

Documento de Arquitectura de Software

Software para tienda de libros

Coronel Jorge

Noguera German  
Salazar Belén

Documento de Arquitectura de Software

**Contenido**

[1 Introducción 3](#_Toc161416909)

[2 Vista de Casos de Uso 4](#_Toc161416910)

[2.1 Casos de uso identificados 4](#_Toc161416911)

[2.2 Diagrama de Casos de Uso Críticos 5](#_Toc161416912)

[2.3 Especificación de Casos de Uso Críticos 6](#_Toc161416913)

[3 Vista Lógica 8](#_Toc161416914)

[3.1 Estilo Arquitectónico 8](#_Toc161416915)

[3.2 Diagrama de Componentes del Sistema 9](#_Toc161416916)

[3.3 Subsistemas y Componentes 10](#_Toc161416917)

[3.4 Diseño de Interfaces 14](#_Toc161416918)

[4 Vista de Procesos 20](#_Toc161416919)

[4.1 Consultar Productos 20](#_Toc161416920)

[4.2 Gestionar Datos del Cliente 21](#_Toc161416921)

[4.3 Registrar Cliente 21](#_Toc161416922)

[4.4 Gestionar Carrito de Compra 22](#_Toc161416923)

[4.5 Comprar 22](#_Toc161416924)

[4.6 Pagar 23](#_Toc161416925)

[4.7 Validar Usuario 23](#_Toc161416926)

[4.8 Gestionar Productos 24](#_Toc161416927)

[5 Vista de Distribución – Deployment 25](#_Toc161416928)

[6 Referencias 27](#_Toc161416929)

# Introducción

El presente documento tiene como objetivo proporcionar una visión detallada de la arquitectura de software propuesta para la aplicación de venta electrónica de libros, desarrollada por la empresa InfoSecurity en colaboración con [nombre de la empresa]. Esta aplicación busca mantener y mejorar la posición de la compañía en el mercado, ofreciendo a los clientes una experiencia de compra fluida y segura.

La arquitectura de software es fundamental para el éxito de cualquier aplicación, ya que define la estructura y el comportamiento del sistema, así como las interacciones entre sus componentes. En este contexto, se ha diseñado una arquitectura basada en un modelo de tres capas: presentación, proceso y datos, que permitirá una gestión eficiente de los distintos tipos de usuarios y funcionalidades requeridas.

A lo largo de este documento, se detallarán los requisitos de negocio y técnicos establecidos para la aplicación, así como la arquitectura propuesta para cumplir con dichos requisitos. Se explicarán las decisiones de diseño tomadas, los componentes principales del sistema, las tecnologías utilizadas y las interacciones entre ellos.

El Documento de Arquitectura de Software servirá como guía para el equipo de desarrollo durante el proceso de implementación, facilitando la comprensión de la estructura del sistema y asegurando la coherencia y la calidad del código. Además, proporcionará una base sólida para futuras modificaciones y expansiones del sistema, garantizando su escalabilidad y mantenibilidad a largo plazo.

# Vista de Casos de Uso

En esta sección se detalla la representación de los requisitos del sistema a ser construido en forma de casos de uso. Se describe el comportamiento de estos como también la información necesaria que requiere el sistema por parte del usuario.

## Casos de uso identificados

* + 1. **Buscar Productos**

Este caso de uso permite a los clientes buscar productos disponibles en la tienda electrónica. Los clientes pueden utilizar filtros de búsqueda, como título, autor, género, o palabras clave, para encontrar los libros deseados.

* + 1. **Gestionar Pedidos**

Los clientes tienen la capacidad de gestionar sus pedidos a través de la aplicación. Pueden ver el estado de sus pedidos, agregar o eliminar productos del carrito de compras, y realizar acciones relacionadas con la entrega y el pago.

* + 1. **Registrarse**

Antes de poder realizar un pedido, los clientes deben registrarse en la aplicación creando una cuenta de usuario. Este caso de uso permite a los clientes proporcionar la información necesaria para crear una cuenta, como nombre, dirección, información de contacto, y método de pago preferido.

* + 1. **Autenticarse**

Una vez registrados, los clientes deben autenticarse en la aplicación utilizando sus credenciales de cuenta de usuario. Este caso de uso verifica las credenciales del cliente contra la base de datos implementada en el backend de la compañía, a través de una interfaz de servicios web.

* + 1. **Realizar Pago**

Después de seleccionar los productos deseados y agregarlos al carrito de compras, los clientes pueden proceder al pago de su pedido. Pueden utilizar tarjetas de crédito, débito o realizar una transferencia bancaria para completar la transacción.

* + 1. **Acceso de Administrador TI**

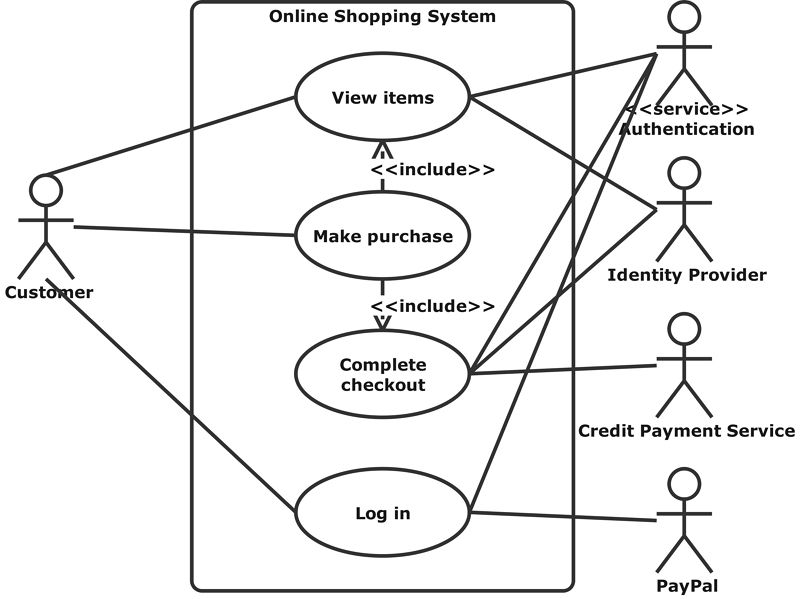
Este caso de uso permite al administrador de TI acceder a funcionalidades administrativas de la aplicación, como gestionar cuentas de usuario, supervisar el rendimiento del sistema, y realizar tareas de mantenimiento y actualización.

* + 1. **Acceso de Agente de Ventas**

Los agentes de ventas tienen acceso a herramientas y funcionalidades específicas que les permiten gestionar el inventario de productos, atender consultas de clientes, y realizar ventas a través de la tienda web o llamadas telefónicas.

## Diagrama de Casos de Uso Críticos

En esta sección se detalla las relaciones entre los casos de uso descritos. Debido a que los casos de usos fueron específicamente tomados con el objetivo de poder englobar una serie de funcionalidades particulares y distintas del sistema, éstos presentan una notable característica de independencia entre sí.



## Especificación de Casos de Uso Críticos

En esta sección se detallan los flujos de ejecución de los casos de uso paso a paso. Con sus condiciones requeridas para la ejecución de los mismos.

|  |  |
| --- | --- |
| **Nombre:** | Buscar Productos |
| **Objetivo:** | Permitir a los clientes buscar y encontrar productos disponibles en la tienda electrónica. |
| **Actores:** | Cliente |
| **Precondiciones:** | El cliente debe estar autenticado en la aplicación. |
| **Descripción:** | El cliente utiliza la funcionalidad de búsqueda para encontrar productos específicos dentro de la tienda electrónica. Puede utilizar filtros de búsqueda como título, autor, género, o palabras clave para refinar los resultados. |
| **Flujo Normal:** | 1. El cliente ingresa a la sección de búsqueda de la aplicación. 2. El cliente introduce los criterios de búsqueda (título, autor, género, etc.). 3. La aplicación realiza una búsqueda en la base de datos de productos. 4. La aplicación muestra los resultados de la búsqueda al cliente. |
| **Flujos Alternativos:** | - Si no se encuentran resultados para la búsqueda, la aplicación muestra un mensaje indicando que no se encontraron resultados. |
| **Postcondiciones:** | El cliente visualiza los resultados de la búsqueda y puede proceder a ver los detalles de los productos encontrados. |

|  |  |
| --- | --- |
| **Nombre:** | Gestionar Pedidos |
| **Objetivo:** | Permitir a los clientes gestionar sus pedidos, agregar o eliminar productos del carrito de compras, y realizar acciones relacionadas con la entrega y el pago. |
| **Actores:** | Cliente |
| **Precondiciones:** | El cliente debe haber iniciado sesión en la aplicación. |
| **Descripción:** | El cliente tiene la capacidad de gestionar sus pedidos, incluyendo agregar productos al carrito de compras, eliminar productos del carrito, actualizar la cantidad de productos, seleccionar la dirección de envío y el método de pago, y confirmar la compra. |
| **Flujo Normal:** | 1. El cliente accede a la sección de gestión de pedidos en la aplicación. 2. El cliente selecciona un pedido existente o inicia un nuevo pedido. 3. El cliente agrega productos al carrito de compras, actualiza la cantidad de productos, selecciona la dirección de envío y el método de pago. 4. El cliente confirma la compra. |
| **Flujos Alternativos:** | - Si el cliente desea cancelar un pedido, puede eliminarlo del carrito de compras antes de confirmar la compra. |
| **Postcondiciones:** | El pedido se registra en el sistema y se procesa para su entrega. |

