

**Universidad Autónoma de Nuevo León**  
**Facultad de Ingeniería Mecánica y Eléctrica**

**DISEÑO Y ESPECIFICACION DE SOFTWARES**

**Manual Técnico**

**Equipo #5**

Nombre	Matricula	Carrera
Dana Ximena Mendoza Medellín	2009817	IAS
Carlos Alberto Reyes Martínez	2082344	IAS
Alejandra Hernández Rodríguez	1994078	IAS
Jair Antonio Ruelas Rosales	1895676	IAS

**Grupo 001**

**Docente: Patricia Garza Garza**

**Semestre Enero - Junio 2024**

**Cd. Universitaria, San Nicolás de los Garza**

## Índice

Introducción.....	3
Requisitos del Sistema: .....	3
Hardware: .....	3
Software:.....	3
Funcionalidades: .....	4
Implementación: .....	19
1. Configuración del Entorno de Desarrollo: .....	19
2. Desarrollo del Backend: .....	19
3. Desarrollo del Frontend:.....	19
4. Pruebas y Depuración:.....	19
5. Implementación en Producción: .....	19
Mantenimiento y Soporte:.....	20
1. Actualizaciones: .....	20
1. Soporte Técnico: .....	20
2. Monitoreo y Seguridad: .....	20

## Introducción

Este manual técnico proporciona una guía detallada para el desarrollo y mantenimiento de un sistema de gestión de inventario para la tienda online de llaveros de Funko de Aaron Emmanuel Almaguer Mata. El sistema permitirá organizar el inventario por series y películas, ajustar el stock según la popularidad de ventas y priorizar los productos más vendidos para promocionarlos. Se utilizarán tecnologías como Xampp, Visual Studio Code, PHP, SQL, HTML, CSS y JavaScript para implementar y mantener la página web.

### Requisitos del Sistema:

#### Hardware:

- Computadora con al menos 4GB de RAM
- Espacio de almacenamiento suficiente para el desarrollo y almacenamiento de datos

#### Software:

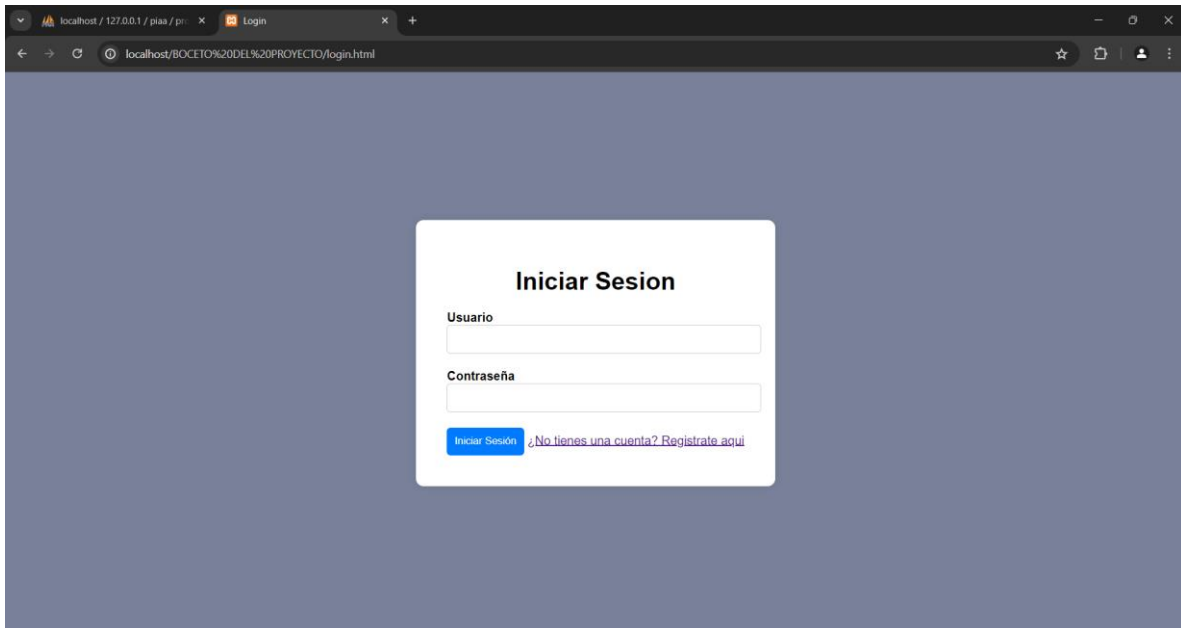
- Xampp: Servidor local para el desarrollo y ejecución de la página web.
- Visual Studio Code: Editor de código para el desarrollo de la página web.
- PHP: Lenguaje de programación para el desarrollo del backend.
- SQL: Lenguaje de consulta estructurado para la gestión de la base de datos.
- HTML, CSS y JavaScript: Tecnologías para el desarrollo del frontend de la página web.



# Funcionalidades:

## 1. Login:

Permite a los usuarios autenticarse en la página web utilizando sus credenciales de acceso.



## HTML LOGIN

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  Buscar
El modo restringido está destinado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones.  Administrar  Más información

login.html X
C:\xampp\htdocs> BOCETO DEL PROYECTO > login.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Login</title>
7      <link rel="stylesheet" href="login.css">
8  </head>
9  <body>
10     <form method="post" action="conexion.php">
11         <div class="container">
12             <div class="login-container">
13                 <h1>Iniciar Sesión</h1>
14
15                 <div class="input-container">
16                     <label for="username">Usuario</label>
17                     <input type="text" id="username" name="txtusuario" required>
18                 </div>
19                 <div class="input-container">
20                     <label for="password">Contraseña</label>
21                     <input type="password" id="password" name="txtpassword" required>
22                 </div>
23                 <button type="submit">Iniciar Sesión</button>
24             </div>
25             <a href="registrar.html">¿No tienes una cuenta? Regístrate aquí</a>
26         </div>
27     </form>
28 </body>
29 </html>
31
```

## CSS LOGIN

```
Archivo Editar Selección Ver Ir Ejecutar ...
El modo restringido está desactivado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información

# login.css
C:\xampp\htdocs> BOCEITO DEL PROYECTO > # login.css > body

1 body {
2   margin: 0;
3   padding: 0;
4   font-family: Arial, sans-serif;
5   background-color: #f7f8f9;
6 }
7
8
9 .container {
10  display: flex;
11  justify-content: center;
12  align-items: center;
13  height: 100vh;
14 }
15
16 .login-container {
17  background-color: #fff;
18  border-radius: 10px;
19  padding: 40px;
20  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
21 }
22
23 .login-container h1 {
24  text-align: center;
25  margin-bottom: 20px;
26 }
27
28 form {
29  display: flex;
30  flex-direction: column;
31 }
32
33 .input-container {
34  margin-bottom: 20px;
35 }
36
37 .input-container label {
38  font-weight: bold;
39 }
40
41 .input-container input {
42  padding: 10px;
43  border: 1px solid #ccc;
44  border-radius: 5px;
45  width: 100%;
46 }
```

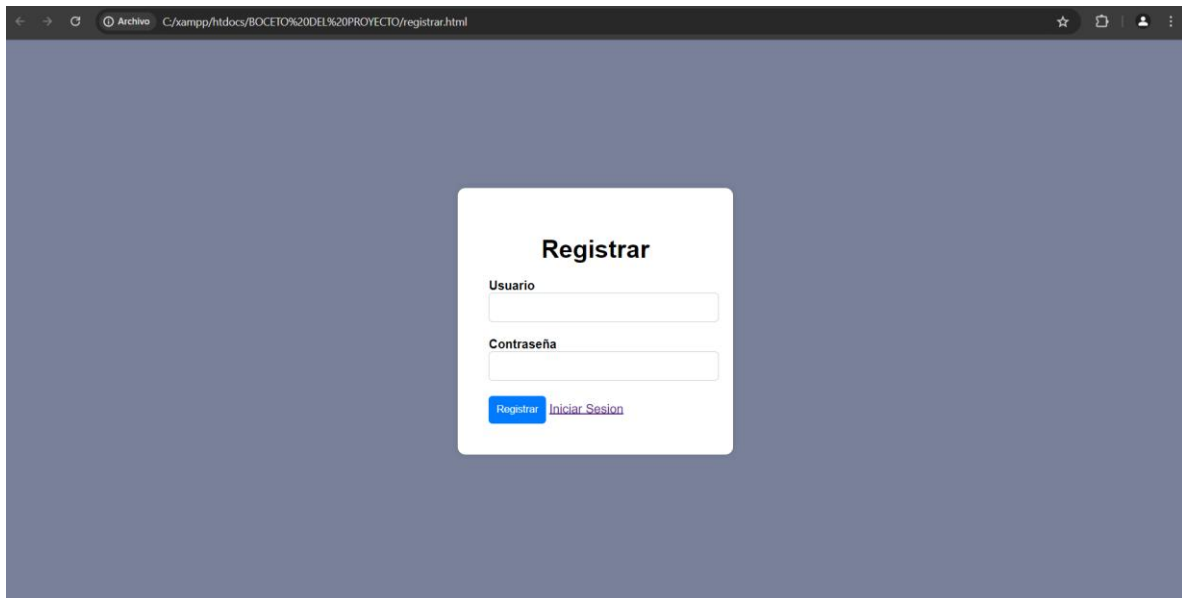
## PHP LOGIN

