

TECNOLÓGICO NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO SUPERIOR DE APATZINGÁN



Proyecto Final FEBRERO 2022 - JULIO 2022

MATERIA:

Programación web

ALUMNO:

Almazán Barajas Salma

N° DE CONTROL:

19020078

CARRERA:

Ing. En Sistemas Computacionales

TURNO:

Matutino

SEMESTRE:

Sexto

DOCENTE:

I.S.C Uber Gerardo Reyes Dueñas

UNIDAD:

5

25 de mayo de 2022, Apatzingán, Michoacán.

INDICE

1. Introducción	3
2. Practica	4
3. Conclusión	20



Introducción

El proyecto final consiste en optimizar el carrito de compra, en esta ocasión de tomaran los conocimientos obtenidos durante las clases y lo que se hará será poner una base de datos a la página web en la que te permita guardar los productos que tienes en tu sitio, esta sea hará mediante el uso de php, la página web tendrá un inicio de sesión donde también se pondrán un formulario que nos permita agregar los productos que queramos mediante un formulario pero con la condición de que conozcas el nombre del usuario y su contraseña que esta guardada en la base de datos y una vez que insertes los productos estos irán apareciendo en la página principal del sitio.

Proyecto Final

Descripción

- Código JS app

Se crea una clase que contendrá todas las funciones

```
class Carrito{
```

Se crea la función comprarProducto, dentro de este pondremos un if que indique que en caso de dar al botón que contiene el id agregar-carrito este que leera el dato y se guardara en la variable producto.

```
    comprarProducto(e){
        e.preventDefault();
        if(e.target.classList.contains("agregar-
        carrito")){
            const
            producto=e.target.parentElement.parentElement;
            this.leerDatosProducto(producto);
        }
    }
```

En la función de leerDatosProducto se pondrán los parámetros que tomaran la información de cada campo en este caso sera la imagen, el nombre, el precio y la cantidad de cada producto y así al tener ya la información con el parámetro this se agregara lo que trae infoProducto a la función insertarCarrito que se usara posteriormente.

```
    leerDatosProducto(producto){
        const
        infoProducto={
            imagen: producto.querySelector("img").src,
            nombre: producto.querySelector("h4").textContent,
            precio: producto.querySelector(".precio span").textContent,
            id: producto.querySelector("a").getAttribute("data-id"),
            cantidad: 1
        }
        this.insertarCarrito(infoProducto);
    }
```

En la función insertarCarrito que se le dara como parámetro producto que es el que trae la información . se creara una variable row y a esta le daremos el parámetro innerHTML que nos permitira insertar la información en el archivo html en la parte del carrito, solo ponemos los td que es donde se insertara la información correspondiente con el parámetro producto y la variable que le corresponde, al final ponemos un listaProductos que anteriormente se creo y con el parámetro appendChild le damos lo que trae row.

```

    insertarCarrito(producto){          const
row=document.createElement("tr");
row.innerHTML = `
    <td>
        
    </td>

    <td>${producto.nombre}</td>
    <td>${producto.precio} </td>
    <td>${producto.cantidad}</td>
    <td>
        <a href="#" class="borrar-producto fa fa-times-circle"
dataid="${producto.id}"></a>
    </td>
`;
listaProductos.appendChild(row);
}

```

En la función eliminarProducto se pondrá un if que indique que si el usuario da en la parte que contiene borrar-producto este eliminara el producto del carrito con la función remove.

```

eliminarProducto(e){
    e.preventDefault();          let producto,
productoID;          if(e.target.classList.contains("borrar-
producto")){
        e.target.parentElement.parentElement.remove();
producto=e.target.parentElement.parentElement;
productoID=producto.querySelector("a").getAttribute("data-id");          }
}

```

En la función vaciarCarrito se pondrá un while que te diga que mientras listaProductos tenga algo al dar en esa opción los de listaProductos se removera y después retornara el resultado

```

vaciarCarrito(e){
    e.preventDefault();
while(listaProductos.firstChild){          listaProductos.removeChild(
listaProductos.firstChild);          }
    return false;
} }

```

□ Código JS app2

En esta parte se invoca la clase que está en el otro archivo js.

```
const carro=new Carrito();
```