|  |  |
| --- | --- |
| **Nombre:** | Registrarse |
| **Objetivo:** | Permitir a los clientes crear una cuenta de usuario en la aplicación. |
| **Actores:** | Cliente |
| **Precondiciones:** | Ninguna |
| **Descripción:** | El cliente proporciona la información necesaria para crear una cuenta de usuario, incluyendo nombre, dirección, información de contacto, y método de pago preferido. |
| **Flujo Normal:** | 1. El cliente accede a la página de registro de la aplicación.  2. El cliente completa el formulario de registro con la información requerida.  3. La aplicación valida la información proporcionada por el cliente. 4. La aplicación crea una cuenta de usuario para el cliente. |
| **Flujos Alternativos:** | - Si la información proporcionada por el cliente es inválida o está incompleta, la aplicación muestra un mensaje de error y solicita al cliente que corrija los campos necesarios. |
| **Postcondiciones:** | El cliente tiene una cuenta de usuario registrada en la aplicación. |

|  |  |
| --- | --- |
| **Nombre:** | Autenticarse |
| **Objetivo:** | Permitir a los clientes autenticarse en la aplicación utilizando sus credenciales de cuenta de usuario. |
| **Actores:** | Cliente, Administrador TI |
| **Precondiciones:** | El cliente debe haber creado una cuenta de usuario previamente. |
| **Descripción:** | El cliente proporciona su nombre de usuario y contraseña para iniciar sesión en la aplicación. El administrador de TI también puede autenticarse utilizando credenciales especiales para acceder a funcionalidades administrativas. |
| **Flujo Normal:** | 1. El cliente accede a la página de inicio de sesión de la aplicación. 2. El cliente ingresa su nombre de usuario y contraseña. 3. La aplicación valida las credenciales del cliente. 4. El cliente es autenticado y tiene acceso a las funcionalidades de la aplicación. |
| **Flujos Alternativos:** | - Si las credenciales proporcionadas son incorrectas, la aplicación muestra un mensaje de error y solicita al cliente que vuelva a intentarlo. |
| **Postcondiciones:** | El cliente tiene acceso a las funcionalidades de la aplicación como usuario autenticado. |

|  |  |
| --- | --- |
| **Nombre:** | Realizar Pago |
| **Objetivo:** | Permitir a los clientes realizar el pago de sus pedidos utilizando diferentes métodos de pago. |
| **Actores:** | Cliente |
| **Precondiciones:** | El cliente debe haber iniciado sesión en la aplicación y tener al menos un producto en el carrito de compras. |
| **Descripción:** | El cliente selecciona el método de pago deseado y proporciona la información necesaria para completar la transacción, como detalles de la tarjeta de crédito o débito, o información de la cuenta bancaria para transferencias. |
| **Flujo Normal:** | 1. El cliente accede al proceso de pago desde el carrito de compras. 2. El cliente selecciona el método de pago deseado (tarjeta de crédito, tarjeta de débito, transferencia bancaria, etc.). 3. El cliente proporciona la información necesaria para completar la transacción. 4. La aplicación procesa el pago y confirma la compra al cliente. |
| **Flujos Alternativos:** | - Si el método de pago seleccionado es una tarjeta de crédito o débito, la aplicación realiza una verificación de la tarjeta antes de procesar el pago. |
| **Postcondiciones:** | El pedido se registra como pagado en el sistema y se procesa para su entrega. |

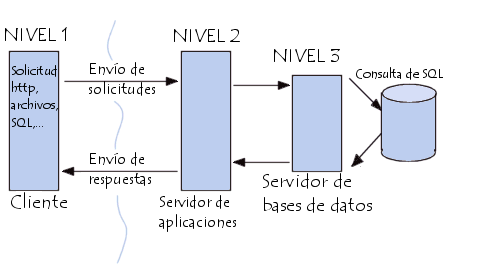
Documento de Arquitectura de Software

# Vista Lógica

Basados en los Casos de Uso, la vista lógica se centra en la estructura y la organización interna del sistema, incluyendo la lógica de negocio y los componentes que componen la aplicación. En este caso, como estamos trabajando con una arquitectura de tres capas, vamos a dividir la vista lógica en las capas de presentación, proceso y datos.

## Estilo Arquitectónico

En el caso de la aplicación de venta electrónica de libros, se ha seleccionado un estilo arquitectónico de tres capas, que es un enfoque comúnmente utilizado para aplicaciones web y que se adapta bien a los requisitos del sistema.



El estilo de tres capas divide la aplicación en tres componentes principales, cada uno con un propósito específico y bien definido:

**Capa de Presentación**

Esta capa se encarga de la interfaz de usuario y la interacción directa con los clientes. Aquí se incluyen todas las páginas web, formularios y componentes de la interfaz que los usuarios utilizan para interactuar con la aplicación. La capa de presentación se centra en la presentación de la información de una manera clara y atractiva para los usuarios finales.

**Capa de Proceso**

La capa de proceso contiene la lógica de negocio de la aplicación. Aquí se encuentran los controladores y servicios que manejan las solicitudes de los clientes, procesan la información y coordinan las acciones entre la capa de presentación y la capa de datos. Esta capa es responsable de implementar las reglas de negocio y asegurar que el sistema funcione de acuerdo a lo esperado.

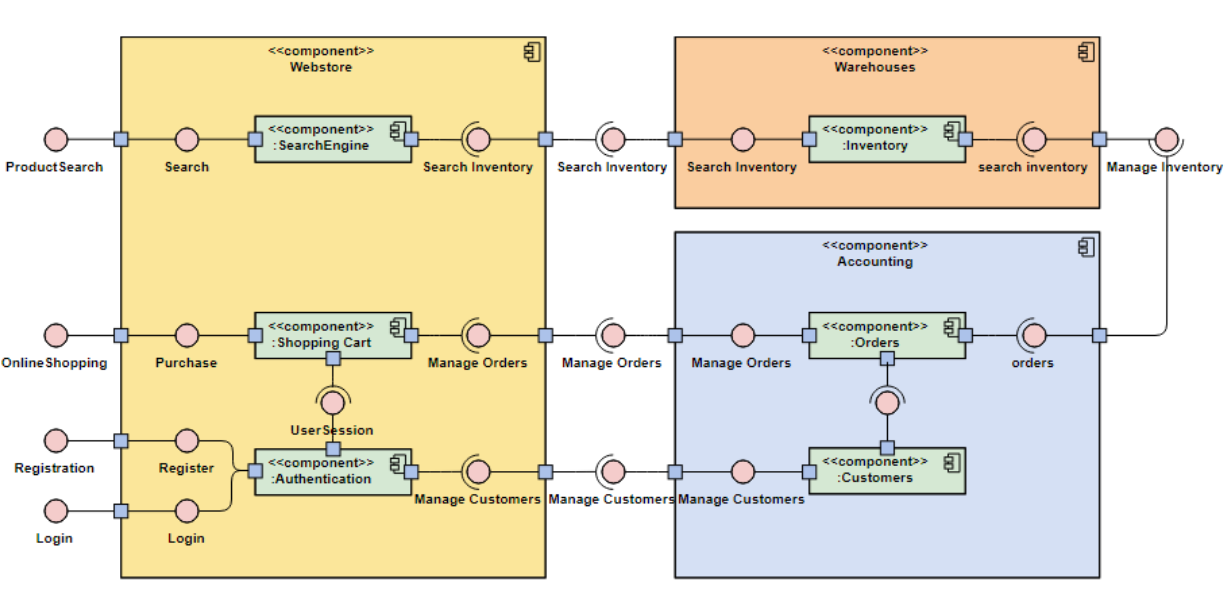
**Capa de Datos**

En esta capa se almacena y gestiona la información utilizada por la aplicación. Aquí se encuentra la base de datos que contiene los datos de productos, clientes, pedidos, etc. La capa de datos proporciona una interfaz para acceder y manipular los datos de manera eficiente y segura. Además de la base de datos, esta capa puede incluir otros sistemas de almacenamiento de datos, como sistemas de archivos o servicios en la nube.

## Diagrama de Componentes del Sistema

Si simplificamos el esquema de tres capas de un sistema de tienda online, centrándonos en los componentes lógicos y destacando los subsistemas esenciales para implementar la solución, obtendremos un diagrama de componentes que refleja la arquitectura del sistema. Este enfoque permite realizar refinamientos iterativos en la vista lógica del sistema.

A continuación, el diagrama con sus capas o subsistemas y componentes que lo integran.



El diagrama de componentes del sistema representa los distintos subsistemas y componentes que componen la aplicación de venta electrónica de libros, así como las relaciones y dependencias entre ellos. Este diagrama proporciona una vista de alto nivel de la estructura del sistema y ayuda a entender cómo están organizados los diferentes elementos que lo componen.

