

# 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

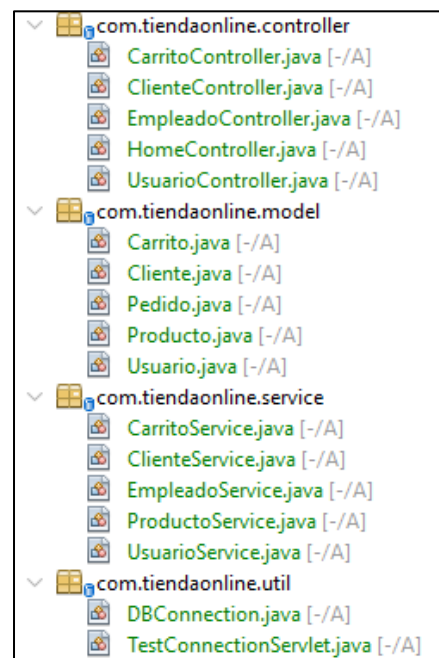
## Informe sobre la Aplicación Web

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.

```
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

@WebServlet("/carrito")
public class CarritoController extends HttpServlet {
    private final ProductoService productoService = new ProductoService();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String action = request.getParameter("action");

        HttpSession session = request.getSession();
        List<Producto> carrito = (List<Producto>) session.getAttribute("carrito");

        if (carrito == null) {
            carrito = new ArrayList<>();
            session.setAttribute("carrito", carrito);
        }

        if ("add".equals(action)) {
            int productId = Integer.parseInt(request.getParameter("id"));
            Producto producto = productoService.obtenerProductoPorId(productId);

            if (producto != null) {
                carrito.add(producto);
            }
            response.sendRedirect("home.jsp");
        } else if ("view".equals(action)) {
            request.getRequestDispatcher("carrito.jsp").forward(request, response);
        }
    }
}
```

### 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

**Iniciar Sesión**

Usuario:

Contraseña:

**Iniciar Sesión**

¿No tienes cuenta? [Regístrate aquí](#)

**Credenciales incorrectas.**

**Iniciar Sesión**

Usuario:

Contraseña:

**Iniciar Sesión**

¿No tienes cuenta? [Regístrate aquí](#)

#### Registro

**Registro**

Nombre:

Apellido:

Usuario:

Contraseña:

Correo:

Teléfono:

**Registrar**

Página Principal del Usuario

 **HuancayoTec**

### Productos Disponibles

Nombre	Descripción	Precio	Acciones
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99	<a href="#">Agregar al Carrito</a>
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99	<a href="#">Agregar al Carrito</a>
Zapatillas Running	Zapatillas para correr, color negro	S/. 59.99	<a href="#">Agregar al Carrito</a>
Mochila Escolar	Mochila con varios compartimentos, 25L	S/. 29.99	<a href="#">Agregar al Carrito</a>
Reloj Inteligente	Reloj inteligente con múltiples funciones de salud	S/. 199.99	<a href="#">Agregar al Carrito</a>
Auriculares Bluetooth	Auriculares inalámbricos con cancelación de ruido	S/. 89.99	<a href="#">Agregar al Carrito</a>

[Ver Carrito](#) [Cerrar sesión](#)

Pagina para ver el Carrito

 **HuancayoTec**

### Carrito de Compras

Nombre	Descripción	Precio
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Camiseta Deportivo	Camiseta deportiva de algodón, talla M	S/. 19.99
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99
Pantalón de Jeans	Pantalón de jeans de corte moderno, talla 32	S/. 39.99

[Regresar a productos](#)

Pagina Principal para el Empleado



Pagina para ver clientes



Pagina para agregar Clientes



# HuancayoTec

## Agregar Cliente

ID Usuario:

Nombre:

Apellido:

Correo:

Teléfono:

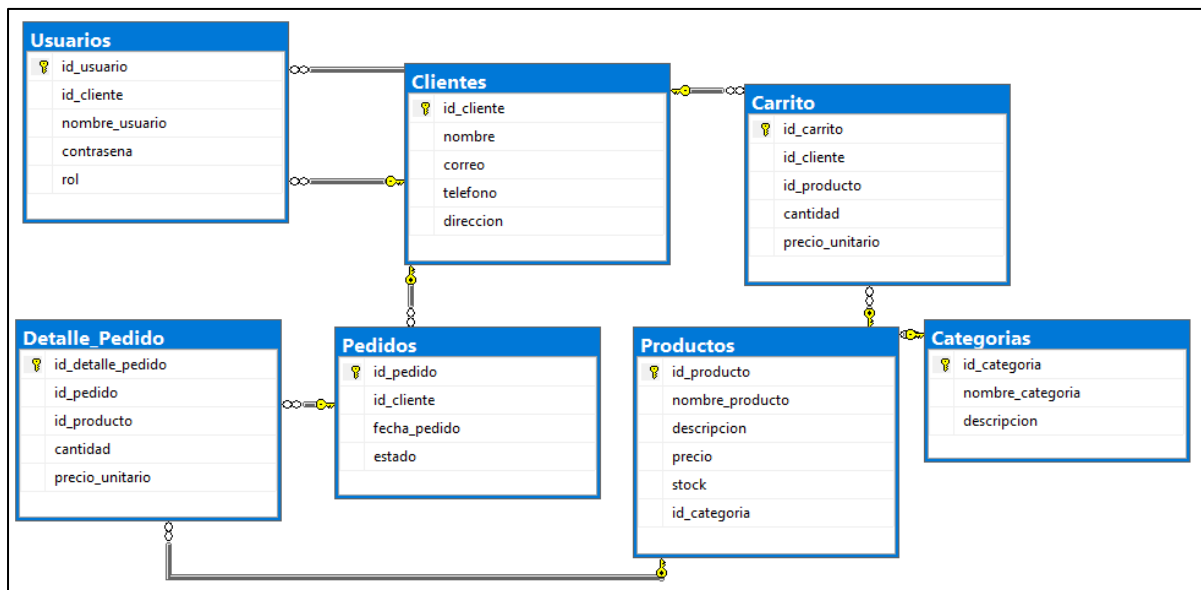
[Agregar Cliente](#)

[Volver a la lista](#)

## 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

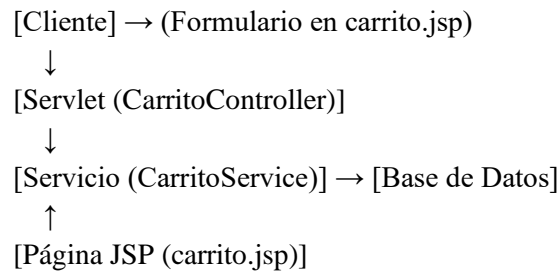
- carrito.jsp muestra la lista de productos actualizada, utilizando los datos reenviados por el Servlet.

#### 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:



#### 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.