
EVALUACIÓN 4 TALLER DE APLICACIONES

Documentación de la Aplicación - Código Fuente y su Ejecutable

NOMBRE: Diego Aguilera, Mitchel Bustamante, Alexis Meneses.

CARRERA: Ingeniería en informática.

ASIGNATURA: Taller de desarrollo de apps.

Profesor: Mauricio Alberto Daza Ródenas.

FECHA: 22-07-2024

Introducción

Este informe presenta el desarrollo de una aplicación para la gestión de un taller de confección de ropa, siguiendo un modelo n-capas para mejorar la organización y mantenimiento del software. La aplicación se desarrolló en el marco del Taller de Desarrollo de Aplicaciones, parte de la carrera de Ingeniería en Informática.

El objetivo principal fue crear una aplicación eficiente para gestionar usuarios, productos y categorías. El trabajo se dividió en cuatro actividades principales:

1. **Diseño de interfaces de los mantenedores:** Se desarrollaron módulos para la autenticación de usuarios, gestión de datos y servicios adicionales, utilizando el modelo n-capas para estructurar estos módulos.
2. **Conexión a la base de datos y operaciones CRUD:** Se estableció la conexión con la base de datos y se realizaron pruebas exhaustivas para validar las operaciones CRUD y la comunicación efectiva con la base de datos.
3. **Diseño de interfaces transaccionales:** Se construyeron interfaces para la gestión de inventario y compras, y se llevaron a cabo pruebas integrales para asegurar su correcto funcionamiento.
4. **Documentación de la aplicación:** Se generó documentación detallada de las clases del proyecto, un manual de instalación y un manual de usuario, garantizando que cualquier persona pueda entender, instalar y utilizar la aplicación.

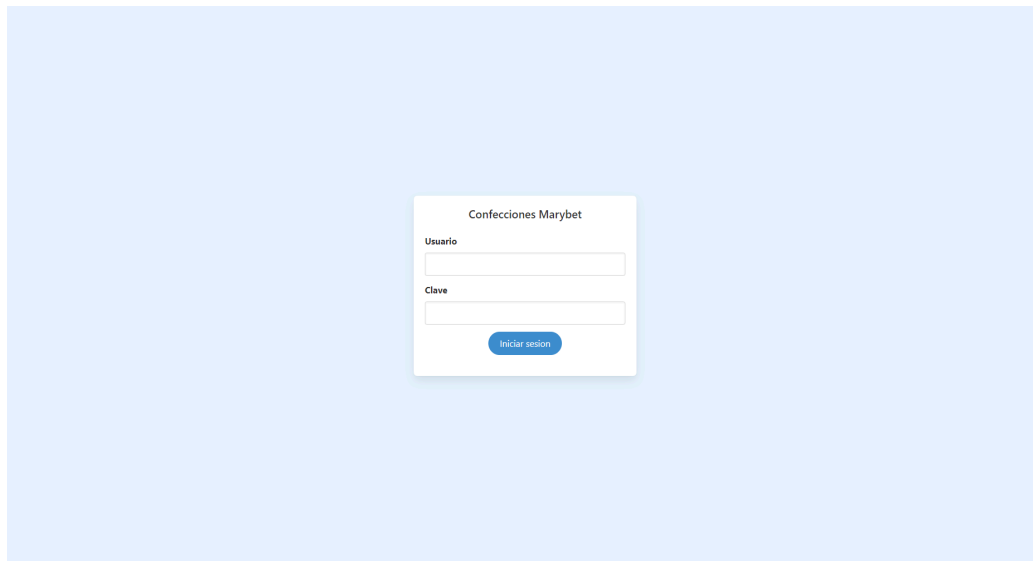
Cada una de estas actividades fue fundamental para el éxito del proyecto. A continuación, se detalla el proceso seguido en cada actividad, describiendo los pasos específicos y los resultados obtenidos.

Actividades

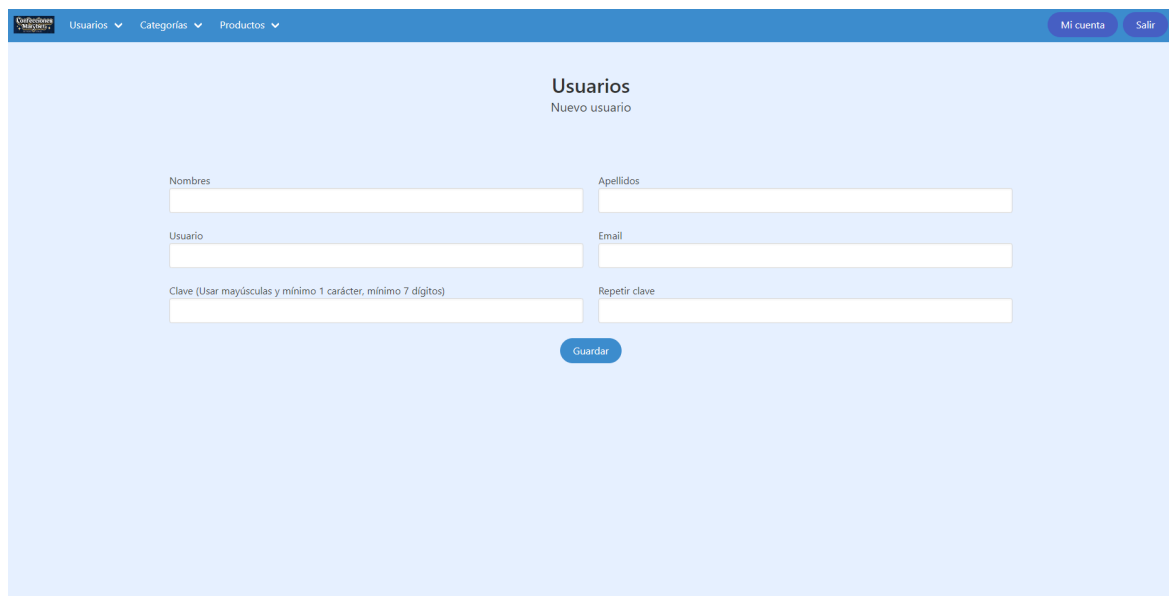
Actividad 1: Diseñar interfaces de los mantenedores de la aplicación.

Paso 1: Desarrollar los módulos de la aplicación.

Módulo de Usuario/Autenticación: Maneja registro, inicio de sesión y gestión de usuarios.



A login form titled "Confecciones Marybet" centered on a light blue background. The form contains two input fields: "Usuario" and "Clave". Below the "Clave" field is a blue button labeled "Iniciar sesión".



A user registration form titled "Usuarios" with the subtitle "Nuevo usuario". The form is set against a light blue background. It features a header bar with the "Confecciones Marybet" logo, navigation links for "Usuarios", "Categorías", and "Productos", and user links for "Mi cuenta" and "Salir". The registration fields are arranged in two columns: "Nombres", "Apellidos", "Usuario", "Email", "Clave (Usar mayúsculas y mínimo 1 carácter, mínimo 7 dígitos)", and "Repetir clave". A blue "Guardar" button is positioned at the bottom center of the form.

Usuarios

Categorías

Productos

Mi cuenta

Salir

Mi cuenta

Actualizar datos de cuenta

Regresar atrás

Nombres

Administrador

Apellidos

Administrador

Usuario

Administrador

Email

Si desea actualizar la clave de este usuario por favor llene los 2 campos. Si NO desea actualizar la clave deje los campos vacíos.

Clave

Repetir clave

Para poder actualizar los datos de este usuario por favor ingrese su USUARIO y CLAVE con la que ha iniciado sesión

Usuario

Clave

Actualizar

Módulo de gestión de datos: Ingresa, modifica y elimina datos (CRUD) y maneja la interacción con la base de datos.

(EJEMPLO INGRESO DE NUEVO PRODUCTO)

1. Se visualiza el ingreso de los datos.

Collection

Management

Usuarios

Categorías

Productos

Mi cuenta

Salir

Productos

Nuevo producto

Código de tela

0123

Nombre

Softshell

Precio

10000

Stock

20

Categoría

Deportivo

Foto o imagen del producto

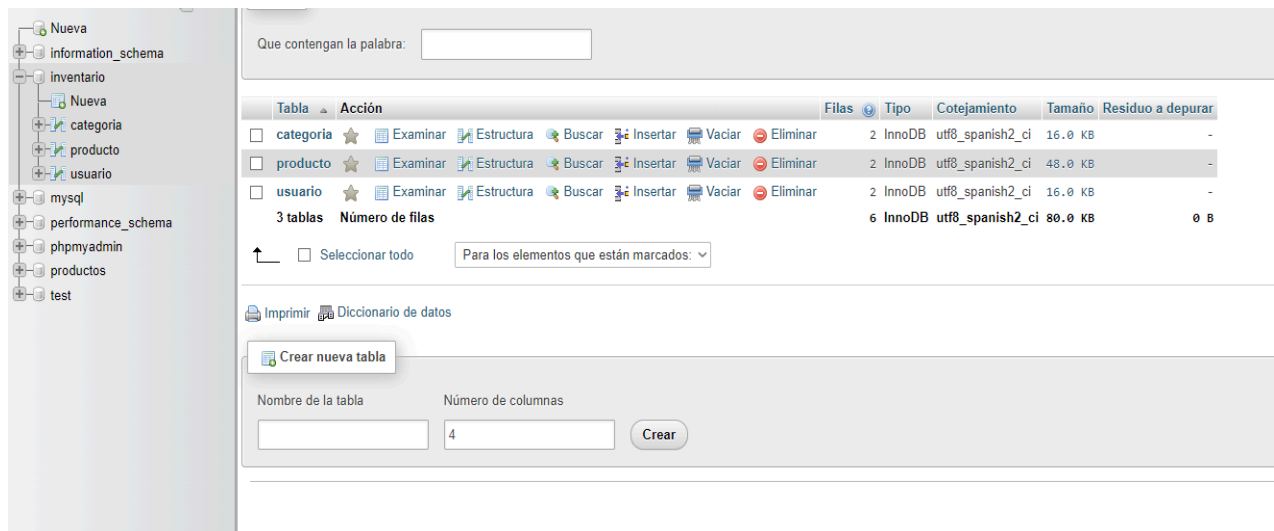
Imagen

JPG, JPEG, PNG, (MAX 3MB)

Guardar

(INTERACCIÓN CON BASE DE DATOS)

2. Se ingresa a una de las tablas.



Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> categoria	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> producto	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish2_ci	48.0 KB	-
<input type="checkbox"/> usuario	Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish2_ci	16.0 KB	-
3 tablas	Número de filas	6	InnoDB	utf8_spanish2_ci	80.0 KB	0 B

☐ Seleccionar todo Para los elementos que están marcados:

[Imprimir](#) [Diccionario de datos](#)

[Crear nueva tabla](#)

Nombre de la tabla Número de columnas [Crear](#)

3. Se visualiza la modificación en la base de datos.



`SELECT * FROM "producto"`

☐ Perfilando [\[Editar en línea \]](#) [\[Editar \]](#) [\[Explicar SQL \]](#) [\[Crear código PHP \]](#) [\[Actualizar \]](#)

☐ Mostrar todo | Número de filas: | Filtrar filas: | Ordenar según la clave:

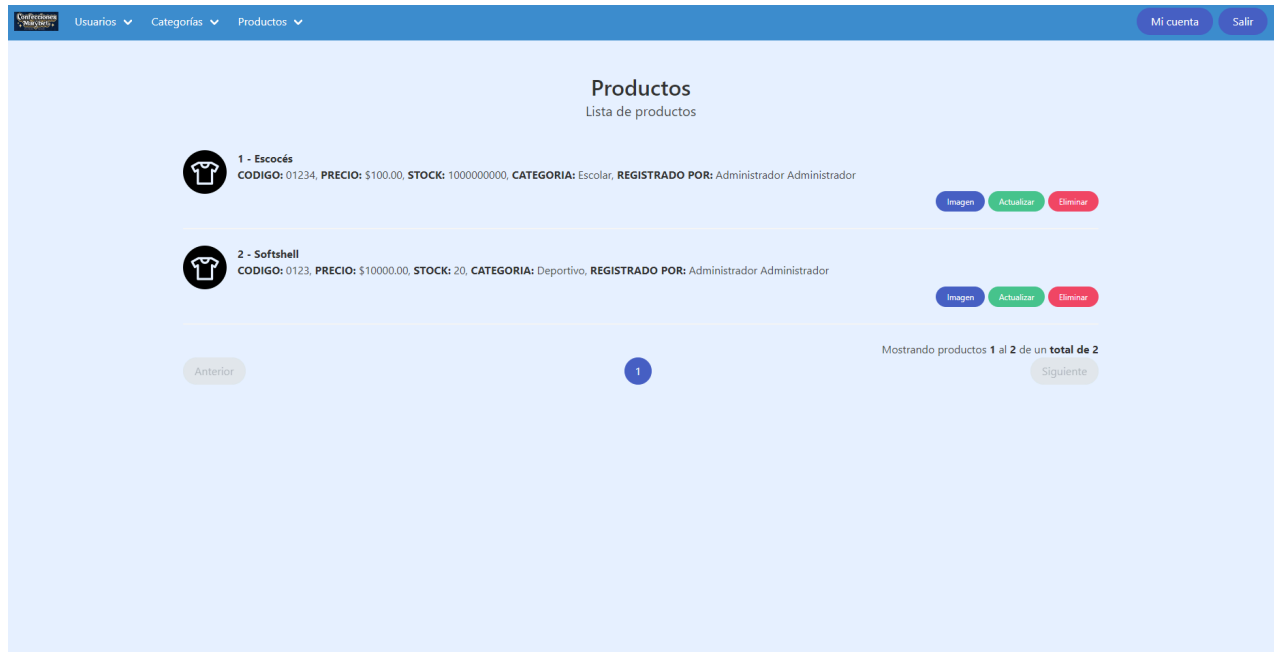
Opciones extra

	producto_id	producto_codigo	producto_nombre	producto_precio	producto_stock	producto_foto	categoria_id	usuario_id
<input type="checkbox"/> Editar Copiar Borrar	4	0123	Softshell	10000.00	20		9	3
<input type="checkbox"/> Editar Copiar Borrar	5	01234	Escocés	100.00	1000000000	Escocés_28.jpg	10	3

☐ Seleccionar todo Para los elementos que están marcados: [Editar](#) [Copiar](#) [Borrar](#) [Exportar](#)

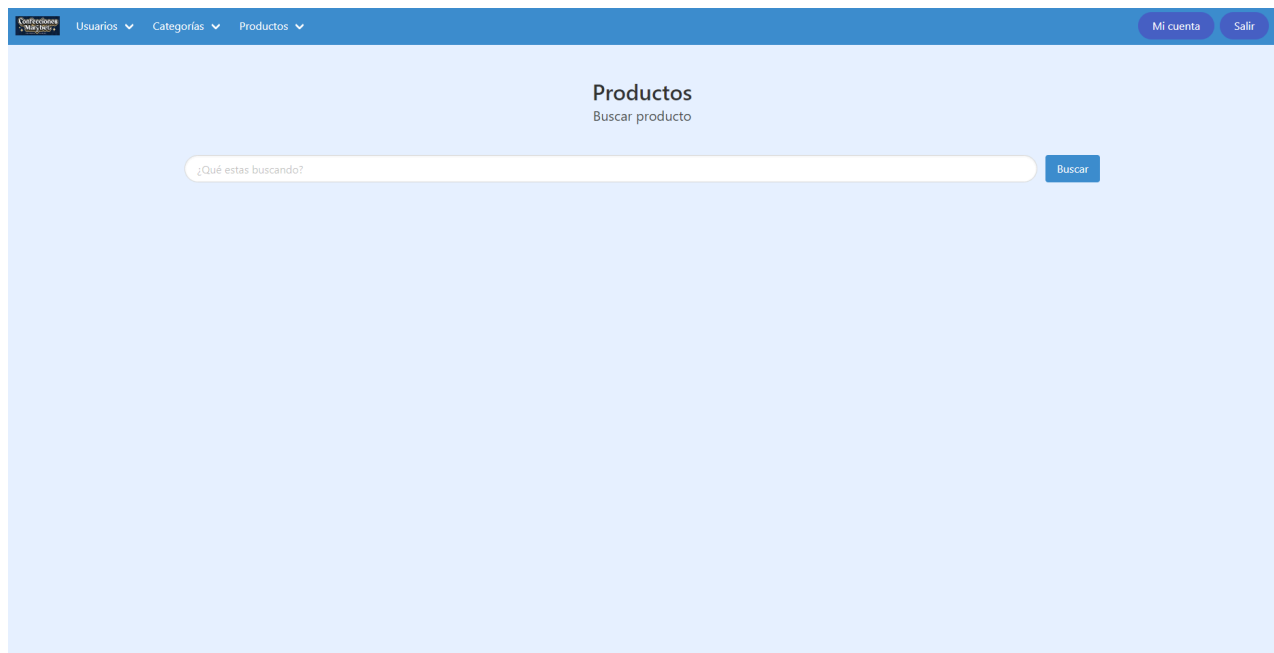
☐ Mostrar todo | Número de filas: | Filtrar filas: | Ordenar según la clave:

Módulo de servicios: Proporciona funcionalidades adicionales como la visualización a tiempo real de los productos ingresados, los precios, la categoría y la ubicación de estos además de proporcionarnos el usuario que hizo el registro.



Módulo de búsqueda: Proporciona la funcionalidad de buscar entre los productos del taller filtrado tanto por categoría como por nombre. Esta funcionalidad está también integrada en las categorías.

Ejemplo búsqueda de producto:



Confesiones Maybelline

Usuarios ▼ Categorías ▼ Productos ▼


Mi cuenta Salir

Productos

Buscar producto

Estas buscando "soft"

Eliminar búsqueda



1 - Softshell
CODIGO: 0123. PRECIO: \$10000.00. STOCK: 20. CATEGORIA: Deportivo. REGISTRADO POR: Administrador Administrador

Imagen

Actualizar

Eliminar

Anterior

1

Mostrando productos 1 al 1 de un total de 1

Siguiente

Ejemplo búsqueda por categorías:

Confesiones Maybelline

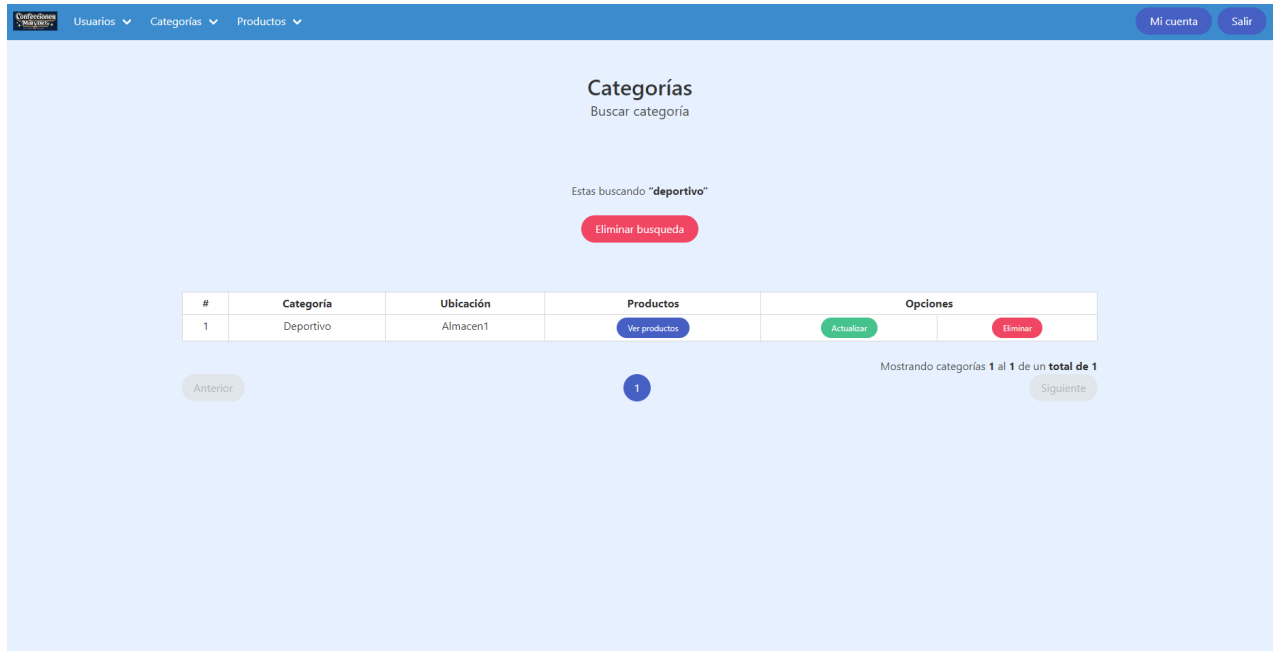
Usuarios ▼ Categorías ▼ Productos ▼

Mi cuenta Salir

Categorías

Buscar categoría

Buscar



Paso 2: Usar modelo de capas para la creación de los mantenedores de la aplicación

- **Capa de Presentación (UI):**
 - **Responsabilidad:** Interfaz de usuario y manejo de la interacción con el usuario.
 - **Tecnologías:** HTML, CSS, JavaScript, el framework como bulma.
 - **Ejemplo:** Diseñar formularios y vistas para la gestión de usuarios, productos y categorías.
- **Capa de Negocio (BL o BLL):**
 - **Responsabilidad:** Lógica de negocio y reglas de la aplicación.
 - **Tecnologías:** Lenguajes de programación como JavaScript, HTML, CSS y PHP.
 - **Ejemplo:** Implementar validaciones, cálculos y procesamiento de datos.
- **Capa de Acceso a Datos (DAL):**
 - **Responsabilidad:** Comunicación con la base de datos.
 - **Tecnologías:** SQL.
 - **Ejemplo:** Crear funciones para insertar, actualizar, eliminar y recuperar datos.
- **Capa de Datos (Database):**
 - **Responsabilidad:** Almacenamiento y gestión de los datos.
 - **Tecnologías:** Sistemas de gestión de bases de datos como MySQL.
 - **Ejemplo:** Diseño de tablas y relaciones entre ellas.

Paso 3: Diseñar interfaces con los debidos componentes que permitan el CRUD para las tablas de parámetros

Tablas que necesitan CRUD:

1. Usuarios.
2. Categorías.
3. Productos.

Lenguajes usados:

Frontend: HTML, CSS, JavaScript, PHP.

Backend: PHP.

Base de Datos: SQL.

Los siguientes pantallazos son una pequeña fracción del código back end realizado para interactuar con la base de datos.

INSERT

Usuario

```
126  /*== Guardando datos ==*/
127  $guardar_usuario=conexion();
128  $guardar_usuario=$guardar_usuario->prepare("INSERT INTO usuario(usuario_nombre,usuario_apellido,usuario_usuario,usuario_clave,usuario_email) VALUES(:nombre,:apellido,:usuario,:clave,:email)");
129
130
131  $marcadores=[
132      ":nombre"=>$nombre,
133      ":apellido"=>$apellido,
134      ":usuario"=>$usuario,
135      ":clave"=>$clave,
136      ":email"=>$email
137  ];
```

Producto

```
193  /*== Guardando datos ==*/
194  $guardar_producto=conexion();
195  $guardar_producto=$guardar_producto->prepare("INSERT INTO producto(producto_codigo,producto_nombre,producto_precio,producto_stock,producto_foto,categoria_id,usuario_id) VALUES(:codigo,:nombre,:precio,:stock,:foto,:categoria,:usuario)");
196
197  $marcadores=[
198      ":codigo"=>$codigo,
199      ":nombre"=>$nombre,
200      ":precio"=>$precio,
201      ":stock"=>$stock,
202      ":foto"=>$foto,
203      ":categoria"=>$categoria,
204      ":usuario"=>$SESSION['id']
205  ];
```

Categoría

```
60  /*== Guardando datos ==*/
61  $guardar_categoria=conexion();
62  $guardar_categoria=$guardar_categoria->prepare("INSERT INTO categoria(categoria_nombre,categoria_ubicacion) VALUES(:nombre,:ubicacion)");
63
64  $marcadores=[
65      ":nombre"=>$nombre,
66      ":ubicacion"=>$ubicacion
67  ];
68
```

Read

Usuario

```
5 if(isset($busqueda) && $busqueda!=""){
6
7     $consulta_datos="SELECT * FROM usuario WHERE ((usuario_id='".$$_SESSION['id']."' ) AND (usuario_nombre LIKE '%$busqueda%' OR usuario_apellido LIKE '%$busqueda%' OR usuario_usuario LIKE '%$busqueda%' OR usuario_email LIKE '%$busqueda%')) ORDER BY usuario_nombre ASC LIMIT $inicio,$registros";
8
9     $consulta_total="SELECT COUNT(usuario_id) FROM usuario WHERE ((usuario_id='".$$_SESSION['id']."' ) AND (usuario_nombre LIKE '%$busqueda%' OR usuario_apellido LIKE '%$busqueda%' OR usuario_usuario LIKE '%$busqueda%' OR usuario_email LIKE '%$busqueda%'))";
10
11 }else{
12
13     $consulta_datos="SELECT * FROM usuario WHERE usuario_id='".$$_SESSION['id']."' ORDER BY usuario_nombre ASC LIMIT $inicio,$registros";
14
15     $consulta_total="SELECT COUNT(usuario_id) FROM usuario WHERE usuario_id='".$$_SESSION['id']."'";
16
17 }
```

Producto

```
7 if(isset($busqueda) && $busqueda!=""){
8
9     $consulta_datos="SELECT *campos FROM producto INNER JOIN categoria ON producto_categoria_id=categoria_categoria_id INNER JOIN usuario ON producto_usuario_id=usuario_usuario_id WHERE producto_producto_codigo LIKE '%$busqueda%' OR producto_producto_nombre LIKE '%$busqueda%' ORDER BY producto_producto_nombre ASC LIMIT $inicio,$registros";
10
11     $consulta_total="SELECT COUNT(producto_id) FROM producto WHERE producto_codigo LIKE '%$busqueda%' OR producto_nombre LIKE '%$busqueda%'";
12
13 }elseif($categoria_id!=0){
14
15     $consulta_datos="SELECT *campos FROM producto INNER JOIN categoria ON producto_categoria_id=categoria_categoria_id INNER JOIN usuario ON producto_usuario_id=usuario_usuario_id WHERE producto_categoria_id= $categoria_id ORDER BY producto_producto_nombre ASC LIMIT $inicio,$registros";
16
17     $consulta_total="SELECT COUNT(producto_id) FROM producto WHERE categoria_id= $categoria_id";
18
19 }else{
20
21     $consulta_datos="SELECT *campos FROM producto INNER JOIN categoria ON producto_categoria_id=categoria_categoria_id INNER JOIN usuario ON producto_usuario_id=usuario_usuario_id ORDER BY producto_producto_nombre ASC LIMIT $inicio,$registros";
22
23     $consulta_total="SELECT COUNT(producto_id) FROM producto";
24
25 }
```

Categoría

```
5 if(isset($busqueda) && $busqueda!=""){
6
7     $consulta_datos="SELECT * FROM categoria WHERE categoria_nombre LIKE '%$busqueda%' OR categoria_ubicacion LIKE '%$busqueda%' ORDER BY categoria_nombre ASC LIMIT $inicio,$registros";
8
9     $consulta_total="SELECT COUNT(categoria_id) FROM categoria WHERE categoria_nombre LIKE '%$busqueda%' OR categoria_ubicacion LIKE '%$busqueda%'";
10
11 }else{
12
13     $consulta_datos="SELECT * FROM categoria ORDER BY categoria_nombre ASC LIMIT $inicio,$registros";
14
15     $consulta_total="SELECT COUNT(categoria_id) FROM categoria";
16
17 }
```

Update

Usuario

```
221  /*== Actualizar datos ==*/
222  $actualizar_usuario=conexion();
223  $actualizar_usuario=$actualizar_usuario->prepare("UPDATE usuario SET usuario_nombre=:nombre,usuario_apellido=:apellido,usuario_usuario=:usuario,usuario_clave=:clave,usuario_email=:email WHERE usuario_id=:id");
224
225  $marcadores=[
226      ":nombre"=>$nombre,
227      ":apellido"=>$apellido,
228      ":usuario"=>$usuario,
229      ":clave"=>$clave,
230      ":email"=>$email,
231      ":id"=>$id
232  ];
233
```

Producto

```
140  /*== Actualizando datos ==*/
141  $actualizar_producto=conexion();
142  $actualizar_producto=$actualizar_producto->prepare("UPDATE producto SET producto_codigo=:codigo,producto_nombre=:nombre,producto_precio=:precio,producto_stock=:stock,categoria_id=:categoria WHERE producto_id=:id");
143
144  $marcadores=[
145      ":codigo"=>$codigo,
146      ":nombre"=>$nombre,
147      ":precio"=>$precio,
148      ":stock"=>$stock,
149      ":categoria"=>$categoria,
150      ":id"=>$id
151  ];
152
```

Categoría

```
82
83  /*== Actualizar datos ==*/
84  $actualizar_categoria=conexion();
85  $actualizar_categoria=$actualizar_categoria->prepare("UPDATE categoria SET categoria_nombre=:nombre,categoria_ubicacion=:ubicacion WHERE categoria_id=:id");
86
87  $marcadores=[
88      ":nombre"=>$nombre,
89      ":ubicacion"=>$ubicacion,
90      ":id"=>$id
91  ];
92
```

Delete

Usuario

```
10  if($check_usuario->rowCount()==1){
11  ...
12      $check_productos=conexion();
13      $check_productos=$check_productos->query("SELECT usuario_id FROM producto WHERE usuario_id='$user_id_del' LIMIT 1");
14
15      if($check_productos->rowCount()<=0){
16
17          $eliminar_usuario=conexion();
18          $eliminar_usuario=$eliminar_usuario->prepare("DELETE FROM usuario WHERE usuario_id=:id");
19
20          $eliminar_usuario->execute(["id"=>$user_id_del]);
21
22          if($eliminar_usuario->rowCount()==1){
23              echo '
24                  <div class="notification is-info is-light">
25                      <strong>¡USUARIO ELIMINADO!</strong><br>
26                      Los datos del usuario se eliminaron con exito
27                  </div>
28              ';
29          }else{
30              echo '
31                  <div class="notification is-danger is-light">
32                      <strong>¡Ocurrio un error inesperado!</strong><br>
33                      No se pudo eliminar el usuario, por favor intente nuevamente
34                  </div>
35              ';
36          }
37          $eliminar_usuario=null;
38      }else{
39          echo '
40              <div class="notification is-danger is-light">
41                  <strong>¡Ocurrio un error inesperado!</strong><br>
42                  No podemos eliminar el usuario ya que tiene productos registrados por el
43              </div>
44          ';
45      }
46      $check_productos=null;
47  }else{
48      echo '
49          <div class="notification is-danger is-light">
50              <strong>¡Ocurrio un error inesperado!</strong><br>
51              El USUARIO que intenta eliminar no existe
52          </div>
53      ';
54  }
```

Producto

```
9      if($check_producto->rowCount()==1){
10          ...
11          $datos=$check_producto->fetch();
12
13          $eliminar_producto=conexion();
14          $eliminar_producto=$eliminar_producto->prepare("DELETE FROM producto WHERE producto_id=:id");
15
16          $eliminar_producto->execute([":id"=>$product_id_del]);
17
18          if($eliminar_producto->rowCount()==1){
19              ...
20              if(is_file("./img/producto/".$datos['producto_foto'])){
21                  chmod("./img/producto/".$datos['producto_foto'], 0777);
22                  unlink("./img/producto/".$datos['producto_foto']);
23              }
24
25              echo '
26                  <div class="notification is-info is-light">
27                      <strong>PRODUCTO ELIMINADO!</strong><br>
28                      Los datos del producto se eliminaron con exito
29                  </div>
30              ';
31          }else{
32              echo '
33                  <div class="notification is-danger is-light">
34                      <strong>Ocurrio un error inesperado!</strong><br>
35                      No se pudo eliminar el producto, por favor intente nuevamente
36                  </div>
37              ';
38          }
39          $eliminar_producto=null;
40      }else{
41          echo '
42              <div class="notification is-danger is-light">
43                  <strong>Ocurrio un error inesperado!</strong><br>
44                  El PRODUCTO que intenta eliminar no existe
45              </div>
46          ';
47      }
```

Categoría

```
9      if($check_categoria->rowCount()==1){
10
11          $check_productos=conexion();
12          $check_productos=$check_productos->query("SELECT categoria_id FROM producto WHERE categoria_id='$category_id_del' LIMIT 1");
13
14          if($check_productos->rowCount()<=0){
15
16              $eliminar_categoria=conexion();
17              $eliminar_categoria=$eliminar_categoria->prepare("DELETE FROM categoria WHERE categoria_id=:id");
18
19              $eliminar_categoria->execute([":id"=>$category_id_del]);
20
21              if($eliminar_categoria->rowCount()==1){
22                  echo '
23                      <div class="notification is-info is-light">
24                          <strong>CATEGORIA ELIMINADA!</strong><br>
25                          Los datos de la categoría se eliminaron con éxito
26                      </div>
27                  ';
28              }else{
29                  echo '
30                      <div class="notification is-danger is-light">
31                          <strong>Ocurrió un error inesperado!</strong><br>
32                          No se pudo eliminar la categoría, por favor intente nuevamente
33                      </div>
34                  ';
35              }
36              $eliminar_categoria=null;
37          }else{
38              echo '
39                  <div class="notification is-danger is-light">
40                      <strong>Ocurrió un error inesperado!</strong><br>
41                      No podemos eliminar la categoría ya que tiene productos asociados
42                  </div>
43              ';
44          }
45          $check_productos=null;
46      }else{
47          echo '
48              <div class="notification is-danger is-light">
49                  <strong>Ocurrió un error inesperado!</strong><br>
50                  La CATEGORIA que intenta eliminar no existe
51              </div>
52          ';
53      }
```

Actividad 2: Efectuar la conexión a la base de datos y efectuar las operaciones CRUD.

Paso 1: Probar las conexiones a la base de datos.