En la siguiente sección se detallan los diferentes subsistemas entrando en detalle en sus componentes y las interfaces que implementan.

## Subsistemas y Componentes

En esta sección se describen cada uno de los subsistemas identificados anteriormente. El diagrama de componentes muestra en detalle aquellos que corresponden a cada subsistema.

* + 1. **Interfaz de Usuario (UI)**
       1. **Descripción**

Este subsistema comprende la interfaz de usuario de la aplicación, incluyendo todas las páginas web, formularios y elementos de diseño que permiten a los usuarios interactuar con el sistema.

* + - 1. **Componentes**

Incluye componentes de frontend que manejan la presentación de la información y la interacción del usuario, como páginas HTML, hojas de estilo CSS y scripts JavaScript.

**Páginas Web:** Cada página web representa una parte específica de la aplicación y proporciona funcionalidades como la búsqueda de productos, la visualización de detalles del producto, la gestión del carrito de compras, etc. Estas páginas están diseñadas utilizando HTML para la estructura, CSS para el estilo y JavaScript para la interactividad.

**Formularios:** Los formularios permiten a los usuarios ingresar y enviar información, como detalles de inicio de sesión, información de envío y detalles de pago. Están diseñados con HTML y CSS, y pueden incluir validación de datos del lado del cliente para mejorar la experiencia del usuario.

**Elementos de Navegación:** Estos elementos, como menús de navegación, barras laterales y botones de acción, facilitan la navegación y la interacción del usuario con la aplicación. Se implementan utilizando HTML, CSS y a menudo se agregan interacciones dinámicas con JavaScript para mejorar la usabilidad.

**Estilos CSS:** Los estilos CSS se utilizan para diseñar y dar formato a los elementos de la interfaz de usuario, como texto, imágenes, botones y fondos. Se pueden aplicar estilos personalizados para garantizar una apariencia coherente y atractiva en toda la aplicación.

**Scripts JavaScript:** Los scripts JavaScript se utilizan para agregar interactividad y funcionalidad dinámica a la interfaz de usuario. Se pueden utilizar para validar formularios, realizar solicitudes AJAX para cargar datos de forma asíncrona, y crear efectos de animación y transición para mejorar la experiencia del usuario.

* + 1. **Controladores y Servicios (Backend)**
       1. **Descripción**

Este subsistema contiene los controladores y servicios que implementan la lógica de negocio de la aplicación. Aquí se encuentra la lógica que procesa las solicitudes de los clientes, interactúa con la base de datos y coordina las acciones entre la interfaz de usuario y la capa de datos.

* + - 1. **Componentes**

Incluye componentes de backend, como controladores de ruta, servicios de autenticación, servicios de gestión de pedidos, etc.

**Controladores de Ruta:** Los controladores de ruta son componentes que gestionan las solicitudes de los clientes y definen las acciones a realizar en respuesta a esas solicitudes. Por ejemplo, un controlador de ruta puede manejar la solicitud de búsqueda de productos, la solicitud de registro de un nuevo cliente, etc.

**Servicios de Autenticación:** Son componentes encargados de gestionar el proceso de autenticación de los usuarios. Estos servicios verifican las credenciales de inicio de sesión proporcionadas por los clientes y generan tokens de sesión válidos para permitir el acceso a las funcionalidades protegidas de la aplicación.

**Servicios de Gestión de Pedidos:** Los servicios de gestión de pedidos son componentes que se encargan de procesar las solicitudes relacionadas con la gestión de pedidos de los clientes. Esto incluye la creación de nuevos pedidos, la actualización del estado de los pedidos existentes, la gestión de la información de envío, etc.

**Servicios de Gestión de Productos:** Son responsables de la gestión de la información de productos disponibles en la tienda electrónica. Esto incluye la consulta de detalles de productos, la actualización de inventario, la gestión de categorías y etiquetas, etc.

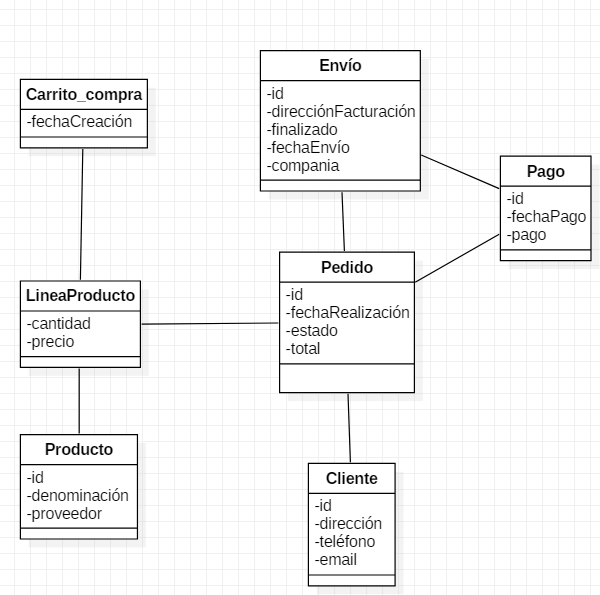
**Servicios de Integración Externa:** Estos componentes se encargan de integrar la aplicación con servicios externos, como procesadores de pagos, sistemas de envío y servicios de autenticación externos. Estos servicios permiten a la aplicación interactuar con sistemas externos para llevar a cabo funciones específicas, como procesar pagos con tarjeta de crédito o gestionar la autenticación de usuarios mediante proveedores de identidad externos como OAuth.

* + 1. **Base de Datos**
       1. **Descripción**

Este subsistema almacena y gestiona la información utilizada por la aplicación, como datos de productos, clientes, pedidos, etc. Aquí se encuentra la base de datos que proporciona acceso y manipulación de los datos de manera eficiente y segura.

* + - 1. **Diagrama de clase**

Este subsistema contiene los controladores y servicios que implementan la lógica de negocio de la aplicación. Aquí se encuentra la lógica que procesa las solicitudes de los clientes, interactúa con la base de datos y coordina las acciones entre la interfaz de usuario y la capa de datos.



* + 1. **Servicios Externos**
       1. **Descripción**

El subsistema de Servicios Externos es una parte fundamental de la aplicación de venta electrónica de libros, ya que permite la integración con sistemas externos para llevar a cabo funciones específicas que no son manejadas internamente por la aplicación. Este subsistema facilita la interacción con servicios externos como procesadores de pagos, sistemas de envío y servicios de autenticación externos, entre otros.

* + - 1. **Componentes**
* **Servicio de Procesamiento de Pagos**

Este componente se encarga de interactuar con un procesador de pagos externo para procesar los pagos realizados por los clientes. Puede ser una pasarela de pago como PayPal, Stripe, o cualquier otro proveedor de servicios de pago.

* **Servicio de Envío**

El servicio de envío se encarga de interactuar con sistemas externos de envío y logística para gestionar la entrega de los productos a los clientes. Puede integrarse con servicios de mensajería como UPS, FedEx, DHL, etc.

* **Servicio de Autenticación Externa**

Este servicio permite la autenticación de los usuarios mediante proveedores de identidad externos, como OAuth. Permite a los usuarios iniciar sesión en la aplicación utilizando sus cuentas de redes sociales u otros servicios de autenticación.

## Diseño de Interfaces

El diseño de interfaces para la aplicación de venta electrónica de libros se centraría en crear una experiencia de usuario intuitiva, atractiva y fácil de usar. Esto incluiría el diseño de las páginas web y la interfaz de usuario tanto para la versión de escritorio como para dispositivos móviles.

* + 1. **Diseño responsivo**

Las interfaces se adaptan a diferentes tamaños de pantalla y dispositivos, asegurando una experiencia coherente y funcional para los usuarios en computadoras de escritorio, tabletas y teléfonos móviles.