En esta parte se crea una variable que tendrá el id de la parte donde se insertaran los productos

```
const carrito=document.getElementById("carrito");
```

En esta parte crearemos una variable que tendrá el id de donde serán tomados los parámetros de los productos

```
const productos=document.getElementById("lista-productos");
```

En esta parte se creara una variable que la clase donde se insertaran los capos y después tendrá la etiqueta de la parte de donde quieres que se inserte la información que en este caso será en el tbody.

```
const listaProductos=document.querySelector("#lista-carrito tbody");
```

En esta parte se creara una variable que sera la que tendrá el id del botón de vaciar-carrito que sera la que nos dejara borrar toda la información que este dentro del carrito de compras

```
const vaciarCarritoBtn=document.getElementById("vaciar-carrito");
```

En esta parte se mandara llamar a la función que se creara mas adelante.

```
cargarEvento();
```

En esta parte se crea la función que sera la que contenga los eventos que harán que funcionen las funciones creadas en el archivo anterior de js.

La primera sera la que al darle click al botón de comprar se agregue el producto al carrito, la siguiente sera la que al darle click a la figura de la x se eliminara únicamente ese producto y no todos, y la ultima que sera la nos que nos permita vaciar el carritp por completo al darle click al botón de vaciar carrito y esto se hizo con la instrucción de addEventListener.

```
function cargarEvento(){    productos.addEventListener("click",  
(e)=>{carro.comprarProducto(e)});    carrito.addEventListener("click",  
(e)=>{carro.eliminarProducto(e)});  
vaciarCarritoBtn.addEventListener("click",(e)=>{carro.vaciarCarrito(e)});  
}
```

□ Imagen de la pagina

Carrito

MiViaje.com

Carrito

Carrito Compras en JavaScript y

127.0.0.1:5500/PROYECTO-JAVASCRIPT-CARRITO-DE-COMPRAS/index.html#

Ktown4u

Busca de tu grupo favorito

Albumes, Lightstick y mas

¿Que quisieras encontrar?

Mas de 20,000 Articulos

Encuentra los mejores precios

Completamente originales

Con beneficios de preventa

Envios gratis en la compra

de mas de 3 articulo

Articulos de Kpop

Square Up Ver. Black

BLACKPINK

★★★★★

\$690

\$537

AGREGAR AL CARRITO

Kill This Love Ver. Black/Pink

BLACKPINK

★★★★★

\$849

\$680

AGREGAR AL CARRITO

The Album Ver. #3 y #4

BLACKPINK

★★★★★

\$824

\$780

AGREGAR AL CARRITO

LALISA ver. Gold

LISA

★★★★★

\$670

\$550

What Is Love? Ver. A y B

TWICE

★★★★★

\$840

\$790

Merry & Happy Ver. Happy/Merry

TWICE

★★★★★

\$675

\$600

7

8

- **Iniciar sesión**

Php

Abrimos las etiquetas de php y dentro ponemos un if, la condición de este sera que con la variable post obtenga el nombre que tiene el botón entonces al dar clic al botón se obtendrá el nombre y entrara a esta.

```
<?php
//Utilizando interrogantes para los valores
if(isset($_POST['submit'])){
    try{
```

si entra en la condición se conectara a la base de datos que tiene por nombre tienda y hara una consulta en la tabla de usuarios para obtener el contenido de esos campos.

```
$conexion=new PDO('mysql:host=localhost:3308;dbname=tienda','root','');
$enunciado=$conexion->prepare("SELECT * FROM usuarios");
$enunciado->setFetchMode(PDO::FETCH_ASSOC);
$enunciado->execute();
```

Pondremos un ciclo para recorrer lo que se obtuvo, y le iremos asignando los valores a una variable. Al terminar el ciclo tomaremos lo que usuario ingresan en los campos del formulario con la variable post y poniéndolos en una variable.

```
while($row= $enunciado->fetch()){
    $nombre=$row["nombre"];
    $password=$row["password"];
}
$usuario=$_POST['usuario'];
$numero=$_POST['password'];
```

