

D1-UD3 Carrito de la compra

1.Temática

Se trata de un formulario de una tienda de consolas de videojuegos donde escoges cuantas consolas quieres de las que hay en stock y además te vaya diciendo la cantidad total de euros que llevas, además se guardará en un archivo data toda la información. Aquí están los arrays del producto con su precio y cantidad:

```
$precios = ["Playstation5" => 700, "Xbox360" => 200, "Switch" => 400, "Nintendo3DS" => 150];  
$stock = ["Playstation5" => 50, "Xbox360" => 30, "Switch" => 60, "Nintendo3DS" => 10];
```

2. Formulario y código: Carrito

2.1 Formulario

Tienda de consolas

Carrito:

Compra realizada con éxito

- Playstation5 - Precio: 700 € - Stock: 50
- Xbox360 - Precio: 200 € - Stock: 30
- Switch - Precio: 400 € - Stock: 60
- Nintendo3DS - Precio: 150 € - Stock: 10

Pago total: 0

El formulario tiene el nombre del producto, su precio y el stock que tiene, además incluye una parte llamada Pago total que según el valor que pongamos en el input y cuando le demos al botón Actualizar compra, este se multiplicará por el precio y se añadirá ahí para saber cuánta cantidad de dinero tenemos acumulado. Una vez que le demos al botón Finalizar compra, se descontará del stock la cantidad que hayamos escogido.

Tienda de consolas

Carrito:

- Playstation5 - Precio: 700 € - Stock: 50
- Xbox360 - Precio: 200 € - Stock: 30
- Switch - Precio: 400 € - Stock: 60
- Nintendo3DS - Precio: 150 € - Stock: 10

Pago total: 2300

Actualizar compra

Finalizar compra

Aquí vemos que cuando le damos al primer botón, según lo que pongamos en el input, se multiplicará por el precio y se añadirá al pago total.

Tienda de consolas

Carrito:

Compra realizada con éxito

- Playstation5 - Precio: 700 € - Stock: 49
- Xbox360 - Precio: 200 € - Stock: 28
- Switch - Precio: 400 € - Stock: 57
- Nintendo3DS - Precio: 150 € - Stock: 10

Pago total: 0

Actualizar compra

Finalizar compra

Cuando se pulse el segundo botón, se finaliza la compra y se descontará la cantidad puesta en cada producto y pondrá un mensaje que la compra ha sido un éxito.

2.2 Errores

Tienda de consolas

Carrito:

No hay suficiente stock de Playstation5

No hay suficiente stock de Xbox360

No hay suficiente stock de Nintendo3DS

- Playstation5 - Precio: 700 € - Stock: 49
- Xbox360 - Precio: 200 € - Stock: 28
- Switch - Precio: 400 € - Stock: 57
- Nintendo3DS - Precio: 150 € - Stock: 10

Pago total: 0

Actualizar compra

Finalizar compra

Si llegamos a pasarnos del stock puesto, al actualizar compra o al intentar finalizar compra, no te dejará y nos mostrará un mensaje de error diciendo que no hay suficiente producto y pondrá el input al máximo de stock que haya.

2.3 Código

```
$pagoTotal=0;
$precios = ["Playstation5" => 700, "Xbox360" => 200, "Switch" => 400, "Nintendo3DS" => 150];
define('RUTA_FICHERO', "stock.data");

// Cargar stock desde el archivo o crear uno por defecto
if (file_exists(RUTA_FICHERO)) {
    $stock = unserialize(file_get_contents(RUTA_FICHERO));
} else {
    $stock = ["Playstation5" => 50, "Xbox360" => 30, "Switch" => 60, "Nintendo3DS" => 10];
    file_put_contents(RUTA_FICHERO, serialize($stock));
}
```

Creamos un array asociativa con todos los productos y sus precios y definimos un fichero llamado stock.data si no existe con un array asociativo con los productos y su stock, en el caso de que ya exista, simplemente los abrimos con el comando unserialize y guardamos en la variable \$stock el contenido del fichero.

```

$cantidades = [];
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if(isset($_POST['actualizar'])){
        foreach($_POST['cantidad'] as $producto => $cantidad){
            if ($cantidad >= 0 && $cantidad <= $stock[$producto]) {
                $pagoTotal=$pagoTotal+($cantidad * $precios[$producto]);
                $cantidades[$producto] = $cantidad;
            }else{
                echo "<p style=color:red>No hay suficiente stock de $producto</p>";
                $cantidades[$producto] = $cantidad;
            }
        }
    }
}

```

Primero que nada inicializamos un array llamado \$cantidades donde guardaremos los valores de los inputs para que al darle al botón, recordemos los valores puestos y no haya que ponerlos otra vez.

