<<VYCTPC (Venta y compra de tenis PinCop)>>

**Documento de Especificación de Arquitectura**

Realizado por:

Juan David Romero Gordo

Michael Campos

Pablo Andrés Mondragón

Santiago Gómez Estepa

***HISTORIAL DE REVISIONES***

| **Fecha** | **Versión** | **Autor** | **Descripción** | **Revisado Por** |
| --- | --- | --- | --- | --- |
| 14/03/2023 | 1.0 | Pablo Mondragon  Santiago Estepa | Ajustes a la información requerida | Juan Romero  Santiago Estepa  Pablo Mondragon  Michael Campos |
| 16/03/2023 | 1.1 | Michael Campos Juan romero | Ajustes a la información requerida | Juan Romero  Santiago Estepa  Pablo Mondragon  Michael Campos |
| 22/03/2023 | 1.2 | Santiago Estepa  Juan Romero | Ajustes a la información requerida | Juan Romero  Santiago Estepa  Pablo Mondragon  Michael Campos |
| 24/03/2023 | 1.3 | Pablo Mondragon  Michael Campos | Ajustes a la información requerida | Juan Romero  Santiago Estepa  Pablo Mondragon  Michael Campos |
| 25/03/2023 | 1.4 | Juan Romero  Santiago Estepa  Pablo Mondragon  Michael Campos | Ajustes a la información requerida | Juan Romero  Santiago Estepa  Pablo Mondragon  Michael Campos |

**Contenido**

[1.](#_heading=h.3znysh7) Documento de Arquitectura de Software 4

[1.1.](#_heading=h.2et92p0) Introducción 4

[1.2.](#_heading=h.tyjcwt) Propósito 4

[1.3.](#_heading=h.3dy6vkm) Alcance 4

[1.4.](#_heading=h.1t3h5sf) Referencias 4

[1.5.](#_heading=h.4d34og8) Definiciones acrónimos y abreviaciones 4

[2.](#_heading=h.2s8eyo1) Generalidades del Proyecto 5

[2.1.](#_heading=h.17dp8vu) Problema a Resolver 5

[2.2.](#_heading=h.3rdcrjn) Descripción General del Sistema a Desarrollar 5

[2.3.](#_heading=h.26in1rg) Identificación de los Stakeholders y sus responsabilidades 5

[3.](#_heading=h.lnxbz9) Vistas de la arquitectura 5

[3.1.](#_heading=h.35nkun2) Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el S.I.) 5

[3.2.](#_heading=h.1ksv4uv) VISTA DE PROCESOS 5

[3.3.](#_heading=h.44sinio) VISTA LÓGICA 5

[3.4.](#_heading=h.z337ya) VISTA DE IMPLEMENTACIÓN 6

[3.5.](#_heading=h.3j2qqm3) VISTA DE DESPLIEGUE 6

[4.](#_heading=h.1y810tw) Arquitectura en capas 6

[5.](#_heading=h.4i7ojhp) VISTA DE DATOS 6

[5.1.](#_heading=h.2xcytpi) Modelo Relacional normalizado tercera forma normal de la base de datos. 6

[6.](#_heading=h.1ci93xb) Definición de Interfaces de Usuario 6

[7.](#_heading=h.3whwml4) Características Generales de Calidad 6

[7.1.](#_heading=h.2bn6wsx) Tamaño y performance 6

[7.2.](#_heading=h.1pxezwc) Calidad 6

[7.3.](#_heading=h.49x2ik5) Usabilidad 7

[7.4.](#_heading=h.2p2csry) Eficiencia 7

[7.5.](#_heading=h.147n2zr) Seguridad 7

[7.6.](#_heading=h.3o7alnk) Confiabilidad 7

[7.7.](#_heading=h.23ckvvd) Mantenimiento 7

[7.8.](#_heading=h.ihv636) Estándares (Normas de calidad): Mencionar dos o tres normas, y argumentar cómo esas normas se aplican para el sistema de información. 7

# Documento de Arquitectura de Software

## Introducción

Este documento de arquitectura de software está destinado a proporcionar una descripción detallada de la arquitectura del software utilizado por una empresa de calzado. Este documento proporcionará una visión general de la arquitectura de software utilizada por la empresa de calzado, incluyendo los diagramas, la c[alidad](#_heading=h.1pxezwc), u[sabilidad, eficiencia, seguridad y confiabilidad](#_heading=h.49x2ik5).

## Propósito

El propósito de este documento es mejorar la comprensión de cómo funciona y se construye el software de la empresa. A través de este documento, se obtendrá una visión general de los componentes clave de la arquitectura de software, su estructura y patrones de diseño. El objetivo de este documento es proporcionar una comprensión clara y completa de la arquitectura de software de la empresa de calzado, para ayudar a impulsar el éxito y el crecimiento sostenible de la empresa.

## Alcance

El alcance de este documento es proporcionar una visión general del sistema de software y su estructura para ayudar a los desarrolladores a comprender cómo deben construir y mantener el sistema. También puede ser utilizado como una guía para futuras mejoras o actualizaciones del sistema, asegurando la coherencia y continuidad de la arquitectura del software.

## Referencias

1. Documento de historias de usuario.
2. Documento de Visión del Proyecto.
3. Planilla de análisis del software.

## Definiciones acrónimos y abreviaciones

**ARQUITECTURA DE SOFTWARE:** conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño Arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

**DESCRIPCIÓN DE ARQUITECTURA**: colección de productos de documentación.

**VISTAS:** es una representación de un área de interés o perspectiva del sistema en alto nivel.

**TIPOS DE VISTAS:** especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

**STAKEHOLDER:** Individuo, equipo u organización con intereses relativos al sistema.

# Generalidades del Proyecto

## Problema a Resolver

## Descripción General del Sistema a Desarrollar(en general y por módulo).

## Identificación de los Stakeholders y sus responsabilidades

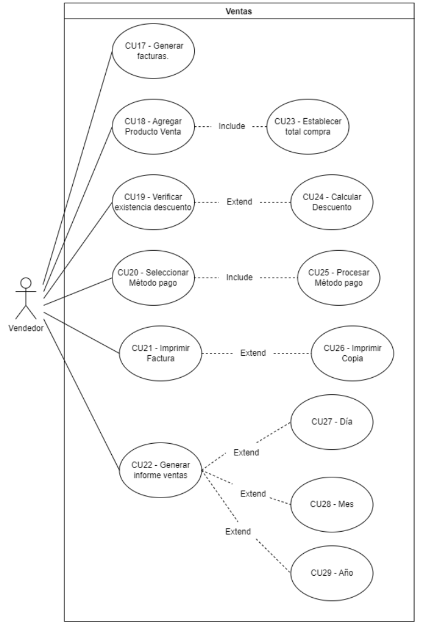
| **STAKEHOLDER** | **DESCRIPCIÓN** | **ESCENARIO** | **Caso de Uso** |
| --- | --- | --- | --- |
| Cliente | Propietario de la tienda de zapatos online | Desea ofrecer a los clientes una experiencia de compra en línea fácil y conveniente. | 1. Crear una cuenta en la tienda en línea. 2. Navegar por el catálogo de zapatos en línea.  3. Seleccionar y agregar zapatos al carrito de compras.  4. Realizar el pago en línea.  5. Realizar un seguimiento del estado de la entrega del pedido. |
| Desarrollador | Persona o equipo responsable del desarrollo del software | Desarrollar y mantener el software de la tienda de zapatos en línea. | 1. Desarrollar la arquitectura del software.  2. Desarrollar y probar las funcionalidades del software.  3. Corregir errores y solucionar problemas de rendimiento. |
| Diseñador | Persona responsable del diseño de la interfaz de usuario (UI) | Crear una interfaz de usuario atractiva, intuitiva y fácil de usar para los clientes. | 1. Diseñar la interfaz de usuario para la tienda en línea.  2. Crear prototipos y maquetas para validar el diseño.  3. Trabajar en conjunto con el equipo de desarrollo para implementar el diseño. |
| Analista de negocios | Persona encargada de analizar el negocio y los requerimientos | Analizar los requerimientos de la tienda de zapatos y traducirlos en requerimientos técnicos. | 1. Realizar el análisis de requerimientos y traducirlos en requerimientos técnicos.  2. Trabajar con el equipo de desarrollo para asegurarse de que los requerimientos se implementen de manera efectiva. |
| Proveedor de pagos | Empresa que proporciona servicios de pago en línea | Procesar pagos en línea de manera segura y confiable. | 1. Proporcionar opciones de pago en línea seguras y confiables.  2. Integrar el servicio de pago en línea con la tienda en línea.  3. Asegurar la privacidad y seguridad de la información del cliente. |
| Proveedor de envíos | Empresa que proporciona servicios de envío y logística | Proporcionar opciones de envío y seguimiento de entregas confiables. | 1. Integrar el servicio de envío y seguimiento de entregas con la tienda en línea.  2. Proporcionar opciones de envío y seguimiento de entregas confiables y eficientes. |