Ahora haremos la comparación de que si lo que usuario ingreso es igual que lo que se obtuvo de la base de datos se iniciara sesión te enviara a la página para agregar los productos.

```
if($nombre==$usuario && $password==$numero){

    session_start();

    $_SESSION['nombre']=$nombre;

    header("location: agregar.php");

}
```

De lo contrario te enviara una alerta para informarte que algo está mal, lo cual puede ser porque dejaste algún campo vacío o porque ingresaste mal la contraseña

```
else{
    echo '<script>
    alert ("Revisa tus respuestas");
    </script>';
}

}
```

Y por último el catch para alertar si hay un problema en la conexión de la base de datos.

```
catch(PDOException $e){
    echo "error".$e->getMessage();
}
?>
```

HTML

Abrimos las etiquetas de html y en el títulos ponemos iniciar sesión.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags-->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="Colorlib Templates">
    <meta name="author" content="Colorlib">
    <meta name="keywords" content="Colorlib Templates">

    <!-- Title Page-->
    <title>Iniciar Sesion</title>
```

Hacemos los links a las clases con los estilos que queremos usar para nuestro formulario.

```
<!-- Icons font CSS-->
    <link href="vendor/mdi-font/css/material-design-iconic-font.min.css"
rel="stylesheet" media="all">
```

```
<link href="vendor/font-awesome-4.7/css/font-awesome.min.css" rel="stylesheet"
media="all">
<!-- Font special for pages-->
<link
href="https://fonts.googleapis.com/css?family=Roboto:100,100i,300,300i,400,400i,500,
500i,700,700i,900,900i" rel="stylesheet">

<!-- Vendor CSS-->
<link href="vendor/select2/select2.min.css" rel="stylesheet" media="all">
<link href="vendor/datepicker/daterangepicker.css" rel="stylesheet" media="all">

<!-- Main CSS-->
<link href="css/main.css" rel="stylesheet" media="all">
</head>
```

Ponemos los contenedores que va tener para la estructura del formulario y ponemos las clases de estilos que sirvan para poner los colores y el posicionamiento del contenedor

```
<body>
<div class="page-wrapper bg-blue p-t-100 p-b-100 font-robo">
  <div class="wrapper wrapper--w680">
    <div class="card card-1">
      <div class="card-heading">

    </div>
    <div class="card-body">
      <h2 class="title">Iniciar Sesión</h2>
```

Abrimos el formulario y en la parte del metodo ponemos post para que en la parte del php se puedan obtener los valores.

Ponemos dos input que serán las entradas de datos en el formulario le pinemos un name para identificar cada campo y una clase de estilo para darle formato.

Al final ponemos un botón y le ponemos nombre.

```
<form action="" method="POST">
  <div class="input-group">
    <input class="input--style-1" type="text" placeholder="Usuario" id="usuario"
name="usuario">
  </div>

  <div class="input-group">
```

```

        <input class="input--style-1" type="password" placeholder="Contraseña"
        id="contraseña" name="password">
    </div>

    <div class="p-t-20">
        <button class="btn btn--radius btn--color" type="submit" name="submit" >
Submit </button>
    </div>
</form>

```

Ya solo cerramos todas la etiquetas y listo.