Se realizaron pruebas de conexión para asegurar la comunicación efectiva entre la aplicación y la base de datos.

```
php > main.php
1  <?php
2
3      # Conexion a la base de datos #
4      function conexion(){
5          $pdo = new PDO('mysql:host=localhost;dbname=inventario', 'root', '');
6          return $pdo;
7      }
8
```

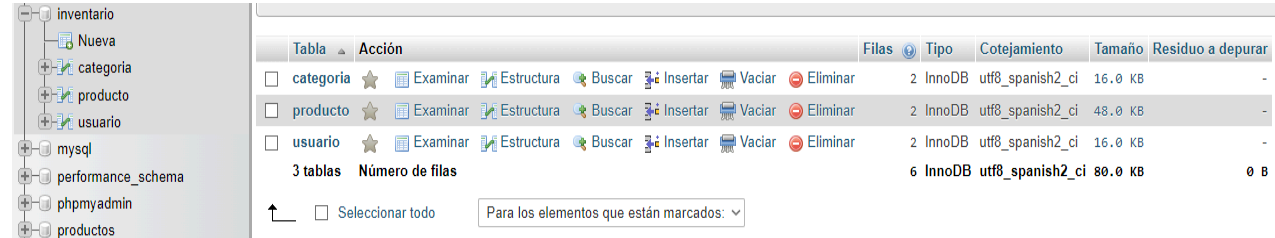
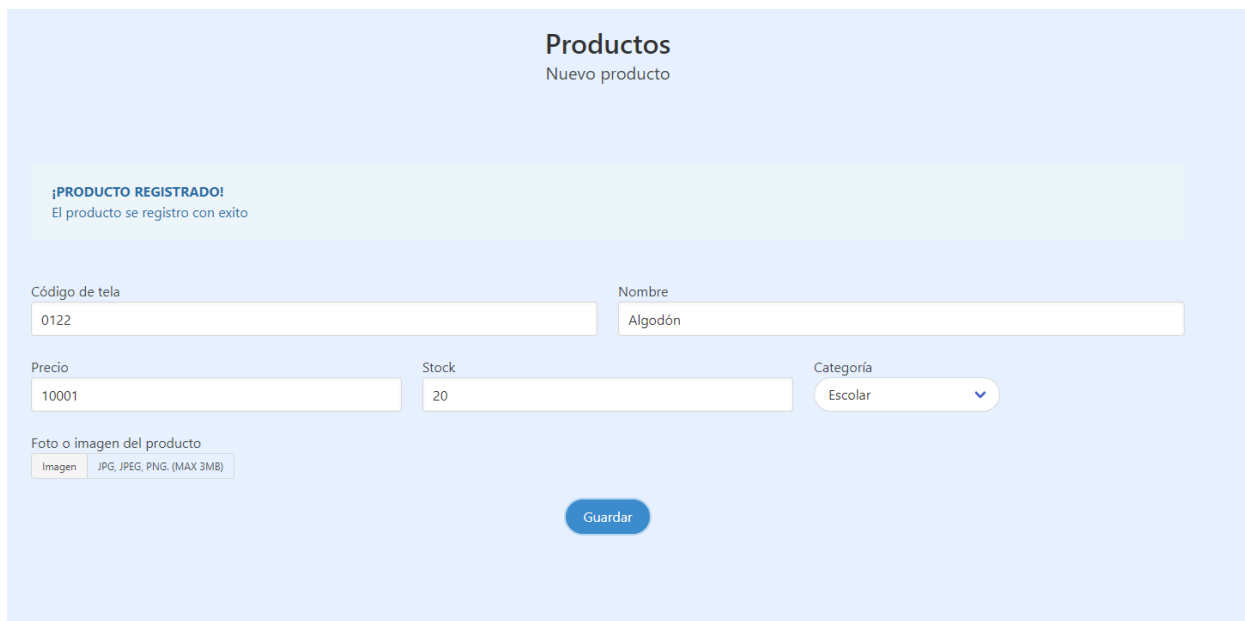


Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> categoria	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish2_ci	16.0 KB	-
<input type="checkbox"/> producto	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish2_ci	48.0 KB	-
<input type="checkbox"/> usuario	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_spanish2_ci	16.0 KB	-
3 tablas Número de filas		6	InnoDB	utf8_spanish2_ci	80.0 KB	0 B

Paso 2: Validar los datos de entrada a través de las interfaces.

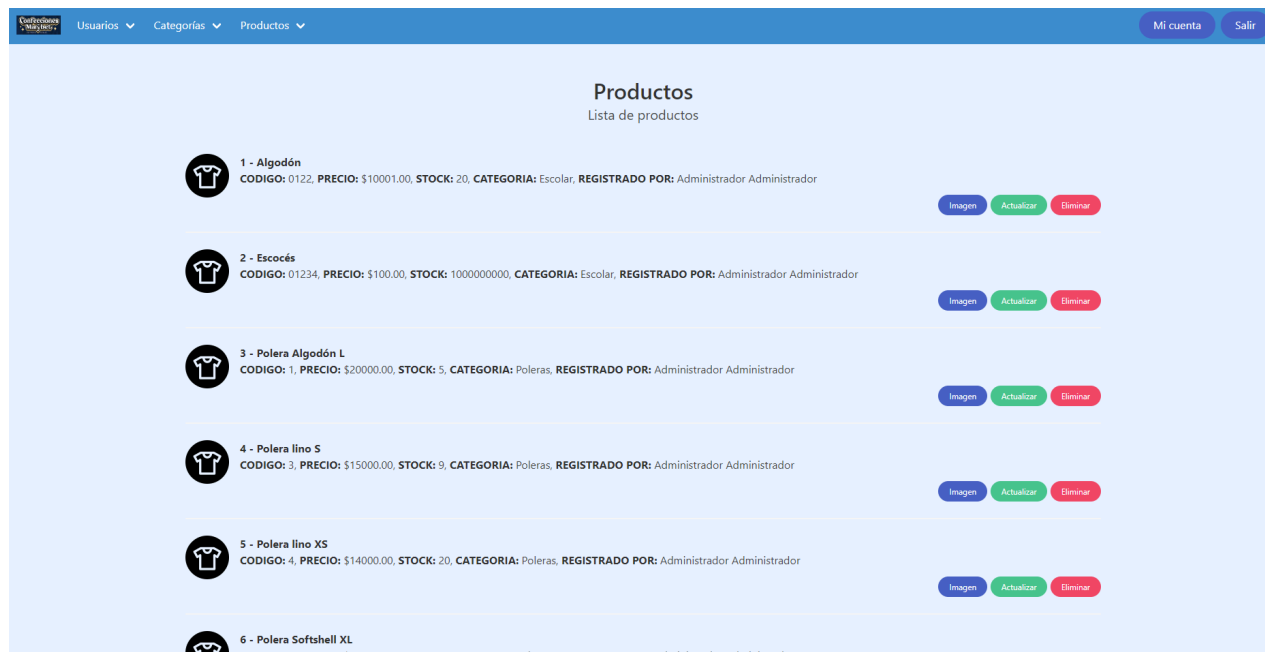
Se implementaron validaciones para garantizar que los datos ingresados a través de las interfaces sean correctos y completos.



Actividad 3: Diseñar las interfaces de procesos transaccionales.

