

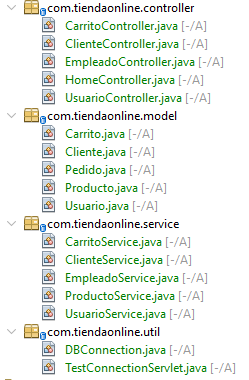
**Universidad Peruana Los Andes  
Ingeniería de Sistemas y Computación**

**Huancayo Tec – Tienda Online**

**Curso:** Taller de Aplicaciones II  
**Docente**: Raul Enrique Fernandez Bejarano **Alumno:** Mosquera Zevallos Valerio

**Informe sobre la Aplicación Web**

La aplicación web implementa un sistema de gestión de usuarios, clientes, empleados, productos, pedidos y carritos de compra. Está construida usando tecnologías de Jakarta EE, incluyendo Servlets, JSP, y una base de datos relacional para gestionar los datos. Este informe detalla cómo interactúan los distintos componentes de la aplicación.

**1. Arquitectura General**

La arquitectura sigue el patrón MVC (Modelo-Vista-Controlador):

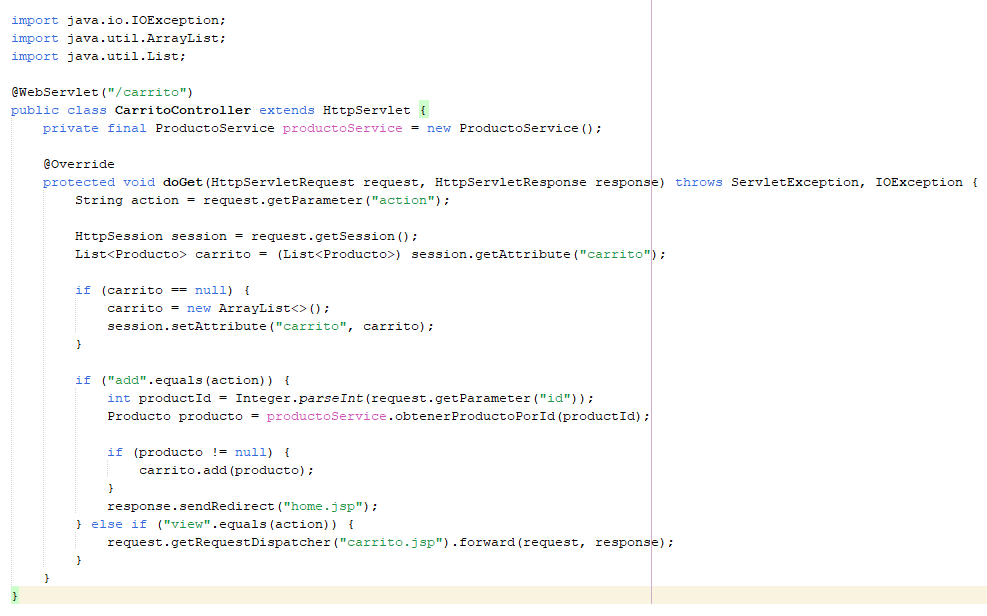
* **Modelo (Model)**: Representa la lógica de negocio y las estructuras de datos.
* **Vista (View)**: Formada por las páginas JSP, que presentan información al usuario y reciben entradas.
* **Controlador (Controller)**: Implementado con Servlets para procesar peticiones del cliente y coordinar las respuestas.

**2. Componentes Principales**

**2.1 Servlets (Controladores)**

Los Servlets actúan como intermediarios entre las vistas (JSP) y el modelo (Base de datos):

* **Responsabilidades:**
  + Reciben las peticiones HTTP desde el navegador.
  + Procesan la lógica necesaria llamando a servicios y modelos.
  + Envían datos a las páginas JSP para su presentación.
* **Ejemplo: CarritoController**
  + Recibe solicitudes para agregar, eliminar o actualizar productos en el carrito.
  + Llama al CarritoService para ejecutar operaciones en la base de datos.
  + Redirige al usuario a carrito.jsp con la lista actualizada de productos.

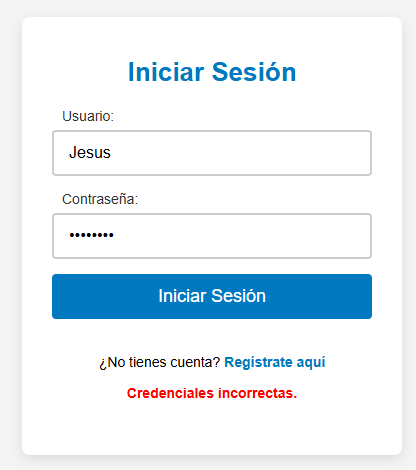
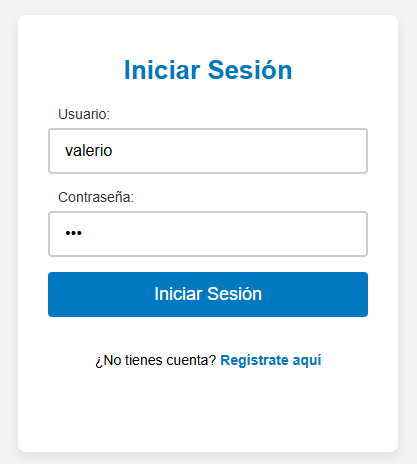


