

# PROYECTO TIENDA

Alumno: César Díaz Valtueña

22/11/2019

Módulo: Desarrollo Entorno Cliente

Ciclo: DAW2

## RESUMEN

El presente Proyecto trata sobre una tienda online, para controlar la gestión de mantenimiento de tablas. Su funcionamiento consiste en que un cliente pueda iniciar sesión y podrá comprar unos productos. Cada producto tendrá su nombre, precio, stock, descripción y una casilla para indicar la cantidad que quieres de ese producto. Una vez seleccionado ese producto, el usuario dará click sobre el botón de añadir al carrito y le llevará a una página donde mostrará el resultado de su compra.

## ÍNDICE

RESUMEN .....	1
ÍNDICE DE FIGURAS .....	3
Tienda online.....	4
1. Manual .....	4
1.1. Modelo entidad-relación.....	4
1.2. Seguridad.....	4
1.3. Conexión a la base de datos.....	5
1.4. Inicio de sesión de un usuario .....	5
1.5. Productos .....	6
1.6. Compra .....	7
1.7. Cookies .....	9
2. Manual de usuario. ....	9

## ÍNDICE DE FIGURAS

Ilustración 1. Diagrama entidad relación .....	4
Ilustración 2. Seguridad.....	4
Ilustración 3. Seguridad principal.....	5
Ilustración 4. Conexión a la base de datos.....	5
Ilustración 5. Método POST .....	5
Ilustración 6. Inicio de sesión .....	6
Ilustración 7. Cierre de consulta .....	6
Ilustración 8. Productos .....	6
Ilustración 9. Tabla productos.....	7
Ilustración 10. Consulta.....	7
Ilustración 11. Tabla ticket .....	8
Ilustración 12. Insertar a la base de datos .....	8
Ilustración 13. Sentencia update.....	8
Ilustración 14.Comprobación consulta2 .....	8
Ilustración 15.Comprobación consulta3 .....	9
Ilustración 16.Comprobar stock .....	9
Ilustración 17. Cierre de consultas .....	9
Ilustración 18.Cookies .....	9
Ilustración 19. Inicio de sesión .....	10
Ilustración 20.Página principal .....	10
Ilustración 21. Productos .....	11
Ilustración 22. Ticket compra .....	11
Ilustración 23. Cierre sesión .....	11

## TIENDA ONLINE

### 1. Manual

#### 1.1. Modelo entidad-relación

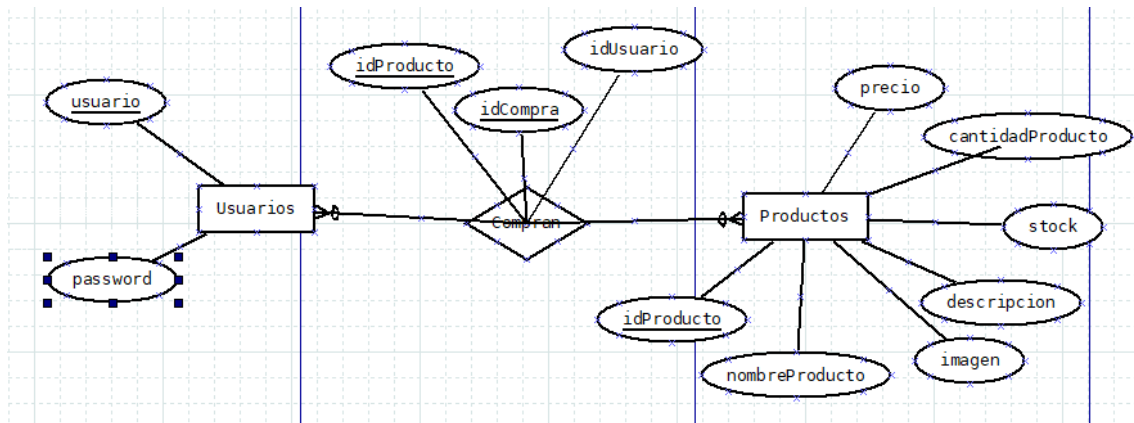


Ilustración 1. Diagrama entidad relación

Fuente: *Elaboración propia*

Usuarios: usuario (Primary key), password.

