inicio con **cd ~**

```
zell@zell-VirtualBox:~/Descargas/ejercicio$ cd ~
zell@zell-VirtualBox:~$
```

Instalamos apache con el comando **sudo apt install apache2**

Comprobamos que se ha instalado con **sudo systemctl status apache2** (pulsa q para salir)

Si su estado es inactivo, con **sudo systemctl start apache2** lo activamos.

```
zell@zell-VirtualBox:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.52-1ubuntu4.7).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libept1.6.0 libflashrom1 libftdi1-2 libgc1 libxapian30
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 424 no actualizados.
zell@zell-VirtualBox:~$
```

```

zell@zell-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
   Active: active (running) since Wed 2024-01-17 23:41:34 CET; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 19191 (apache2)
     Tasks: 6 (limit: 4606)
    Memory: 10.1M
       CPU: 64ms
    CGroup: /system.slice/apache2.service
            └─19191 /usr/sbin/apache2 -k start
              └─19193 /usr/sbin/apache2 -k start
                └─19194 /usr/sbin/apache2 -k start
                  └─19195 /usr/sbin/apache2 -k start
                    └─19196 /usr/sbin/apache2 -k start
                      └─19197 /usr/sbin/apache2 -k start

ene 17 23:41:34 zell-VirtualBox systemd[1]: Starting The Apache HTTP Server...
ene 17 23:41:34 zell-VirtualBox apachectl[19190]: AH00558: apache2: Could not r>
ene 17 23:41:34 zell-VirtualBox systemd[1]: Started The Apache HTTP Server.
zell@zell-VirtualBox:~$

```

Permisos en el firewall

Con `sudo ufw app list`, vemos las app a las que podemos dar los permisos Apache estará entre ellas.

Le damos permiso con `sudo ufw allow 'Apache'`. (estas comillas dan error, cambiarlas antes de ejecutar comando, son las simples)

Con `sudo ufw status`, comprobamos que el estado es inactivo.

```

zell@zell-VirtualBox:~$ sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS
zell@zell-VirtualBox:~$ sudo ufw allow 'Apache'
ERROR: Puerto erróneo
zell@zell-VirtualBox:~$ sudo ufw allow 'Apache'
Reglas actualizadas
Reglas actualizadas (v6)
zell@zell-VirtualBox:~$

```

Lo siguiente es instalar MySQL.

`sudo apt install mysql-server`

Iniciamos MySQL

`sudo systemctl start mysql.service`

```

zell@zell-VirtualBox:~$ sudo apt install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mysql-server ya está en su versión más reciente (8.0.35-0ubuntu0.22.04.1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no
son necesarios.
  libept1.6.0 libflashrom1 libftdi1-2 libgc1 libxapian30
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 424 no actualizados.
zell@zell-VirtualBox:~$ sudo systemctl start mysql.service

```

sudo mysql

Y creamos el usuario root con la contraseña ''

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY '';

(estas comillas dan error, cambiarlas antes de ejecutar comando, son las simples)

Salimos de MySQL

exit

```
zell@zell-VirtualBox:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.35-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY ''
;
Query OK, 0 rows affected (0,02 sec)

mysql> exit
Bye
zell@zell-VirtualBox:~$
```

Para identificarse como el usuario root en MySQL escribimos **mysql -u root -p**

Introducimos la contraseña y entra en MySQL.

Ahora creamos la base de datos.

Copiar y pegar el archivo .sql

Para comprobar que sea creado **SELECT * FROM familia;**

```
mysql> INSERT INTO `usuarios` (`usuario`, `contrasena`) VALUES
-> ('dwes', 'e8dc8ccd5e5f9e3a54f07350ce8a2d3d');
Query OK, 1 row affected (0,01 sec)

mysql> SELECT * FROM familia;
+-----+-----+
| cod   | nombre                                     |
+-----+-----+
| CAMARA | Cámaras digitales                         |
| CONSOL | Consolas                                 |
| EBOOK  | Libros electrónicos                       |
| IMPRES | Impresoras                               |
| MEMFLA | Memorias flash                           |
| MP3    | Reproductores MP3                        |
| MULTIF | Equipos multifunción                     |
| NETBOK | Netbooks                                 |
| ORDENA | Ordenadores                             |
| PORTAT | Ordenadores portátiles                   |
| ROUTER | Routers                                  |
| SAI    | Sistemas de alimentación ininterrumpida |
| SOFTWA | Software                                |
| TV     | Televisores                             |
| VIDEOC | Videocámaras                             |
+-----+-----+
15 rows in set (0,00 sec)

mysql>
```

Ahora creamos el Usuario dwes para poder conectarnos a la BBDD.