# Vistas de la arquitectura

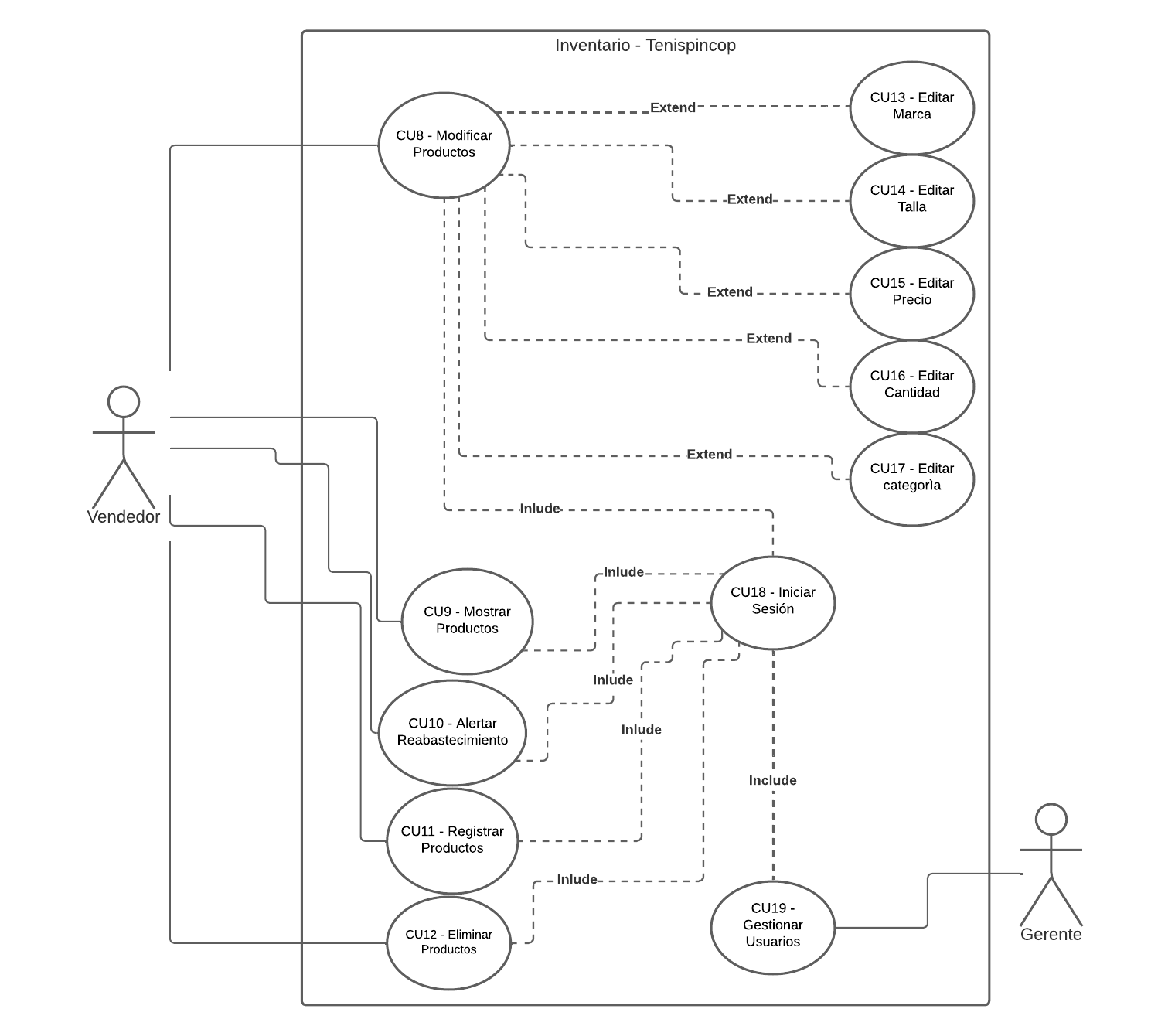
## Vista de Casos de Uso (Se debe colocar cada uno de los casos de uso que han sido elaborados y requeridos para el software)

Para cada caso de uso, se debe argumentar como valor agregado a la comprensión del mismo(imagen y observaciones importante a tener en cuenta)

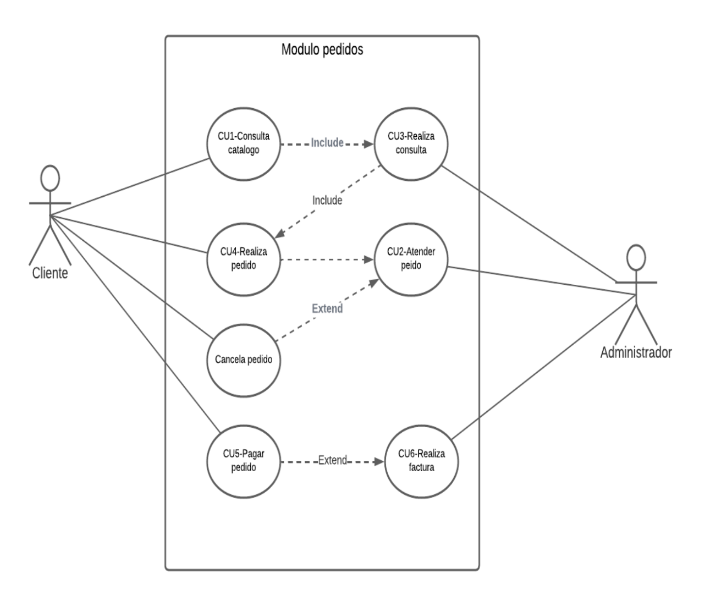
Casos de uso del módulo Ventas



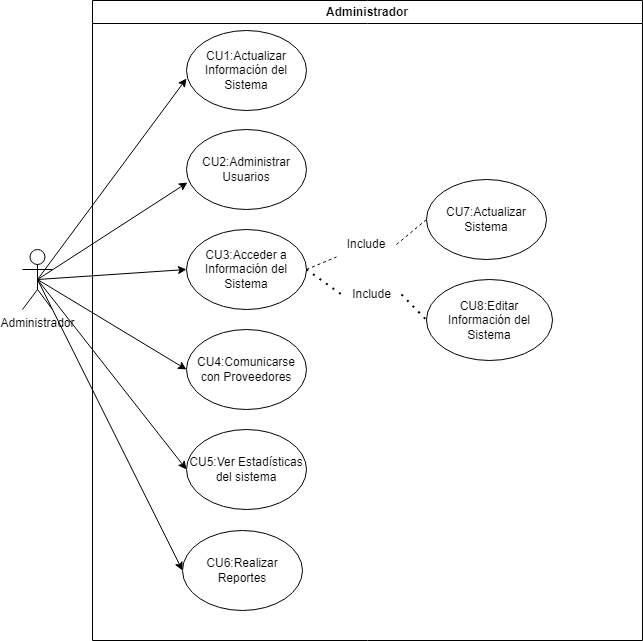
Casos de uso del módulo Inventario



Casos de uso del módulo Pedidos



Casos de uso del módulo Administrador

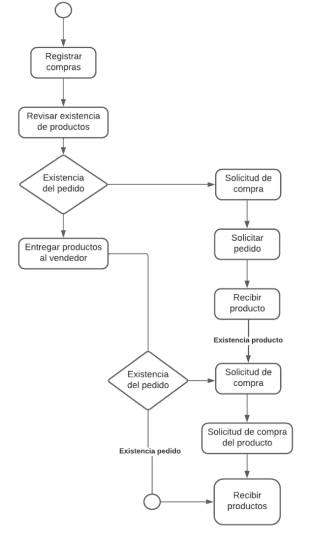


## VISTA DE PROCESOS

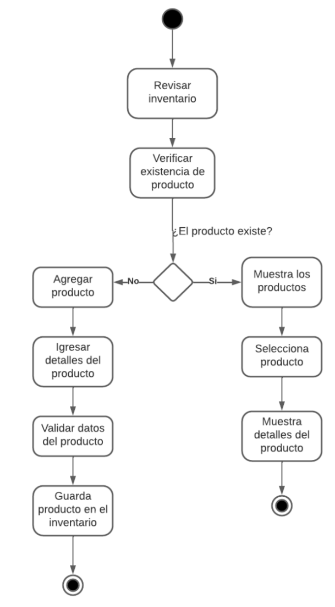
Todos los diagramas que hayan elaborado en fase II, por módulo (aplica para los puntos 3.2.1 al 3.3.2), debidamente argumentados cada uno de los diagramas

### Diagrama de Actividades

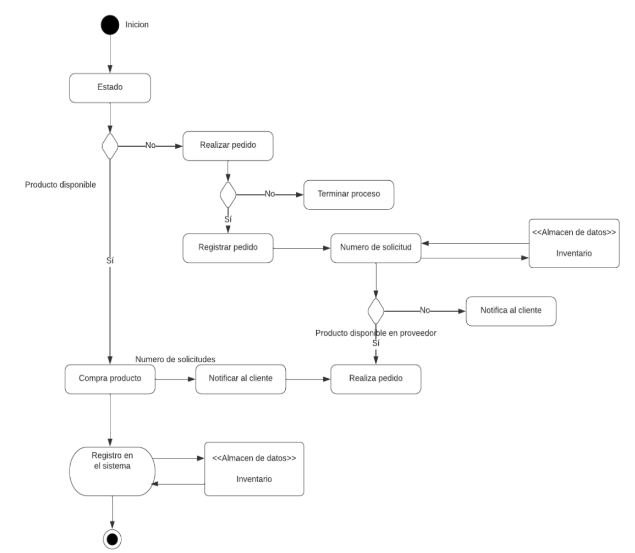
Diagramas de actividades del módulo ‘ventas’



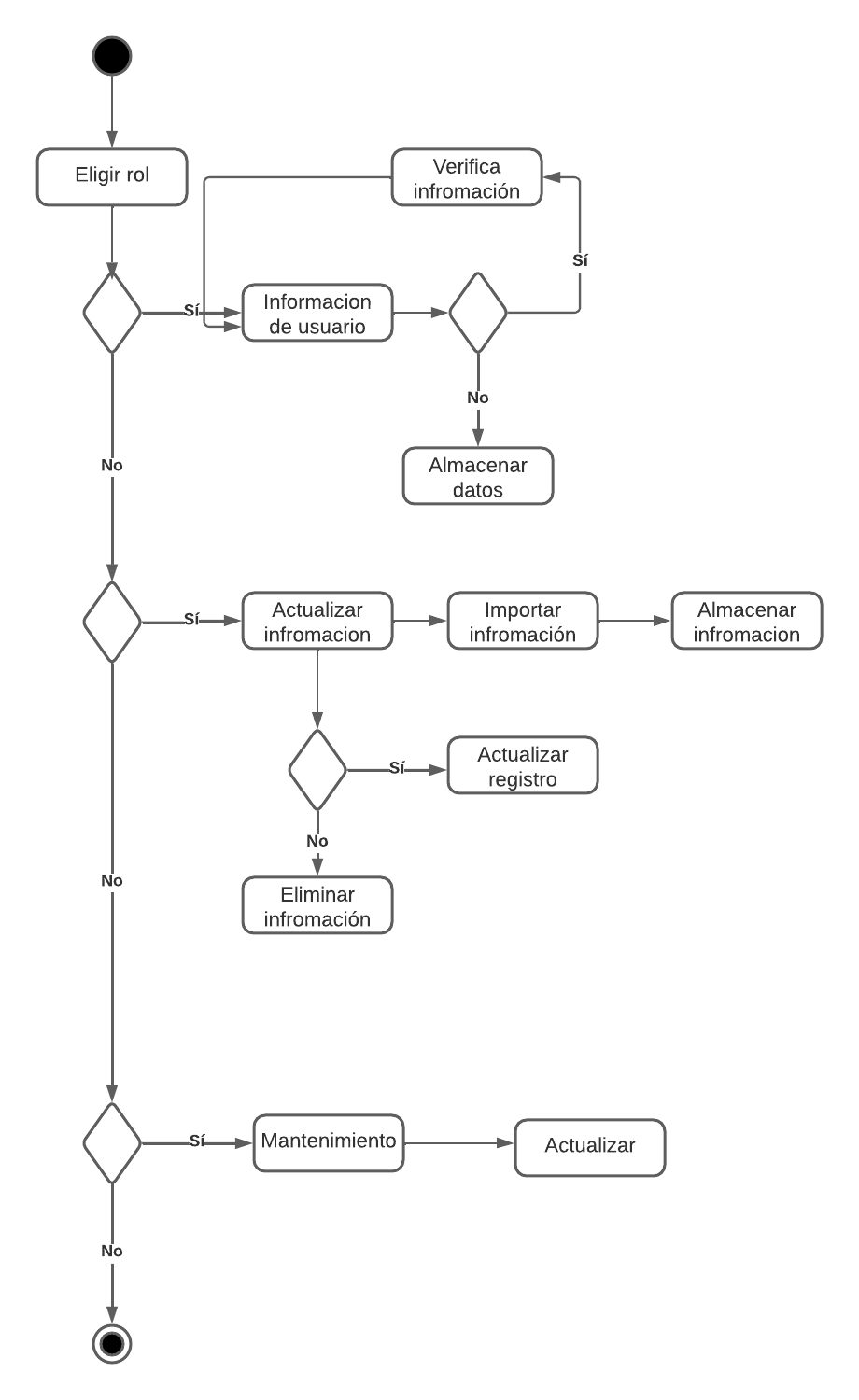
Diagramas de actividades del módulo ‘inventario’



Diagramas de actividades del módulo ‘Pedidos’