Compras: (idProducto, idCompra, idUsuario) Primary key

Productos: idProducto(Primary key), nombreProducto, imagen, descripción, stock, cantidadProducto, precio

#### 1.2. Seguridad

Primero, creamos el archivo de seguridad en el que indicamos el localhost, el usuario de la base de datos, la clave y el nombre de la base de datos. Este archivo lo encontraremos en la carpeta de seguridad, fuera de la carpeta www.

```
<?php
define("IP","127.0.0.1");
define("USUARIO","tienda");
define("CLAVE","tiendaBueno");
define("BD","tienda");
?>
```

Ilustración 2. Seguridad

Fuente: *Elaboración propia*

Una vez incluido este archivo, en nuestro archivo principal, tienda.php lo incluiremos y ahí es donde empezaremos a usar código php.

```
//Seguridad
include "../../seguridad/tema03/datosBDTienda.php";
```

**Ilustración 3. Seguridad principal**

*Fuente: Elaboración propia*

### 1.3. Conexión a la base de datos

Una vez establecida la seguridad, nos conectaremos a la base de datos ya creada en MySQL. Con el comando `mysqli_connect()` asignaremos los nombre del archivo de seguridad. Si la conexión es correcta no dará ningún error, si no, lanzará un mensaje con el error que se ha producido.

Este bloque de código lo incluiremos en todos los archivos, en los cuales necesitemos obtener datos de las tablas de la base de datos.

```
//Vamos a conectarnos a la base de datos
$canal = @mysqli_connect(IP,USUARIO,CLAVE,BD);
if (!$canal){
    echo "Ha ocurrido un error: ".mysqli_connect_errno()." ".mysqli_connect_error()."<br />";
    exit;
}
mysqli_set_charset($canal, "utf8");
```

**Ilustración 4. Conexión a la base de datos**

*Fuente: Elaboración propia*

### 1.4. Inicio de sesión de un usuario

Lo primero, recibimos los datos introducidos por el usuario, mediante el método POST guardaremos los datos en una variable y con la función `isset` comprobamos si la variable está definida y no es null.

```
//Comprobación usuario
$usuario="";
if(isset($_POST['usuario'])){
    $usuario=strip_tags(trim($_POST['usuario']));
}
//Comprobación contraseña
$password="";
if(isset($_POST['password'])){
    $password=strip_tags(trim($_POST['password']));
}
```

**Ilustración 5. Método POST**

*Fuente: Elaboración propia*

Después, seleccionaremos los campos usuario y password de la tabla usuarios y los guardaremos en la variable `$sql` y comprobaremos si la consulta se hace correctamente. Después, ejecutaremos la consulta y comprobaremos si los valores insertados por el usuario existen en la base de datos, si no, devolverá un error. Todos los datos se pasarán por el método POST.

```
$sql = "select usuario, password from usuarios where usuario=? and password=?";
$consulta = mysqli_prepare($canal, $sql);
if (!$consulta) {
    echo "Ha ocurrido un error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br/>";
}

//Comprobar si existe el usuario y la contraseña de la base de datos, se la pasamos a la sentencia sql
mysqli_stmt_bind_param($consulta, "ss", $usuario, $password);

mysqli_stmt_execute($consulta);
mysqli_stmt_bind_result($consulta, $usuario, $password);

mysqli_stmt_store_result($consulta);

//Comprueba el número de filas que la consulta ha encontrado
$n=mysqli_stmt_num_rows($consulta);
if ($n!=1) {
    $http="Location: login.php?mensaje=".urlencode("Usuario o contraseña incorrecto");
    $http.="&usuario=$usuario";
    header($http);
    exit;
} else if (empty($usuario) or empty($password)) {
    $http="Location: login.php?mensaje=".urlencode("Alguno de los datos están vacíos");
    $http.="&usuario=pipas";
    header($http);
    exit;
}
```

**Ilustración 6. Inicio de sesión**

*Fuente: Elaboración propia*

Y por último cerramos la consulta.

```
//Cerramos la consulta
mysqli_stmt_close($consulta);
mysqli_close($canal);
```

**Ilustración 7. Cierre de consulta**

*Fuente: Elaboración propia*

## 1.5. Productos

En productos, seleccionaremos los campos nombreProducto, imagen, descripción, precio, cantidad y el idProducto y los metemos dentro de la variable \$sql. Preparamos la consulta y comprobamos que si hay algún error en la consulta salte un mensaje de error. En el caso de que esté todo correcto, ejecutaremos la consulta y declaramos las variables a los campos solicitados a la base de datos.

```
$sql="select nombreProducto, imagen, descripcion, precio, cantidad, idProducto from productos";

$consulta = mysqli_prepare($canal, $sql);
if (!$consulta) {
    echo "Ha ocurrido un error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br />";
    exit;
}

mysqli_stmt_execute($consulta);
mysqli_stmt_bind_result($consulta, $nombreProducto, $imagen, $descripcion, $precio, $cantidad, $idProducto);
```