CREATE USER IF NOT EXISTS `dwes` IDENTIFIED BY 'abc123.';

GRANT ALL ON `dwes`.* TO 'dwes';

FLUSH PRIVILEGES;

Salimos de MySQL **exit**

(ojo con las comillas)

```
mysql> CREATE USER IF NOT EXISTS `dwes` IDENTIFIED BY 'abc123.';
Query OK, 0 rows affected (0,01 sec)

mysql> GRANT ALL ON `dwes`.* TO 'dwes';
Query OK, 0 rows affected (0,02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,00 sec)

mysql> exit
Bye
zell@zell-VirtualBox:~$
```

Volvemos a actualizar el sistema

sudo apt-get update

sudo apt-get upgrade

Reiniciamos el servidor Apache

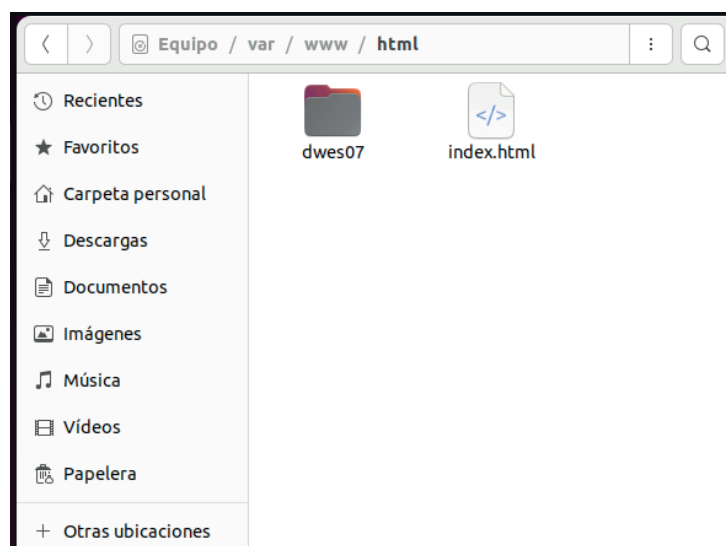
sudo systemctl restart apache2

Cerramos la consola y en el explorar de archivos **Otras ubicaciones** → **var** → **www** → **html**, aquí es donde ponemos la carpeta con el ejercicio (es el equivalente al htdocs del Xampp).