```
Archivo Editar Selección Ver Ir Ejecutar ...
El modo restringido está desactivado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información

# login.css  # conexion.php
C:\xampp\htdocs> BOCEITO DEL PROYECTO > # conexion.php

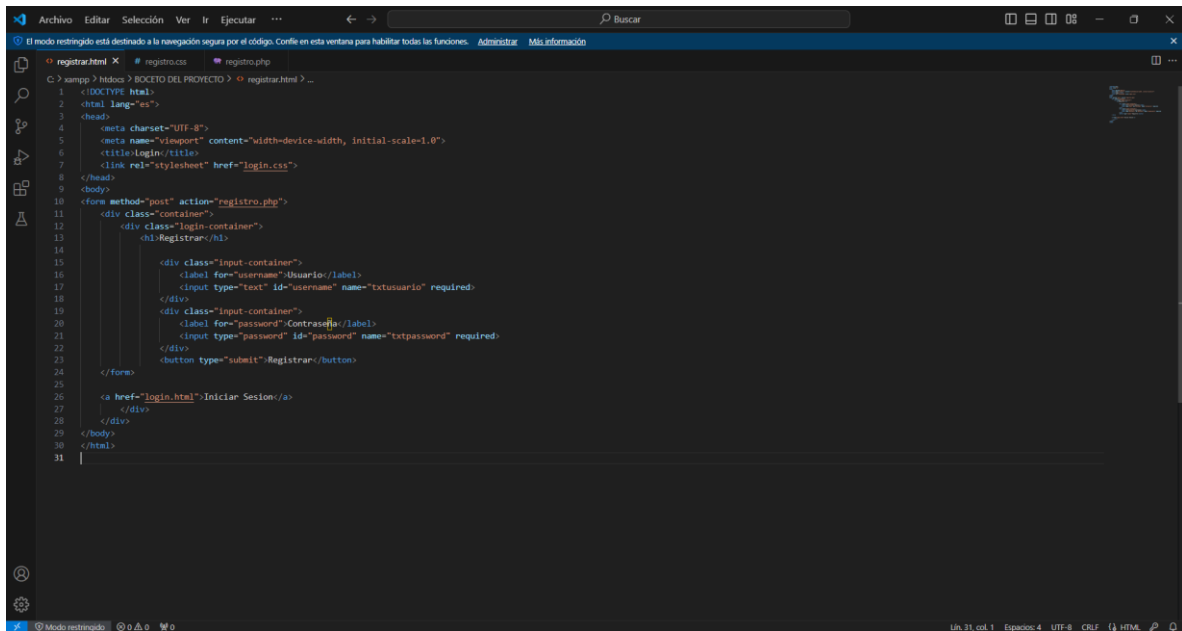
1 <?php
2
3 $dbhost = "localhost";
4 $dbuser = "root";
5 $dbpass = "";
6 $dbname = "pilas";
7
8 $conn = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
9 if (!$conn)
10 {
11     die("No hay conexión: " . mysqli_connect_error());
12 }
13
14 $nombre = $_POST["txtusuario"];
15 $pass = $_POST["txtpassword"];
16
17 $query = mysqli_query($conn, "SELECT * FROM login WHERE usuario = '" . $nombre . "' and password = '" . $pass . "'");
18 $nr = mysqli_num_rows($query);
19
20 if ($nr == 1)
21 {
22     // Redirige a menu.html si los datos son válidos
23     header("location: menu.html");
24     exit; // Asegúrate de que el script se detenga después de la redirección
25 }
26 else
27 {
28     // Si no hay coincidencias en la base de datos, redirige de nuevo a login.html con una alerta
29     echo "<script>alert('Error: Nombre de usuario o contraseña incorrectos');</script>";
30     echo "<script>window.location= 'login.html';</script>";
31 }
32
33
34
```

# REGISTRAR



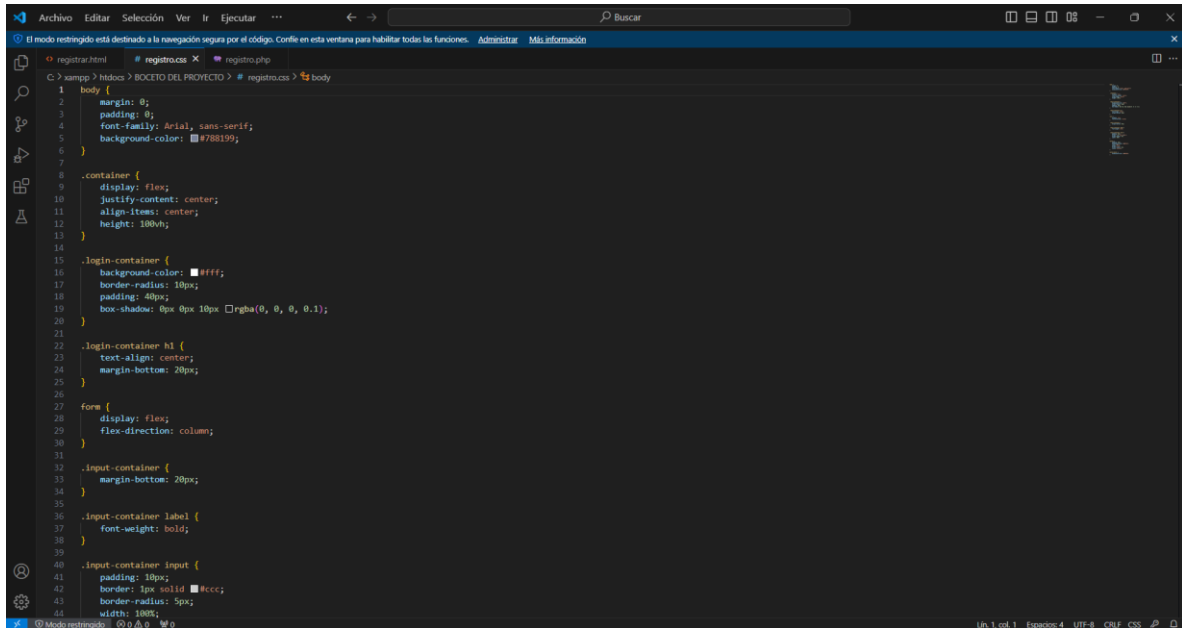
A screenshot of a web browser displaying a registration form. The browser's address bar shows the URL: C:\xampp\htdocs\BOCETO%20DEL%20PROYECTO\registrar.html. The form is centered on a light blue background and is titled "Registrar". It contains two input fields: "Usuario" and "Contraseña". Below the "Contraseña" field, there is a blue button labeled "Registrar" and a link labeled "Iniciar Sesión".

## HTML REGISTRAR



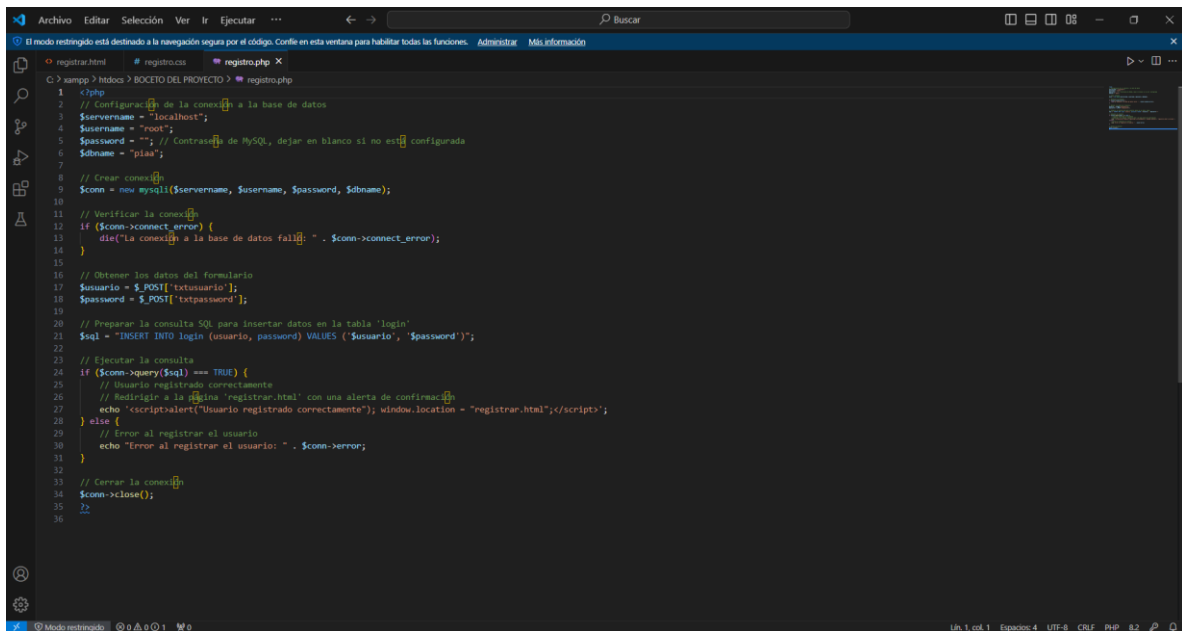
```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>login//title>
7   <link rel="stylesheet" href="login.css">
8 </head>
9 <body>
10  <form method="post" action="registro.php">
11    <div class="container">
12      <div class="login-container">
13        <h1>Registrar</h1>
14
15        <div class="input-container">
16          <label for="username">Usuario</label>
17          <input type="text" id="username" name="txtusuario" required>
18        </div>
19        <div class="input-container">
20          <label for="password">Contraseña</label>
21          <input type="password" id="password" name="txtpassword" required>
22        </div>
23        <button type="submit">Registrar</button>
24      </form>
25
26      <a href="login.html">Iniciar Sesión</a>
27    </div>
28  </body>
29 </html>
```

## CSS REGISTRAR