**Ilustración 8. Productos**

*Fuente: Elaboración propia.*

Con estos datos recogidos lo que haremos será una tabla con todos los productos que se ofrecen.

```
<form action="pedido.php" method="post">
  <p>?=$mensaje?</p>
  <table border="1">
    <tr>
      <th>Nombre producto</th>
      <th>Imagen</th>
      <th>Descripción</th>
      <th>Precio</th>
      <th>Cantidad</th>
    </tr>
    <?php
      while (mysqli_stmt_fetch($consulta)){
        echo "<tr>";
        echo "<td>$nombreProducto</td><td><img src='".$imagen.'" />"
          . "<td>$descripcion</td><td>".$precio."</td>"
          . "<td><input type='number' name='cantidad$idProducto' max='".$cantidad.'" min='0' /></td>";
        echo "</tr>";
      }
      mysqli_stmt_close($consulta);
      unset($consulta);
    </?php>
  </table>
</form>
```

**Ilustración 9. Tabla productos**

*Fuente: Elaboración propia*

## 1.6. Compra

En este caso, seleccionaremos los campos nombreProducto, precio, cantidad y el idProducto y los metemos dentro de la variable \$sql. Preparamos la consulta y comprobamos que si hay algún error en la consulta salte un mensaje de error. En el caso de que esté todo correcto, ejecutaremos la consulta y declaramos las variables a los campos solicitados a la base de datos.

```
$sql="select precio, cantidad, idProducto, nombreProducto from productos";
$consulta=mysqli_prepare($canal,$sql);
if (!$consulta){
  echo "Ha ocurrido el error: ".mysqli_errno($canal). " ".mysqli_error($canal). "<br />";
  exit;
}

mysqli_stmt_execute($consulta);
mysqli_stmt_bind_result($consulta, $precio, $stock, $idProducto, $nombreProducto);
```

**Ilustración 10. Consulta**

*Fuente: Elaboración propia*

Después, según lo que el cliente haya seleccionado de productos, crearemos un pequeño ticket resumen, con los productos comprados. Además, pasaremos a una variable \$cantidadProducto, la cantidad de producto mandado por POST y comprobaremos que la variable no sea null y sea mayor a 0 para poder realizar la tabla.



```

<table border="1">
  <tr>
    <th>Nombre producto</th>
    <th>Cantidad</th>
    <th>Precio</th>
    <th>Precio productos</th>
  </tr>

<?php
$precioFinal = 0;
while (mysqli_stmt_fetch($consulta)){
    $cantidadProducto = $_POST["cantidad$idProducto"];
    if (isset($cantidadProducto) && $cantidadProducto>0){
        echo "<tr>";
        echo "<td>".$nombreProducto."</td><td>$cantidadProducto</td>"
        . "<td>$precio €</td><td>Precio total: ".$precio*$cantidadProducto. "€</td>";
        echo "</tr>";
        $precioFinal += $precio*$cantidadProducto;
    }
}
```

**Ilustración 11. Tabla ticket**

*Fuente: Elaboración propia*

Ahora, modificaremos y actualizaremos las tablas de la base de datos. Mediante la sentencia insert, insertaremos los datos de la compra a la tabla compran.

```

//Canal para que no de error y podamos hacer otra consulta
$canal2 = @mysqli_connect(IP,USUARIO,CLAVE,BD);
$insertar = "insert into compran (cantidadProducto, fecha, idProducto, idUsuario, precio total)"
. " values ($cantidadProducto, CURRENT_DATE, $idProducto, '$usuario' , $precioFinal)";
$consulta2 = mysqli_prepare($canal2, $insertar);
mysqli_stmt_execute($consulta2);
```

**Ilustración 12. Insertar a la base de datos**

*Fuente: Elaboración propia*

Y seguido, con la sentencia update, actualizaremos los campos de la tabla productos.

```

//Cantidad es la tabla de productos y es la tabla que vamos a modificar
$update = "update productos set cantidad = $stock - $cantidadProducto where idProducto = $idProducto";
$consulta3 = mysqli_prepare($canal2, $actualizar);
mysqli_stmt_execute($consulta3);
```

**Ilustración 13. Sentencia update**

*Fuente: Elaboración propia*

Comprobaremos que las consultas se realizan correctamente, y en caso negativo, lanzará un mensaje de error

```

//Si hay un error en la consulta2, mostrará un mensaje de error
if (!$consulta2){
    echo "Ha ocurrido el error: ".mysqli_errno($canal2). " ".mysqli_error($canal2). "<br />";
    exit;
}
```