Si no deja pegar, abrimos la consola y con

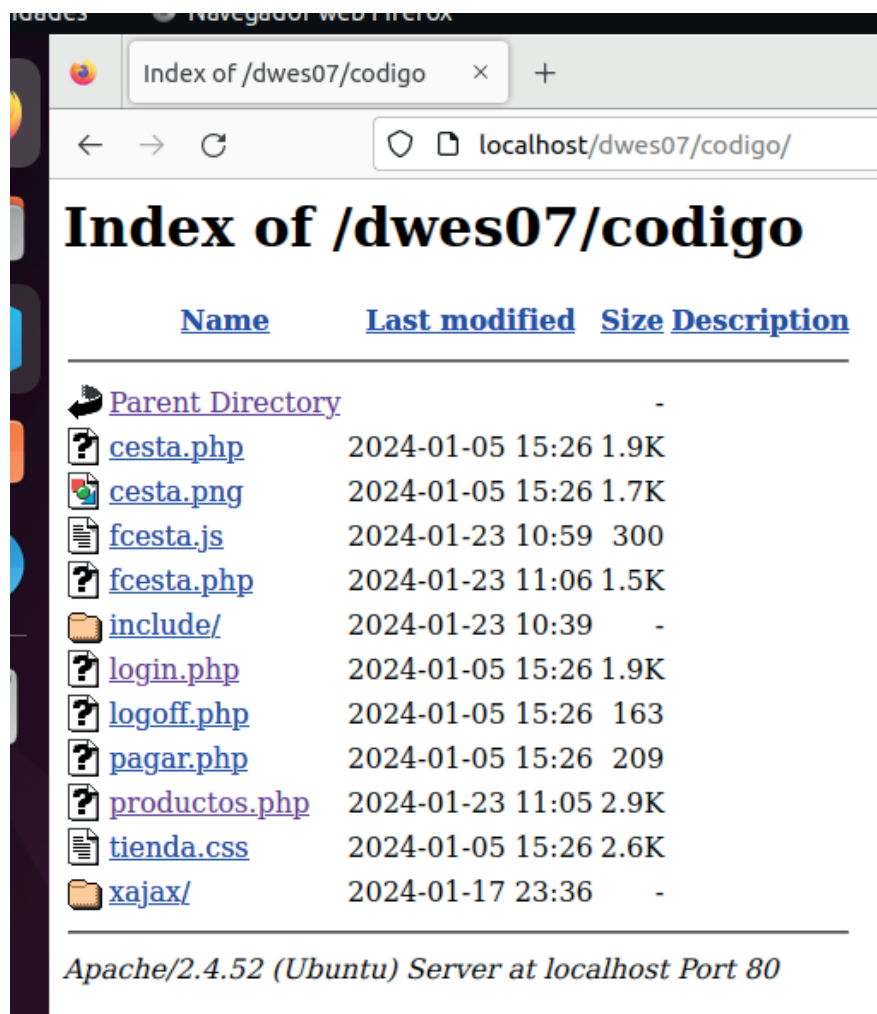
sudo chmod 777 /var/www/html -R

ya tendríamos permiso.



Abrimos el navegador web y escribimos la **IP de la máquina virtual o localhost / nombre de la carpeta**.

Y con esto funciona XAJAX



Ahora vamos a realizar las modificaciones del código.

Primero vamos a **CestaCompra.php** y creamos una nueva función para recoger el código del producto y pasarlo a **fiesta.php** que luego los mostrará.

```
/* ----- html de la cesta de la compra ----- */

public function html() {
    $html = "";
    // Si la cesta está vacía, mostramos un mensaje
    if (count($this->productos)==0) {
        $html .= "<p>Cesta vacía</p>";
        return $html;
    } else {
        // y si no está vacía, mostramos su contenido
        foreach ($this->productos as $producto) {
            $html .= '<p>'.$producto->getcodigo().'</p>';
        }
    }
    return $html;
}
```

Pasamos a crear **fcesta.js** y las dos funciones que necesitamos, la de añadir Producto a la Cesta y la de Vaciar Cesta.

```
/* ----- Añadir Producto a la Cesta ----- */
function addProducto(codigo) {
    //recogemos el id con el código del producto
    var boton_codigo = document.getElementById("btn-"+codigo)
    boton_codigo.disabled = true
    //y se lo pasamos a xajax para enviarselo al php
    xajax_addProducto(codigo)
}

/* ----- Vaciar Cesta ----- */
function vaciarCesta() {
    //recoge el id del boton de vaciar cesta
    var boton_vaciar = document.getElementById("btn-vaciarCesta")
    boton_vaciar.disabled = true
    //llamamos al php para vaciar la cesta
    xajax_vaciarCesta();
}
```