Esta parte del código, una vez que le demos al botón actualizar, recorreremos todos los productos y las cantidades puestas en los inputs y si la cantidad es igual o mayor a 0 y menor o igual al stock, se multiplicará el precio del producto por la cantidad puesta y se guardará en la variable \$pagoTotal y guardamos el valor de los inputs en la variable \$cantidades. Si no se cumple la condición saltará el error.

```

$exito="true";
if(isset($_POST['finalizar'])){
    foreach ($_POST['cantidad'] as $producto => $cantidad) {
        // Restar del stock solo si hay suficiente stock
        if ($cantidad >= 0 && $cantidad <= $stock[$producto]) {
            $stock[$producto] -= $cantidad;
        }else{
            echo "<p style=color:red>No hay suficiente stock de $producto</p>";
            $exito="false";
        }
    }
    if($exito=="true"){
        echo "Compra realizada con exito";
    }
    // Guardar el stock actualizado
    file_put_contents(RUTA_FICHERO, serialize($stock));
}
}

```

En esta parte cuando le demos al botón finalizar, volvemos a poner la misma condición y si no hay errores, se restará el stock de cada producto y mostrará un mensaje de la compra ha sido un éxito utilizando un booleano que si detecta un valor mayor al stock, se pondrá false y con esa condición, si el booleano es true mostrará el mensaje de éxito y si no, no se muestra. En el caso de que no, se mostrará el mismo error de que no hay stock.

```
<h1>Tienda de consolas</h1>
<h2>Carrito:</h2>
```

```
// Mostrar productos
echo "<ul>";
foreach ($precios as $producto => $precio) {
    if (isset($cantidades[$producto])) {
        $valorCantidad = $cantidades[$producto];
    } else {
        $valorCantidad = 0;
    }
    echo "<li>";
    echo "<label for=\"\$producto\">$producto - Precio: $precio € - Stock: {$stock[$producto]}</label>";
    echo "<input type=\"number\" name=\"cantidad[$producto]\" value=\"\$valorCantidad\">";
    echo "</li>";
}
echo "</ul>";
echo "<label for=total>Pago total:$_pagoTotal</label><br><br>";
?>
<input type="submit" name="actualizar" value="Actualizar compra">
<input type="submit" name="finalizar" value="Finalizar compra">
```

Una vez hecho todo lo anterior, vamos a mostrar todos los datos del formulario: el producto, su precio y stock. \$valorCantidad es una variable que pondremos en cada input que si hemos introducido un número a la hora de actualizar compra, recuerde el número. Si no, se pondrá 0.

3. Administración

Esta página solo hará que si no hay stock, se pueda añadir más y la cantidad deseada

3.1 Formulario

Reponer Stock de Consolas

- Playstation5 - Stock actual: 50
- Xbox360 - Stock actual: 30
- Switch - Stock actual: 60
- Nintendo3DS - Stock actual: 10

Reponer Stock

Aquí decidimos que tanta cantidad queremos añadir al stock en los inputs y cuando le demos al botón reponer stock se añadirá esa cantidad puesta, por ejemplo vamos a añadir 10 playstation 5 más:

Reponer Stock de Consolas

Stock actualizado correctamente.

- Playstation5 - Stock actual: 60
- Xbox360 - Stock actual: 30
- Switch - Stock actual: 60
- Nintendo3DS - Stock actual: 10

Reponer Stock

3.2 Código

```
// Cargar stock desde el archivo o crear uno por defecto
if (file_exists(RUTA_FICHERO)) {
    $stock = unserialize(file_get_contents(RUTA_FICHERO));
} else {
    $stock = ["Playstation5" => 50, "Xbox360" => 30, "Switch" => 60, "Nintendo3DS" => 10];
    file_put_contents(RUTA_FICHERO, serialize($stock));
}
```

Como con el código anterior, comprobamos si ya existe el fichero stock.data y guardamos su contenido en la variable ~\$stock. En el caso de que no, se creará dicho archivo y se creará un array asociativo solo con los productos y su stock.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if(isset($_POST['reponer'])){
        foreach($_POST['cantidad'] as $producto => $cantidad){
            if ($cantidad >= 0) {
                $stock[$producto] += $cantidad; // Aumentar el stock
            }
        }
        // Guardar el stock actualizado
        file_put_contents(RUTA_FICHERO, serialize($stock));
        echo "<p>Stock actualizado correctamente.</p>";
    }
}
```

Si presionamos el botón reponer, recorreremos con un foreach todas las cantidades escritas en los inputs y se sumarán al stock de cada producto. Si todo es correcto, saldrá un mensaje diciendo que el stock se ha actualizado correctamente.

Esto hará que cuando volvamos al programa del carrito, se nos mostrará el nuevo stock actualizado.