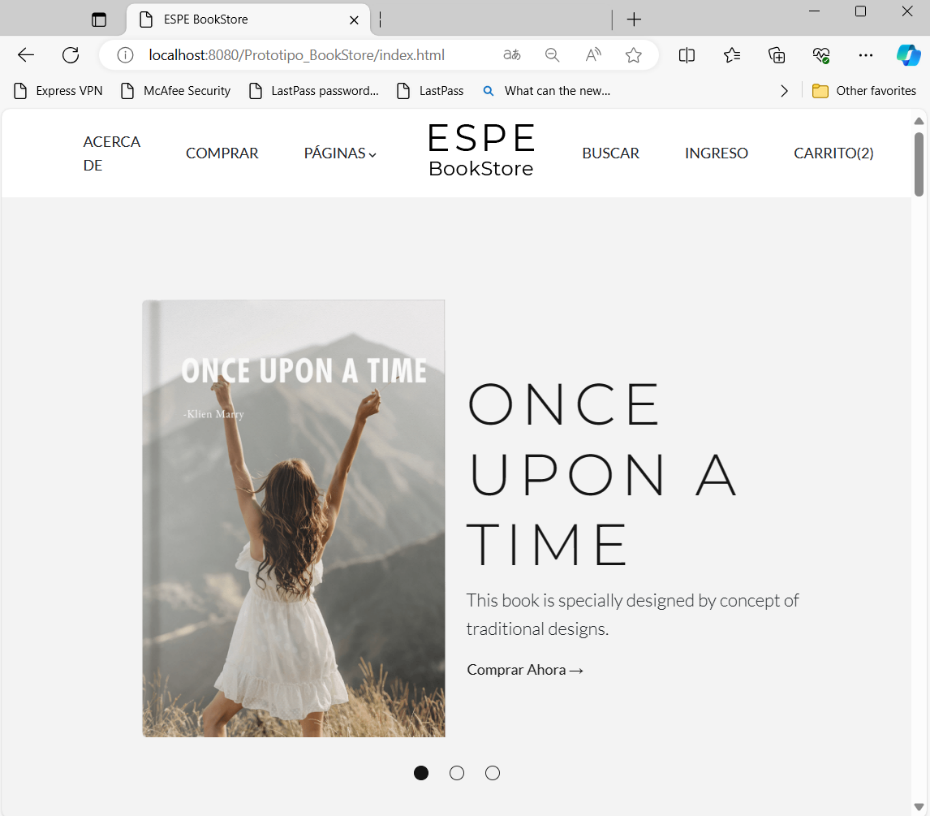
* + 1. **Navegación Intuitiva**

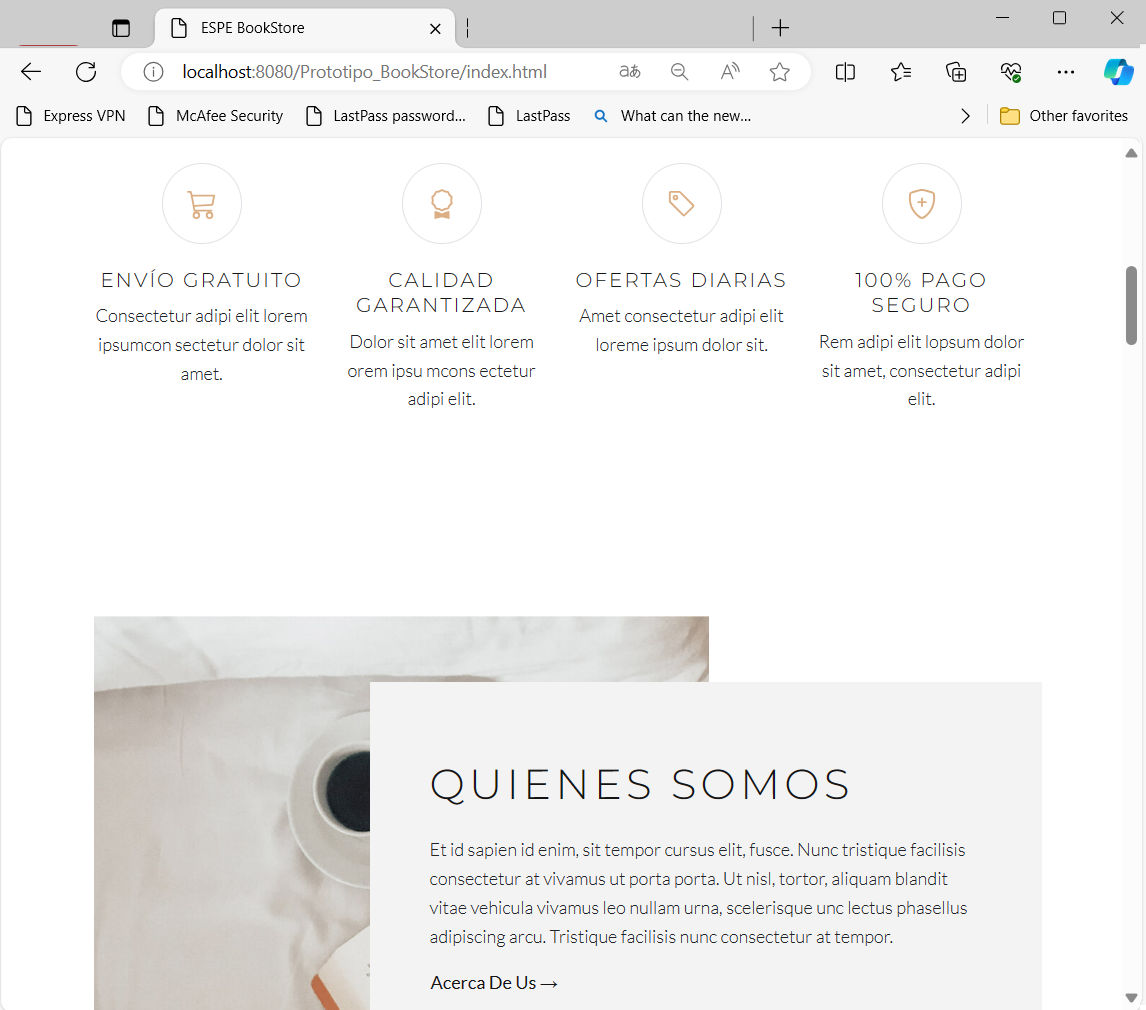
Se diseña una navegación clara y fácil de usar que permita a los usuarios encontrar rápidamente lo que están buscando. Esto incluye un menú de navegación bien organizado, botones de llamada a la acción claros y una estructura de páginas lógica.

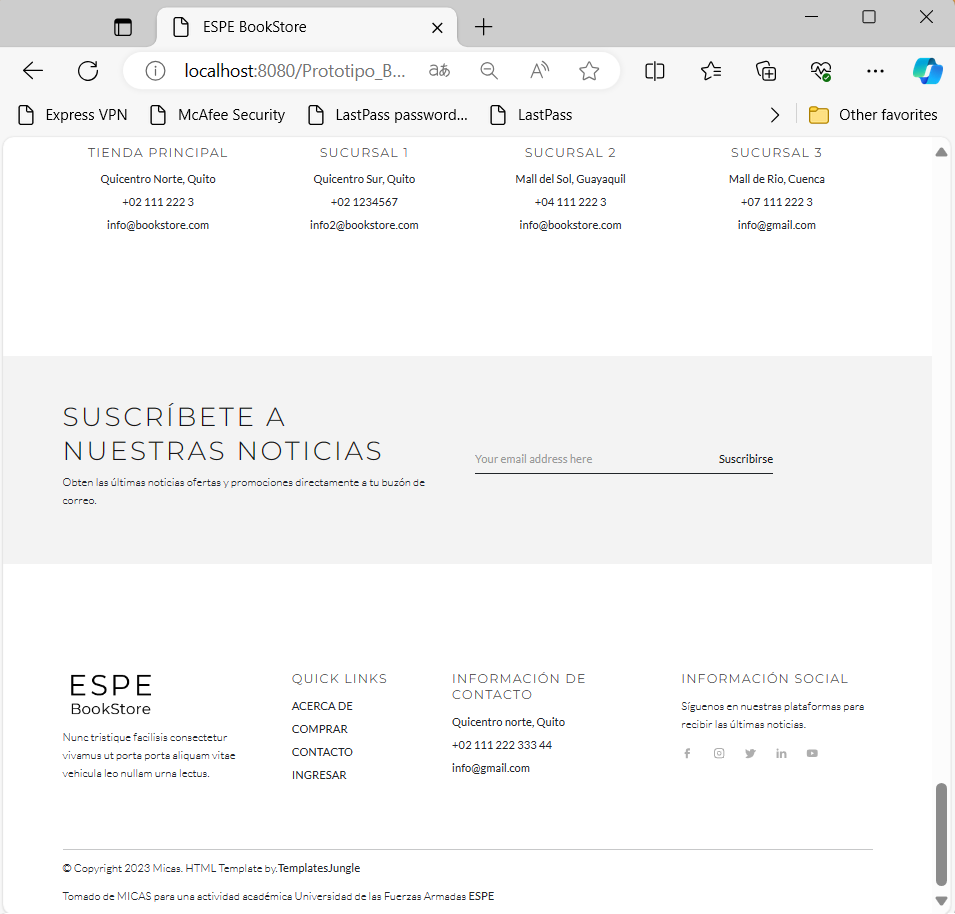
* + 1. **Diseño atractivo**

El diseño visual de las interfaces es atractivo y agradable a la vista, utilizando colores, tipografías y elementos visuales que reflejen la identidad de la marca y creen una experiencia de usuario agradable.