El siguiente archivo que debemos crear es **fcesta.php** con tres funciones, la primera es la de añadir producto a la cesta, la segunda es vaciar la cesta y, el último servirá para mostrar el html de la cesta de la compra.

```
<?php
require_once('include/CestaCompra.php');
require_once('productos.php');

/* ----- Añadir Producto a la Cesta ----- */
function nuevoProducto($codigo) {
    //session_start();

    $cesta = CestaCompra::carga_cesta();

    //objeto de respuesta xajax
    $respuesta = new xajaxResponse();

    //métodos de la CestaCompra
    $cesta->nuevo_articulo($codigo);

    $cesta->guarda_cesta();

    //asignamos los valores a las variables con xajax
    $id_btn = "btn-".$codigo;
    $respuesta->assign($id_btn, "value", "Añadir");
    $respuesta->assign($id_btn, "disabled", false);

    //pasamos la informacion para que sea mostrada en la cesta de la compra
    $html = html_Cesta($cesta);
    $respuesta->assign("cesta", "innerHTML", $html);

    //devolvemos el objeto respuesta
    return $respuesta;
}

/* ----- Vaciar Cesta ----- */
function vaciarCesta() {
    //session_start();

    $cesta = CestaCompra::carga_cesta();

    //objeto de respuesta xajax
    $respuesta = new xajaxResponse();

    //session de la cesta
    unset($_SESSION['cesta']);

    $cesta = new CestaCompra();

    //pasamos la informacion para que sea mostrada en la cesta de la compra
    $html = html_Cesta($cesta);
    $respuesta->assign("cesta", "innerHTML", $html);
```

```

    $respuesta->assign("btn-vaciarCesta" ,"value","Vaciar Cesta");

    //devolvemos el objeto respuesta

    return $respuesta;
}

/* ----- HTML Cesta ----- */
function html_Cesta($cesta) {
    //metemos todo el html en una variable y lo usamos
    $html = "<h3><img src='cesta.png' alt='Cesta' width='24' height='21'> Cesta</h3>" . "<hr
/>";

    $html .= $cesta->html();

    $html .= "<form id='vaciar' action='productos.php' method='post'>";
    $html .= "<input type='submit' name='vaciar' value='Vaciar Cesta' id='btn-vaciarCesta' ";
    if ($cesta->vacia()) {$html .= "disabled='true'";}
    $html .= " onclick='vaciarCesta()'></form>";

    $html .= "<form id='comprar' action='cesta.php' method='post'>";
    $html .= "<input type='submit' name='comprar' value='Comprar' ";
    if ($cesta->vacia()) {$html .= "disabled='true'";}
    $html .= "</></form>";

    return $html;
}

?>

```

Por último modificamos **productos.php**

Primero añadimos la biblioteca y el archivo fcesta.php

```

require_once('fcesta.php');
require_once('xajax/xajax_core/xajax.inc.php');

```

Añadimos la instancia de xajax y los métodos para las funciones.

```

$xajax = new xajax();
$xajax->register(XAJAX_FUNCTION,"vaciarCesta");
$xajax->register(XAJAX_FUNCTION,"nuevoProducto");
$xajax->configure('javascript URI', './xajax');
$xajax->processRequest();

```

En la parte head del HTML incluimos fcesta.js y el método print para poder mostrar por pantalla.

```

<script type="text/javascript" src="fcesta.js"></script>
<?php $xajax->printJavascript(); ?>

```

En la función creaFormularioProductos, en el botón Añadir añadimos un atributo id y un evento onclick, quedando así.

```
function creaFormularioProductos(){
    $productos = DB::obtieneProductos();
    foreach ($productos as $p) {
        echo "<p><form id='\" . $p->getcodigo() . \"' action='productos.php' method='post'>";
        echo "<input type='hidden' name='cod' value='\" . $p->getcodigo() . \"'/>";

        echo "<input type='submit' name='enviar' value='Añadir' id='\" . \"btn-\" . $p->getcodigo() . \"'\"
onclick='nuevoProducto(\"\" . $p->getcodigo() . \"\")' />";

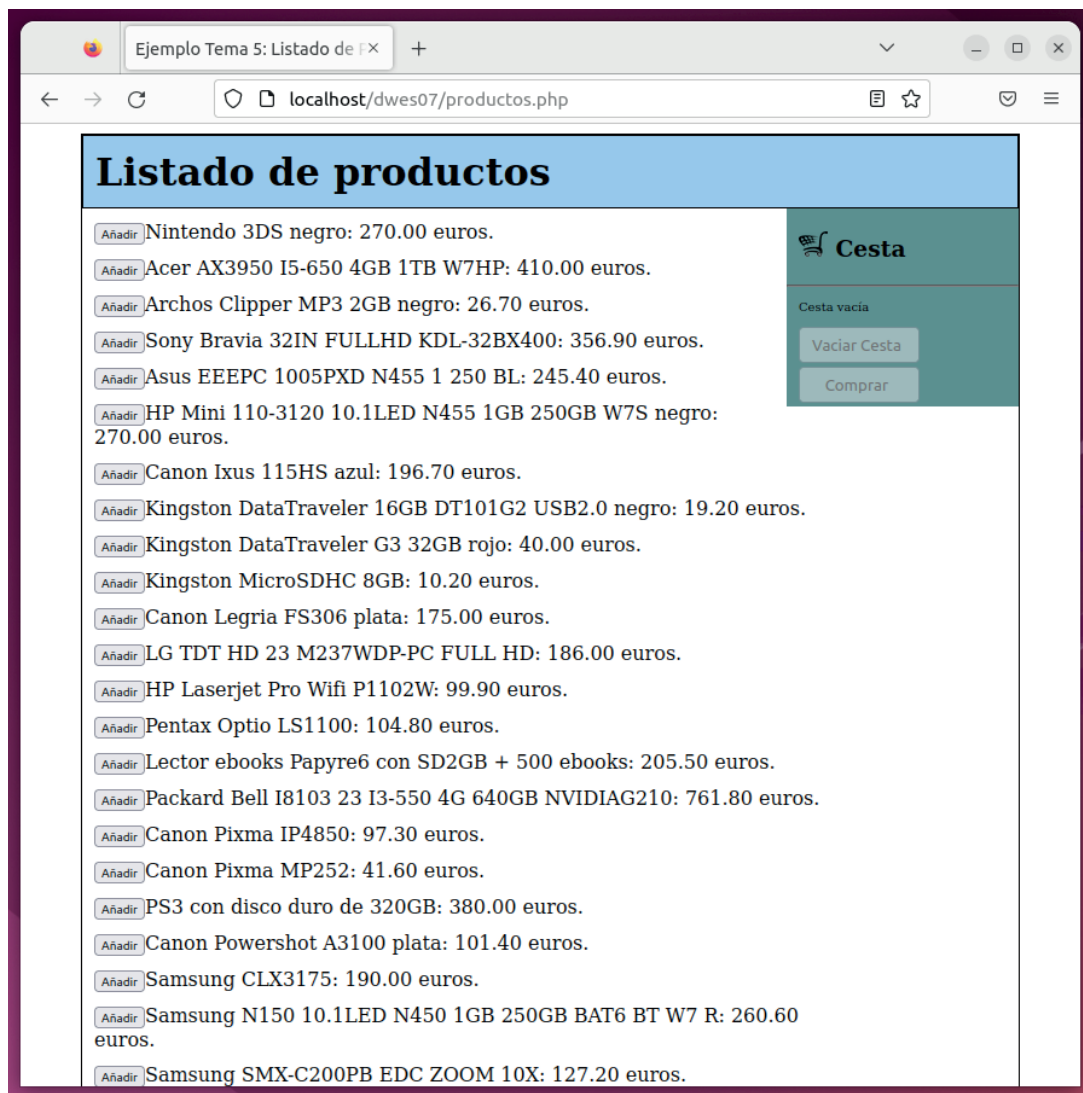
        echo $p->getnombrecorto() . ": ";
        echo $p->getPVP() . " euros.";
        echo "</form>";
        echo "</p>";
    }
}
```

Hacemos lo mismo con la función muestraCestaCompra

```
function muestraCestaCompra($cesta) {
    echo "<h3><img src='cesta.png' alt='Cesta' width='24' height='21'> Cesta</h3>";
    echo "<hr />";
    $cesta->muestra();
    echo "<form id='vaciar' action='productos.php' method='post'>";
    echo "<input type='submit' name='vaciar' value='Vaciar Cesta' id='btn-vaciarCesta' ";
    if ($cesta->vacía()) echo "disabled='true'";
    echo " onclick='vaciarCesta()'></form>";
    echo "<form id='comprar' action='cesta.php' method='post'>";
    echo "<input type='submit' name='comprar' value='Comprar' ";
    if ($cesta->vacía()) echo "disabled='true'";
    echo "/></form>";
}
```

Para comprobar que hemos hecho todo bien abrimos el navegador y vamos al localhost y ejecutamos el código.

Nos logueamos y pasamos a productos.php donde aparecerá la cesta vacía.



Probamos que los botones Añadir funcionan y la página no se recarga al añadir.



Si pinchamos en vaciar cesta productos.php deberá volver a su estado original, sin recargar también.