**Ilustración 14. Comprobación consulta2**

*Fuente: Elaboración propia*

```
//Con esta comprobación , si hay un error, la consulta no se ejecuta y lo devuelve.  
if(!mysqli_stmt_execute($consulta3)){  
    mysqli_rollback($canal2);  
    $http="Location: productos.php?mensaje=".urlencode("Error al descontar la cantidad");  
    header($http);  
    exit;  
}
```

**Ilustración 15.Comprobación consulta3**

*Fuente: Elaboración propia*

También comprobaremos que en el caso de que no haya stock suficiente, se lance un mensaje de error.

```
//Comprobamos que haya stock suficiente  
if($stock<$cantidadProducto){  
    mysqli_rollback($canal2);  
    $http="Location: productos.php?mensaje=".urlencode("No hay stock suficiente");  
    header($http);  
    exit;  
}
```

**Ilustración 16.Comprobar stock**

*Fuente: Elaboración propia*

Por último, cerramos todas las consultas.

```
//Cerramos todas las consultas realizadas  
mysqli_stmt_close($consulta);  
unset($consulta);  
mysqli_stmt_close($consulta2);  
unset($consulta2);  
mysqli_stmt_close($consulta3);  
unset($consulta3);
```

**Ilustración 17. Cierre de consultas**

*Fuente: Elaboración propia.*

## 1.7. Cookies

Las cookies, en este caso, las utilizaremos para pasar el usuario que inicie sesión por todas las paginas php.

```
//Guarda el usuario  
setcookie('usuario',$usuario);
```

**Ilustración 18.Cookies**

*Fuente: Elaboración propia*

## 2. Manual de usuario.

El usuario, primero, deberá iniciar sesión para poder acceder a la tienda. Deberá rellenar los campos con su nombre de usuario y contraseña y le dará al botón Entrar. Si los datos son correctos le dejará entrar, si no, aparecerá un mensaje de error.



**Ilustración 19. Inicio de sesión**

*Fuente: Elaboración propia*

Una vez haya iniciado sesión, se encontrará con la página principal, con una breve descripción de la tienda e imágenes de algunos de los productos que se venden.



**Ilustración 20. Página principal**

*Fuente: Elaboración propia*

Para comprar productos, accederemos a la página productos, situada en la barra de navegación. Ahí encontraremos todos los productos disponibles. Para comprar uno o más productos, solo hay que introducir la cantidad que se desea comprar. Una vez introducida, en la parte inferior encontraremos el botón Finalizar Compra, el cuál pulsaremos para realizar la compra.

Samsung galaxy S10		El Galaxy S10 tiene una pantalla QHD+ Dynamic AMOLED de 6.1 pulgadas, y está potenciado por el nuevo procesador Exynos 9820 de ocho núcleos o bien un Snapdragon 855, con 8GB de RAM y 128GB o 512GB de almacenamiento.	500€	<input type="text" value="1"/>
Samsung galaxy S10+		El Samsung Galaxy S10+ es el más poderoso de la serie Galaxy S10. Con una pantalla AMOLED QHD+ de 6.4 pulgadas, el Galaxy S10+ está potenciado por el procesador Exynos 9820 octa-core o Snapdragon 855, con opciones de 6GB o 12GB de RAM y 128GB, 512GB o 1TB de almacenamiento.	600€	<input type="text" value="1"/>

[Finalizar compra](#)

### Ilustración 21. Productos

*Fuente: Elaboración propia*

Una vez pulsado el botón Finalizar compra, accederemos al ticket, el cuál mostrará los productos comprados por el usuario. En el caso de que quiera volver a realizar otra compra, debemos pulsar en el enlace Realizar otra compra, ese enlace nos volverá a llevar a los productos disponibles.

#### Ticket de compra:

Nombre producto	Cantidad	Precio	Precio productos
Samsung galaxy S10	1	500 €	Precio total:500€
Samsung galaxy S10+	1	600 €	Precio total:600€
Precio final:1100 €			

¡Gracias por su compra!

[Realizar otra compra](#)

### Ilustración 22. Ticket compra

*Fuente: Elaboración propia*

Si el usuario desea salir de la tienda, deberá pulsar el botón Cerrar sesión, situado en la parte superior derecha de la pantalla, y nos redirigirá a la pantalla de inicio de sesión.



### Ilustración 23. Cierre sesión

*Fuente: Elaboración propia.*

## Inicio sesión

Usuario:

Contraseña: