

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES CARRERA DE
INFORMÁTICA



SISTEMA DE GESTIÓN DE VENTAS E INVENTARIO PARA CHIRK

S.R.L.

Estudiante: - Cristhian Andres Escobar Herrera

- Cristian Gonzalo Mamani Espinoza
- Esther Mayerly Carvajal Quispe
- Sarahi Nicol Mamani Gutierrez
- Jhon Vargas Quispe

Carrera: Informática

Docente: Lic. Gladys Vargas Herrera

Materia: Ingeniería de Software

La Paz - Bolivia

2025

1. CAPÍTULO I

MARCO INTRODUCTORIO

CHIRK S.R.L. es una destacada tienda de ropa situada en Bolivia, reconocida por ofrecer productos textiles de alta calidad que reflejan las tendencias de moda más actuales. La empresa combina operaciones tradicionales en su tienda física con un moderno canal de ventas en línea, brindando a sus clientes la flexibilidad de adquirir productos desde cualquier lugar. Su enfoque principal está en proporcionar una experiencia de compra eficiente, ágil y personalizada, lo que la ha posicionado como una de las opciones preferidas por los consumidores en el mercado nacional.

A través de los años, CHIRK S.R.L. ha enfrentado el reto de mantener la eficiencia operativa a medida que su catálogo de productos y su base de clientes han crecido exponencialmente. Con una visión de futuro clara, la empresa busca implementar soluciones tecnológicas que le permitan consolidarse como un líder en el sector retail, optimizando procesos como la gestión de inventarios, ventas y promociones, mientras garantiza la satisfacción del cliente en todos sus puntos de contacto.

1.1. Antecedentes

1.1.1. Antecedentes internacionales

Estos Software que a continuación se mencionan, cuya función es la administración de movimientos internos de material en el almacén (entradas, salidas, expediciones y ubicaciones) manejado por técnicas de inventario con FIFO, FEFO, LIFO, etc. El sistema es un conjunto de herramientas que posee dos mecanismos de optimización: uno dedicado a optimizar el espacio de almacenaje mediante una adecuada gestión de ubicaciones y otro destinado a optimizar los movimientos o flujos de material, bien sean estos realizados por máquinas o por personas.

- “ContaBit” que ofrece un sistema de gestión contable y de almacenes basados en la nube permitiendo a sus clientes tener acceso a la información en el instante, brindando seguridad dentro de servidores alojados estratégicamente para el acceso a la información de manera efectiva, esta empresa tiene como clientes a la

Universidad Adventista de Bolivia, Quántica y Barion Ltda ambos de la ciudad de Cochabamba.

- “Epicor”, basado en estándares internacionales con más de 40 años de experiencia brinda a sus clientes un software en la nube, alojadas o en las instalaciones. A medida que el software recolecta información, gestiona la complejidad, aumenta la eficiencia y liberan recursos para centrarse en el crecimiento del cliente.

1.1.2. Antecedentes de proyectos anteriores

- “Control de ventas e inventario para el monitoreo de pedidos CASO: Empresa distribuidora VMCC”; desarrollado para el control de ventas e inventarios, con un objetivo claro, distribuir de manera eficiente los pedidos realizados por la empresa VMCC, desarrollado con la metodología XP, y haciendo uso de lenguajes de programación como php, JavaScript y utilizando un gestor de Base de Datos MySQL. (Cruz, 2013).
- “Sistema de Distribución, Ventas y Control de Inventarios para SOBOLMA LTDA”, plantea el diseño e implementación del sistema de control de inventarios, distribución y ventas mediante inventarios perpetuos, para desarrollar este proyecto se utilizó la metodología R.U.P. (Condori, 2007).
- “Sistema de Control de ventas e inventarios para almacenes de aluminio utilizando dispositivos móviles”, plantea el diseño e implementación de un sistema de control de ventas e inventarios, mediante el modelo de inventarios sin déficit, para el presente proyecto se utilizó la metodología de Scrum en su desarrollo
- “Propuesta de un sistema de control de inventario de Stock de seguridad para mejorar la gestión de compras de materia prima, repuestos e insumos de la empresa Balgres C.A.”, Se propuso un software administrativo con la finalidad de utilizar el modelo de control de inventario, stock de seguridad que facilita el sistema y con ello mantener un registro más estricto de las entradas y salidas de los productos albergados en los distintos almacenes que manipula Balgres C.A. (Universidad Simon Bolivar, Ysabel L. Cabriles G.,2014).

1.2. Planteamiento del Problema

La falta de un sistema automatizado en CHIRK S.R.L. genera diversas problemáticas que impactan tanto en las operaciones internas como en la satisfacción del cliente. Entre las principales dificultades se encuentran:

- Errores en la gestión de inventarios: La administración manual de los inventarios provoca inconsistencias en los datos, lo que puede derivar en desabastecimientos o sobre existencias de productos.
- Desconexión entre canales de venta: Las ventas realizadas en la tienda física y en la plataforma virtual no están sincronizadas, lo que genera problemas para mantener un control unificado.
- Dificultades en la administración de cupones y promociones: La falta de un sistema que registre y controle los cupones promocionales impide aprovechar al máximo esta herramienta de marketing.
- Reportes financieros y operativos ineficientes: La generación de informes es lenta y poco precisa, lo que afecta la capacidad de tomar decisiones informadas en tiempo real.
- Falta de alertas para la gestión del stock: No se cuenta con un sistema de alarmas que notifique cuando un producto está cerca de agotarse, lo que lleva a pérdidas de ventas potenciales.
- Estas problemáticas no solo generan ineficiencias internas, sino que también afectan la reputación de la empresa al no cumplir con las expectativas de los clientes.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar e implementar un sistema integral de gestión para CHIRK S.R.L. que optimice las operaciones de inventario, ventas, almacenes y cupones promocionales, garantizando una experiencia de compra eficiente y satisfactoria para los clientes.

1.3.2. Objetivos Específicos

- Automatizar el registro y la actualización del inventario: Garantizar que cada transacción de venta o compra se refleje automáticamente en el sistema.

- Sincronizar los canales de venta: Integrar la tienda física y la plataforma virtual para mantener un control unificado de inventarios y ventas.
- Implementar un módulo de gestión de cupones: Permitir la creación, administración y seguimiento del uso de cupones promocionales.
- Generar reportes en tiempo real: Proveer información precisa y oportuna sobre ventas, inventarios y promociones para apoyar la toma de decisiones.
- Diseñar un sistema de alertas: Notificar al personal sobre niveles bajos de inventario para evitar desabastecimientos.
- Facilitar la experiencia de compra virtual: Asegurar que los clientes puedan realizar compras en línea de manera rápida y segura.

1.4. Justificación

En un mercado competitivo como el de la moda, donde la satisfacción del cliente es clave, la implementación de un sistema de gestión automatizado permitirá a CHIRK S.R.L.:

- Mejorar la eficiencia operativa: Reducir errores humanos y optimizar los procesos relacionados con inventarios y ventas.
- Incrementar la satisfacción del cliente: Garantizar que los productos estén siempre disponibles y que las promociones sean claras y fáciles de usar.
- Aumentar las ventas: Al ofrecer una experiencia de compra ágil y aprovechar al máximo las estrategias promocionales.
- Tomar decisiones estratégicas: Basarse en datos precisos y actualizados para planificar estrategias comerciales y logísticas.
- Estar a la vanguardia tecnológica: Adaptarse a las demandas del comercio electrónico, ampliando su alcance y visibilidad en el mercado.

1.5. Alcances y Límites

1.5.1. Alcances

El sistema propuesto incluirá las siguientes funcionalidades:

- Control de inventarios en tiempo real: Registro automático de entradas y salidas de productos.
- Gestión de ventas integradas: Sincronización de datos entre la tienda física y la plataforma virtual.

- Administración de promociones y cupones: Creación, seguimiento y redención de cupones promocionales.
- Generación de reportes detallados: Informes sobre ventas, inventarios, promociones y tendencias de compra.
- Sistema de alertas: Notificaciones automáticas para productos con bajo nivel de inventario.
- Facilidad de uso: Interfaces intuitivas y amigables tanto para el personal como para los clientes.

1.5.2. Límites

El sistema estará sujeto a las siguientes restricciones:

- Control limitado al inventario y ventas: No incluirá módulos para la gestión de recursos humanos o contabilidad avanzada.
- Implementación inicial restringida: Se limitará a la tienda principal y a la plataforma de comercio electrónico.
- Moneda local: Todas las transacciones serán procesadas en bolivianos (Bs) sin soporte para conversiones automáticas.

2. CAPÍTULO II

MARCO TEÓRICO

En este capítulo, se presenta el marco teórico que sustenta el desarrollo del sistema de gestión de ventas e inventario para la empresa CHIRK S.R.L. Se abordan conceptos clave relacionados con la gestión de inventarios, sistemas de ventas y las tecnologías utilizadas en el desarrollo del proyecto.

2.1. Gestión de Inventarios

La gestión de inventarios es una función esencial en cualquier empresa que maneje productos físicos. Consiste en supervisar y controlar las existencias de productos para garantizar una disponibilidad óptima, evitando tanto el exceso como la escasez de stock. Una gestión eficiente del inventario contribuye a la reducción de costos operativos y mejora la satisfacción del cliente al asegurar la disponibilidad de productos.