```

    </div>
    </div>
    </div>
</div>

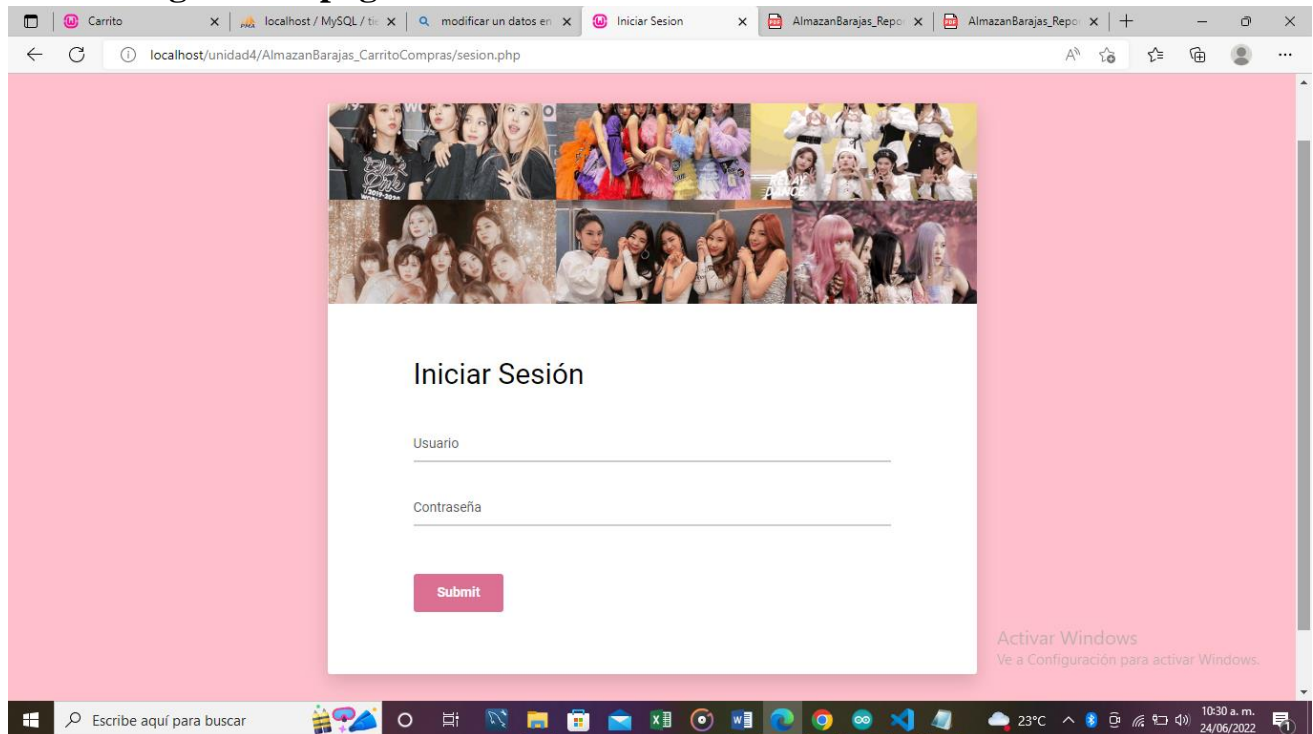
<!-- Main JS-->
<!--<script src="js/global.js"></script>-->

</body>

</html>

```

• Imagen de la pagina



- **Agregar**

PHP

Abrimos las etiquetas de php y ponemos de condición que si damos clic en el botón que lleva por nombre submit entrara en la condición y hará la conexión a la base de datos, después hará una inserción a la base de datos.

```
<?php

if(isset($_POST['submit'])){

    try{
        $conexion=new PDO('mysql:host=localhost:3308;dbname=tienda','root','');
        //$enunciado=$conexion->prepare("INSERT INTO `albumes` (`producto`, `artista`,
`precioant`, `precioact`, `img`) values (?, ?, ?, ?, ?)");
        $enunciado=$conexion->prepare("INSERT INTO albumes(producto, artista, precioant,
precioact, img) values (?, ?, ?, ?, ?)");
```

Con el método post tomaremos la información que se va insertando en los campos y los ponemos a una variable.

Después sanitizaremos los campos con la opción de FILTER_SANITIZE_STRING y lo repetimos en todos los campos.

```
$p=$_POST['producto1'];
$p=filter_var($p, FILTER_SANITIZE_STRING);
$a=$_POST['artista1'];
$a=filter_var($a, FILTER_SANITIZE_STRING);
$pr=$_POST['precioant1'];
$pr=filter_var($pr, FILTER_SANITIZE_STRING);
$prb=$_POST['precioact1'];
$prb=filter_var($prb, FILTER_SANITIZE_STRING);
$i=$_POST['img1'];
$i=filter_var($i, FILTER_SANITIZE_STRING);
```

Insertaremos lo que ya traen las variables en la base de datos poniendo las posiciones y ya solo ejecutamos

```
$enunciado->bindParam(1,$p);
$enunciado->bindParam(2,$a);
$enunciado->bindParam(3,$pr);
$enunciado->bindParam(4,$prb);
$enunciado->bindParam(5,$i);
```

```
$enunciado->execute();
}
```

En el catch ponemos una alerta para que nos avise si hay un error al conectar la base de datos.