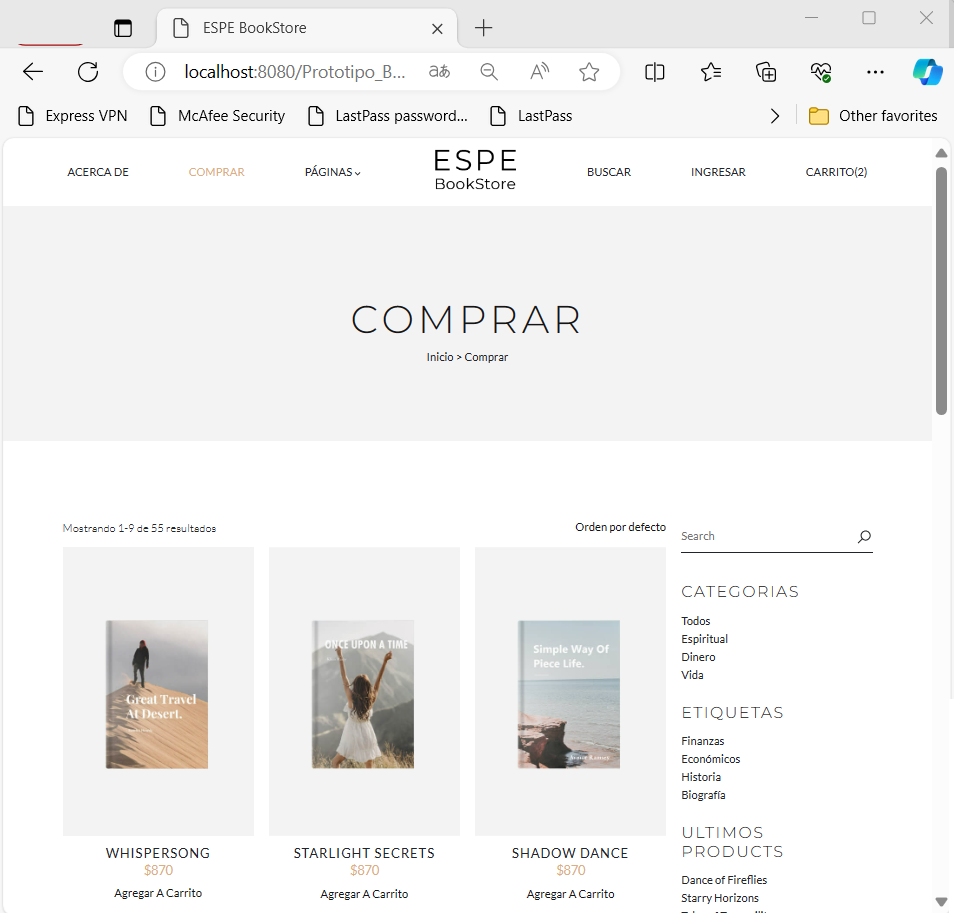
Landing Page:



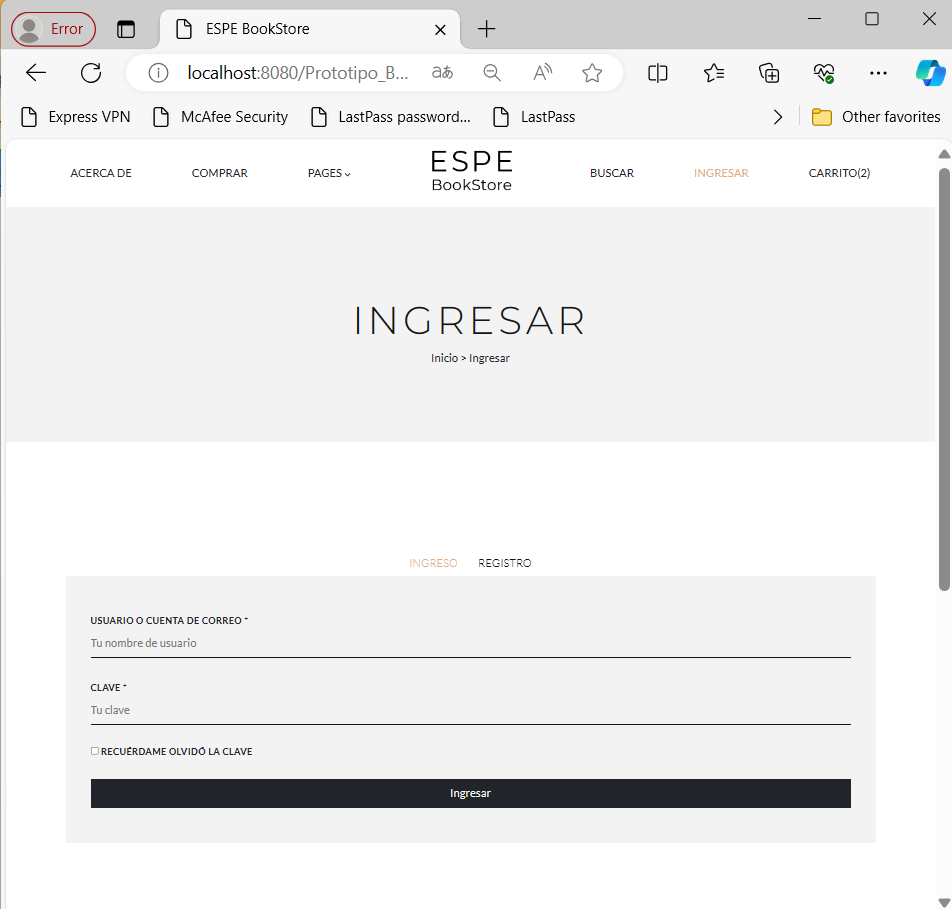


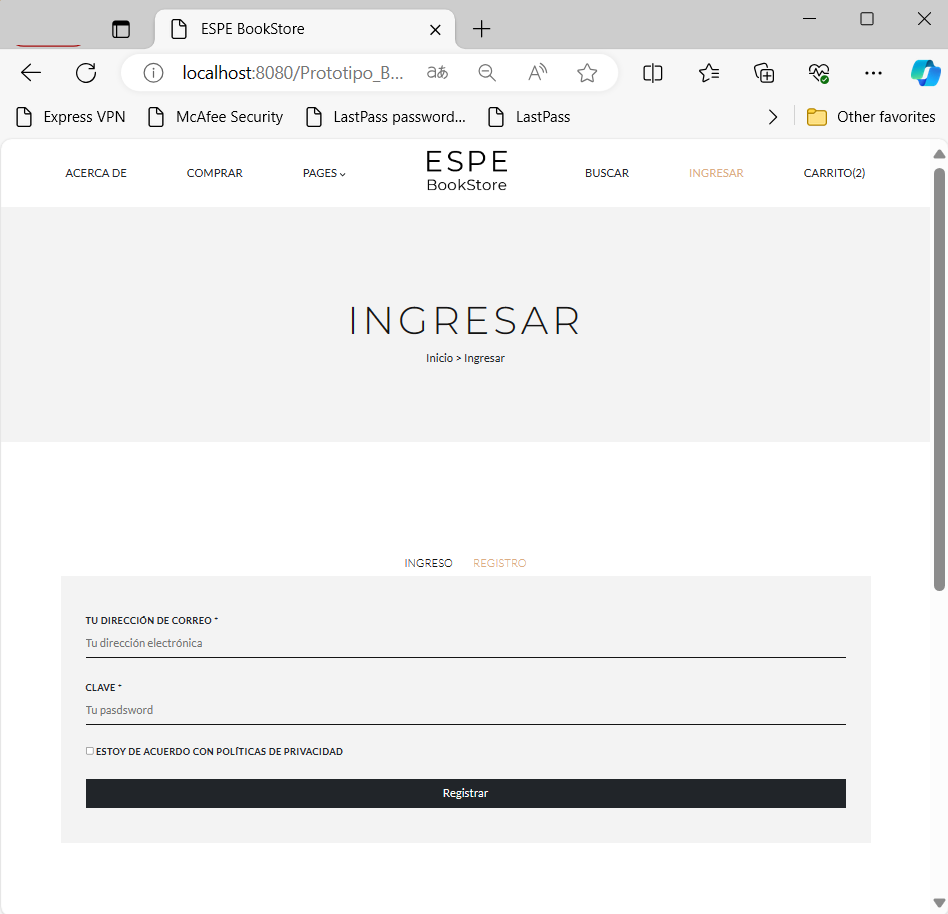


Página de compras:

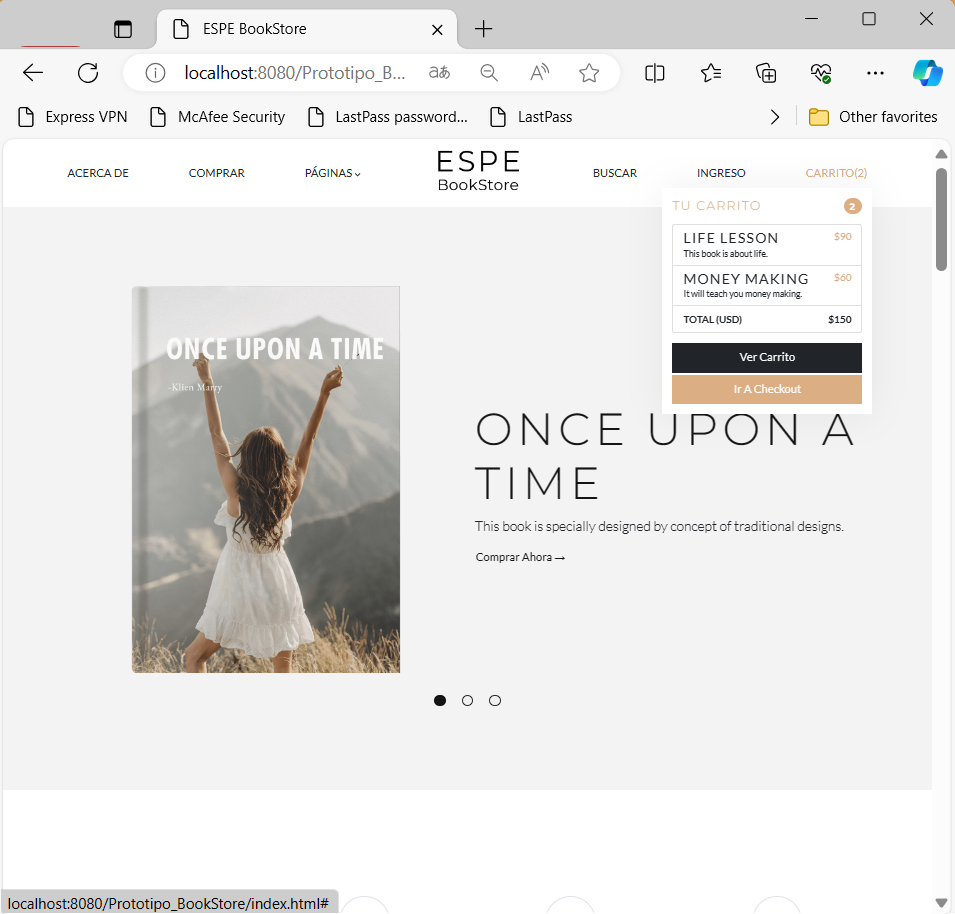


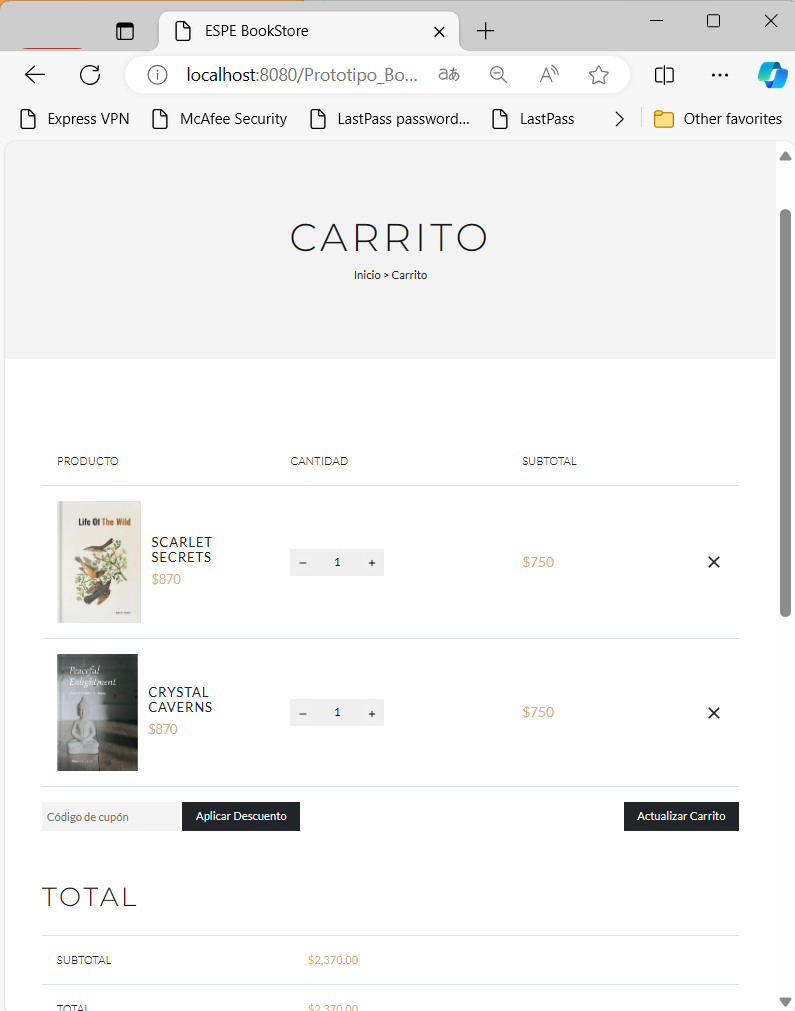
Página de Ingreso de Administradores y Clientes y Registro de Clientes.





Carrito de compras:







# Vista de Procesos

La vista de procesos trata con aspectos dinámicos del sistema. Representa cómo se comunican entre sí los subsistemas y sus componentes. Se enfoca en el comportamiento del sistema en tiempo de ejecución.

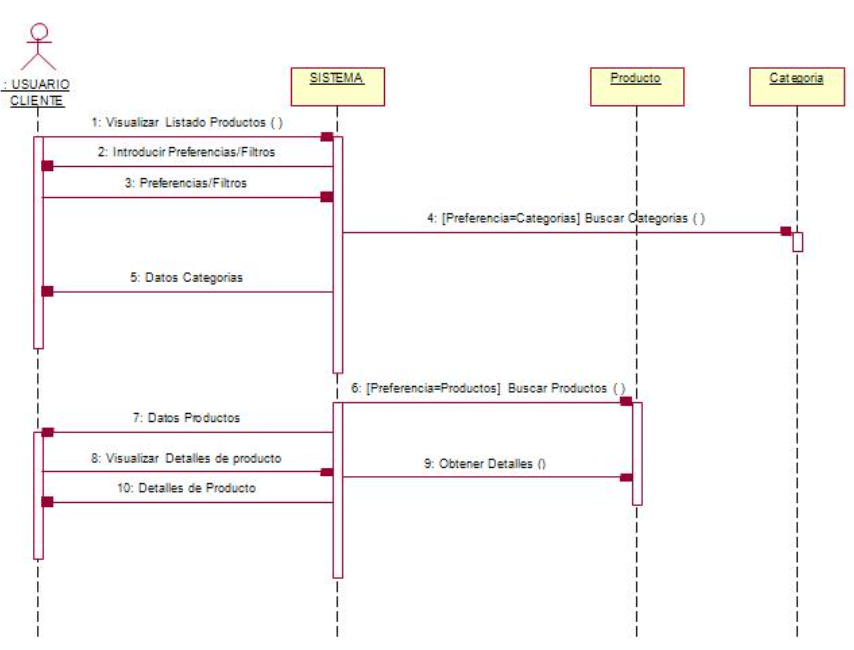
## Consultar Productos

**Objetivo**

Permitir al usuario cliente buscar y visualizar productos disponibles en la tienda electrónica según sus preferencias y criterios de búsqueda.

**Descripción**

El usuario solicita la visualización de productos, establece preferencias de búsqueda y filtros de información, y el sistema muestra una lista de productos que cumplen con las condiciones establecidas.



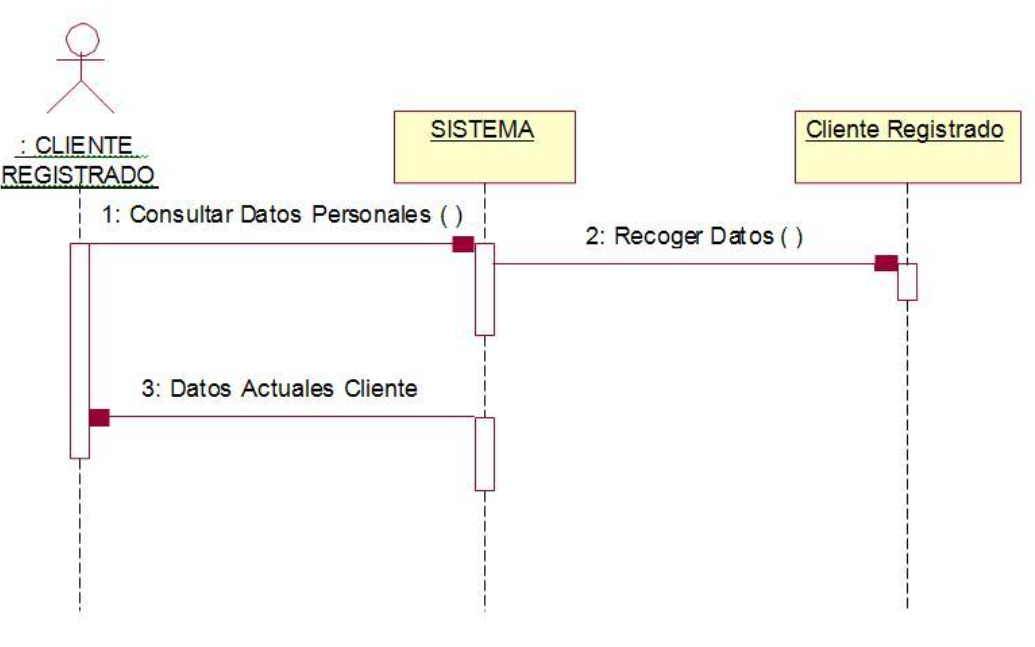
## Gestionar Datos del Cliente

**Objetivo**

Permitir al usuario cliente ver y modificar sus datos personales registrados en el sistema.