2.1.1. Métodos de Control de Inventario

Existen diferentes formas de manejar el inventario en una empresa, pero las más comunes son:

- **Inventario Perpetuo:** Se actualiza automáticamente cada vez que un producto entra o sale del almacén. Esto permite tener siempre datos actualizados sobre las existencias.
- **Inventario Periódico:** Se revisa manualmente cada cierto tiempo para verificar el stock disponible. Aunque es menos costoso, no es tan preciso como el sistema perpetuo.

En este proyecto, se utilizará el método de inventario perpetuo para ofrecer información en tiempo real tanto a los administradores como a los clientes en la tienda online.

2.2. Sistemas de Ventas

Los sistemas de ventas son herramientas que facilitan la administración de las transacciones y el seguimiento de los pedidos. Un sistema bien diseñado permite no solo vender

productos de manera eficiente, sino también conocer mejor a los clientes, ofrecerles productos que realmente necesitan y mantener un control sobre las operaciones comerciales.

En CHIRK S.R.L., el sistema de ventas integrará funciones como:

- Filtros para buscar productos por fabricante, género y categoría.
- Opciones de compra online.
- Almacenamiento de imágenes de los productos en la nube, para reducir la carga en la base de datos.

2.3. Importancia de los Filtros en la Tienda Online

En una tienda online, los clientes suelen buscar productos específicos. Para facilitar esta experiencia, el sistema incluirá filtros que permitan:

- Buscar productos según su **fabricante**.
- Seleccionar productos según su **categoría** (camisas, vestidos, etc.).
- Identificar productos según su **uso** (masculino o femenino).

Esto no solo mejora la experiencia del cliente, sino que también aumenta la probabilidad de ventas al facilitar la navegación en la tienda online.

2.4. Propuesta de Valor del Sistema

La propuesta de valor del sistema de gestión de ventas e inventario para CHIRK S.R.L. se basa en dos elementos clave:

- **Almacenamiento de imágenes de productos en la nube**
- **Búsqueda avanzada por filtros en la tienda online**

Ambos componentes se integran para ofrecer una solución tecnológica eficiente, moderna y alineada con las expectativas del mercado actual. A continuación, se detalla cómo estos elementos aportan valor al negocio y mejoran la experiencia del cliente.

2.4.1. Almacenamiento de Imágenes en la Nube

El almacenamiento de imágenes en la nube implica que las fotografías de los productos no se guardan directamente en la base de datos del sistema, sino que se alojan en servidores externos especializados. Este enfoque tiene varios beneficios clave:

a) Optimización del rendimiento del sistema:

Al mantener la base de datos enfocada exclusivamente en la gestión de información estructurada (nombres, precios, categorías, stock, etc.), se mejora considerablemente la velocidad y la capacidad de respuesta del sistema. Esto garantiza que tanto los empleados como los clientes puedan acceder rápidamente a la información que necesitan.

b) Escalabilidad:

Conforme la empresa amplíe su catálogo de productos, no habrá preocupaciones por la capacidad de almacenamiento. Los servicios de nube permiten aumentar el espacio disponible sin afectar el funcionamiento del sistema.

c) Seguridad y respaldo de datos:

La nube ofrece mayores niveles de seguridad frente a pérdidas de información causadas por fallos locales. Los proveedores de servicios en la nube garantizan la disponibilidad de las imágenes a través de múltiples copias de seguridad en servidores redundantes.

d) Acceso global:

Los empleados y clientes pueden visualizar las imágenes de los productos desde cualquier lugar y en cualquier dispositivo, siempre que tengan acceso a Internet. Esto es especialmente importante para la tienda online de CHIRK S.R.L., donde las fotografías desempeñan un papel crucial en la decisión de compra.

2.4.2. Búsqueda Avanzada por Filtros en la Tienda Online

La tienda online de CHIRK S.R.L. incorpora un sistema de búsqueda avanzada que permite a los clientes encontrar productos de manera rápida y eficiente. Este sistema se basa en la implementación de filtros de búsqueda que clasifican los productos según criterios relevantes para el cliente.

a) Filtros disponibles:

- **Fabricante:** Ideal para clientes que prefieren marcas específicas.
- **Categoría de ropa:** Permite clasificar los productos en camisas, pantalones, vestidos, etc., facilitando la navegación según las necesidades del cliente.
- **Género:** Los clientes pueden elegir productos masculinos, femeninos o unisex, mejorando la personalización de la experiencia de compra.

b) Beneficios para los clientes:

- **Rapidez:** Los clientes pueden encontrar lo que buscan en pocos clics, lo que reduce la frustración y mejora su experiencia de compra.
- **Decisión informada:** Gracias a los filtros, los clientes tienen acceso directo a los productos que cumplen con sus criterios, lo que acelera el proceso de compra y reduce dudas.
- **Satisfacción:** Una navegación clara y efectiva aumenta la confianza de los clientes en la tienda online, favoreciendo su fidelización.

c) Beneficios para la empresa:

- **Aumento de ventas:** Al facilitar la búsqueda, los clientes tienen más probabilidades de encontrar productos que les interesen, lo que incrementa las conversiones.
- **Análisis de preferencias:** Los datos recopilados sobre las búsquedas de los clientes permiten a CHIRK S.R.L. conocer mejor sus preferencias y ajustar su oferta de productos en consecuencia.

2.5. Integración de la Propuesta de Valor

La combinación del almacenamiento en la nube y la búsqueda avanzada por filtros se traduce en una experiencia de compra moderna y eficiente, al mismo tiempo que simplifica la gestión interna del negocio.

Por un lado, los clientes disfrutan de una tienda online que les permite encontrar productos con facilidad, con imágenes atractivas y accesibles desde cualquier dispositivo. Por otro lado, el equipo de CHIRK S.R.L. cuenta con un sistema robusto y eficiente que optimiza el manejo del inventario y reduce la carga administrativa.

2.6. Tecnologías Utilizadas

Para garantizar un desarrollo eficiente y una implementación efectiva, se seleccionaron tecnologías modernas y confiables. A continuación, se explican las herramientas y componentes principales del proyecto:

2.6.1. XAMPP

- **Qué es:** Es un servidor local que integra Apache, MySQL, PHP y Perl.
- **Función en el proyecto:** XAMPP se utiliza para simular un servidor en el entorno de desarrollo, permitiendo ejecutar y probar el sistema de manera local antes de implementarlo en un servidor real.

2.6.2. PHP

- **Qué es:** PHP (Hypertext Preprocessor) es un lenguaje de programación del lado del servidor, diseñado para el desarrollo de aplicaciones web dinámicas.
- **Función en el proyecto:** PHP será el lenguaje principal para construir las funcionalidades del sistema, como el manejo del inventario y la gestión de usuarios.

2.6.3. MySQL y Workbench

- **Qué es:** MySQL es un sistema de gestión de bases de datos relacional, mientras que Workbench es una herramienta gráfica que permite diseñar, modelar y administrar bases de datos.
- **Función en el proyecto:**
 - MySQL almacenará todos los datos del sistema, como productos, clientes y pedidos.
 - Workbench se usará para diseñar el esquema de la base de datos y optimizar su estructura.

2.6.4. Composer

- **Qué es:** Es un administrador de dependencias para PHP que permite instalar y gestionar bibliotecas y frameworks necesarios para el desarrollo.
- **Función en el proyecto:** Composer se utilizará para integrar librerías como Dompdf, necesarias para generar funcionalidades adicionales.

2.6.5. Dompdf

- **Qué es:** Es una biblioteca de PHP que convierte contenido HTML y CSS en documentos PDF.

- **Función en el proyecto:** Dompdf permitirá generar reportes personalizados en formato PDF, como inventarios, resúmenes de ventas y órdenes de compra, que los usuarios pueden descargar o imprimir fácilmente.

2.7. Software

Software se define como un conjunto de programas, instrucciones y reglas para ejecutar ciertas tareas en una computadora u ordenador (Real Academia Española, 2015).

Estos conjuntos de instrucciones también se conocen como programas y cada uno se desarrolla con un propósito específico.

Sus principales componentes son:

- Instrucciones : este componente proporciona la funcionalidad deseada y el rendimiento adecuado cuando se ejecute
- Datos: este componente es necesario para manejar y poner en marcha el software desarrollado, este componente es algo esencial para el software.
- Documentos: Este componente describe el funcionamiento y operatividad del programa o software.

2.8. Ingeniería de Software

La ingeniería de software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción del software. (Somerville, 2011).

Los procesos del software real son secuencias entrelazadas de actividades técnicas, colaborativas administrativas con la meta general de especificar, diseñar, implementar y probar un sistema de software.

Dentro de la ingeniería de software existen cuatro actividades básicas que son: el proceso de especificación, desarrollo, validación, y evolución que se organiza de diversa manera en los diferentes procesos de desarrollo. La forma en la que se lleva a cabo estas actividades depende del tipo de software que se realizará y la inclusión de estructuras organizativas

2.9. Especificaciones del Software

Las especificaciones del software o la ingeniería de requerimientos consisten en un proceso de comprender y definir los servicios que se requieren en el sistema, así como las restricciones sobre la operatividad del sistema.