```
catch(PDOException $e){
    echo "error".$e->getMessage();
}
```

Pondremos otro if para si damos clic en el otro botón nos envíe a la página de inicio.

```
}
if(isset($_POST['back'])){

    header("location: index.php");

}

?>
```

HTML

Abrimos las etiquetas de html y ponemos los links a las clases de estilos.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <!-- Required meta tags-->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="Colorlib Templates">
    <meta name="author" content="Colorlib">
    <meta name="keywords" content="Colorlib Templates">

    <!-- Title Page-->
    <title>Iniciar Sesion</title>
    <!-- Icons font CSS-->
    <link href="vendor/mdi-font/css/material-design-iconic-font.min.css"
rel="stylesheet" media="all">
    <link href="vendor/font-awesome-4.7/css/font-awesome.min.css" rel="stylesheet"
media="all">
    <!-- Font special for pages-->
    <link
href="https://fonts.googleapis.com/css?family=Roboto:100,100i,300,300i,400,400i,500,
500i,700,700i,900,900i" rel="stylesheet">
```



```
<!-- Vendor CSS-->
<link href="vendor/select2/select2.min.css" rel="stylesheet" media="all">
<link href="vendor/datepicker/daterangepicker.css" rel="stylesheet" media="all">

<!-- Main CSS-->
<link href="css/main.css" rel="stylesheet" media="all">
</head>
```

En el body ponemos los contenedores y les ponemos estilos para que se acomode dónde queremos, y ya solo ponerle en los campos al formulario.

```
<body>
  <div class="page-wrapper bg-blue p-t-100 p-b-100 font-robo">
    <div class="wrapper wrapper--w680">
      <div class="card card-1">
        <div class="card-heading">

          </div>
          <div class="card-body">
            <h2 class="title">Insertar Productos</h2>
```

Agregamos un formulario y en el method le ponemos POST para poder obtener los datos que ingresemos para poder ingresarlos a la base de datos.

Ponemos 5 inputs para ingresar los cambios y ponemos un nombre para identificar cada campo. Y les ponemos los estilos.

```
<form action="" method="POST">
  <div class="input-group">

    <input class="input--style-1" type="text"
placeholder="Nombre del producto" name="producto1" id="producto1">
  </div>

  <div class="input-group">

    <input class="input--style-1" type="text"
placeholder="Nombre del artista" name="artista1" id="artista1">
  </div>
```

```

        <div class="input-group">
            <input class="input--style-1" type="text"
placeholder="Link de la imagen" name="img1" id="img1">
        </div>

        <div class="input-group">
            <input class="input--style-1" type="text"
placeholder="Precio anterior" name="precioant1" id="precioant1">
        </div>

        <div class="input-group">
            <input class="input--style-1" type="text"
placeholder="Precio actual" name="precioact1" id="precioact1">
        </div>

```

Ponemos dos botones y les das formato, les agregas un nombre para identificarlo.

```

<div class="p-t-20">
    <button class="btn btn--radius btn--color" type="submit" name="submit" >
Agregar</button>

    <button class="btnback btn btn--radius btn--color" type="submit" name="back">Pagina
principal</button>
</div>
</form>