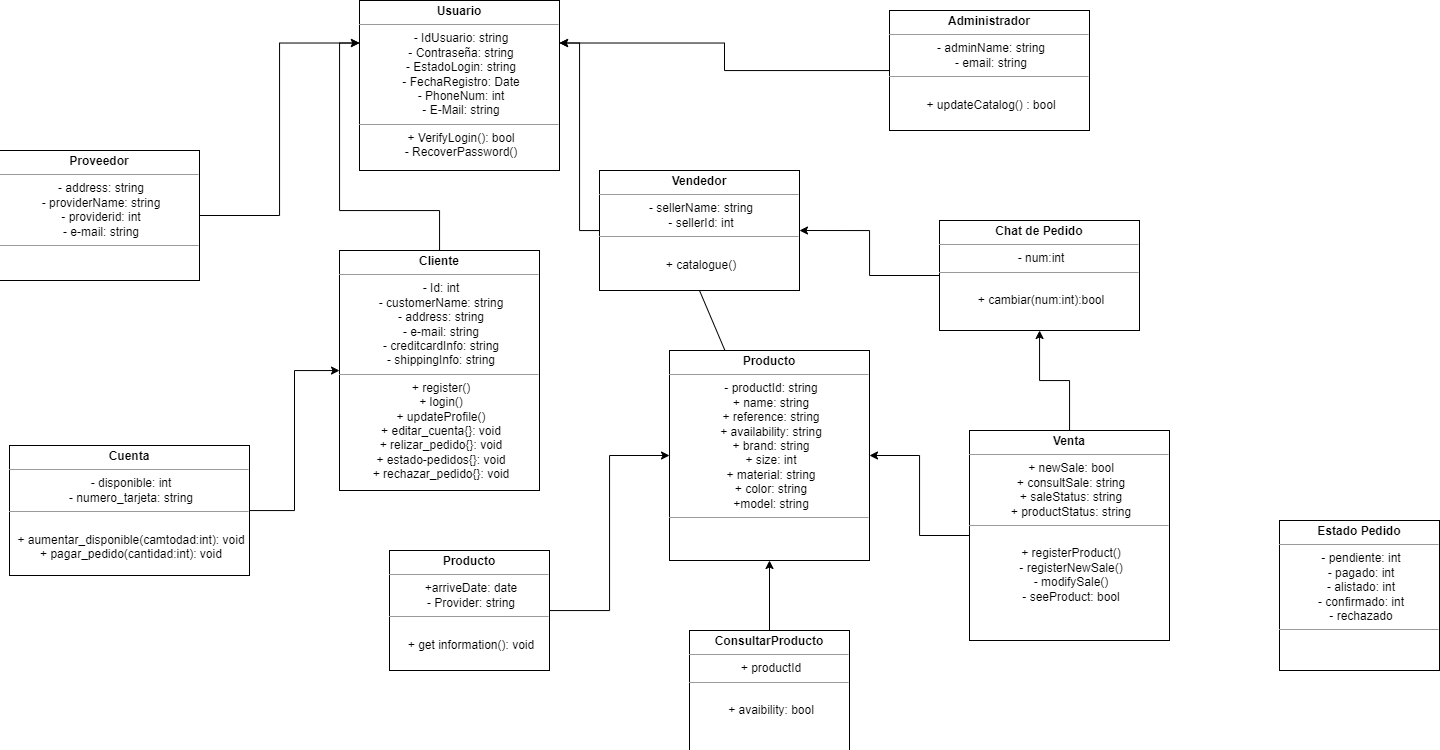


Diagramas de actividades del módulo ‘Administrador’



## VISTA LÓGICA

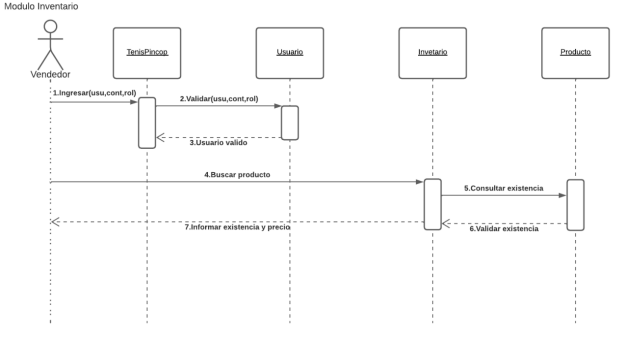
### Diagramas - Clases(Actualizado)



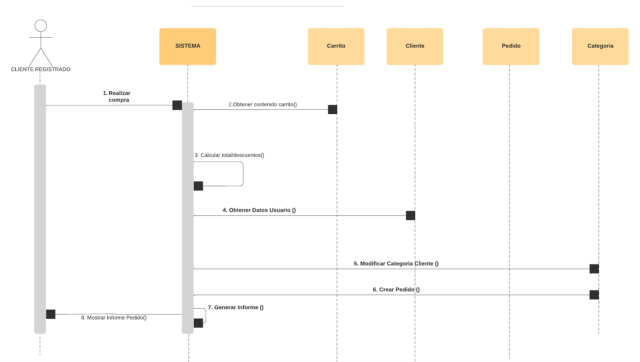
### Diagramas - Secuencia

Los diagramas de los puntos 3.4 y 3.5.1 corresponden a los diagramas que se han elaborado en la fase III