```
1 body {
2   margin: 0;
3   padding: 0;
4   font-family: Arial, sans-serif;
5   background-color: #f0f0f0;
6 }
7
8 .container {
9   display: flex;
10  justify-content: center;
11  align-items: center;
12  height: 100vh;
13 }
14
15 .login-container {
16   background-color: #ffff;
17   border-radius: 10px;
18   padding: 40px;
19   box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
20 }
21
22 .login-container h1 {
23   text-align: center;
24   margin-bottom: 20px;
25 }
26
27 form {
28   display: flex;
29   flex-direction: column;
30 }
31
32 .input-container {
33   margin-bottom: 20px;
34 }
35
36 .input-container label {
37   font-weight: bold;
38 }
39
40 .input-container input {
41   padding: 10px;
42   border: 1px solid #ccc;
43   border-radius: 5px;
44   width: 100%;
45 }
```

## PHP REGISTRAR

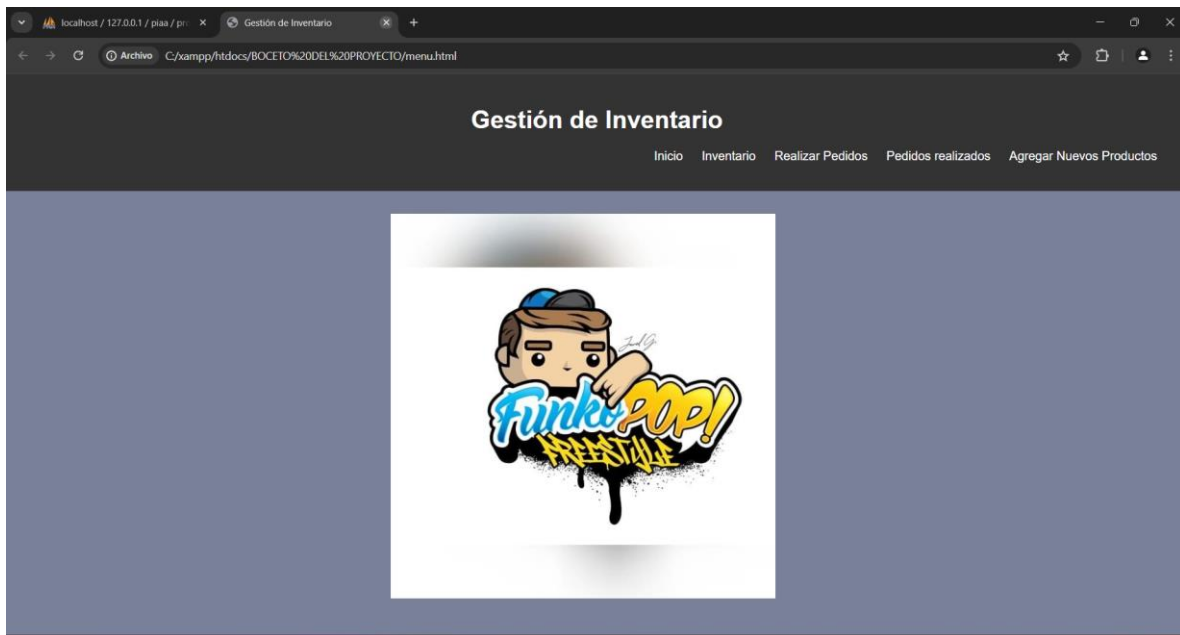


```
1 <?php
2 // Configuraci3n de la conexi3n a la base de datos
3 $servername = "localhost";
4 $username = "root";
5 $password = ""; // Contrasea de MySQL, dejar en blanco si no est3 configurada
6 $dbname = "plata";
7
8 // Crear conexi3n
9 $conn = new mysqli($servername, $username, $password, $dbname);
10
11 // Verificar la conexi3n
12 if ($conn->connect_error) {
13   die("La conexi3n a la base de datos fall3: " . $conn->connect_error);
14 }
15
16 // Obtener los datos del formulario
17 $usuario = $_POST['txtusuario'];
18 $password = $_POST['txtpassword'];
19
20 // Preparar la consulta SQL para insertar datos en la tabla 'login'
21 $sql = "INSERT INTO login (usuario, password) VALUES ('$usuario', '$password')";
22
23 // Ejecutar la consulta
24 if ($conn->query($sql) === TRUE) {
25   // Usuario registrado correctamente
26   // Redirigir a la p3gina 'registrar.html' con una alerta de confirmaci3n
27   echo "<script>alert('Usuario registrado correctamente'); window.location = 'registrar.html';</script>";
28 } else {
29   // Error al registrar el usuario
30   echo "Error al registrar el usuario: " . $conn->error;
31 }
32
33 // Cerrar la conexi3n
34 $conn->close();
35
36 ?>
```

## 2. Inicio:

Presenta una vista general de la tienda y los productos destacados.

Muestra promociones y ofertas especiales.



## HTML INICIO

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Gestión de Inventario</title>
7 <link rel="stylesheet" href="menu.css">
8 </head>
9 <body>
10 <div class="container">
11 <div class="menu-container">
12 <h1>Gestión de Inventario</h1>
13
14 <nav>
15 <ul>
16 <li><a href="menu.html">Inicio</a></li>
17 <li><a href="stock.html">Inventario</a></li>
18 <li><a href="pedidos.html">Realizar Pedidos</a></li>
19 <li><a href="pedidosrealizados.html">Pedidos realizados</a></li>
20 <li><a href="agregar.html">Agregar Nuevos Productos</a></li>
21 </ul>
22 </nav>
23 </div>
24 </div>
25
26
27 
28
29
30
31
32 </body>
33 </html>
34
```



## CSS INICIO

```
Archivo Editar Selección Ver Ir Ejecutar ... Buscar
El modo restringido está destinado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información
C:\xampp\htdocs> BOCETO DEL PROYECTO # menu1.css
1 body {
2   margin: 0;
3   padding: 0;
4   font-family: Arial, sans-serif;
5   background-color: #788199;
6 }
7 .content-container h2 {
8   color: yellow; /* Cambia el color del texto del elemento h1 dentro de ambas clases */
9 }
10
11 .container {
12   display: flex;
13 }
14 .menu-container {
15   width: 100%;
16   background-color: #333;
17   color: #fff;
18   padding: 20px;
19 }
20
21 .menu-container h1 {
22   margin-bottom: 20px;
23   text-align: center;
24 }
25
26 .menu-container nav ul {
27   list-style-type: none;
28   padding: 0;
29   text-align: right;
30 }
31
32 .menu-container nav ul li {
33   display: inline;
34   margin-right: 20px;
35 }
36
37 .menu-container nav ul li a {
38   color: #fff;
39   text-decoration: none;
40 }
41
42 .content-container {
43   width: 100%;
44   padding: 20px;
45 }
Modo restringido 0.0.0 M0 Lin 1, col 1 Espacios 4 UTF-8 CRLF CSS
```

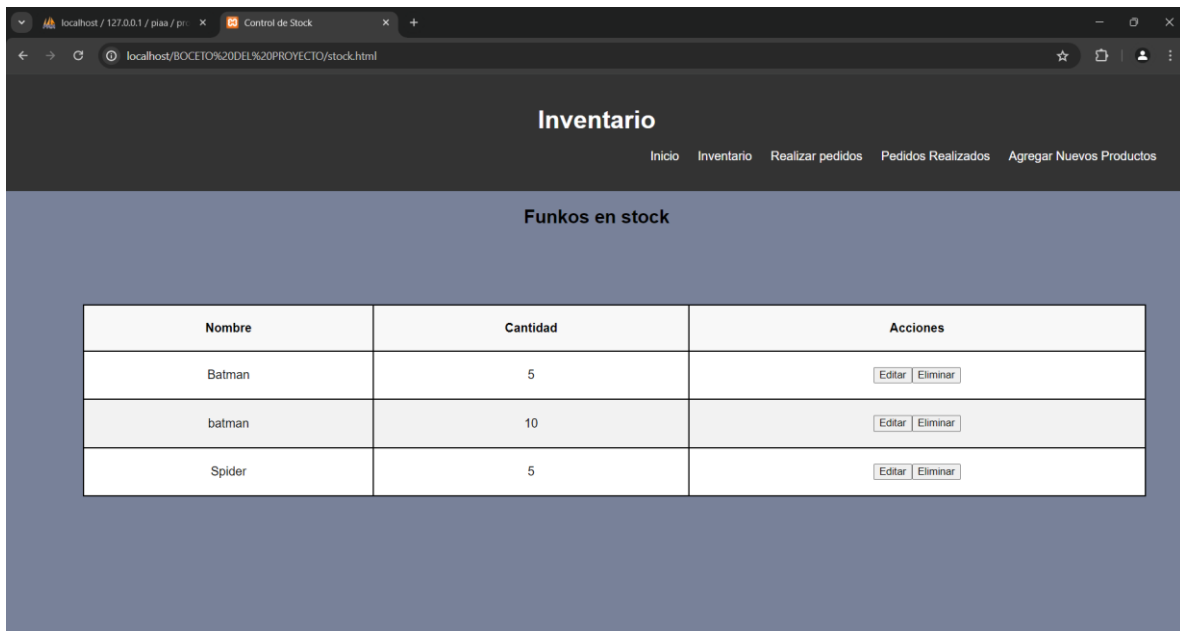
## PHP INICIO

```
Archivo Editar Selección Ver Ir Ejecutar ... Buscar
El modo restringido está destinado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información
C:\xampp\htdocs> BOCETO DEL PROYECTO # menu.php
1 <?php
2   <html lang="es">
3   <body>
4   <div class="content-container">
5   <div class="fundo-container">
6   <div>
7   <div>
8   // Conectar a la base de datos (reemplaza con tus propios datos de conexión)
9   $servername = "localhost";
10  $username = "root";
11  $password = "";
12  $dbname = "pia";
13
14  // Crear conexión
15  $conn = new mysqli($servername, $username, $password, $dbname);
16
17  // Verificar la conexión
18  if ($conn->connect_error) {
19    die("Conexión fallida: " . $conn->connect_error);
20  }
21
22  // Preparar y ejecutar consulta SQL para obtener productos
23  $sql = "SELECT * FROM productos";
24  $result = $conn->query($sql);
25
26  // Mostrar productos en la página HTML
27  if ($result->num_rows > 0) {
28    while ($row = $result->fetch_assoc()) {
29      echo <div class="fundo">;
30      echo <div> . $row["nombre"] . </div>;
31      echo <div> . $row["cantidad"] . </div>;
32      // Aquí puedes mostrar la imagen si la guardaste en la base de datos
33      echo </div>;
34    }
35  } else {
36    echo "No se encontraron productos.";
37  }
38
39  // Cerrar la conexión
40  $conn->close();
41
42 </div>
43 </body>
44 </html>
Modo restringido 0.0.0 M0 Lin 1, col 1 Espacios 4 UTF-8 CRLF PHP 8.2
```