```

Cerramos todas las etiquetas y listo.

```

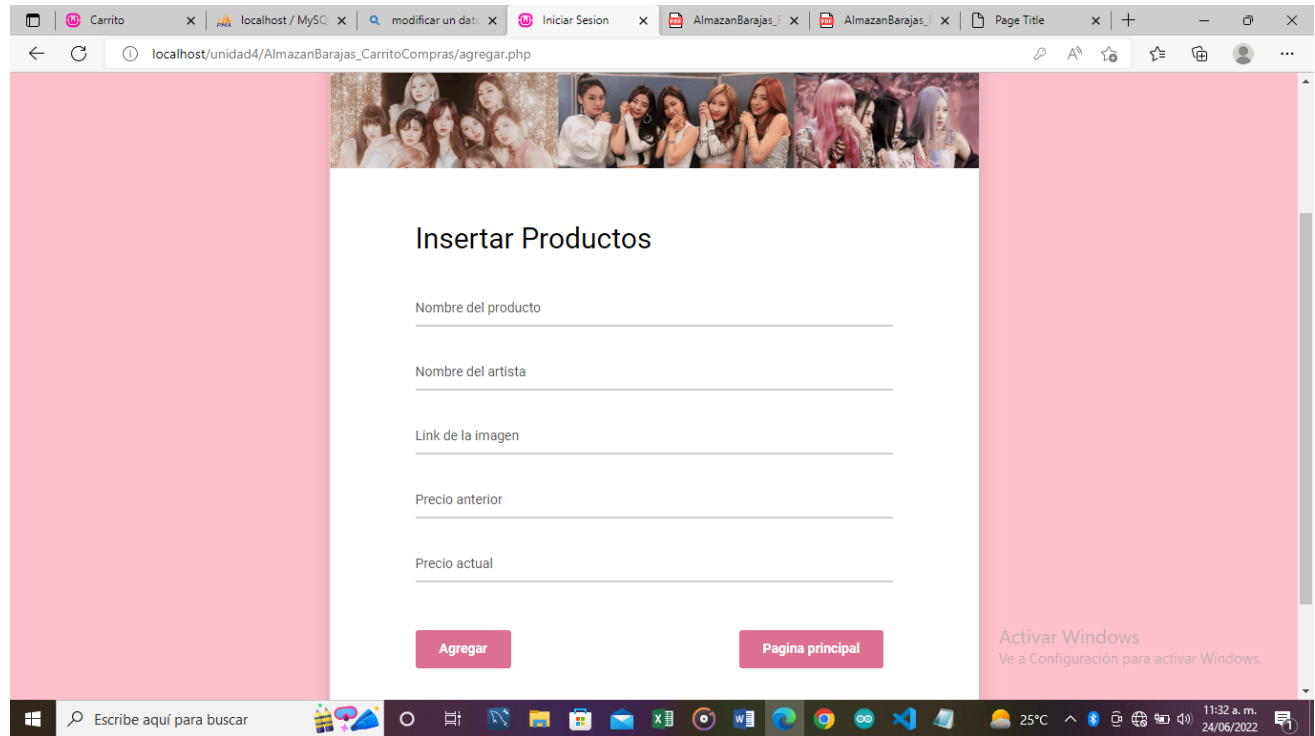
        </div>
    </div>
</div>
</div>

</body>

</html>

```

- **Imagen de la página**



- Index

En el index haremos la impresión de las tarjetas con los datos ingresados en las base de datos, lo que haremos será abrir las etiquetas de php y pondremos un try seguido pondremos en enlace a la base de datos y haremos una consulta para ir obteniendo los datos y ejecutamos la instrucción.

```
<!--de aqui comienza a cargar los productos con php-->
<div id="lista-productos" class="container">
  <h1 id="encabezado" class="encabezado">Articulos de Kpop</h1>

  <!--bueno en realidad aqui jajaja con php-->

  <?php

  try{
    $conexion=new PDO('mysql:host=localhost:3308;dbname=tienda','root','');
    $enunciado=$conexion->prepare("SELECT * FROM albumes");
    $enunciado->setFetchMode(PDO::FETCH_ASSOC);
    $enunciado->execute();
```

Pondremos un Contador para ir contando las tarjetas.

```
$cont=1;

?>
```

En otro php pondemos una condicion de si lo que tiene el contador al dividirlo su residuo es 0 entonces va a poner un div row que nos dejara poner 3 tarjetas en cada fila.

```
<?php if($cont%3==0) echo '<div class="row">'; ?>
```

en la impresion de las tarjetas pondremos una etiqueta de php dentro de la etiqueta donde se imprimen los resultados, en la etiqueta de php pondremos un echo seguido de una variable row con el nombre del campo de la base de datos y asi se ira imprimiendo cada poscion.

```
<div class="four columns">
  <div class="card">
    " class="imagen-curso u-full-width">
    <div class="info-card">
```