**Descripción**

El usuario solicita al sistema visualizar o actualizar sus datos personales, el sistema muestra los datos actuales y permite al usuario realizar las modificaciones necesarias.



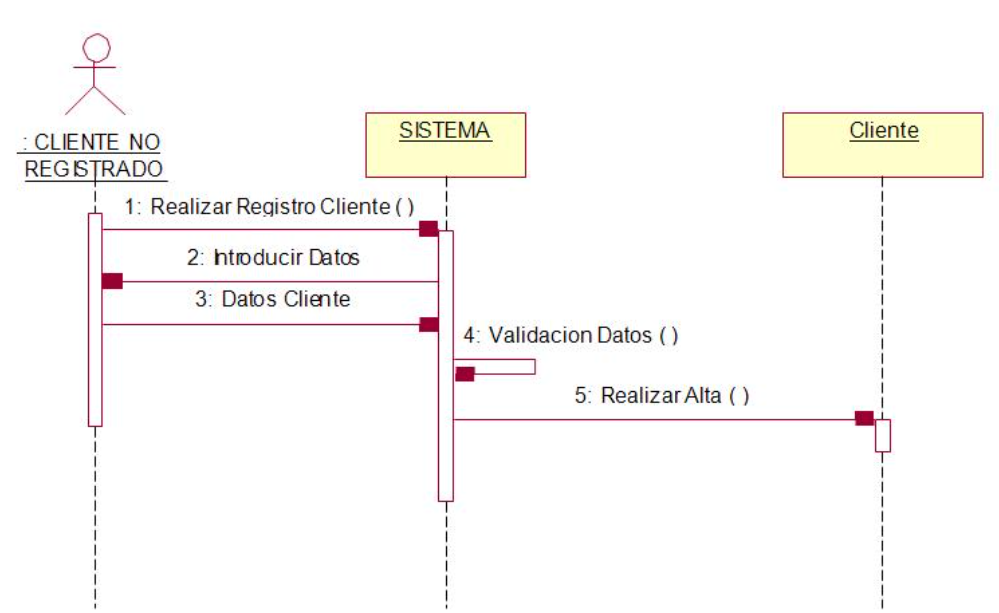
## Registrar Cliente

**Objetivo**

Permitir a un usuario no registrado registrarse como cliente en el sistema.

**Descripción**

El usuario no registrado solicita el registro, proporciona los datos requeridos, y el sistema valida y registra al usuario como cliente.



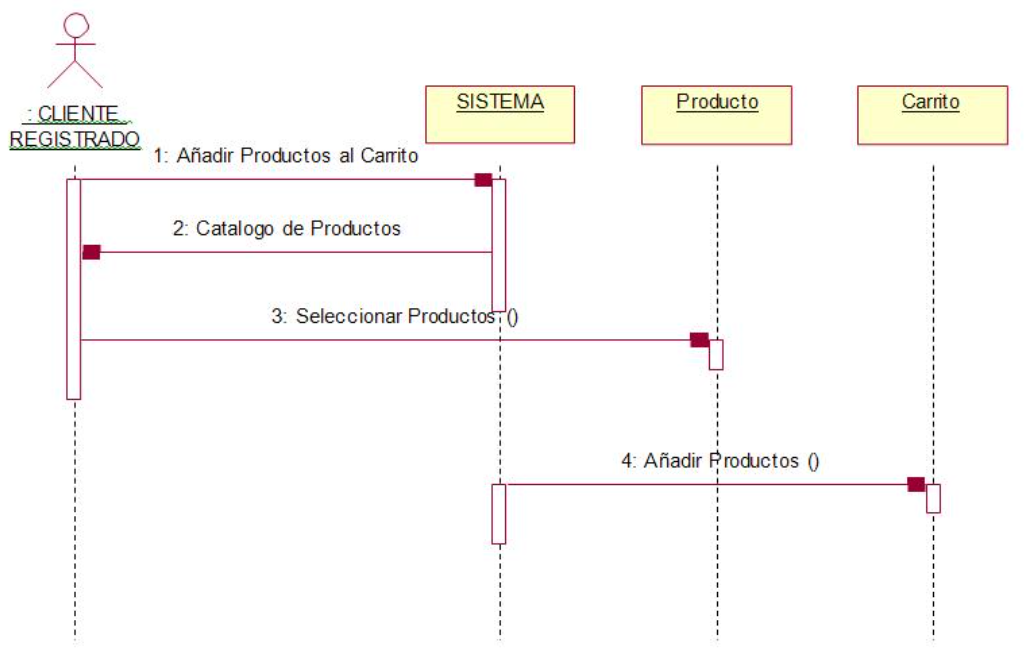
## Gestionar Carrito de Compra

**Objetivo**

Permitir al usuario cliente agregar, visualizar y eliminar productos en su carrito de compra.

**Descripción**

El usuario agrega productos al carrito, visualiza su contenido actual y puede eliminar productos según sea necesario.



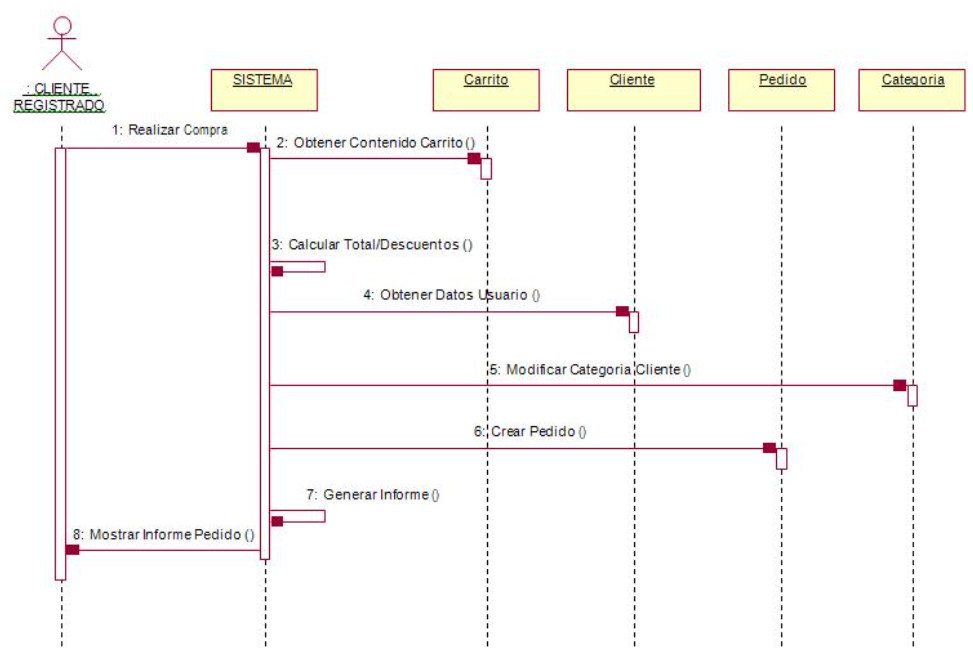
## Comprar

**Objetivo**

Permitir al usuario cliente registrado realizar la compra de los productos en su carrito.

**Descripción**

El usuario realiza la compra de los productos en su carrito, el sistema aplica descuentos, muestra el resumen del pedido y genera un informe que se envía al cliente.



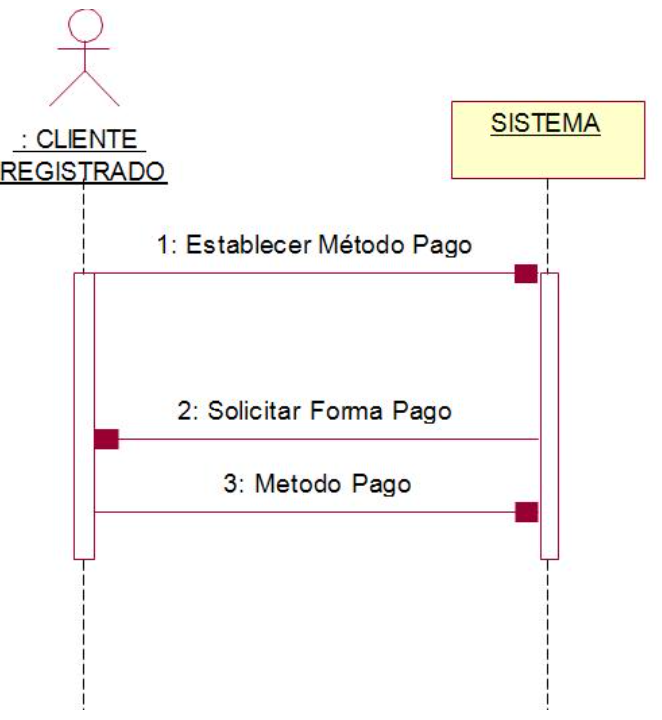
## Pagar

**Objetivo**

Permitir al usuario cliente elegir y guardar un método de pago para realizar sus compras.