**2.2 Comunicación entre Componentes**

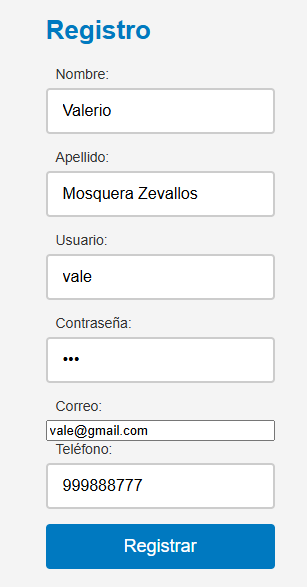
1. **Cliente → Servidor**
   * El cliente realiza solicitudes a través de formularios HTML o botones (acciones GET o POST).
2. **Servidor (Servlet) → Servicios**
   * Los Servlets procesan las solicitudes y llaman a los servicios correspondientes.
   * Los servicios gestionan la lógica de negocio y las interacciones con la base de datos mediante clases DAO (Data Access Object).
3. **Servicios → Base de Datos**
   * Los servicios envían consultas SQL al motor de base de datos a través de DBConnection.
   * Las respuestas de la base de datos se convierten en objetos del modelo (por ejemplo, Usuario, Carrito).
4. **Servidor → Cliente**
   * Los Servlets reenvían los datos procesados a las páginas JSP para mostrarlos.

**2.3 Páginas JSP (Vistas)**

Login

Registro



Página Principal del Usuario



Pagina para ver el Carrito



Pagina Principal para el Empleado



Pagina para ver clientes

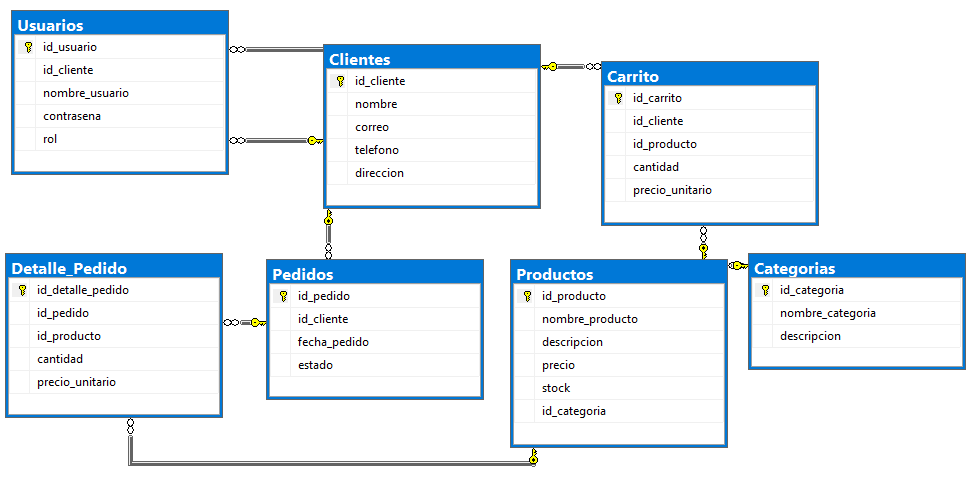


Pagina para agregar Clientes



**2.4 Base de Datos**

* **Estructura Relacional:**
  + La base de datos está diseñada con integridad referencial usando claves primarias y foráneas.
  + Tablas principales:
    - Usuarios: Datos generales de usuarios.
    - Clientes, Empleado: Subtipos especializados de usuario.
    - Productos: Información de los artículos disponibles.
    - Carrito, Pedidos, DetallePedido: Gestionan el flujo de compra.
  + **Ejemplo de Interacción:**
    - Cuando un cliente agrega un producto al carrito:
      * El formulario en carrito.jsp envía una solicitud POST al CarritoController.
      * El controlador llama a CarritoService, que ejecuta una consulta SQL para insertar un registro en la tabla Carrito.



**3. Ejemplo de Flujo Completo: Agregar Producto al Carrito**

1. **Interfaz de Usuario (carrito.jsp)**
   * El cliente selecciona un producto y define la cantidad.
   * Presiona el botón "Agregar al carrito", lo que envía un formulario con datos al servidor.
2. **Solicitud Procesada por el Servlet (CarritoController)**
   * Recibe la solicitud HTTP POST.
   * Extrae los parámetros (id\_usuario, id\_producto, cantidad).
   * Llama a CarritoService para realizar la operación.
3. **Servicio y Modelo**
   * El CarritoService usa la clase DBConnection para interactuar con la base de datos.
   * Ejecuta un INSERT INTO Carrito con los datos proporcionados.
4. **Respuesta del Servlet**
   * Una vez completada la operación, el Servlet redirige al cliente a carrito.jsp.
5. **Vista Actualizada**
   * carrito.jsp muestra la lista de productos actualizada, utilizando los datos reenviados por el Servlet.

**4. Base de Datos: Detalles Técnicos**

* **Esquema Relacional:**
  + Relaciones jerárquicas:
    - Usuarios → Clientes / Empleado.
    - Productos → Carrito / DetallePedido.
  + Claves foráneas garantizan consistencia entre tablas.
* **Procedimientos Almacenados:**
  + BackupDatabase y RestoreDatabase: Aseguran la integridad de los datos y facilitan la recuperación.

**5. Interacción entre Componentes**

El siguiente diagrama muestra la interacción entre los componentes clave:

[Cliente] → (Formulario en carrito.jsp)

↓

[Servlet (CarritoController)]

↓

[Servicio (CarritoService)] → [Base de Datos]

↑

[Página JSP (carrito.jsp)]

**6. Conclusiones**

* **Modularidad y Escalabilidad**: La separación de responsabilidades entre Servlets, servicios, y JSP facilita el mantenimiento y la expansión.
* **Integridad de Datos**: La base de datos está bien diseñada, garantizando relaciones consistentes y coherentes.
* **Interfaz Amigable**: Las páginas JSP permiten una interacción fluida entre el usuario y la aplicación.