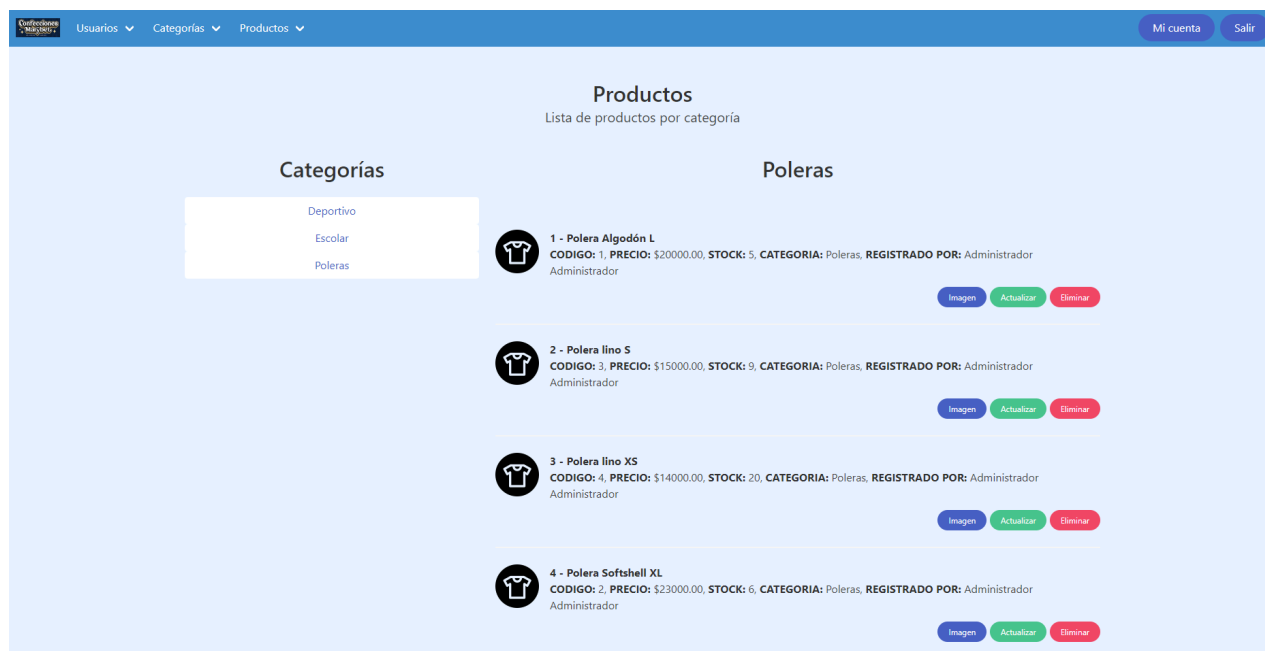
Paso 1: Construir las interfaces que permitan la transacción de los procesos de negocio.

Se desarrollaron interfaces que permiten realizar transacciones relacionadas con la gestión de inventario y compras.



Paso 2: Efectuar pruebas integrales con datos ficticios.

Se llevaron a cabo pruebas integrales utilizando datos ficticios para asegurar el correcto funcionamiento de las transacciones.



Paso 3: Validar el almacenamiento en la base de datos mediante reportes.

Se generaron reportes para validar que los datos se almacenan correctamente en la base de datos.



The screenshot shows a database management interface. On the left is a tree view of the database structure. The main area displays a table view of the 'producto' table. The table has columns: producto_id, producto_codigo, producto_nombre, producto_precio, producto_stock, producto_foto, categoria_id, and usuario_id. There are four rows of data. Each row has a checkbox and three action buttons: Editar, Copiar, and Borrar. Below the table, there are options to select all, edit, copy, delete, and export the selected data.

	producto_id	producto_codigo	producto_nombre	producto_precio	producto_stock	producto_foto	categoria_id	usuario_id
<input type="checkbox"/>	7	1	Polera Algodón L	20000.00	5		11	3
<input type="checkbox"/>	8	2	Polera Softshell XL	23000.00	6		11	3
<input type="checkbox"/>	9	3	Polera lino S	15000.00	9		11	3
<input type="checkbox"/>	10	4	Polera lino XS	14000.00	20		11	3

Actividad 4: Documentar la aplicación.

Clases del proyecto

1. Clase Usuario

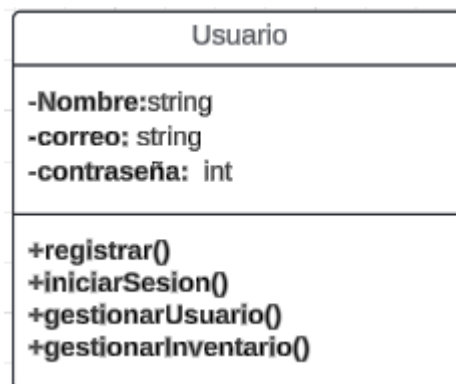
Descripción: Maneja el registro, inicio de sesión y gestión de usuarios.

Atributos:

1. **id:** Identificador único del usuario.
2. **Nombre:** Nombre del usuario.
3. **correo:** Correo electrónico del usuario.
4. **contraseña:** Contraseña del usuario (almacenada de forma segura).

Métodos:

1. **registrar():** Registra un nuevo usuario en el sistema.
2. **iniciarSesion():** Permite a un usuario iniciar sesión.
3. **gestionarUsuario():** Permite la gestión de usuarios (actualización y eliminación).
4. **gestionarInventario():** Permite la gestión del inventario (ingreso, actualización, eliminación).



2. Clase Producto

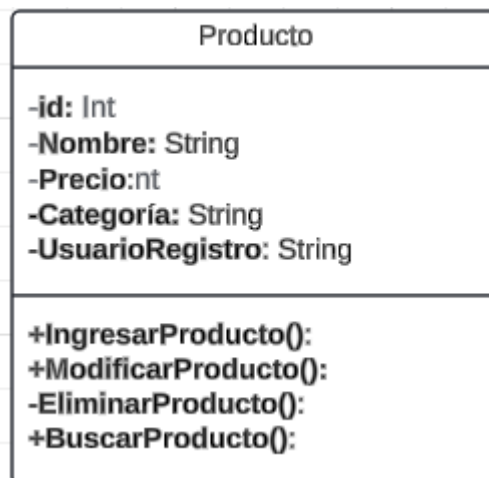
Descripción: Gestiona los productos del taller de confección.

Atributos:

1. **id:** Identificador único del producto.
2. **Nombre:** Nombre del producto.
3. **Precio:** Precio del producto.
4. **Categoría:** Categoría a la que pertenece el producto.
5. **UsuarioRegistro:** Usuario que registró el producto.

Métodos:

1. **IngresarProducto():** Ingresa un nuevo producto en el sistema.
2. **ModificarProducto():** Modifica la información de un producto existente.
3. **EliminarProducto():** Elimina un producto del sistema.
4. **BuscarProducto():** Busca productos en el sistema por nombre o categoría.



3. Clase Categoría

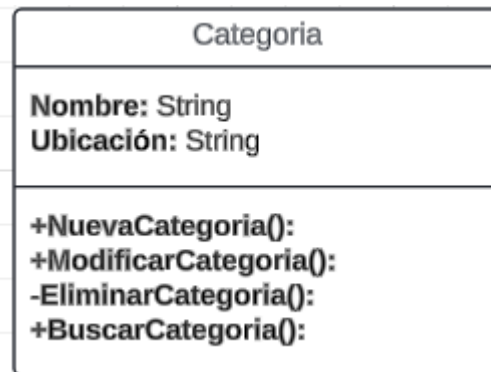
Descripción: Gestiona las categorías de los productos.