### 3. Inventario:

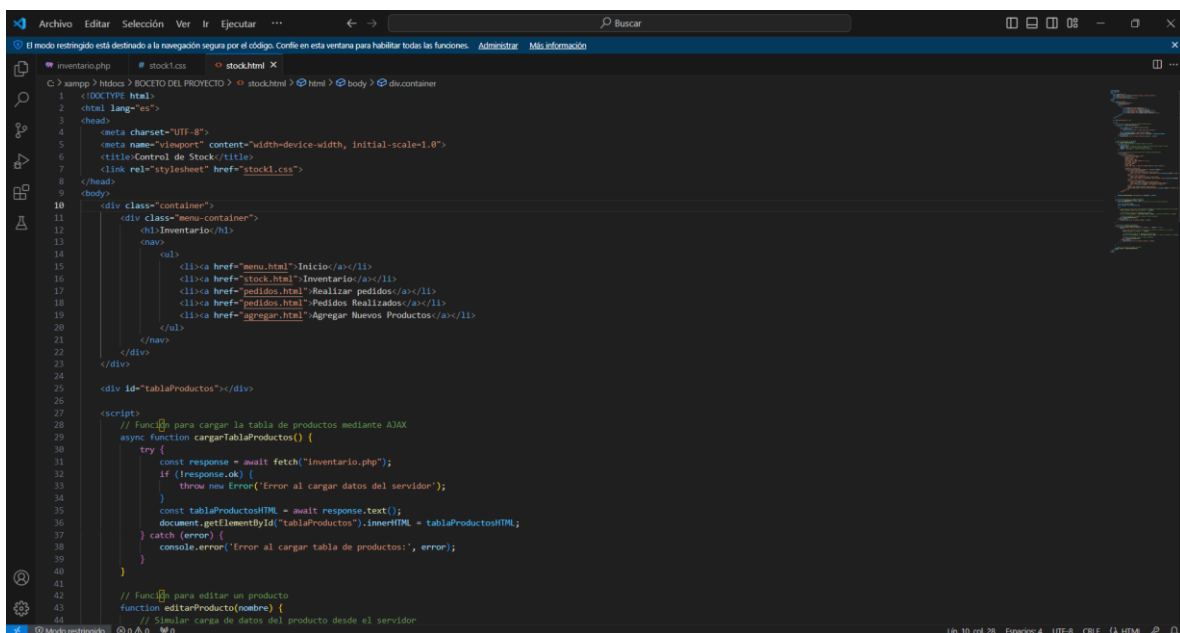
Organiza los productos por series y películas de Funko.

Permite visualizar información detallada de cada producto, como nombre, precio y stock disponible.



Nombre	Cantidad	Acciones
Batman	5	<a href="#">Editar</a> <a href="#">Eliminar</a>
batman	10	<a href="#">Editar</a> <a href="#">Eliminar</a>
Spider	5	<a href="#">Editar</a> <a href="#">Eliminar</a>

### HTML INVENTARIO



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Control de Stock</title>
7   <link rel="stylesheet" href="stock1.css">
8 </head>
9 <body>
10  <div class="container">
11    <div class="menu-container">
12      <h1>Inventario</h1>
13      <nav>
14        <ul>
15          <li><a href="menu.html">Inicio</a></li>
16          <li><a href="stock.html">Inventario</a></li>
17          <li><a href="pedidos.html">Realizar pedidos</a></li>
18          <li><a href="pedidos.html">Pedidos Realizados</a></li>
19          <li><a href="agregar.html">Agregar Nuevos Productos</a></li>
20        </ul>
21      </nav>
22    </div>
23  </div>
24  <div id="tablaProductos"></div>
25
26  <script>
27    // Función para cargar la tabla de productos mediante AJAX
28    async function cargarTablaProductos() {
29      try {
30        const response = await fetch('inventario.php');
31        if (!response.ok) {
32          throw new Error('Error al cargar datos del servidor');
33        }
34        const tablaProductosHTML = await response.text();
35        document.getElementById('tablaProductos').innerHTML = tablaProductosHTML;
36      } catch (error) {
37        console.error('Error al cargar tabla de productos:', error);
38      }
39    }
40
41    // Función para editar un producto
42    function editarProducto(nombre) {
43      // Simular carga de datos del producto desde el servidor
44    }
45  </script>
```

# CSS INVENTARIO

```
Archivo Editar Selección Ver Ir Ejecutar ...
El modo restringido está destinado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información

inventario.php stock1.css stock.html
C:\xampp\htdocs\BOCETO DEL PROYECTO > stock1.css > body

1 body {
2     margin: 0;
3     padding: 0;
4     font-family: Arial, sans-serif;
5     background-color: #788199;
6 }
7 .content-container h2 {
8     color: yellow; /* Cambia el color del texto del elemento h1 dentro de ambas clases */
9 }
10
11 .container {
12     display: flex;
13 }
14 .menu-container {
15     width: 100%;
16     background-color: #8333;
17     color: #ffff;
18     padding: 20px;
19 }
20
21 .menu-container h1 {
22     margin-bottom: 20px;
23     text-align: center;
24 }
25
26 .menu-container nav ul {
27     list-style-type: none;
28     padding: 0;
29     text-align: right;
30 }
31
32 .menu-container nav ul li {
33     display: inline;
34     margin-right: 20px;
35 }
36
37 .menu-container nav ul li a {
38     color: #ffff;
39     text-decoration: none;
40 }
41
42 .content-container {
43     width: 100%;
44     padding: 20px;
45 }
```

# PHP INVENTARIO

```
Archivo Editar Selección Ver Ir Ejecutar ...
El modo restringido está destinado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información

inventario.php stock1.css stock.html
C:\xampp\htdocs\BOCETO DEL PROYECTO > inventario.php