Haremos los mismo, en cada etiqueta donde se imprimían anteriormente los datos ahora pondremos la etiqueta de php seguido de un echo que tendrá la variable row y dentro el nombre de la columna de la base de datos que tiene la información que queremos poner, esto lo repetiremos en todas la etiquetas que imprimen la información.

```
<h4><?php echo $row["producto"]; ?></h4>
<p><?php echo $row["artista"]; ?></p>

<p class="precio"><?php echo "$".$row["precioant"]; ?> <span class="u-pull-right ">
<?php echo "$".$row["precioact"]; ?></span></p>
<a href="#" class="u-full-width button-primary button input agregar-carrito" data-
id="1">Agregar Al Carrito</a>
  </div>
</div> <!-- .card -->
</div>
```

Después pondremos otras etiquetas de php y lo que haremos ser poner un if como el anterior que te diga que, si el Contador al dividirlo su residuo es 0, entonces imprime la etiqueta que Cierra la el div anterior. Y pondremos el aumento del contador que ira de uno en uno.

```
<?php if($cont%3==0) echo '</div>'; ?>
```

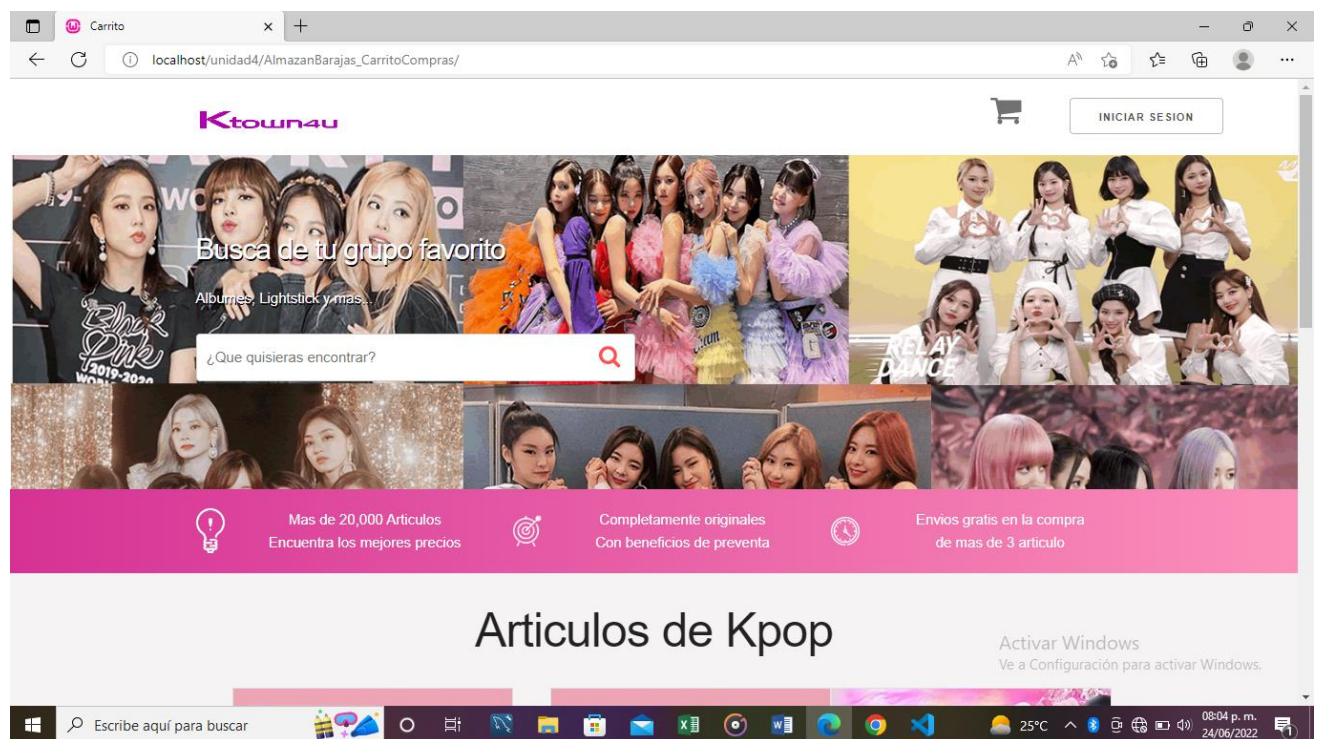
```
<?php
    $cont++;
}
}
```

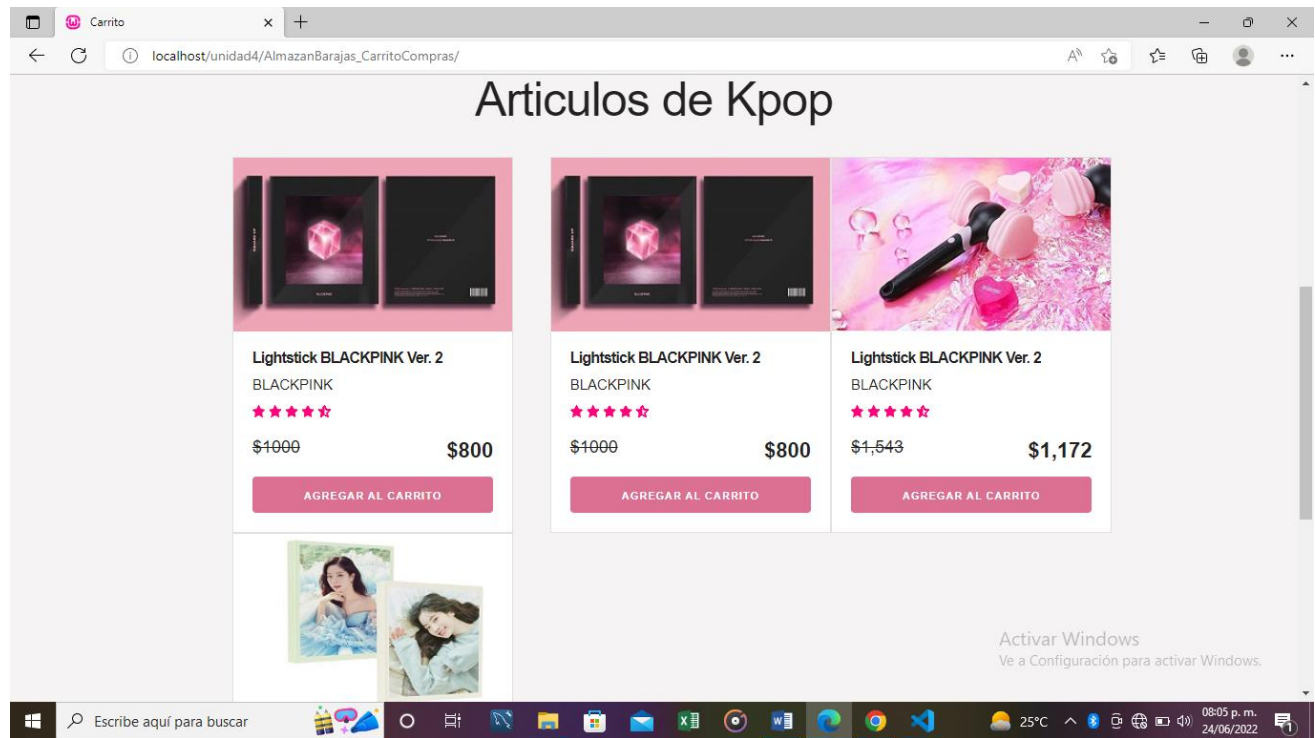
Por ultimo en el catch pondremos una advertencia que nos avise si hay un error al enlazar la base de datos.

```
catch(PDOException $e){
    echo "error".$e->getMessage();
}

?>
```

- **Imagen de la página**





Conclusión

Esta práctica fue complicada de realizar ya que se tenía que hacer uso de todos los métodos vistos en clase y saber cómo aprovecharlos, ya que eran muchas funciones que se tenían que hacer. El implementar lo que es una base de datos en nuestro sitio hace que sea un poco ya más profesional, el saber manejar estos métodos en sitios web nos ayudara en el campo laboral ya que son funciones básicas que se requiere en las páginas, esto también optimizar tu código ya que no tienes que estar repitiendo código únicamente ir poniendo en la base de datos.