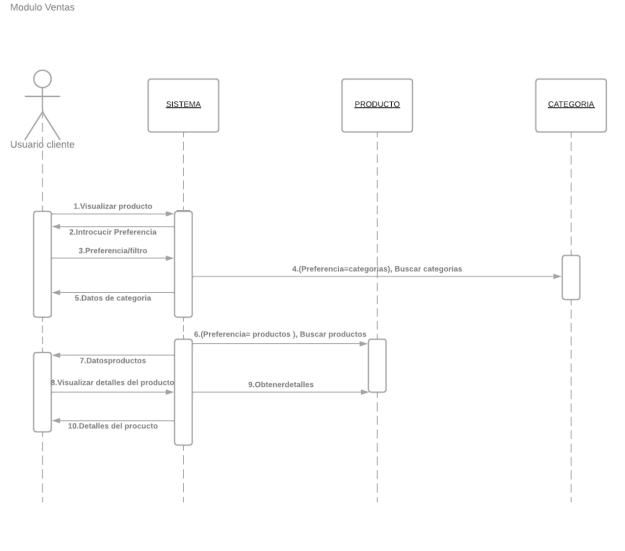
Módulo Inventario

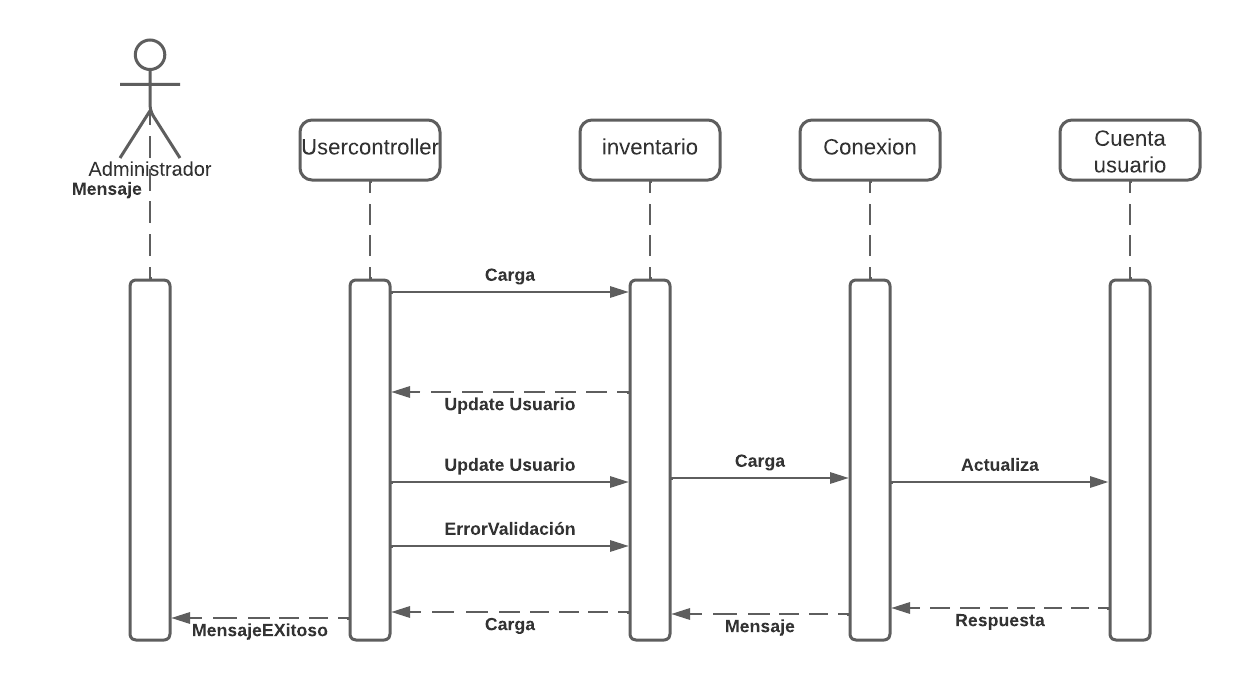


Módulo Pedidos



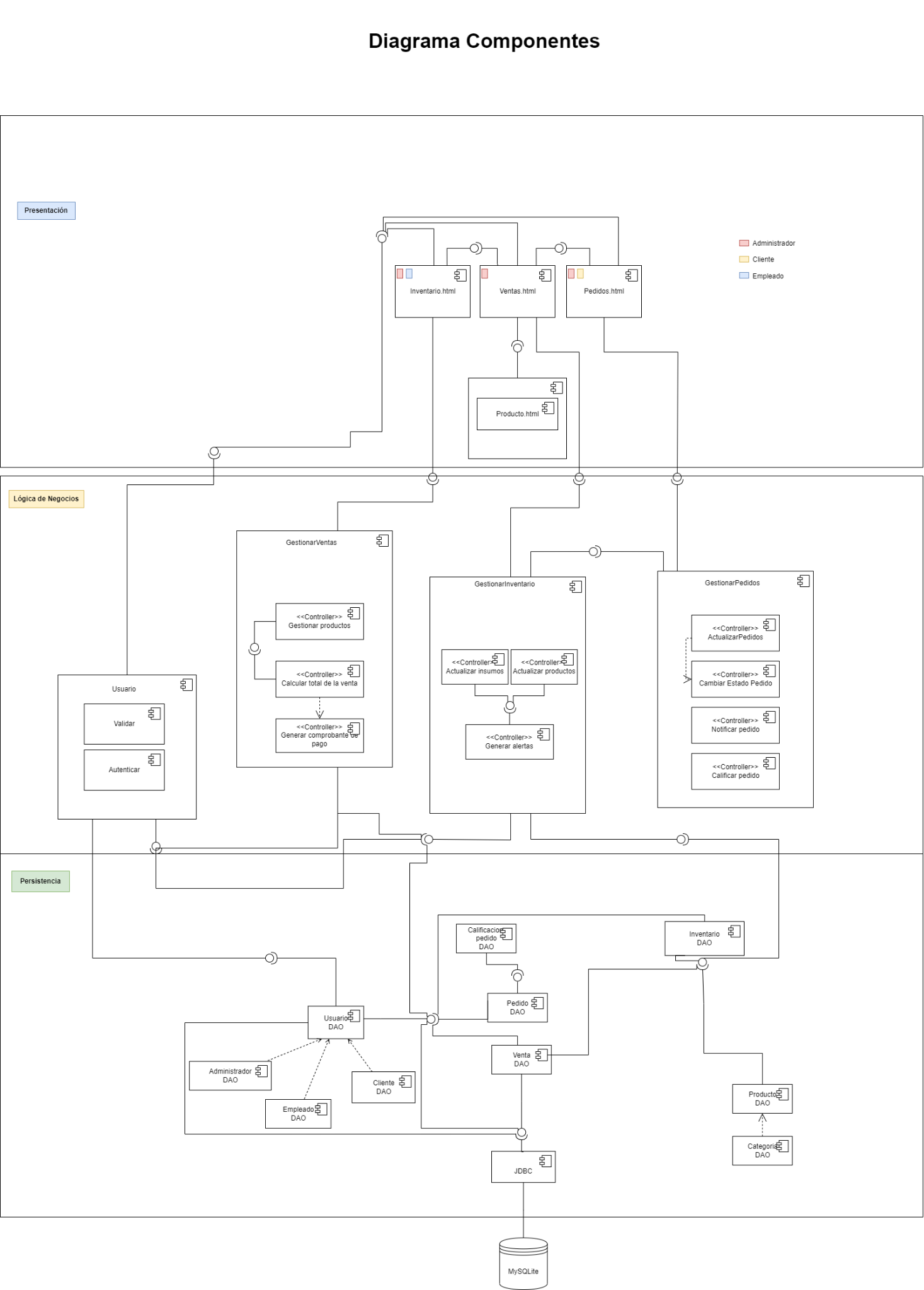
Módulo Ventas



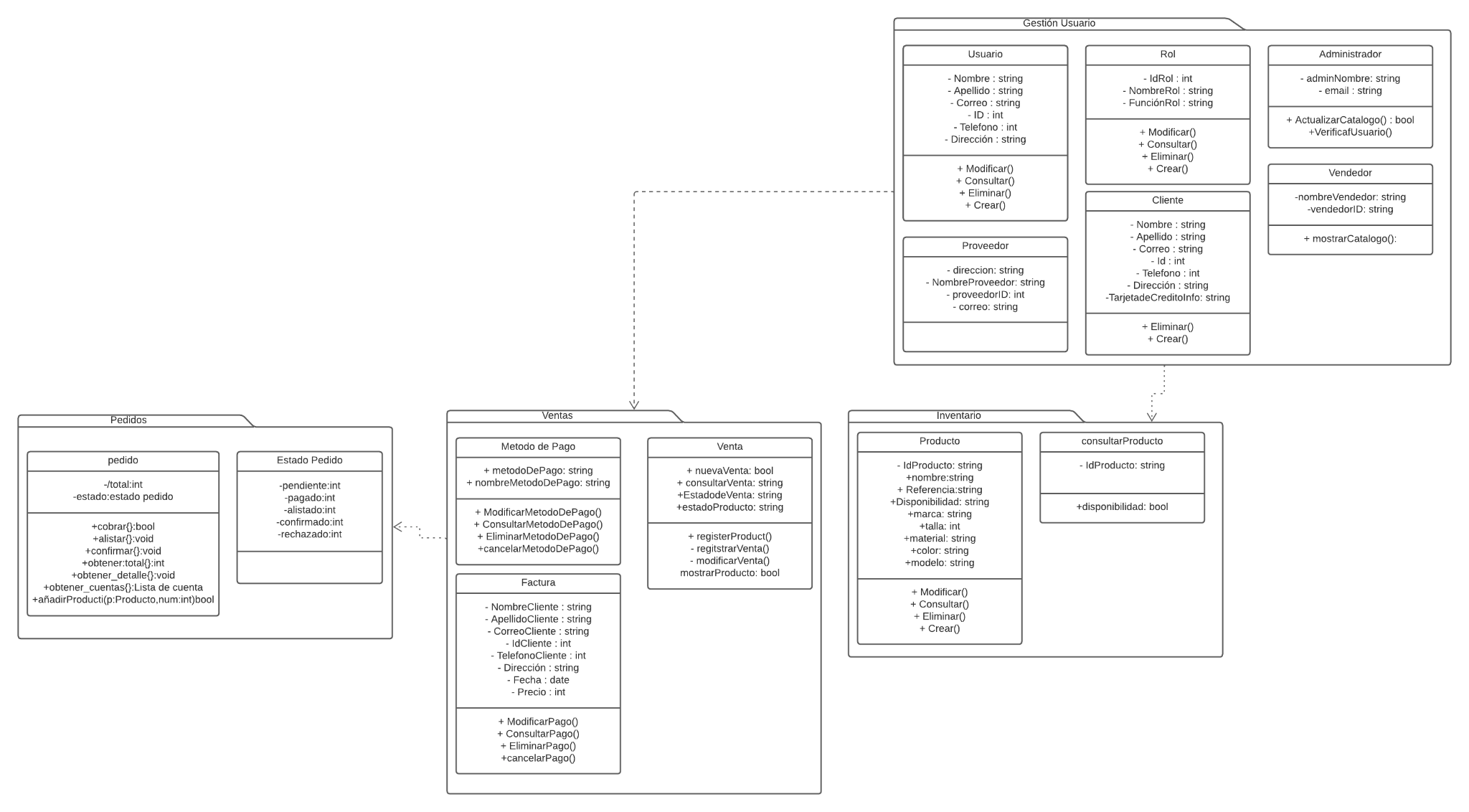
MóduloAdministrador

## VISTA DE IMPLEMENTACIÓN

### Diagrama de Componentes

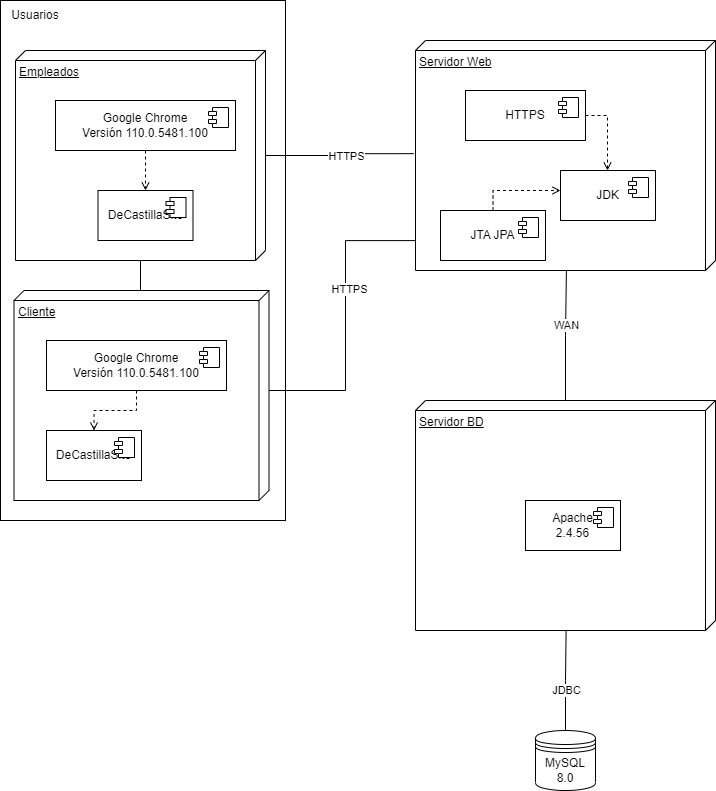