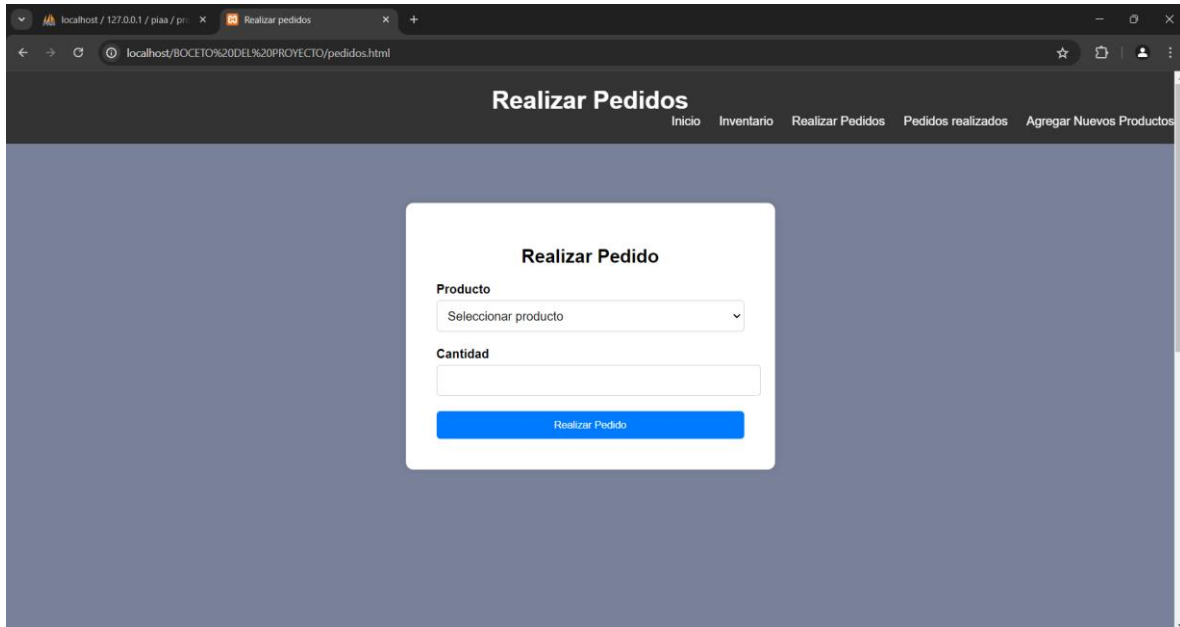
47 <?php
48 // Establecer conexión con la base de datos
49 $conexion = new mysqli('localhost', 'root', '', 'plaa');
50
51 // Verificar la conexión
52 if ($conexion->connect_error) {
53     die("Error en la conexión: " . $conexion->connect_error);
54 }
55
56 // Consulta SQL para obtener los productos
57 $sql = "SELECT nombre, cantidad FROM productos";
58 $resultado = $conexion->query($sql);
59
60 // Mostrar la tabla HTML con los productos y botones de acción
61 echo <table class="styled-table">;
62 echo <thead>;
63 echo <tr>;
64 echo <th>Nombre</th>;
65 echo <th>Cantidad</th>;
66 echo <th>Acciones</th>;
67 echo </tr>;
68 echo </thead>;
69 echo <tbody>;
70
71 if ($resultado->num_rows > 0) {
72     while ($fila = $resultado->fetch_assoc()) {
73         echo <tr>;
74         echo <td> . $fila['nombre'] . </td>;
75         echo <td> . $fila['cantidad'] . </td>;
76         echo <td>
77             <button onclick="editarProducto(\' . $fila['nombre'] . \')">Editar</button>;
78             <button onclick="eliminarProducto(\' . $fila['nombre'] . \')">Eliminar</button>;
79         echo </td>;
80         echo </tr>;
81     }
82 } else {
83     echo <tr><td colspan="3">No se encontraron productos.</td></tr>;
84 }
85
86 echo </tbody>;
87 echo </table>;
88
89 // Cerrar la conexión
90 $conexion->close();
```

#### 4. Realizar Pedidos:

Permite a los usuarios agregar productos al carrito de compras.

Permite seleccionar la cantidad de productos deseados.

Procesa el pedido una vez completado.

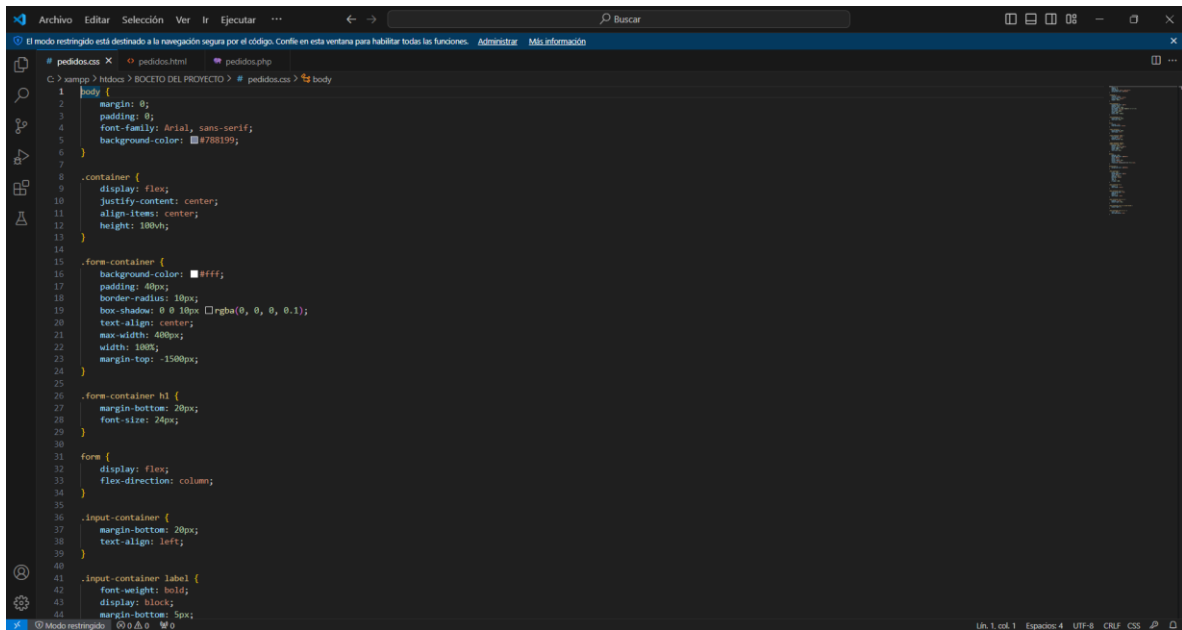


#### HTML REALIZAR PEDIDOS

```
Archivo Editar Selección Ver Ir Ejecutar ...
El modo restringido está diseñado a la navegación segura por el código. Confié en esta ventana para habilitar todas las funciones. Administrar Mis información

# pedidos.css # pedidos.html # pedidos.php
C:\xampp\htdocs\BOCETO DEL PROYECTO > pedidos.html > html > body > div.container > div.menu-container > nav > ul > li
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Realizar pedidos</title>
7 <link rel="stylesheet" href="pedidos.css">
8 </head>
9 <body>
10
11 <div class="container">
12 <div class="menu-container">
13 <h1>Realizar Pedidos</h1>
14
15 <nav>
16 <ul>
17 <li><a href="menu.html">Inicio</a></li>
18 <li><a href="stock.html">Inventario</a></li>
19 <li><a href="pedidos.html">Realizar Pedidos</a></li>
20 <li><a href="pedidosrealizados.html">Pedidos realizados</a></li>
21 <li><a href="agregar.html">Agregar Nuevos Productos</a></li>
22 </ul>
23 </nav>
24 </div>
25
26 <div class="container">
27 <div class="form-container">
28 <div class="form-container">
29 <h1>Realizar Pedido</h1>
30 <form action="pedidos.php" method="post">
31 <div class="input-container">
32 <label for="producto">Producto</label>
33 <select id="producto" name="producto" required>
34 <option value="">Seleccionar producto</option>
35 <option value="funke_spiderman">Funke Spider-Man</option>
36 <option value="funke_batman">Funke Batman</option>
37 <option value="funke_darth_vader">Funke Darth Vader</option>
38 <option value="funke_ant_man">Funke Ant Man</option>
39 <option value="funke_boba_fett">Funke Boba Fett</option>
40 <!-- Agrega más opciones según sea necesario -->
41 </select>
42 </div>
43 <div class="input-container">
44 <label for="cantidad">Cantidad</label>
45 <input type="text">
46 </div>
47 <div class="button">
48 <button type="submit">Realizar Pedido</button>
49 </div>
50 </form>
51 </div>
52 </div>
53 </div>
54 </body>
55 </html>
```

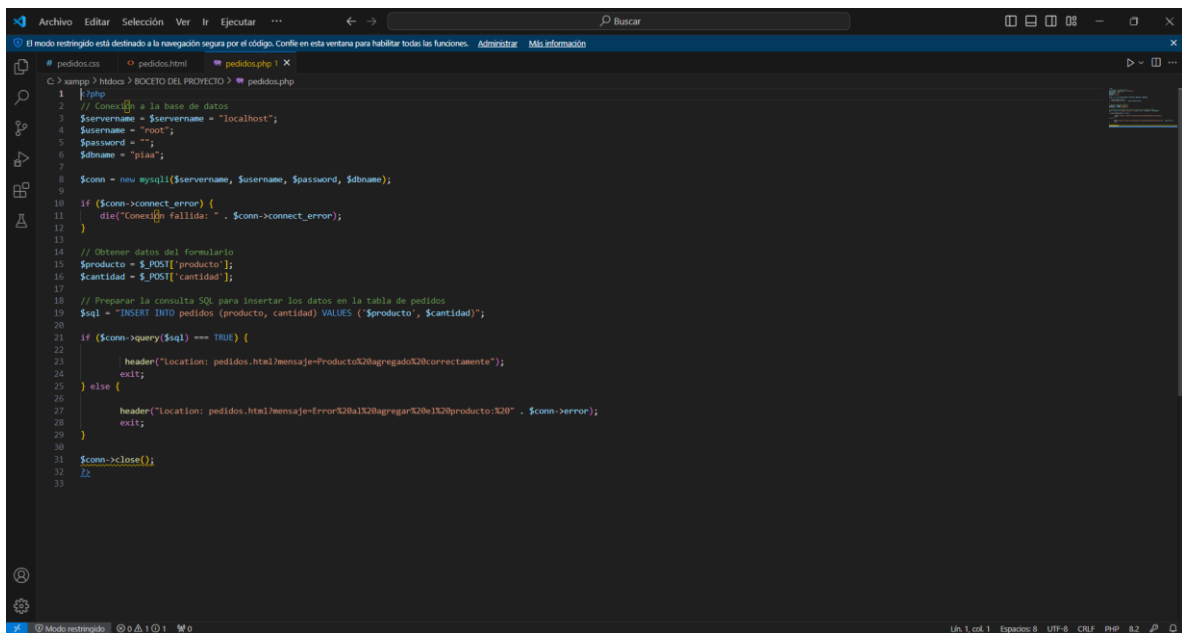
## CSS REALIZAR PEDIDOS



The screenshot shows a code editor with a dark theme. The file explorer on the left shows a project named 'pedidos' with files 'pedidos.html' and 'pedidos.php'. The main editor area displays CSS code for a form. The code includes a 'body' style, a '.container' style, a '.form-container' style, a '.form-container h1' style, and an '.input-container' style. The code is as follows:

```
1 body {
2     margin: 0;
3     padding: 0;
4     font-family: Arial, sans-serif;
5     background-color: #f78199;
6 }
7
8 .container {
9     display: flex;
10    justify-content: center;
11    align-items: center;
12    height: 100vh;
13 }
14
15 .form-container {
16     background-color: #fff;
17     padding: 40px;
18     border-radius: 10px;
19     box-shadow: 0 0 10px #rgba(0, 0, 0, 0.1);
20     text-align: center;
21     max-width: 400px;
22     width: 100%;
23     margin-top: -150px;
24 }
25
26 .form-container h1 {
27     margin-bottom: 20px;
28     font-size: 24px;
29 }
30
31 .form {
32     display: flex;
33     flex-direction: column;
34 }
35
36 .input-container {
37     margin-bottom: 20px;
38     text-align: left;
39 }
40
41 .input-container label {
42     font-weight: bold;
43     display: block;
44     margin-bottom: 5px;
45 }
```

## PHP RELIZAR PEDIDOS



The screenshot shows a code editor with a dark theme. The file explorer on the left shows a project named 'pedidos' with files 'pedidos.html' and 'pedidos.php'. The main editor area displays PHP code for a form. The code includes a database connection, a query to insert data into a table named 'pedidos', and a header function to display the result. The code is as follows:

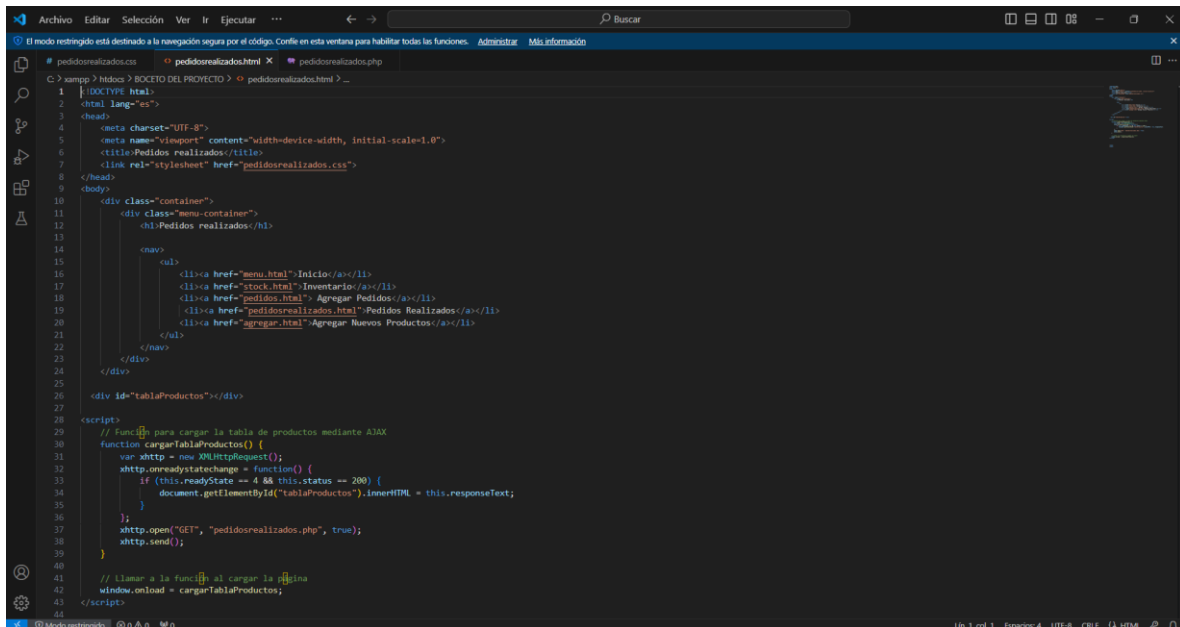
```
1 <?php
2 // Conexión a la base de datos
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "pilas";
7
8 $conn = new mysqli($servername, $username, $password, $dbname);
9
10 if ($conn->connect_error) {
11     die("Conexión fallida: " . $conn->connect_error);
12 }
13
14 // Obtener datos del formulario
15 $producto = $_POST['producto'];
16 $cantidad = $_POST['cantidad'];
17
18 // Preparar la consulta SQL para insertar los datos en la tabla de pedidos
19 $sql = "INSERT INTO pedidos (producto, cantidad) VALUES ('$producto', $cantidad)";
20
21 if ($conn->query($sql) === TRUE) {
22     header("Location: pedidos.html?mensaje=Producto%20agregado%20correctamente");
23     exit;
24 } else {
25     header("Location: pedidos.html?mensaje=Error%20al%20agregar%20el%20producto%20");
26     exit;
27 }
28
29 $conn->close();
30
31 <?
32
33
```

## 5. Pedidos Realizados:

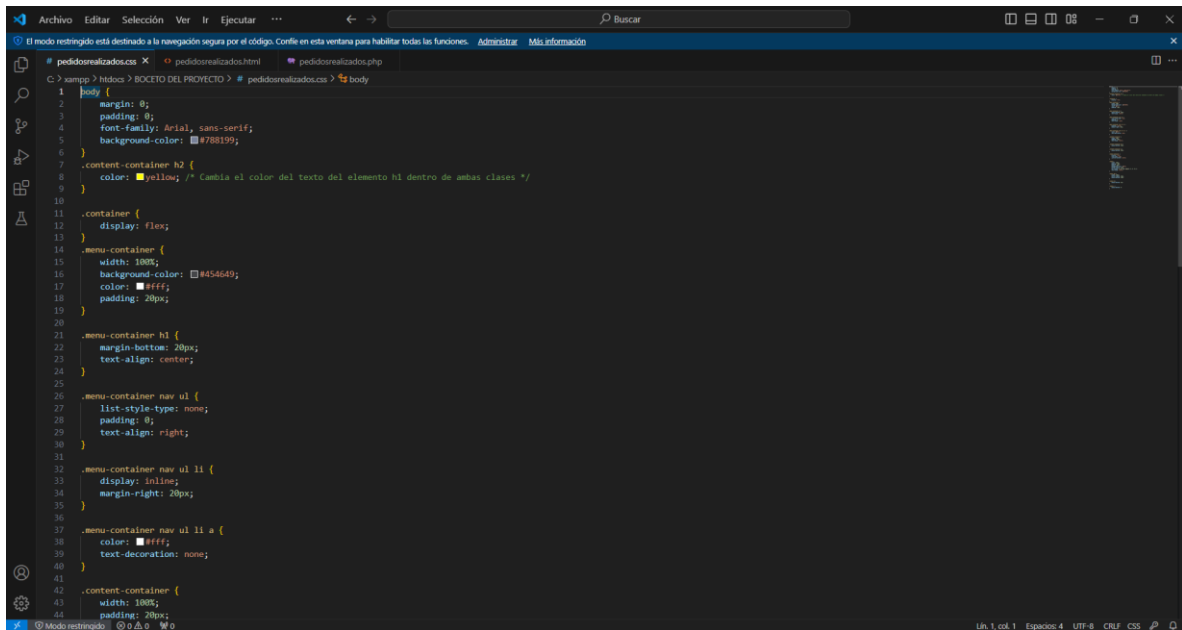
Muestra un historial de los pedidos realizados por el usuario.



## HTML PEDIDOS REALIZADOS

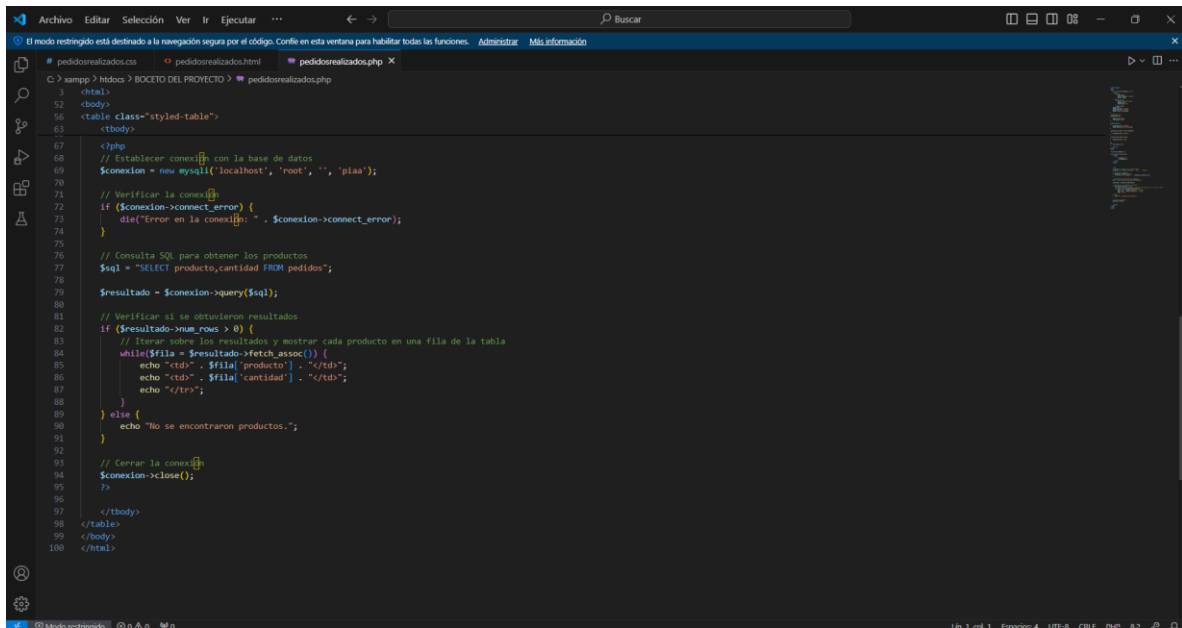


## CSS PEDIDOS REALIZADOS