Existen 4 actividades principales en el proceso de ingeniería de requerimientos:

- **Estudio de factibilidad**

Se evalúa si las necesidades identificadas del usuario se cubren con las tecnologías actuales, el estudio considera si el software tendrá costo-beneficio desde un punto de vista empresarial, y si este puede desarrollarse dentro de las restricciones presupuestales existentes. Este estudio debe informar la decisión respecto a si se continúa o no con un análisis más detallado.

- **Obtención y análisis de requerimientos**

Este proceso de derivar los requerimientos del sistema mediante observación de sistemas similares, discusiones con los usuarios y proveedores potenciales, análisis de tareas, etc.

Esto puede incluir uno o más modelos del sistema y prototipos, esto ayudará a tener más claro el panorama que se desarrollará.

- **Especificación de requerimientos**

Esta Actividad consiste en transcribir la información recopilada durante el análisis, que se debe colocar en un documento, este último incluye 2 tipos de requerimientos. Los requerimientos del usuario son informes abstractos de requerimientos del sistema para el cliente y el usuario final del sistema; y los requerimientos del sistema que son una descripción detallada de la funcionalidad que se va a ofrecer.

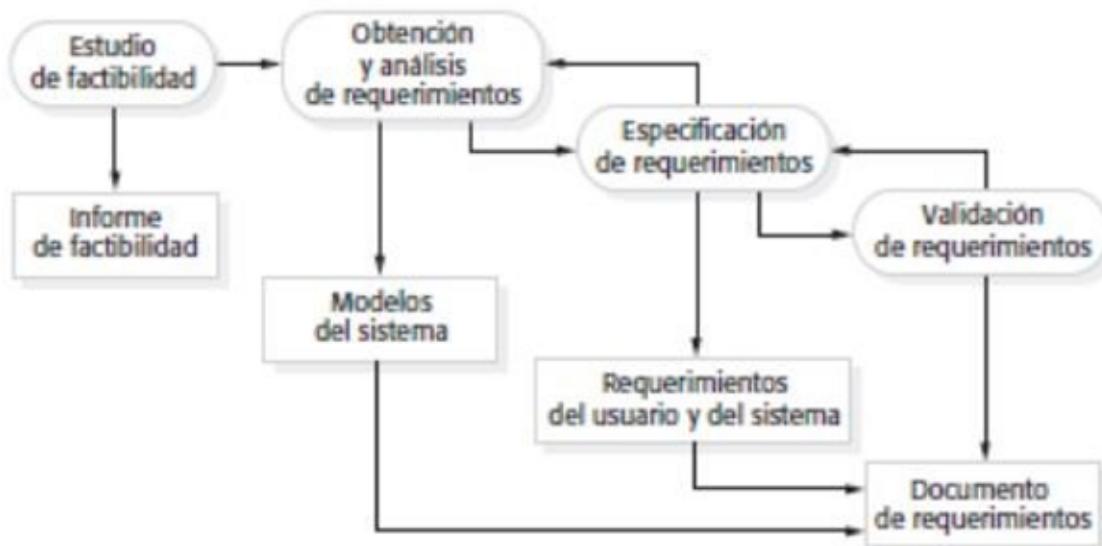
Dentro de la ingeniería de requerimientos se encuentran 2 tipos de requerimientos que son los requerimientos funcionales y requerimientos no funcionales.

→ Requerimientos Funcionales: Son Enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de

cómo debería comportarse el sistema en situaciones específicas. En algunos casos los requerimientos funcionales también explican lo que no debe hacer el sistema.

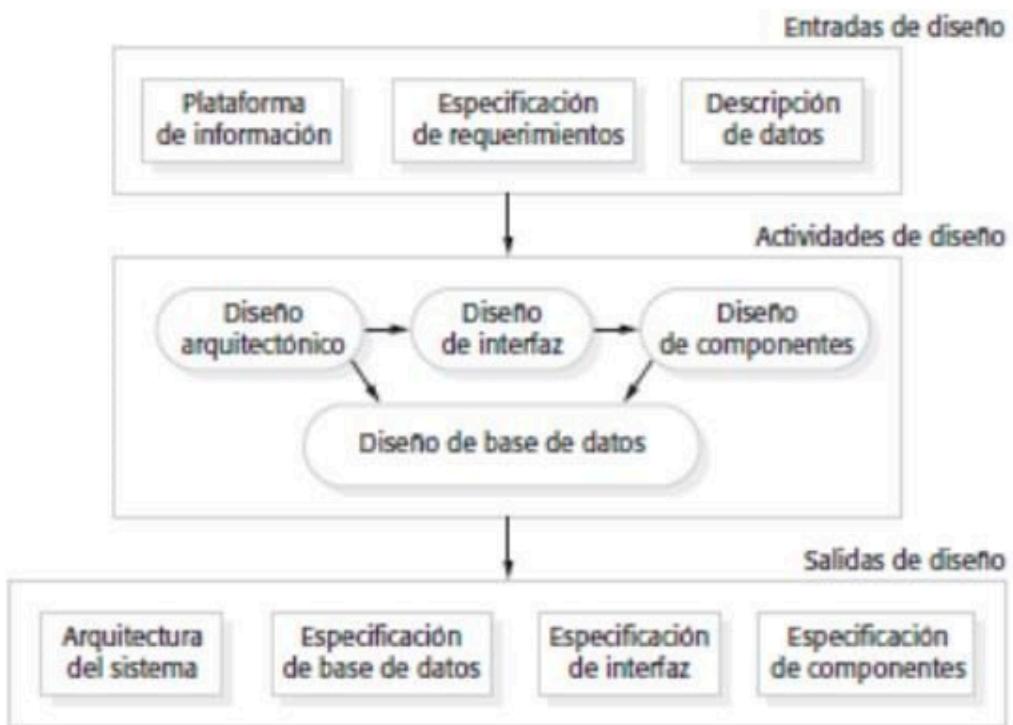
- Requerimientos no funcionales: son limitaciones sobre servicios o funciones que ofrece el sistema que incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que características o a servicios individuales del sistema.
- Validación de requerimientos

Se verifica que los requerimientos sean coherentes, realistas y completos. Durante el proceso es inevitable descubrir errores dentro de los requerimientos con el objetivo de corregir dichos problemas.



2.10. Diseño e implementación del Software

La etapa de implementación de desarrollo de software corresponde en convertir una especificación del sistema a un sistema ejecutable. Siempre incluye procesos de diseño y programación, un diseño de software se entiende como una descripción de la estructura del software que se va a implementar, las interfaces entre componentes del sistema y en ocasiones los algoritmos usados.



Las cuatro actividades principales para desarrollar un sistema de información se definen de la siguiente manera.

- Diseño arquitectónico

Comúnmente en esta etapa se desarrollan los principales componentes llamados subsistemas o módulos, sus relaciones y cómo se distribuyen en todo el sistema.

- Diseño de la interfaz

Esta especificación de interfaz no debe presentar ambigüedades, debe ser una interfaz precisa, los componentes deben ser fáciles de usar para que tenga una mejor interacción con el usuario final.

- Diseño de componentes

Cada componente del sistema se diseña su funcionalidad, estas funcionalidades pueden variar durante la implementación, pero el modelo de diseño sirve para generar en automático una implementación.

- Diseño de base de datos

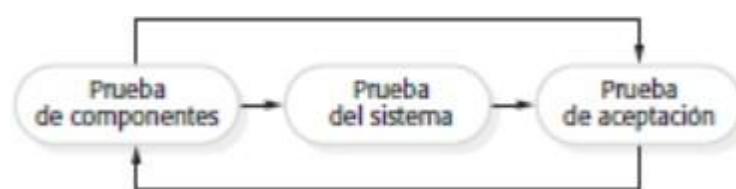
En esta etapa se diseñan las estructuras del sistema de datos y como se representaran en una base de datos, en esta parte se puede reutilizar una base de datos existente o crear una nueva estructura acorde a los requerimientos del sistema.

El diseño de un programa para implementar un sistema se sigue naturalmente de los procesos de elaboración del sistema. Aunque algunas clases de sistemas se diseñan con detalle antes de comenzar cualquier implementación, lo más común es entrelazar los módulos desarrollados con los módulos ya implementados anteriormente, esto dependerá de la metodología que se implemente.

2.11. Validación de software

La prueba de validación o más generalmente conocida como verificación y validación, se crea para mostrar que un sistema cumple tanto como las especificaciones como las expectativas del cliente.

Las pruebas se ejecutan a través de datos de prueba simulados, esta parte puede incluir procesos de comprobación, como inspecciones y revisiones en cada etapa del proceso de software. Se muestran un proceso de prueba de tres etapas donde los componentes del sistema se ponen a prueba y luego, se hace lo mismo con el sistema integrado o ya implementado y finalmente el sistema se pone a prueba con los datos del cliente.