### Diagrama de Paquetes



## VISTA DE DESPLIEGUE

### Diagrama de despliegue

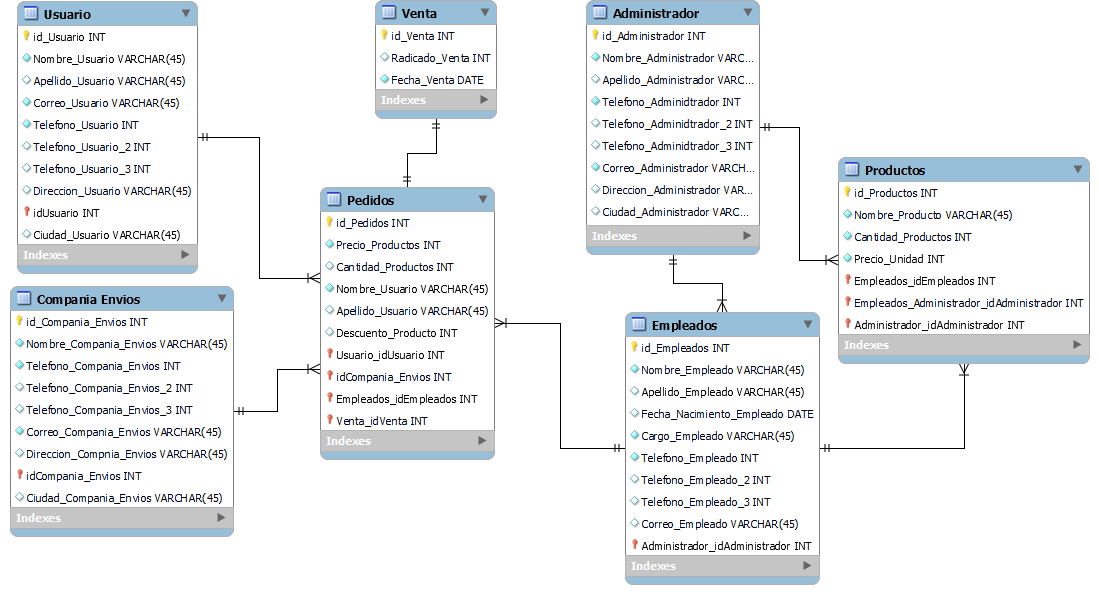


# Arquitectura en capas

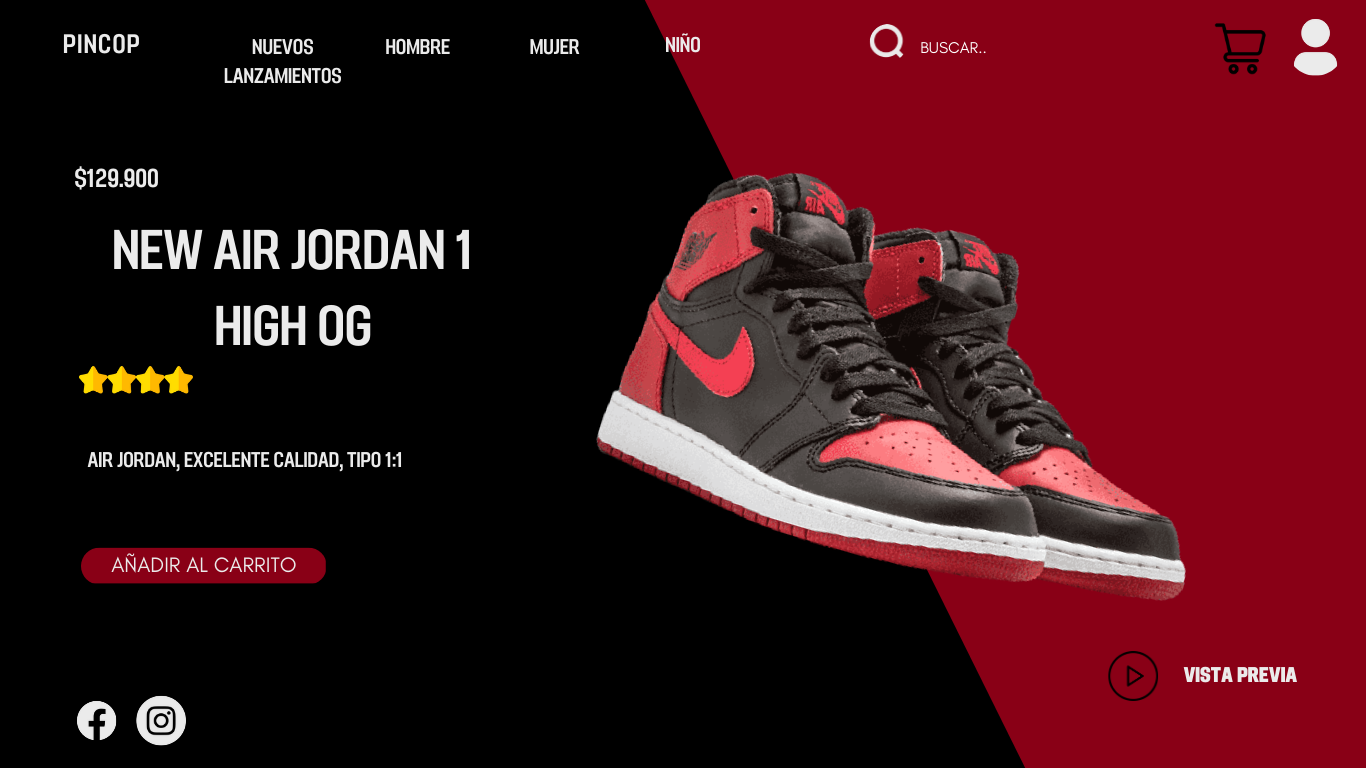
| **Capa** | **Tecnologías a usar** |
| --- | --- |
| Presentación | HTML, CSS, JavaScript |
| Lógica de negocio | Java |
| Persistencia | DAO, JDBC |