```
1 body {
2     margin: 0;
3     padding: 0;
4     font-family: Arial, sans-serif;
5     background-color: #478199;
6 }
7 .content-container h2 {
8     color: yellow; /* Cambia el color del texto del elemento h1 dentro de ambas clases */
9 }
10
11 .container {
12     display: flex;
13 }
14 .menu-container {
15     width: 100%;
16     background-color: #454649;
17     color: #ffff;
18     padding: 20px;
19 }
20
21 .menu-container h1 {
22     margin-bottom: 20px;
23     text-align: center;
24 }
25
26 .menu-container nav ul {
27     list-style-type: none;
28     padding: 0;
29     text-align: right;
30 }
31
32 .menu-container nav ul li {
33     display: inline;
34     margin-right: 20px;
35 }
36
37 .menu-container nav ul li a {
38     color: #ffff;
39     text-decoration: none;
40 }
41
42 .content-container {
43     width: 100%;
44     padding: 20px;
45 }
```

## PHP PEDIDOS REALIZADOS

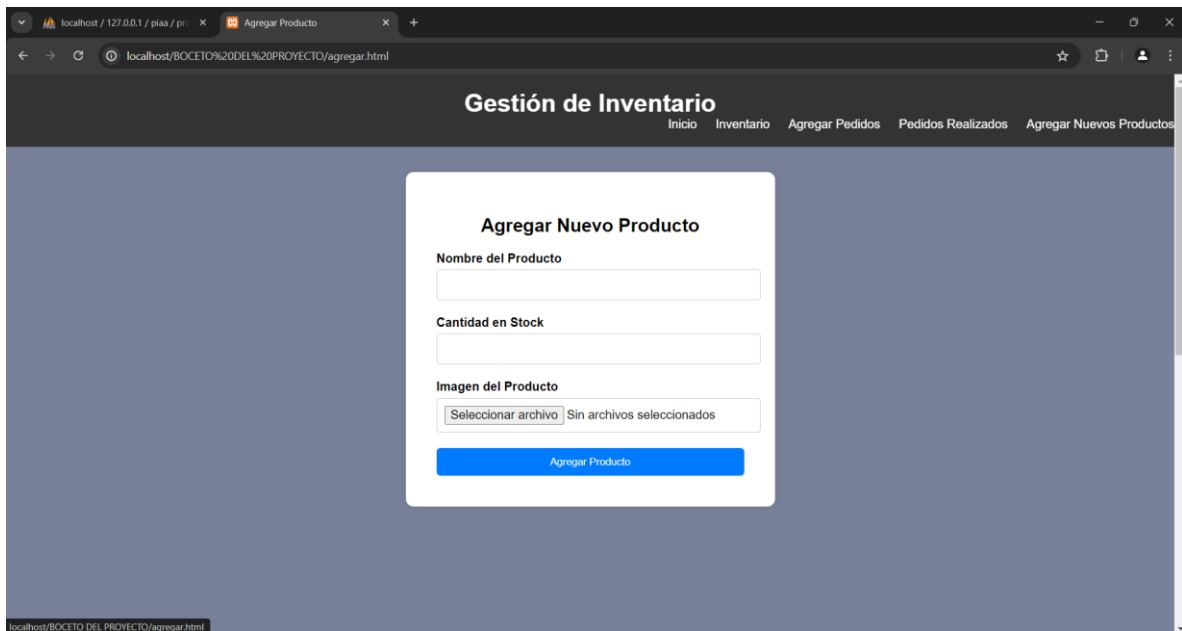


```
1 <html>
2 <body>
3     <table class="styled-table">
4         <tbody>
5             <tr>
6                 <td>
7                     <?php
8                         // Establecer conexión con la base de datos
9                         $conexion = new mysqli("localhost", "root", "", "pilas");
10
11                         // Verificar la conexión
12                         if ($conexion->connect_error) {
13                             die("Error en la conexión: " . $conexion->connect_error);
14                         }
15
16                         // Consulta SQL para obtener los productos
17                         $sql = "SELECT producto,cantidad FROM pedidos";
18
19                         $resultado = $conexion->query($sql);
20
21                         // Verificar si se obtuvieron resultados
22                         if ($resultado->num_rows > 0) {
23                             // Iterar sobre los resultados y mostrar cada producto en una fila de la tabla
24                             while($fila = $resultado->fetch_assoc()) {
25                                 echo "<tr>";
26                                 echo "<td>" . $fila["producto"] . "</td>";
27                                 echo "<td>" . $fila["cantidad"] . "</td>";
28                                 echo "</tr>";
29                             }
30                         } else {
31                             echo "No se encontraron productos.";
32                         }
33
34                         // Cerrar la conexión
35                         $conexion->close();
36                     </?php>
37                 </td>
38             </tr>
39         </tbody>
40     </table>
41 </body>
42 </html>
```

## 6. Agregar Nuevos Productos:

Permite al administrador de la tienda agregar nuevos productos al inventario.

Proporciona un formulario para ingresar la información del producto, como nombre, agregar imagen y cantidad disponible.



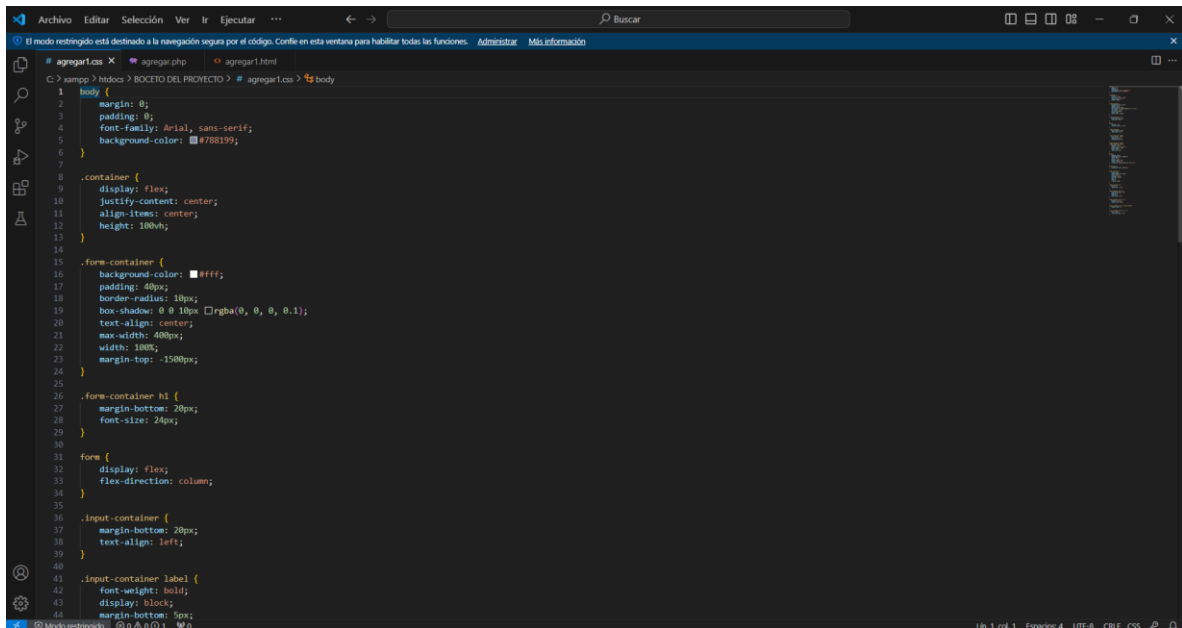
The screenshot shows a web browser window with the address bar displaying 'localhost/BOCETO%20DEL%20PROYECTO/agregar.html'. The page title is 'Gestión de Inventario'. The navigation bar includes links: 'Inicio', 'Inventario', 'Agregar Pedidos', 'Pedidos Realizados', and 'Agregar Nuevos Productos'. The main content area features a white modal box titled 'Agregar Nuevo Producto'. Inside the modal, there are three input fields: 'Nombre del Producto', 'Cantidad en Stock', and 'Imagen del Producto'. The 'Imagen del Producto' field is a file selector showing 'Seleccionar archivo' and 'Sin archivos seleccionados'. At the bottom of the modal is a blue button labeled 'Agregar Producto'.

## HTML AGREGAR NUEVOS PRODUCTOS

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  Buscar
# agregar1.css  # agregar.php  # agregar1.html X
C:\xampp\htdocs\BOCETO DEL PROYECTO > agregar1.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>Agregar Producto</title>
7  <link rel="stylesheet" href="agregar1.css">
8  </head>
9  <body>
10 <div class="container">
11 <div class="menu-container">
12 <h1>Gestión de Inventario</h1>
13
14 <nav>
15 <ul>
16 <li><a href="menu.html">Inicio</a></li>
17 <li><a href="stock.html">Inventario</a></li>
18 <li><a href="pedidos.html">Agregar Pedidos</a></li>
19 <li><a href="pedidosrealizados.html">Pedidos Realizados</a></li>
20 <li><a href="agregar.html">Agregar Nuevos Productos</a></li>
21 </ul>
22 </nav>
23 </div>
24 </div>
25
26 <div class="container">
27 <div class="form-container">
28 <h1>Agregar Nuevo Producto</h1>
29 <div id="mensaje" class="mensaje"></div>
30
31 <form id="agregarForm" enctype="multipart/form-data">
32 <div class="input-container">
33 <label for="nombre">Nombre del Producto</label>
34 <input type="text" id="nombre" name="nombre" required>
35 </div>
36 <div class="input-container">
37 <label for="cantidad">Cantidad en Stock</label>
38 <input type="number" id="cantidad" name="cantidad" min="0" required>
39 </div>
40 <div class="input-container">
41 <label for="imagen">Imagen del Producto</label>
42 <input type="file" id="imagen" name="imagen" accept="image/*" required>
43 </div>
44 <button type="button" onclick="agregarProducto()">Agregar Producto</button>
45 </form>
46 </div>
47 </div>
48 </body>
49 </html>
```