Estas tres etapas se describen a continuación:

- **Prueba de componentes**

Se pone a prueba los componentes desarrollados, cada componente se prueba de manera independiente, es decir sin otros componentes del sistemas. Estos pueden ser simples

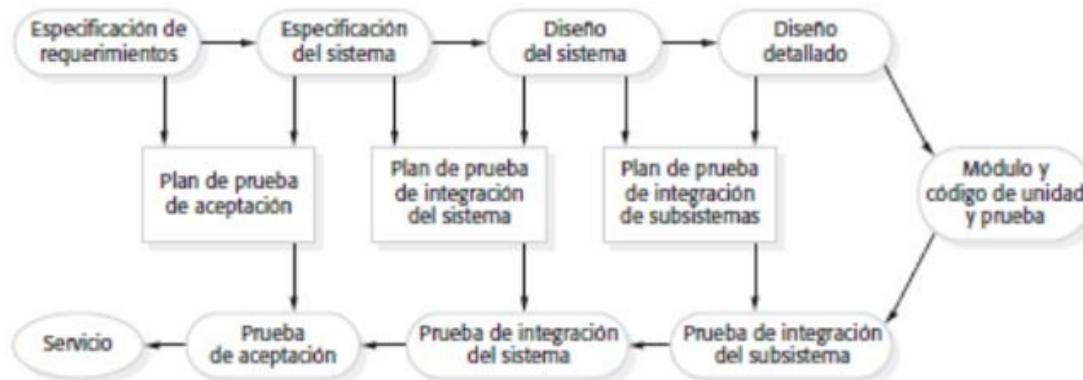
entidades, como funciones o clases de objetos, o agrupamientos coherentes de dichas entidades.

- **Pruebas del sistema**

Una vez realizada la prueba de componentes, se procede a unirlas al sistema en sí para formar un sistema completo. Este proceso tiene la finalidad de descubrir errores que resultan de interacciones no anticipadas entre componentes y problemas de interfaz de componentes, así también mostrar que el sistema cubre sus requerimientos funcionales y no funcionales y poner a prueba las propiedades emergentes del sistema.

- **Pruebas de aceptación**

Una vez integrado el sistema se pone a prueba con datos suministrados por el cliente, las pruebas de aceptación revelan errores y omisiones en la definición de requerimientos del sistema.



2.12. Metodologías Ágiles de Desarrollo de Software

Se considera que un modelo es ágil o liviano cuando se emplea para su construcción una herramienta o técnica sencilla, que apunta a desarrollar un modelo aceptable bueno y

suficiente en lugar de un modelo perfecto y complejo. Un modelo es suficientemente bueno cuando cumple con los objetivos para los que fue creado.

Las metodologías ágiles impulsan generalmente una gestión de proyecto que promueve el trabajo en equipo, la organización y responsabilidad propia, un grupo de buenas prácticas de ingeniería de software, que brinda una calidad efectiva y en un enfoque de negocios que alinea el desarrollo con las necesidades del cliente y los objetivos de la compañía.

2.13. Metodología Ágil Scrum

Scrum se define como una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo, es decir que reduce al máximo la burocracia y actividades no orientadas a producir software que funcione y produce resultado por periodos muy breves de tiempo por medio de iteraciones o Sprint.

Un concepto importante es que esta metodología requiere que exista una fluida comunicación entre el equipo de trabajo y los interesados, logrando de esta forma que los cambios puedan ser implementados de manera rápida y económica.

2.13.1. Elementos de Scrum

❖ Herramientas de Scrum

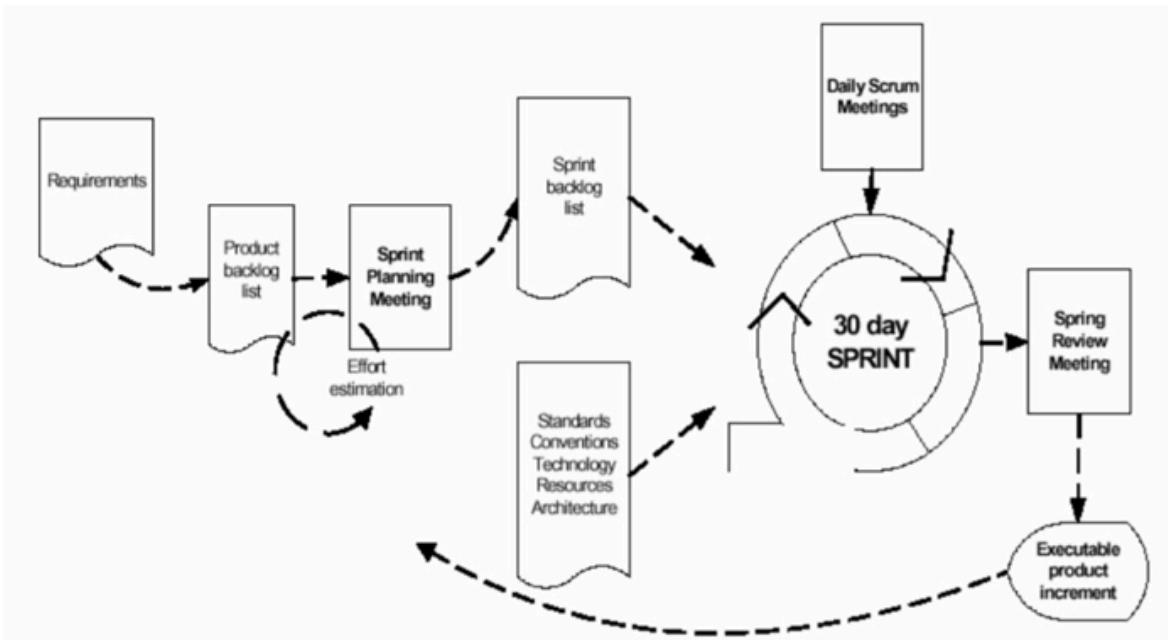
- **Product Backlog:** Es una lista priorizada de funcionalidades técnicas y de negocio. Estas funcionalidades son requisitos a muy alto nivel de lo que debe hacer la aplicación, donde se listan características, funciones, tecnología, mejoras, bugs, etc. que serán aplicadas. Esta etapa es el punto de inicio del proyecto.
A través del Product Backlog se planifica las iteraciones, agrupando de forma coherente funcionalidades de modo que no implique un esfuerzo extra para completar todos los objetivos.
- **Sprint Backlog:** Es la lista de tareas de un Sprint. Identifica y define el trabajo a ser alcanzado por el equipo de desarrollo durante un sprint. A un nivel el Sprint Backlog identifica las características mientras a otro nivel identifica las tareas requeridas para implementar esas características
- **Incremento:** Parte de un sistema desarrollado en un Sprint. Este producto desarrollado es potencialmente entregable al final de cada Sprint, implica que todo está completamente terminado en cada Sprint; y se podría realmente empaquetar o desplegar inmediatamente después de la Revisión del Sprint con mínimas tareas, si bien a veces se necesitan ciertos trabajos de acabado tales como pruebas o documentación.

❖ Roles y Responsables

- **Propietario del Producto (Product Owner):** Es la persona conocedora del entorno del cliente y de la visión del producto. Es el responsable de obtener el mayor valor posible para el cliente o los usuarios; también responsable de la financiación necesaria del proyecto, de la finalización y del retorno de inversión.
- **Equipo Desarrollador (Scrum Team):** Los equipos Scrum deben estar conformados por 4 a 7 personas. Si bien se valora que existan personas con determinados perfiles, lo más importante es que exista una identidad de equipo. Las responsabilidades de los miembros del equipo abarca pruebas, análisis, arquitectura, diseño, programación, planificación y estimación (Ambler & Holitzka, 2012)
- **Scrum Master:** El Scrum Master lidera el equipo y elimina los obstáculos que impiden a los miembros del equipo avanzar con sus actividades. Ayuda al equipo a focalizarse en el trabajo que se debe desarrollar y cumplir los objetivos de la iteración.

❖ Prácticas

- **Sprint:** Es el procedimiento de adaptación de las cambiantes de variables del entorno (requerimientos, tiempo, recursos, conocimiento y tecnología). Son ciclos interactivos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado y su arquitectura y diseño evolucionan durante el desarrollo.
Un Sprint tiene una duración entre una semana y un mes. No es posible introducir cambios durante el Sprint, por tanto para planificar su duración hay que pensar en cuánto tiempo se dedicara cambios fuera del sprint. Dependiendo del tamaño del sistema, la construcción se puede llevar entre 3 y 8 Sprint. Por otra parte podrían firmarse equipos para un desarrollo en paralelo.



➤ **Sprint Backlog:** Es el punto de entrada de cada Sprint. Es una lista que tiene los ítems del Product Backlog que van a ser implementados en el siguiente Sprint.