Atributos:

1. **Nombre:** Nombre de la categoría.
2. **Ubicación:** Ubicación de la categoría en el inventario.

Métodos:

1. **NuevaCategoría():** Crea una nueva categoría en el sistema.
2. **ModificarCategoría():** Modifica la información de una categoría existente.
3. **EliminarCategoría():** Elimina una categoría del sistema.
4. **BuscarCategoría():** Busca categorías en el sistema.



4. Clase servicio

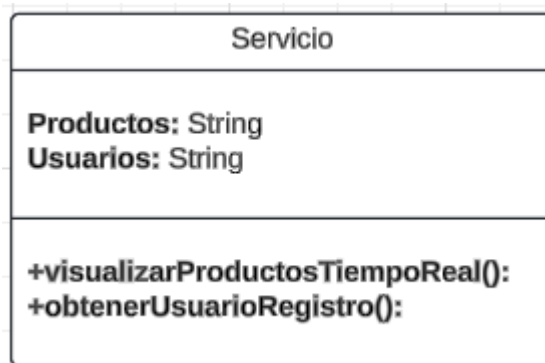
Descripción: Proporciona funcionalidades adicionales como la visualización en tiempo real de los productos, precios, categorías y usuarios.

Atributos:

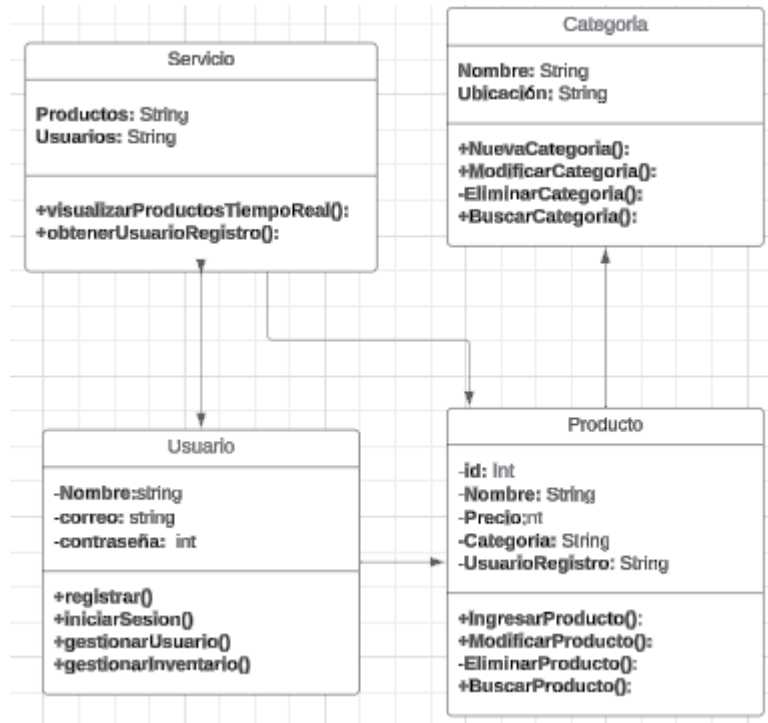
1. **Productos:** Lista de productos registrados.
2. **Usuarios:** Lista de usuarios registrados.

Métodos:

1. **visualizarProductosTiempoReal():** Muestra en tiempo real los productos ingresados, sus precios, categorías y ubicaciones.
2. **obtenerUsuarioRegistro():** Devuelve el usuario que registró un producto específico.



Relación entre las clases del proyecto:



Paso 2: Generar manual de instalación si corresponde.

Se creó un manual de instalación detallado para facilitar la implementación de la aplicación.

- **Requisito de uso (IMPORTANTE):**
 - XAMPP
- **Pasos de Instalación:**
 - Copiar/Mover la Carpeta
 - Copie o mueva la carpeta INVENTARIO previamente extraída de INVENTARIO.ZIP a "C:\xampp\htdocs".
- **Creación de Base de Datos**
 - Abra XAMPP, darle click a "start" a Apache y MySQL.
 - Acceda a phpMyAdmin desde <http://localhost/phpmyadmin>.
 - Cree una base de datos con el nombre "inventario".
 - Seleccione la base de datos creada e importe el archivo de la base de datos que se encuentra en la carpeta DB dentro de "INVENTARIO".
- **Iniciar gestor de inventario**
 - Ir a su navegador preferido (Preferencia chrome), y en la barra de búsqueda escribir "localhost/INVENTARIO/"
- **Iniciar sesión**
 - Inicio de sesión predeterminado:
 - Usuario: Administrador
 - Contraseña: Administrador

Manual de Usuario

Esta aplicación permite agregar, modificar y eliminar usuarios, productos y categorías.

Uso de la Aplicación : Inicio de Sesión

- Ingresa tus credenciales y haz clic en "Iniciar Sesión".

Gestión de cuenta:

- Modificar datos de cuenta:
 - Seleccionar "Mi cuenta" en la esquina superior derecha → Rellenar datos que se requieran modificar, seleccionar el botón "Actualizar"

Gestión de Usuario:

- Navega a la sección "Usuarios".
- Creación de nuevo usuario:
 - Usuarios → Nuevo → Ingresa las credenciales y haz clic en "Guardar".
- Ver lista de usuarios:
 - Usuarios → Lista
- Modificación de usuarios:
 - Usuarios → Lista → Actualizar → Modificar los nuevos datos del usuario siguiendo los pasos que indica la página.
- Eliminar usuario:
 - Usuarios → Lista → Eliminar usuario.
- Buscar usuarios:
 - Usuarios → Buscar → Buscar por nombre, apellido, usuario, o correo (Cualquiera de las opciones es válida).

Gestión de Categorías:

- Navega a la sección "Categorías".
- Agregar nueva categoría:
 - Categorías → Nueva → Agregar nombre de categoría y ubicación (Almacenes).
- Ver lista de categorías:
 - Categorías → Lista.
- Modificación de categorías:
 - Categorías → Lista → Actualizar → Modificar nombre y ubicación de categoría y seleccionar "Actualizar".
- Eliminar categoría:
 - Categorías → Lista → Eliminar categoría.
- Buscar categorías:
 - Categorías → Buscar → Insertar el nombre de la categoría y seleccionar "Buscar".

Gestión de Productos:

- Navega a la sección "Productos".
- Agregar nuevo producto:
 - Productos → Nuevo → Ingresar datos del producto y seleccionar "Guardar".
- Ver lista de productos:
 - Productos → Lista.
- Eliminar producto:
 - Productos → Lista → Eliminar producto.
- Buscar productos por categoría:
 - Productos → Por categoría → Seleccionar categoría.
- Buscar productos:
 - Productos → Buscar → Insertar nombre del producto y seleccionar "Buscar".