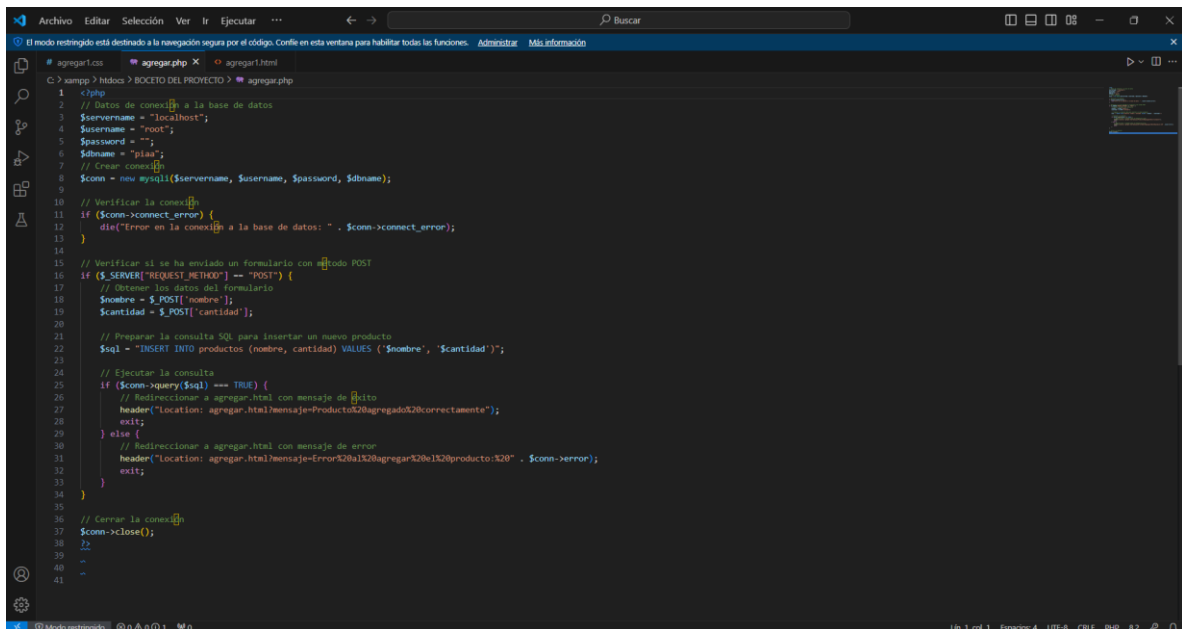


## CSS AGREGAR NUEVOS PRODUCTOS



```
1 body {
2     margin: 0;
3     padding: 0;
4     font-family: Arial, sans-serif;
5     background-color: #f8d199;
6 }
7
8 .container {
9     display: flex;
10    justify-content: center;
11    align-items: center;
12    height: 100vh;
13 }
14
15 .form-container {
16     background-color: #fff;
17     padding: 40px;
18     border-radius: 10px;
19     box-shadow: 0 0 10px #f8d199;
20     text-align: center;
21     max-width: 400px;
22     width: 100%;
23     margin-top: -150px;
24 }
25
26 .form-container h1 {
27     margin-bottom: 20px;
28     font-size: 24px;
29 }
30
31 form {
32     display: flex;
33     flex-direction: column;
34 }
35
36 .input-container {
37     margin-bottom: 20px;
38     text-align: left;
39 }
40
41 .input-container label {
42     font-weight: bold;
43     display: block;
44     margin-bottom: 5px;
```

## PHP AGREGAR NUEVOS PRODUCTOS



```
1 <?php
2 // Datos de conexión a la base de datos
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "pisa";
7 // Crear conexión
8 $conn = new mysqli($servername, $username, $password, $dbname);
9
10 // Verificar la conexión
11 if ($conn->connect_error) {
12     die("Error en la conexión a la base de datos: " . $conn->connect_error);
13 }
14
15 // Verificar si se ha enviado un formulario con método POST
16 if ($_SERVER["REQUEST_METHOD"] == "POST") {
17     // Obtener los datos del formulario
18     $nombre = $_POST['nombre'];
19     $cantidad = $_POST['cantidad'];
20
21     // Preparar la consulta SQL para insertar un nuevo producto
22     $sql = "INSERT INTO productos (nombre, cantidad) VALUES ('$nombre', '$cantidad')";
23
24     // Ejecutar la consulta
25     if ($conn->query($sql) === TRUE) {
26         // Redireccionar a agregar.html con mensaje de éxito
27         header("Location: agregar.html?mensaje=Producto%20agregado%20correctamente");
28         exit;
29     } else {
30         // Redireccionar a agregar.html con mensaje de error
31         header("Location: agregar.html?mensaje=Error%20al%20agregar%20el%20producto%20" . $conn->error);
32         exit;
33     }
34 }
35
36 // Cerrar la conexión
37 $conn->close();
38
39
40
41
```

# BASE DE DATOS

localhost/phpmyadmin/index.php?route=/database/structure&db=pias

phpMyAdmin

Reciente Favoritas

- Nueva
- information\_schema
- logeo
- mysql
- performance\_schema
- phpmyadmin
- pias
  - Nueva
  - login
  - pedidos
  - productos
  - Columnas
    - Nueva
    - cantidad (int)
    - imagen (longblob)
    - nombre (varchar)
- test
- usuarios

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Eventos Disparadores Más

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar	
<input type="checkbox"/> login	Examinar Estructura Buscar Insertar Vaciar Eliminar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-	
<input type="checkbox"/> pedidos	Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	16.0 KB	-	
<input type="checkbox"/> productos	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-	
<b>3 tablas</b>		<b>Número de filas</b>	<b>8</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>48.0 KB</b>	<b>0 B</b>

☐ Seleccionar todo Para los elementos que están marcados

Imprimir Diccionario de datos

Crear nueva tabla

Nombre de la tabla:  Número de columnas:

Consola

Presione Ctrl+Enter para ejecutar la consulta

```
> SELECT * FROM `productos`  
>
```

## Implementación:

### 1. Configuración del Entorno de Desarrollo:

- Instalar y configurar Xampp en la computadora local.
- Configurar un proyecto en Visual Studio Code para el desarrollo de la página web.

### 2. Desarrollo del Backend:

Utilizar PHP para crear scripts que gestionen la lógica de negocio, como la autenticación de usuarios, la gestión de inventario y la administración de pedidos. Utilizar SQL para diseñar y gestionar la base de datos que almacenará la información de la tienda y los usuarios.

### 3. Desarrollo del Frontend:

- Utilizar HTML para estructurar el contenido de la página web.
- Utilizar CSS para aplicar estilos y diseño a la página web.
- Utilizar JavaScript para agregar interactividad y funcionalidades dinámicas a la página web, como la actualización del carrito de compras y la visualización de productos destacados.

### 4. Pruebas y Depuración:

- Realizar pruebas exhaustivas del sistema para identificar y corregir errores.
- Verificar la funcionalidad del sistema en diferentes navegadores y dispositivos.

### 5. Implementación en Producción:

Desplegar la página web en un servidor en vivo para que los usuarios puedan acceder a ella desde cualquier lugar.

## **Mantenimiento y Soporte:**

### **1. Actualizaciones:**

Realizar actualizaciones periódicas del sistema para agregar nuevas funcionalidades, corregir errores y mejorar el rendimiento.

### **1. Soporte Técnico:**

Proporcionar soporte técnico a los usuarios para resolver problemas y responder preguntas relacionadas con el funcionamiento de la página web.

### **2. Monitoreo y Seguridad:**

Monitorear el rendimiento y la seguridad del sistema para detectar y prevenir posibles vulnerabilidades y ataques cibernéticos.

Este manual técnico proporciona una guía completa para el desarrollo y mantenimiento de la página web de la tienda de llaveros de Funko. Para cualquier consulta adicional o asistencia técnica, no dudes en ponerte en contacto con el equipo de desarrollo.