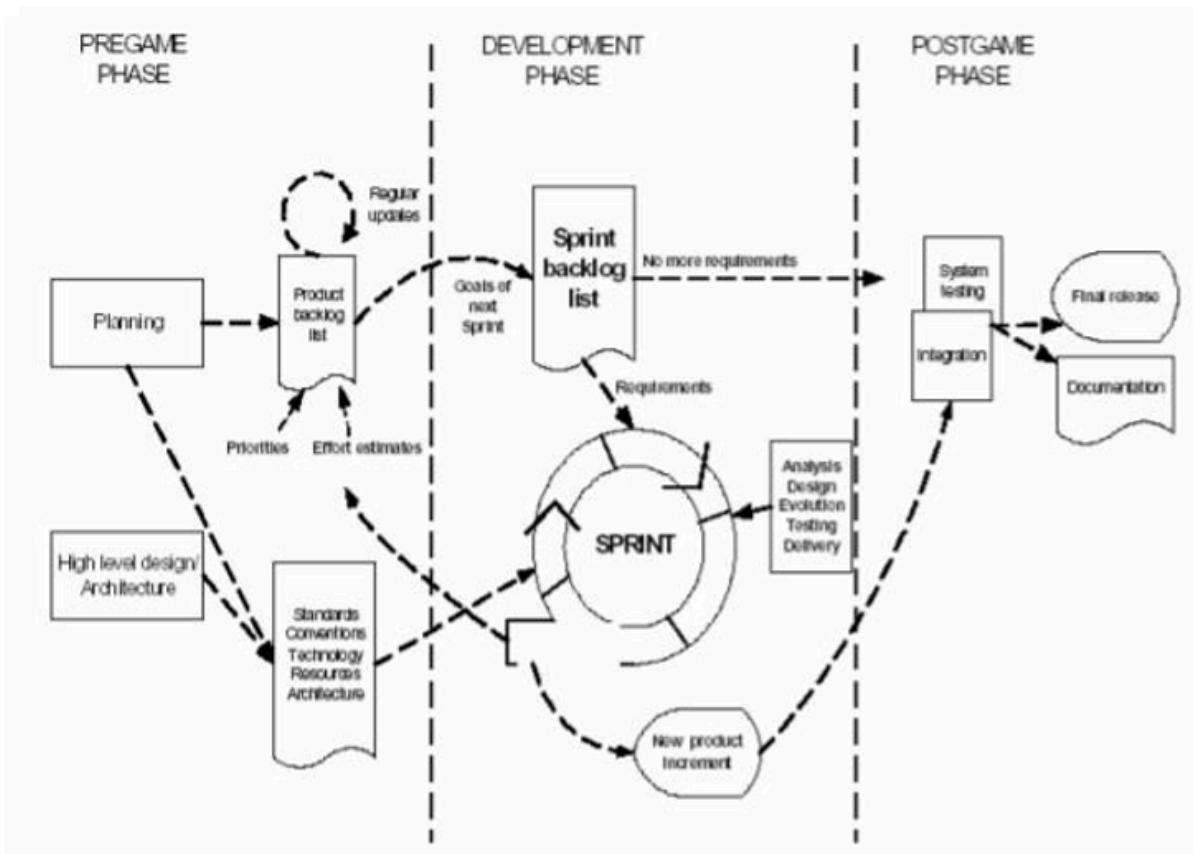
La información y objetivos que se genera a partir de una reunión entre el Scrum Team, Scrum master y el Product Owner se marcan en el sprint Backlog lo que modifica dependiendo de la adición o retiro de información por parte del Scrum Team o del Scrum master.

➤ **Stabilization Sprints:** En estos sprints el equipo se concentra en encontrar defectos, no en agregar funcionalidad. Suelen aplicarse cuando se prepara un producto para el release. Son útiles cuando se están realizando pruebas beta, se está introduciendo a un equipo la metodología Scrum o cuando la calidad de un producto no alcanza los límites esperados.

Una vez asignado los roles a todos y cada uno de los responsables para el desarrollo del software y definiendo los parámetros de trabajo, se procede a detallar cada una de las etapas por las que tiene que atravesar el desarrollo del producto y la combinación de herramientas utilizadas.

2.13.2. Fases del Scrum

Dentro de Scrum se puede observar 3 fases de desarrollo para cada iteración, cada iteración tiene su propia característica



Pre-Game

Consta de 2 subfases que son la planeación y la arquitectura.

- **Planeación:** Consiste en la definición del sistema que será construido. Para esto se crea la lista Product Backlog a partir del conocimiento que actualmente se tiene del sistema en el que se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido.
- **Arquitectura:** El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la lista Product Backlog. Se identifican los cambios necesarios para ser implementar los elementos que aparecen en la lista Product Backlog y el impacto que esta pueda tener en los cambios a implementarse.

Game

En la fase de Game se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificados y pueden cambiar. Este control se realiza durante los Sprints. Dentro de la variable de entorno se puede encontrar los siguientes parámetros: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación.

Post-Game

Contiene el cierre del release. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno como los requerimientos completados. El sistema está listo para ser liberado y es en esta etapa que realiza la integración, pruebas del sistema y la documentación.

2.14. Diseño

2.14.1. UWE

Dentro del modelado del sistema se utilizará la metodología de UWE.

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones Web. Entre los principales modelos de UWE podemos citar: el modelo lógico-conceptual, modelo navegacional, modelo de presentación, visualización de Escenarios Web y la interacción temporal, entre los diagramas: diagramas de estado, secuencia, colaboración y actividad.

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

❖ Actividades de modelado de UWE.

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo navegacional y el modelo de presentación. A estos modelos se pueden sumar otros modelos como lo son el modelo de interacción y la visualización de Escenarios Web.

El modelo que propone UWE está compuesto por etapas o sub-modelos:

- Modelo Lógico-Conceptual.

UWE apunta a construir un modelo conceptual de una aplicación Web, procura no hacer caso en la medida de lo posible de cuestiones relacionadas con la navegación, y de los

aspectos de interacción de la aplicación Web. La construcción de este modelo lógico-conceptual se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos. El modelo conceptual incluye los objetos implicados en las actividades típicas que los usuarios realizan en la aplicación Web.

- **Modelo de Navegación**

Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación. El primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionarán.

- **Modelo de presentación**

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario.

- **Interacción Temporal**

Presenta los objetos que participan en la interacción y la secuencia de los mensajes enviados entre ellos.

- **Escenarios Web**

Permiten detallar la parte dinámica del modelo de navegación, especificando los eventos que disparan las situaciones, definen condiciones y explícitamente incluyen las acciones que son realizadas. Junto con el modelo de interacción temporal, los escenarios Web proveen la representación funcional dinámica del modelo de navegación.

- **Diagramas**

Los diagramas usados por UWE, son diagramas UML puro. Entre los más importantes tenemos: Diagramas de estado, de Secuencia, de colaboración y diagramas de Actividad.

❖ Fases de la UWE.

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

Las fases o etapas a utilizar son:

- **Captura, análisis y especificación de requisitos**

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

- **Diseño del sistema**

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

- **Codificación del software**

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

- **La Instalación o Fase de Implementación**

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

- **El Mantenimiento**

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

2.15. Design Thinking como Método de Diseño

El Design Thinking es un enfoque centrado en el usuario que busca resolver problemas complejos mediante la empatía, la creatividad y la innovación. Este método se utiliza para garantizar que el sistema desarrollado no solo cumpla con las necesidades funcionales de CHIRK S.R.L., sino que también sea intuitivo, accesible y eficiente para los usuarios finales.

- ❖ **Fases del Design Thinking en el Proyecto:**

- **Empatizar:**

Se identificaron las necesidades y expectativas de los usuarios clave: administradores, encargados de inventario y clientes finales. Se realizaron entrevistas y observaciones para entender sus problemas actuales al manejar el inventario y las compras.

- **Definir:**

A partir de la información recopilada, se definieron los problemas principales, como la falta de control en el inventario y la dificultad para encontrar productos en línea.

- **Idear:**

Se generaron soluciones creativas, como el uso de almacenamiento en la nube para imágenes y un sistema de filtros avanzados en la tienda online, para mejorar la experiencia de usuario.

- **Prototipar:**

Se desarrollaron versiones preliminares del sistema que incluyen pantallas básicas para la gestión de productos y la tienda online, con el objetivo de validar las ideas.

- **Probar:**

Los prototipos se probaron con usuarios reales para identificar áreas de mejora, asegurando que el sistema sea funcional y fácil de usar.

2.16. Calidad de Software

La gestión de calidad proporciona una comprobación independiente sobre el proceso de desarrollo de software. El proceso de gestión de calidad verifica los entregables del proyecto para garantizar que sean conscientes con los estándares y las metas de la organización

2.16.1. Web Site QEM

El principal objetivo de esta metodología cuantitativa consiste en evaluar y determinar el nivel de cumplimiento de las características especificadas para lo cual se analizan las preferencias elementales, parciales y globales, utiliza a su es un modelo de calidad jerárquico basado en la ISO 9126-1 y es adaptado a necesidades propias de la web.

El resultado del proceso de evaluación puede ser interpretado como el grado de requerimientos de calidad satisfechos.

Esta metodología comprende de una serie de fases y actividades que los evaluadores deben llevar a cabo en el proceso de evaluación:

- **Definición de metas de evaluación y selección del perfil de usuario**

Se debe definir las metas y establecer el alcance del proyecto en evaluación, La evaluación puede llevarse a cabo tanto en la fase de desarrollo como en la fase de operatividad del proyecto, y se puede valorar la calidad de un producto completo o bien se puede valorar la calidad de un conjunto de características y atributos de un componente. Los resultados podrán ser utilizados para comprender, mejorar controlar o predecir la calidad de los productos.