```
// Mostrar productos
echo "<ul>";
foreach ($stock as $producto => $cantidad) {
    echo "<li>";
    echo "<label for=\"\$producto\">$producto - Stock actual: $cantidad</label>";
    echo "<input type=\"number\" name=\"cantidad[$producto]\" min=\"0\" placeholder=\"Cantidad a reponer\">";
    echo "</li>";
}
echo "</ul>";
?>
<input type="submit" name="reponer" value="Reponer Stock">
</form>
```

4.Inicio sesión

4.1 Formulario

Bienvenido a la Tienda de Consolas

Por favor, ingresa tu nombre para comenzar:

Esto es una página en el que antes de acceder al carrito o administración, tienes que escribir tu nombre y luego darle al botón de la página a la que quieres ir y en ambas te darán la bienvenida con el nombre que has puesto.

4.2 Código

```
<?php
session_start();
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    if(isset($_POST['carrito'])){}
        $_SESSION['nombre'] = $_POST['nombre'];
        header("Location: carrito.php"); // Redirige al carrito
        exit();
    }
    if(isset($_POST['admin'])){}
        $_SESSION['nombre'] = $_POST['nombre'];
        header("Location: admin.php"); // Redirige al carrito
        exit();
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Inicio de la Tienda</title>
</head>
<body>
    <h1>Bienvenido a la Tienda de Consolas</h1>
    <form method="POST">
        <label for="nombre">Por favor, ingresa tu nombre para comenzar:</label>
        <input type="text" name="nombre" required>
        <input type="submit" name="carrito" value="Ir al carrito">
        <input type="submit" name="admin" value="Administracion">
    </form>
</body>
</html>
```

Este es el código php y HTML. Cuando se pulse uno de los botones, el nombre que se escriba se guardará en \$_SESSION['nombre'] que servirá para que podamos llevarnoslo a otras páginas que en este caso sería a carrito.php o admin.php. Header("Location:(nombre página)") nos redirigirá a la página que queramos ir.

Hacemos unos cambios en el código de carrito y admin para incluir el nombre puesto en nombre.php:

Carrito:

```
<h2>¡Hola, <?php echo htmlspecialchars($_SESSION['nombre']); ?>! Bienvenido a tu carrito:</h2>
```

Administración:

```
<h2>¡Hola, <?php echo htmlspecialchars($_SESSION['nombre']); ?>! Es hora de reponer</h2>
```

Cambio en ambas:

```
<?php
session_start();
if (!isset($_SESSION['nombre'])) {
    header("Location: nombre.php"); // Redirige al formulario si no hay nombre en sesión
    exit();
}
```

Esto sirve para que si no se ha escrito un nombre, nos quedemos en la página de nombre.php hasta que escribimos uno.

Resultado:

Tienda de consolas

¡Hola, Angela! Bienvenido a tu carrito:

Carrito:

- Playstation5 - Precio: 700 € - Stock: 60
- Xbox360 - Precio: 200 € - Stock: 30
- Switch - Precio: 400 € - Stock: 60
- Nintendo3DS - Precio: 150 € - Stock: 10

Pago total:0

Actualizar compra

Finalizar compra

Reponer Stock de Consolas

¡Hola, Angela! Es hora de reponer

- Playstation5 - Stock actual: 60
- Xbox360 - Stock actual: 30
- Switch - Stock actual: 60
- Nintendo3DS - Stock actual: 10

Reponer Stock

5.Conclusión

A pesar que parte del ejercicio ya lo hicimos todos juntos en clase,varias partes como la de que se vaciara el stock dependiendo de la cantidad que comprar o la página de administración o la de saludo fueron complicados pero se logró.

Gracias a esta práctica,he podido practicar y entender bien ,usandolo en mi programa,cómo va el \$_SESSION y como esto es una especie de “memoria” para guardar datos que meta el usuario y moviéndose entre diferentes páginas para usarlos en ellas.