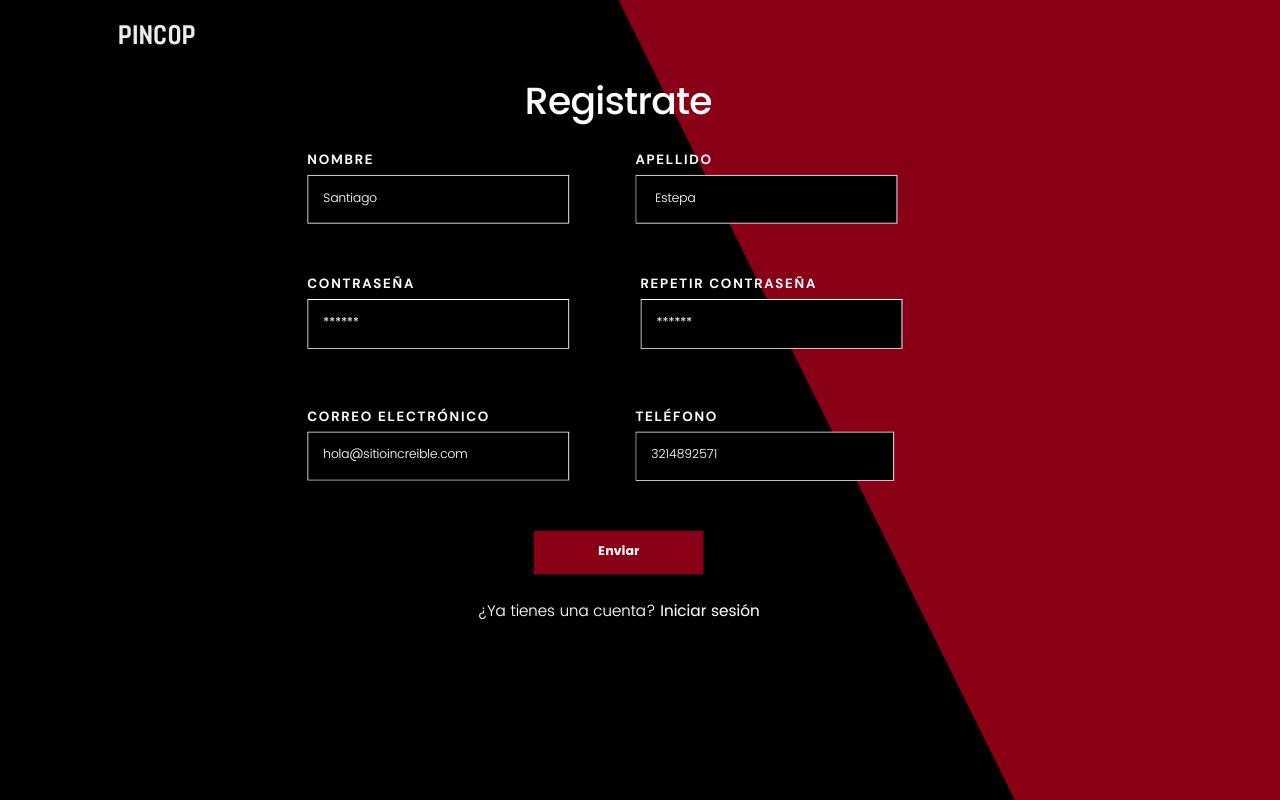
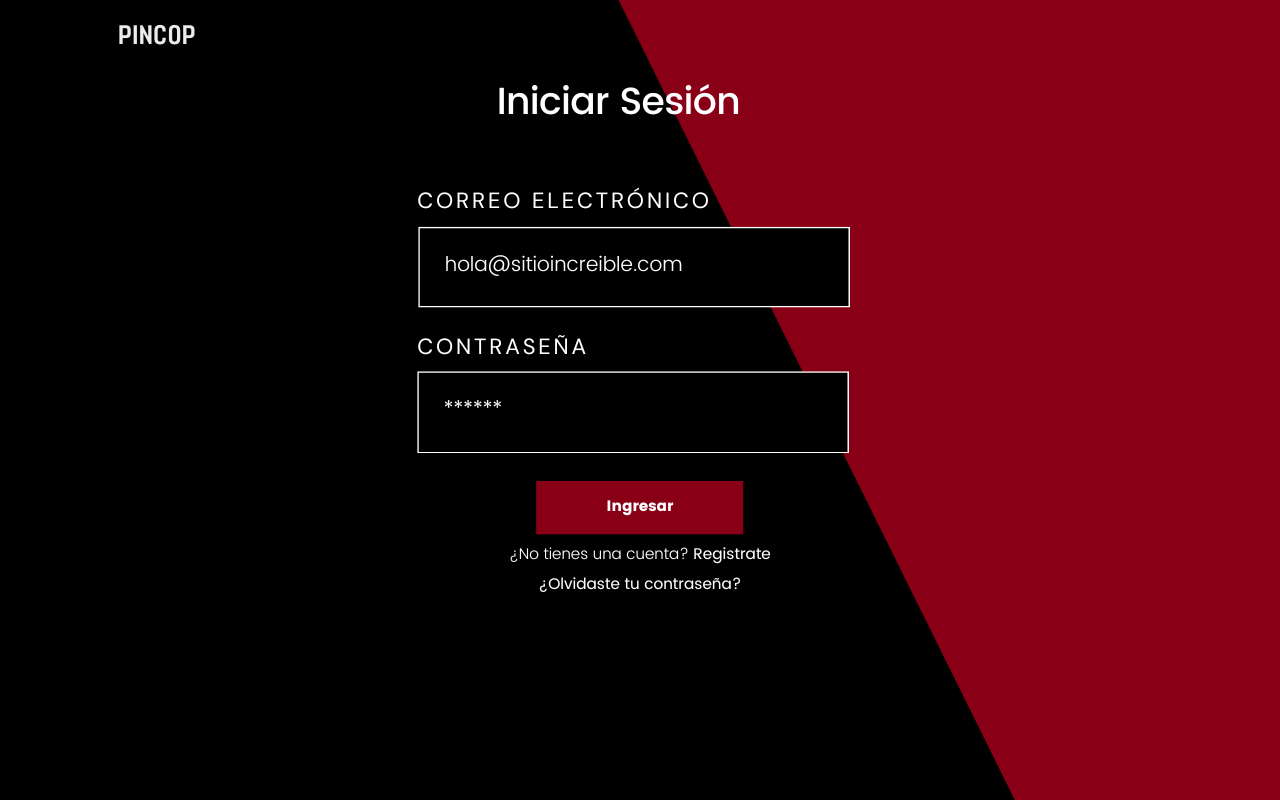
# VISTA DE DATOS