- **Definición de los requerimientos de calidad (y/o costos)**

En esta etapa se debe solicitar, acordar y especificar los atributos y características de calidad que estarán presentes en el proceso agrupando en un árbol de requerimientos, de las características de calidad ISO derivándose en sub características y de estas derivamos a los atributos con un mínimo solapamiento. A cada atributo cuantificable del dominio empírico se le asocia una variable en el dominio numérico esta variable puede tomar valor real, que podrá ser medido y computado.

- **Definición de criterios de preferencia elementales y procedimientos de medición**

Un criterio de evaluación elemental es clara y específica cómo medir atributos cuantificables. El resultado final es una preferencia o indicador elemental, el cual puede ser interpretado como el grado o porcentaje del requerimiento elemental satisfecho. Por lo tanto para cada métrica de un atributo necesitamos establecer un rango de valores aceptables y definir la función de criterio elemental, que producirá una correspondencia entre el valor de la métrica con el nuevo valor que representa la preferencia elemental

- **Definición de estructuras de agregación e implementación de la evaluación global**

En el paso previo se producen preferencias de calidad elemental para los atributos considerados en el árbol de requerimientos. Por lo tanto, aplicando un mecanismo de agregación paso a paso, las preferencias elementales se pueden agrupar convenientemente para producir al final un esquema de agregación. Las preferencias de calidad parciales y global se pueden obtener mediante cálculo conforme al modelo de agregación y puntaje empleado.

- **Análisis de resultados y recomendaciones**

Una vez diseñado e implementado el proyecto de evaluación, el proceso culmina con la documentación de las conclusiones y recomendaciones, se analiza los resultados considerando las metas, el perfil de usuarios establecidos, El proceso de evaluación, mediante el uso de WEBQEM_Tool, que produce información elemental, parcial y global que puede fácilmente ser analizada por medio de un modelo hipertextual y eficientemente empleada en actividades de toma de decisiones.

Está compuesto por tres modelos denominados: composición de aplicación, diseño temprano y post Arquitectura.

2.16.2. ISO/IEC 9126

ISO 9126 es un estándar internacional para la evaluación de la calidad de software. Este estándar está dividido en cuatro partes las cuales dirigen, realidad, métricas externas, métricas internas y calidad en las métricas de uso y expendio. El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

- **Funcionalidad**

Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

- Adecuación - Atributos del software relacionados con la presencia y aptitud de un conjunto de funciones para tareas especificadas.
- Exactitud - Atributos del software relacionados con la disposición de resultados o efectos correctos o acordados.
- Interoperabilidad - Atributos del software que se relacionan con su habilidad para la interacción con sistemas especificados.
- Seguridad - Atributos del software relacionados con su habilidad para prevenir acceso no autorizado ya sea accidental o deliberado, a programas y datos.
- Cumplimiento funcional.

- **Fiabilidad**

Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

- Madurez - Atributos del software que se relacionan con la frecuencia de falla por fallas en el software.

- Recuperabilidad - Atributos del software que se relacionan con la capacidad para restablecer su nivel de desempeño y recuperar los datos directamente afectos en caso de falla y en el tiempo y esfuerzo relacionado para ello.
- Tolerancia a fallos - Atributos del software que se relacionan con su habilidad para mantener un nivel especificado de desempeño en casos de fallas de software o de una infracción a su interfaz especificada.
- Cumplimiento de Fiabilidad - La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.
- Usabilidad - Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
- Aprendizaje- Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Comprensión - Atributos del software que se relacionan al esfuerzo de los usuarios para reconocer el concepto lógico y sus aplicaciones.
- Operatividad - Atributos del software que se relacionan con el esfuerzo de los usuarios para la operación y control del software.
- Atractividad
- **Eficiencia**

Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

- Comportamiento en el tiempo - Atributos del software que se relacionan con los tiempos de respuesta y procesamiento y en las tasas de rendimientos en desempeñar su función.
- Comportamiento de recursos - Usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

- **Mantenibilidad**

Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

- Estabilidad - Atributos del software relacionados con el riesgo de efectos inesperados por modificaciones.
 - Facilidad de análisis - Atributos del software relacionados con el esfuerzo necesario para el diagnóstico de deficiencias o causas de fallos, o identificaciones de partes a modificar.
 - Facilidad de cambio - Atributos del software relacionados con el esfuerzo necesario para la modificación, corrección de falla, o cambio de ambiente.
 - Facilidad de pruebas - Atributos del software relacionados con el esfuerzo necesario para validar el software modificado.
- **Portabilidad**

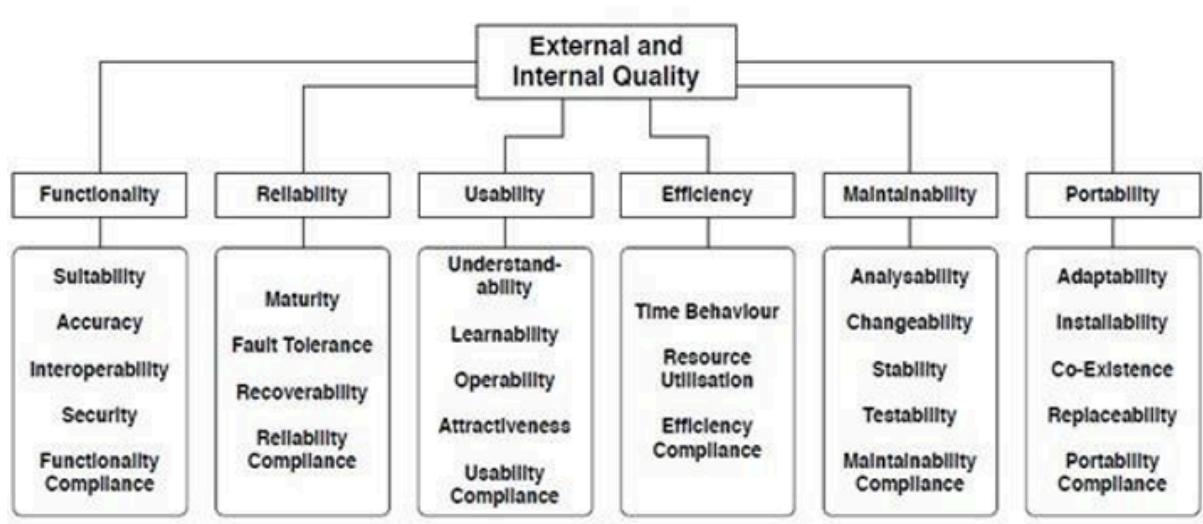
Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra.

- Capacidad de instalación - Atributos del software relacionados con el esfuerzo necesario para instalar el software en un ambiente especificado.
- Capacidad de reemplazamiento - Atributos del software relacionados con la oportunidad y esfuerzo de usar el software en lugar de otro software especificado en el ambiente de dicho software especificado.

- **Calidad en uso**

Conjunto de atributos relacionados con la aceptación por parte del usuario final y Seguridad.

- Eficacia - Atributos relacionados con la eficiencia del software cuando el usuario final realiza los procesos.
- Productividad - Atributos relacionados con el rendimiento en las tareas cotidiana realizadas por el usuario final.
- Seguridad - Atributos para medir los niveles de riesgo.
- Satisfacción - Atributos relacionados con la satisfacción de uso del software



Idealmente, la calidad interna no necesariamente implica la calidad externa y esta a su vez la calidad de uso.

Este estándar proviene del modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

2.17. Historias de Usuario

Se utilizan para especificar los requisitos de las aplicaciones software en las metodologías ágiles (Scrum, XP, FDD, ASD, etc.). Las historias de usuario son tarjetas en donde el interesado describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada historia de usuario debe ser lo

suficientemente comprensible y delimitada para que se pueda implementar en un transcurso de tiempo.

El modelo de Historias de Usuario depende de las características de desarrollo y la información que sea requerida, en la siguiente figura se muestra la tarjeta de historias de usuario que será utilizada.

Historia de Usuario	
No. : Para identificación	Usuario: Persona que utilizará la funcionalidad del sistema descrita en la tarjeta de Historia de Usuario
Nombre de la historia: Descripción de manera general de la tarjeta de historia de Usuario	
Prioridad: La prioridad de la tarea que se plantea	Riego en desarrollo: Riesgo que tiene al momento del desarrollo
Puntos Estimado: puntos o tiempo estimado para el desarrollo	Iteración Asignada: nro. de iteración que se implementará dentro de la metodología
Programador Responsable: Persona encargada de programar cada historia	
Descripción: Información detallada de una historia de usuario	
Observaciones: Aclaraciones necesarias para la historia de Usuario, que le servirá al programador para tenerlas en cuenta al momento del desarrollo	

2.18. Pruebas de Stress

En la ingeniería del software, las pruebas de rendimiento son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Las pruebas de rendimiento son un subconjunto de la ingeniería de pruebas, una

Práctica informática que se esfuerza por mejorar el rendimiento, englobando en el diseño y la arquitectura de un sistema, antes incluso del esfuerzo inicial de la codificación.

La prueba de stress se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los desarrolladores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere la carga esperada

CAPÍTULO 3

MARCO APlicativo

3.1 Introducción

En el diseño y desarrollo de software, existen actividades fundamentales que garantizan la producción, mantenimiento, y evolución de un sistema. Para el desarrollo del Sistema de Gestión de Ventas e Inventory, se implementará la metodología Scrum, que se compone de tres etapas principales: Pre-Game, Sprints y Post-Game.