**Descripción**

El usuario selecciona un método de pago entre las opciones disponibles, el sistema guarda la información del método de pago elegido por el usuario.

****

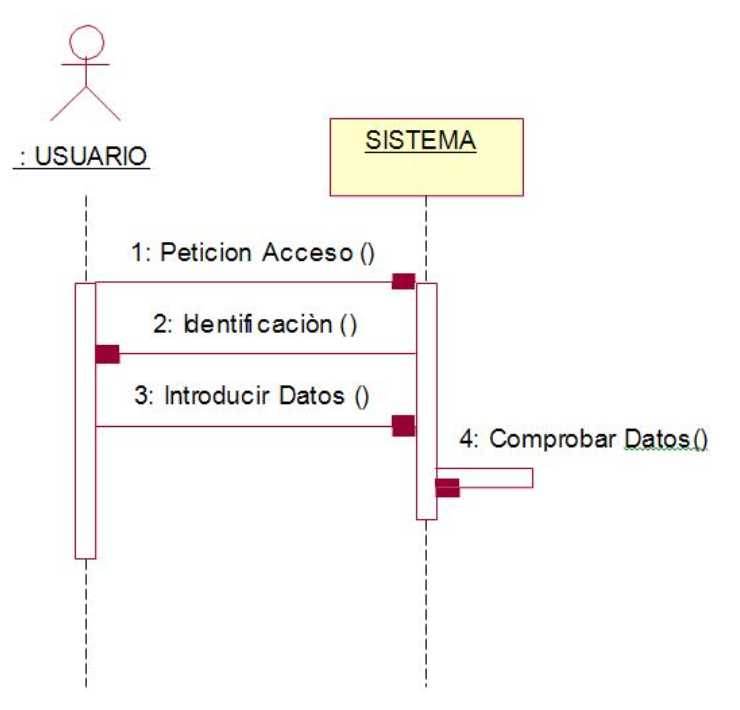
## Validar Usuario

**Objetivo**

Verificar la autenticidad de un usuario cliente para permitirle el acceso a las funcionalidades de la aplicación.

**Descripción**

El usuario realiza una solicitud de acceso, se autentica ante el sistema y el sistema verifica sus credenciales para permitir el acceso.



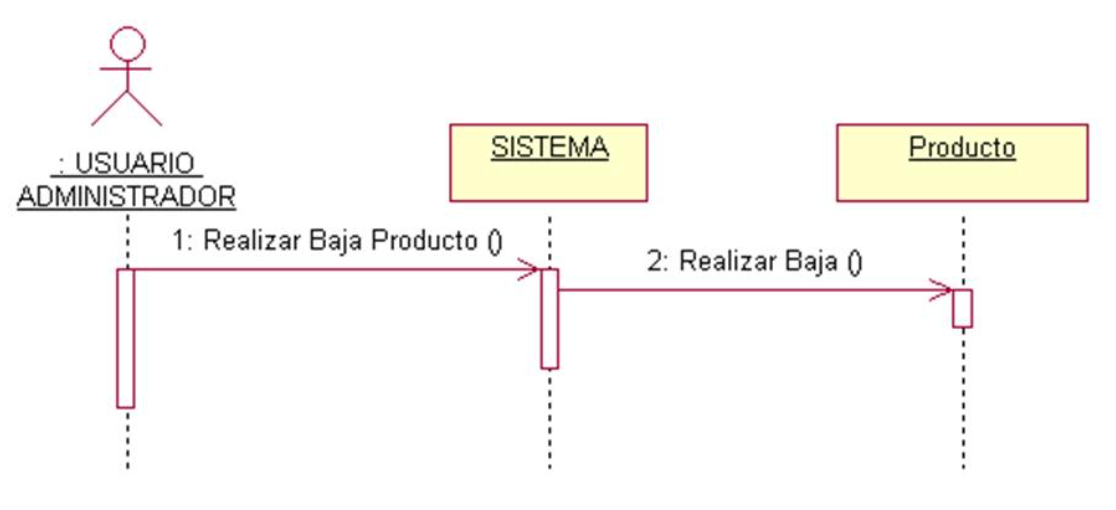
## Gestionar Productos

**Objetivo**

Permitir al usuario administrador dar de alta o baja un producto en el sistema.

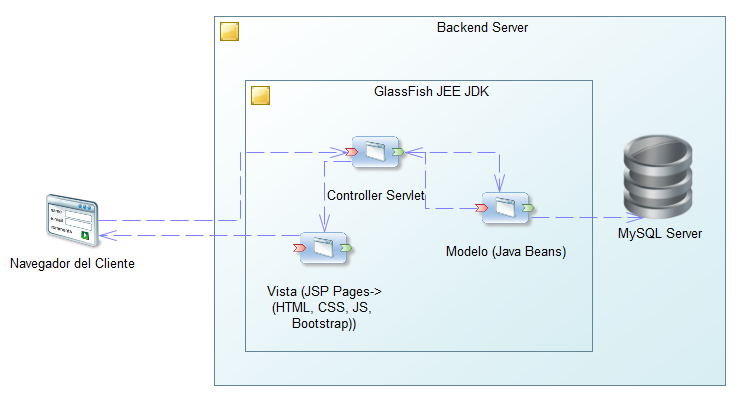
**Descripción**

El usuario administrador realiza la petición correspondiente para dar de alta o baja un producto y el sistema realiza la operación solicitada.



# Vista de Distribución – Deployment

En la vista de la distribución de los componentes y subsistemas en los diferentes nodos tendremos:



Donde se utiliza Java EE (JEE) como la plataforma para el desarrollo de aplicaciones. Esta proporciona un conjunto de especificaciones y APIs para simplificar el desarrollo de aplicaciones web y empresariales. Incluye tecnologías como Servlets, JSP, EJB, CDI, etc.

El servidor de aplicaciones utilizado es GlassFishEs que es un servidor de aplicaciones de código abierto que implementa las especificaciones de Java EE y proporciona el entorno de ejecución para desplegar y ejecutar la aplicación web basada en Java EE.

Como sistema de gestión de bases de datos relacional se usa MySQL para almacenar y gestionar los datos de la aplicación web.

Se utiliza el patrón MVC (Modelo-Vista-Controlador) que separa la lógica de la aplicación en tres componentes principales: el Modelo, la Vista y el Controlador. Este enfoque promueve la modularidad y la reutilización del código.

El Modelo representa los datos y la lógica de negocio de la aplicación. En una aplicación Java EE, los modelos pueden ser representados por objetos Java, beans gestionados por el contenedor, entidades JPA, etc. En este caso, los modelos interactuarían con la base de datos MySQL para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos.

La Vista como la interfaz de usuario de la aplicación, son las páginas JSP (JavaServer Pages) que contienen HTML, CSS, JavaScript y código Java embebido. En este caso, las páginas JSP se utilizan para generar la interfaz de usuario que se mostrará en el navegador web. Bootstrap se utiliza para mejorar el diseño y la usabilidad de estas páginas, ya que proporciona un conjunto de componentes y estilos CSS predefinidos.

El Controlador como responsable de manejar las solicitudes del usuario, procesar la entrada, interactuar con el modelo y actualizar la vista en consecuencia. En esta aplicación web, los controladores serían los Servlets o componentes gestionados por el contenedor. Estos controladores manejarían las solicitudes HTTP, llamarían a los métodos del modelo para realizar operaciones en la base de datos y luego seleccionarían la vista adecuada para mostrar al usuario.

# Referencias

1. Fernández, Luis Felipe. "Arquitectura de software." Software Guru 2.3 (2006). Recuperado de   
   https://ozarate.net/articulos/arquitectura\_sw\_sg\_2006.pdf
2. Reynoso, Carlos Billy. "Introducción a la Arquitectura de Software." *Universidad de Buenos Aires* 33 (2004). Recuperado de https://www.academia.edu/download/64699769/Arquitectura\_software.pdf
3. Castro, Luis. *Arquitectura del Software*. Cengage Learning Editores, 2015. Recuperado de  
   https://upload.wikimedia.org/wikipedia/commons/b/be/AS1-arquitecturas-no-distribuidas.pdf
4. Jimenez-Torres, Victor Hugo, Wilman Tello-Borja, and Jorge Iván Rios-Patiño. "Lenguajes de patrones de arquitectura de software: una aproximación al estado del arte." *Scientia et technica* 19.4 (2014): 371-376.Recuperado de  
   https://www.redalyc.org/pdf/849/84933912003.pdf
5. Cervantes Maceda, Humberto, PERLA VELASCO, and LUIS CASTRO. "Arquitectura de Software." (2016). Recuperado de   
   https://dspace.itsjapon.edu.ec/jspui/bitstream/123456789/3811/1/Arquitectura%20de%20Software%20-%20Humberto%20Cervantes%20Maceda%20%28FreeLibros%29.pdf
6. Copia instantánea de volumen. Recuperado Marzo, 2021, de [https://es.wikipedia.org/wiki/Copia\_instant%C3%A1nea\_de\_volumen](https://es.wikipedia.org/wiki/Copia_instantÃ¡nea_de_volumen)