## Modelo Relacional normalizado tercera forma normal de la base de datos (SGBD).



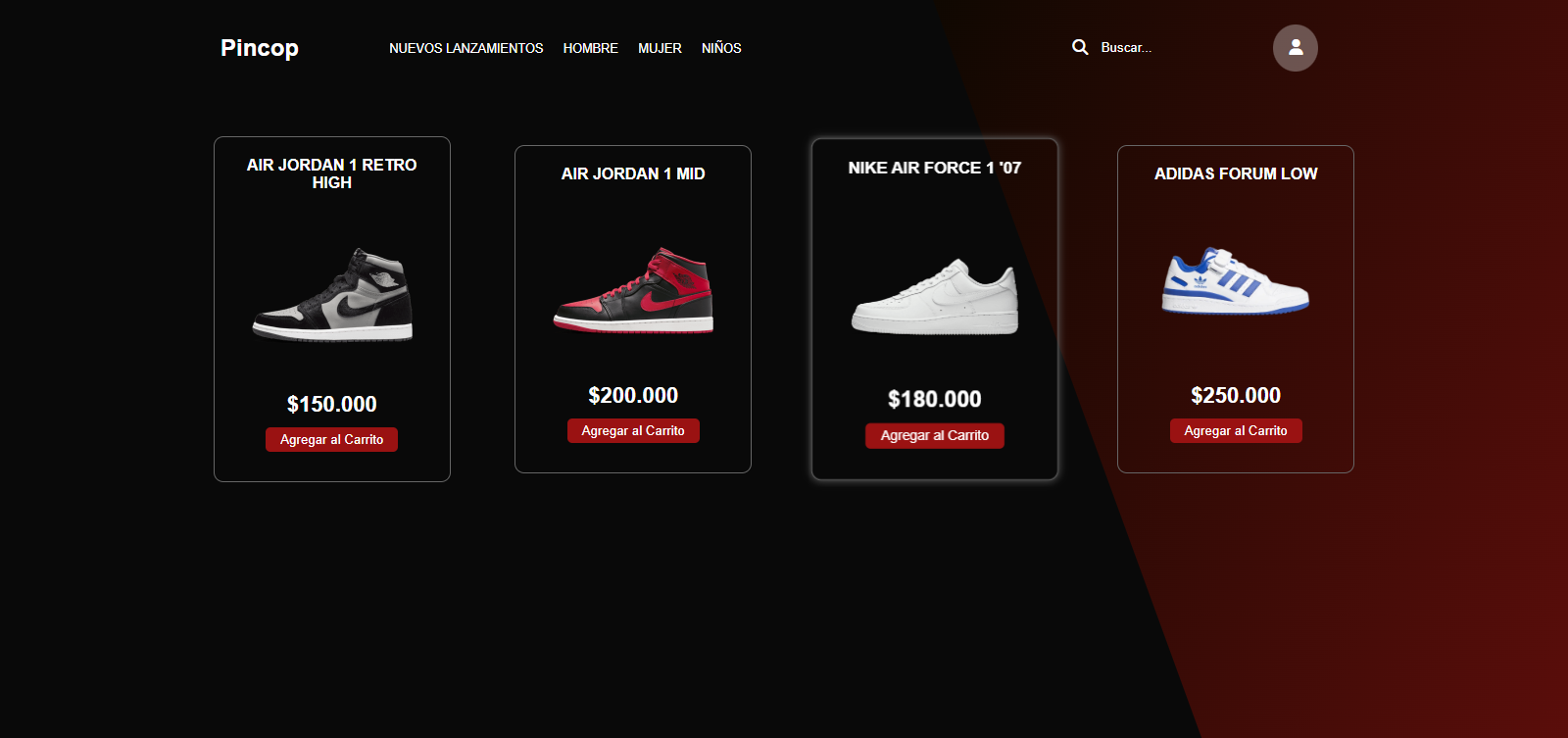
# Definición de Interfaces de Usuario



****

****

****

****

# Características Generales de Calidad

## Tamaño y performance

* Tiempo de respuesta en el acceso a la Base de Datos:
* Tiempo de respuesta de transacciones:
* Espacio en disco para el cliente:
* Espacio en disco para el servidor de Base de datos:

Puntos 7.2. al 7.7., se debe argumentar por cada atributo de calidad, como se cumple el mismo para el Software.

## Calidad: Capacidad del software para cumplir los requisitos y expectativas del usuario de manera eficiente y efectiva. Es crucial para garantizar que el software cumpla con los requisitos del usuario y tenga un impacto positivo en el negocio.

## Usabilidad: Fácil de usar y comprender para el usuario. Un software con una buena usabilidad puede tener una serie de beneficios para el usuario y para el negocio como: Mayor satisfacción al usuario, mayor productividad y la reducción de errores y costos.

## Eficiencia: Utilizar de manera efectiva los recursos disponibles, como el tiempo, el espacio de memoria y el poder de procesamiento. Esto es importante ya que puede proporcionar una mejor experiencia de usuario, ser más rápido y consumir menos recursos, lo que puede ser especialmente importante en entornos con recursos limitados.

## Seguridad: Proteger datos y recursos del usuario y del sistema en general, de posibles ataques y vulnerabilidades. Es importante que se implementen medidas de seguridad adecuadas durante todo el ciclo de vida del software, desde la fase de diseño hasta la fase de mantenimiento. Entre las medidas que se pueden implementar se incluye: Autenticación y autorización, encriptación, pruebas de seguridad, actualizaciones de seguridad y la restricción de acceso a los recursos de sistema.

## Confiabilidad: Realizar sus funciones de manera consistente y sin errores, incluso en situaciones críticas o en condiciones adversas. Implementar prácticas de desarrollo y pruebas rigurosas para garantizar que el software funcione de manera confiable en todas las situaciones. Entre las medidas que se pueden dar para garantizar una confiabilidad se incluyen: Pruebas de software, monitoreo constante, respaldo de datos y las actualizaciones de software.

## Mantenimiento: Capacidad de un software para ser mantenido y mejorado de manera efectiva de manera eficiente y efectiva a lo largo del tiempo, a medida que surjan nuevas necesidades y requisitos. Entre las medidas que se pueden implementar para garantizar el mantenimiento del software se incluyen: Diseño modular, documentación adecuada, pruebas de regresión, control de versiones y actualizaciones regulares.

## Estándares (Normas de calidad): Mencionar dos normas de calidad de Desarrollo de Software y argumentar cómo esas normas aplican para el software.

**La norma ISO 25010:** Es una norma internacional que establece los requisitos y criterios de calidad para el software y los sistemas de información. Esta norma se centra en la calidad interna y externa del software y proporciona un marco de referencia para evaluar y mejorar la calidad del software.

**La norma ISO 25012:** En el desarrollo de software ayuda a garantizar que los datos utilizados en los productos de software y servicios sean de alta calidad y cumplan con los requisitos del usuario. La evaluación de cada una de las ocho características mencionadas anteriormente.