Durante el Pre-Game, se definirán los requerimientos funcionales y no funcionales del sistema, identificando las historias de usuario que describen las interacciones clave para los roles principales. Además, se diseñará la base de datos como núcleo para almacenar y gestionar toda la información relacionada con productos, clientes y pedidos.

En la etapa de Sprints, se empleará la metodología UWE (UML-based Web Engineering) para organizar el desarrollo en tres modelos principales: lógico-conceptual, navegacional y de presentación. Este enfoque facilitará el diseño del sistema, codificación del software, pruebas y eventual implementación.

Finalmente, en el Post-Game, se ejecutarán pruebas de rendimiento, análisis de calidad y validación de seguridad del software, utilizando herramientas como test de stress.

3.2 Pre-Game

Para la fase de planificación se especificarán los roles de usuario que tenga el sistema, los requerimientos obtenidos se realizarán con los métodos especificados en la ingeniería de requerimientos las cuales son funcionales y no funcionales, obteniendo tales requisitos se procederá a crear historias de usuarios, y así obtener los requerimientos para empezar la siguiente fase del proceso.

3.2.1 Roles de Usuario

Se identifican los roles principales del sistema para estructurar los accesos y permisos de manera eficiente:

- **Usuario:** Puede explorar el catálogo de productos organizados en categorías, añadirlos al carrito o a una lista de favoritos, y realizar pedidos. También podrá registrarse e iniciar sesión.
- **Administrador del Sistema:** Administra productos, clientes, categorías, pedidos, ganancias, y cupones. Gestiona el inventario y supervisa el estado de los pedidos, dividiéndolos en pendientes y completados.

3.2.2 Requerimientos del Software

a) Requerimientos Funcionales

i) Administración del sistema

- R1-1 Diseño de la Base de datos: diseñar la base de datos según requerimientos planteados.
- R1-2 Acceso de usuarios: Definirá los permisos de acceso que tiene el usuario a diferentes módulos del sistema

ii) Módulo de Gestión de Productos

- R2-1 Ingreso de una nueva categoría: cada producto se ordena de manera categórica, se crearan, editaran, eliminaran categorías.
- R2-2 Productos: se insertaran, editaran, eliminaran lógicamente productos, estas se asignan a distintas categorías, se medirán tanto su precio, unidades de medida, proveedores principales y secundarios.

iii) Gestión de usuarios

- R3-1: Registro de usuarios (nombre, correo, contraseña).
- R3-2: Inicio de sesión con autenticación segura.

iv) Catálogo y carrito de compras

- R4-1: Mostrar catálogo de productos por categorías y subcategorías.
- R4-2: Agregar productos al carrito o a la lista de favoritos.
- R4-3: Ver y gestionar el carrito (eliminar productos o actualizar cantidades).

v) Gestión del administrador

- R5-1: Crear, editar o eliminar productos, categorías.
- R5-2: Gestionar el inventario (actualización de stock y precios).
- R5-3: Ver informes de ventas y ganancias.
- R5-4: Supervisar pedidos (pendientes y completados).

b) Requerimientos No Funcionales

- El sistema será desarrollado como una aplicación web, basado en PHP, compatible con navegadores modernos como Google Chrome y Firefox.
- La interfaz será intuitiva y amigable tanto para usuarios como para administradores.
- La base de datos estará diseñada en MySQL para garantizar robustez y escalabilidad.

3.2.3. Historias de usuario

En esta fase de desarrollo se evaluarán las prioridades de las historias de usuario, con el objetivo de establecer un plan de entregas. Asimismo, se definirán los requerimientos específicos de cada usuario para diseñar una estrategia que permita su correcta implementación en el sistema.

A partir de los requerimientos recopilados, surge la necesidad de implementar un control de usuarios basado en roles asignados dentro del sistema. Por ello, es fundamental desarrollar una historia de usuario que contemple el control de acceso al sistema.

Esta historia asegura que el sistema cuente con un mecanismo seguro de inicio de sesión para los usuarios, quienes podrán autenticarse utilizando credenciales únicas. Para garantizar la privacidad de la información, las contraseñas deben estar encriptadas antes de almacenarse en la base de datos. Además, se debe incluir una opción de recuperación de contraseña en caso de que el usuario la olvide, siguiendo un flujo seguro.

Historia de Usuario	
No. :1	Usuario: Administradores, Vendedores
Nombre de la historia: Control de Acceso al Sistema	
Prioridad: Alta	Riego en desarrollo: Bajo
Puntos Estimado: 2	Iteración Asignada: 1
Programador Responsable: Cristhian Andres Escobar Herrera	
Descripción: Los usuarios podrán iniciar sesión en el sistema con credenciales únicas. Las contraseñas estarán encriptadas para garantizar la seguridad de los datos.	
Requerimiento R1-2	
Observaciones: Es fundamental implementar la encriptación y realizar pruebas de acceso con distintos roles para garantizar el correcto funcionamiento.	

Ahora bien, la siguiente historia permite a los administradores manejar el inventario del sistema de manera eficiente. Los usuarios pueden agregar nuevos productos con detalles como nombre, código, categoría, precio y cantidad disponible en stock. También incluye la posibilidad de modificar información de productos existentes y eliminarlos si ya no se usan. Esto facilita el control y actualización de los productos manejados en el sistema.

Historia de Usuario	
No. : 2	Usuario: Administradores
Nombre de la historia: Gestión de Productos	
Prioridad: Alta	Riego en desarrollo: Medio
Puntos Estimado: 3	Iteración Asignada: 1

Programador Responsable: Cristian Gonzalo Mamani Espinoza

Descripción: Permite registrar nuevos productos en el inventario, modificar detalles como nombre, código, categoría, precio y stock disponible. También permitirá eliminar productos que ya no se manejan en el sistema.

Requerimiento R2-1, R2-2.

Observaciones: Es necesario validar que no se eliminen productos con ventas activas o stock pendiente.

La siguiente historia permite al administrador del sistema gestionar los usuarios que tienen acceso al sistema de ventas e inventario. Incluye la funcionalidad para registrar nuevos usuarios asignándoles roles específicos como administrador, vendedor o supervisor. También se podrá modificar la información de los usuarios existentes, desactivar cuentas inactivas o eliminar usuarios según las políticas de la empresa. Además, se garantiza que cada usuario tenga credenciales únicas y seguras para acceder al sistema.

Historia de Usuario	
No. : 3	Usuario: Administrador del Sistema
Nombre de la historia: Gestión de Usuarios	
Prioridad: Alta	Riego en desarrollo: Medio
Puntos Estimado: 2	Iteración Asignada: 1
Programador Responsable: Carvajal Quispe Esther Mayerly	
Descripción: Esta funcionalidad permitirá al administrador gestionar usuarios del sistema mediante el registro con roles específicos, la modificación de datos, la activación, desactivación o eliminación de usuarios según las políticas, y la generación de credenciales seguras para garantizar la autenticidad y privacidad. Requerimiento R3-1, R3-2.	
Observaciones: Se debe implementar una validación para evitar usuarios duplicados, almacenar las contraseñas encriptadas en la base de datos.	

Esta funcionalidad permite registrar las ventas realizadas en la tienda. El vendedor seleccionará los productos, especificará las cantidades vendidas y asociará la venta a un cliente registrado o no registrado. Al finalizar la transacción, se generará un comprobante único de la venta con todos los detalles relevantes. Es vital que el sistema valide el stock disponible antes de completar la operación, notificando cualquier irregularidad.

Historia de Usuario	
No. : 4	Usuario: Vendedores
Nombre de la historia: Procesar Ventas	
Prioridad: Alta	Riego en desarrollo: Alta
Puntos Estimado: 4	Iteración Asignada: 2
Programador Responsable: Mamani Gutierrez Sarahi Nicol	
Descripción: Los vendedores podrán registrar las ventas realizadas, seleccionando productos, cantidades, y cliente asociado. Se generará automáticamente un comprobante de venta con un código único.	
Requerimiento R4-1, R4-2, R4-3	
Observaciones: Es indispensable que el sistema valide el stock de los productos seleccionados y notifique al usuario en caso de insuficiencia.	

La funcionalidad de Gestión del Administrador permitirá al administrador principal realizar tareas clave para configurar y supervisar el sistema. Estas incluyen la gestión de usuarios y roles, supervisión de actividades mediante logs, establecimiento de políticas de seguridad avanzadas, generación de reportes globales con opciones de filtrado y exportación, y ajustes de configuraciones globales como impuestos, descuentos, horarios y datos empresariales. El objetivo es centralizar el control del sistema, garantizando seguridad y un funcionamiento eficiente, con acceso exclusivo para el administrador principal.

Historia de Usuario	
No. : 5	Usuario: Administrador del Sistema
Nombre de la historia: Gestión del Administrador	
Prioridad: Alta	Riego en desarrollo: Medio
Puntos Estimado: 3	Iteración Asignada: 2
Programador Responsable: Jhon Vargas Quispe	
<p>Descripción: Esta funcionalidad permitirá al administrador principal gestionar la configuración y operación general del sistema, abarcando la asignación de roles y permisos, supervisión de actividades mediante registros, gestión de políticas de seguridad, generación de informes personalizados y configuración de parámetros generales como impuestos, descuentos y horarios.</p> <p>Requerimiento R5-1, R5-2, R5-3, R5-4</p>	
<p>Observaciones: Esta funcionalidad estará disponible exclusivamente para el administrador principal, asegurando que los cambios realizados impacten dinámicamente en todo el sistema. Además, los registros de actividades deberán ser inalterables y accesibles para auditorías.</p>	

A continuación se muestra un resumen de las historias de usuario, que se utilizará como un Product Backlog.

Id	Nombre	Puntos Estimados	Iteración
1	Control del acceso al sistema	2	1
2	Gestión de productos	3	1
3	Gestion de usuarios	2	1
4	Procesar ventas	4	2
5	Gestion del administrador	3	2

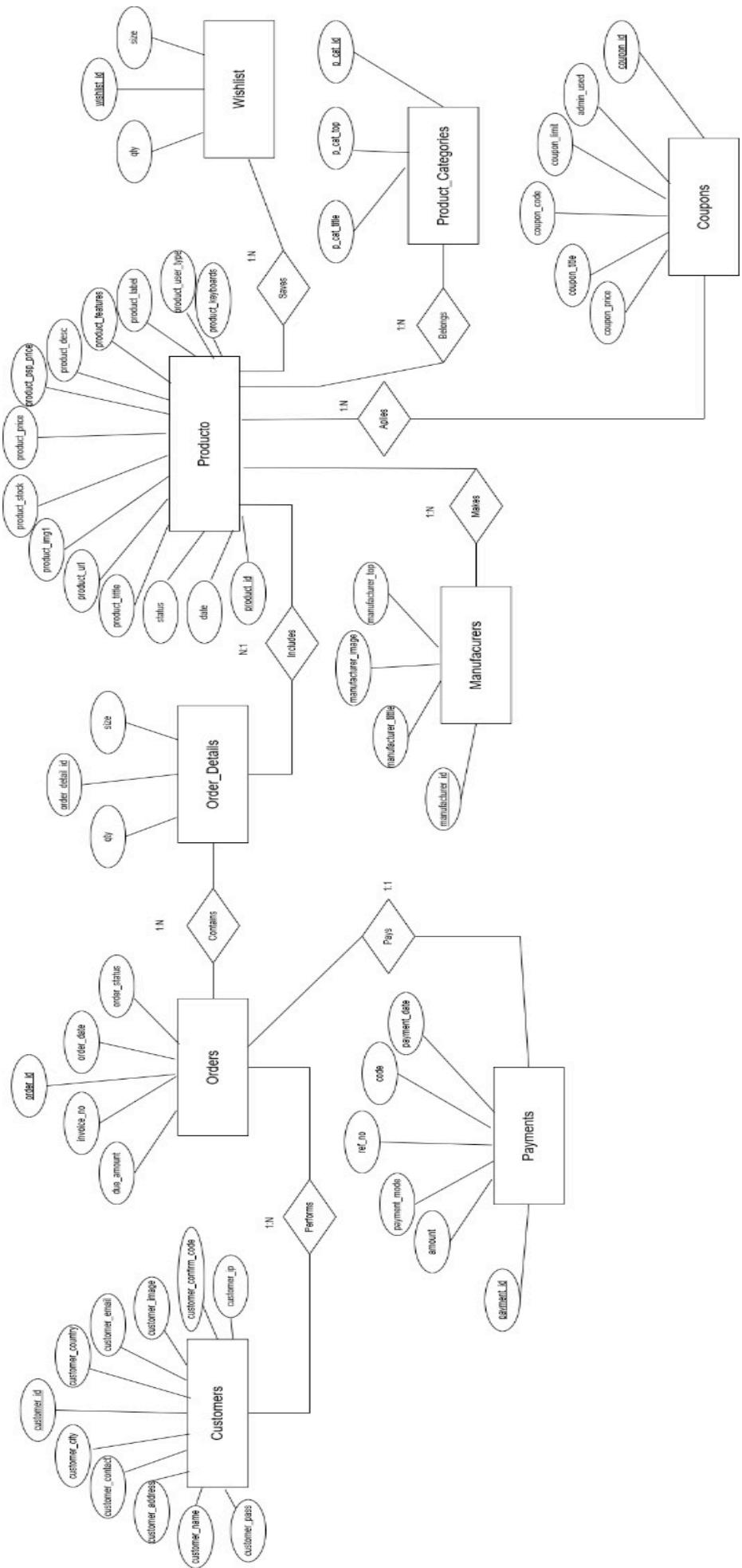
3.2.4. Base de Datos

La base de datos es un componente esencial para el funcionamiento de cualquier sistema de ventas, ya que es responsable de almacenar, organizar y gestionar toda la información relacionada con usuarios, productos, transacciones y otros aspectos clave del sistema. En este apartado, se describe el modelo de datos, la estructura de las tablas y las relaciones entre ellas, así como las tecnologías y herramientas utilizadas.

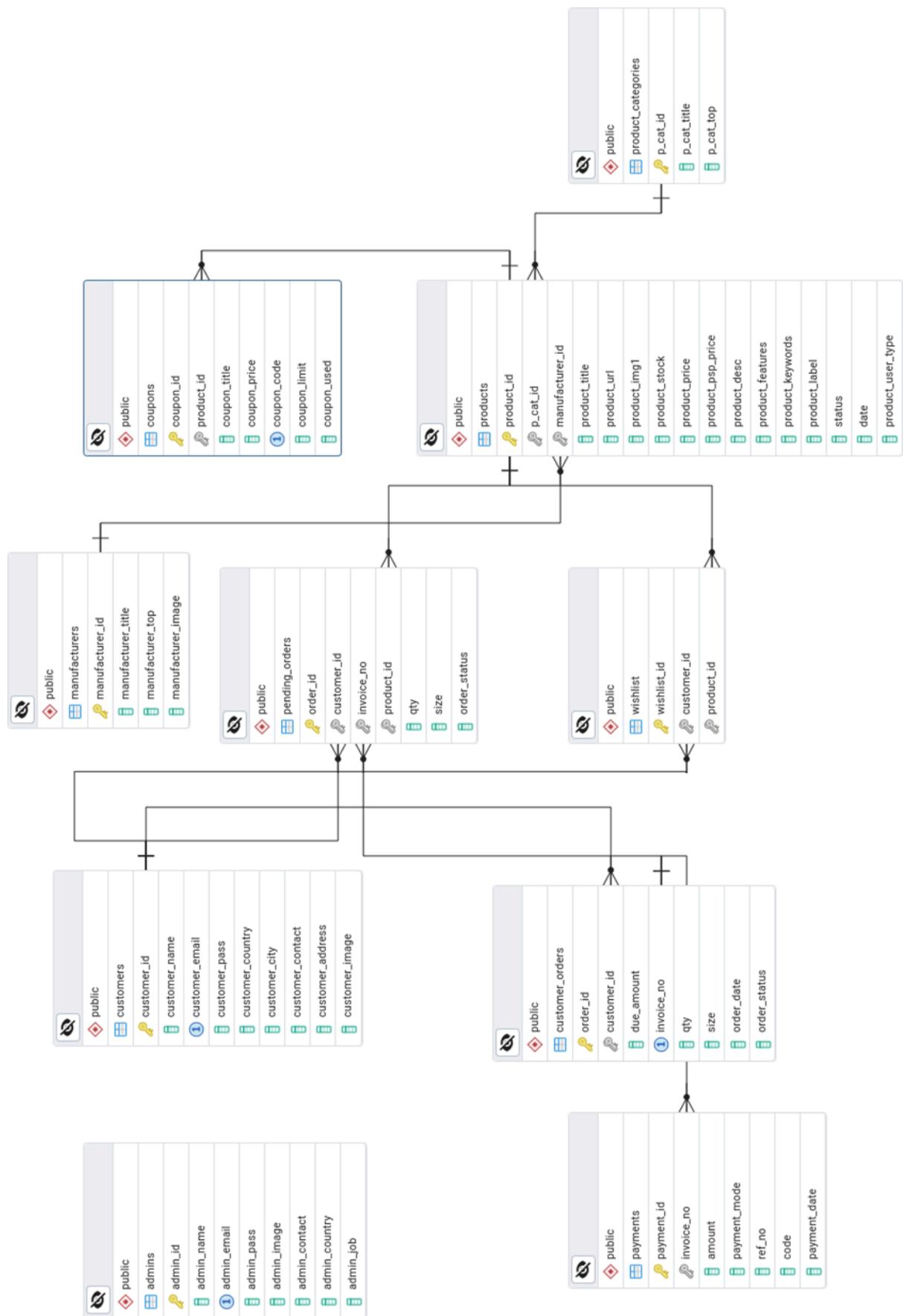
a) Diagrama Entidad-Relación

El sistema de gestión de ventas e inventario cuenta con una base de datos que incluye las siguientes entidades principales:

- **Usuarios:** Administra la información del usuario (nombre, correo, rol).
- **Productos:** Contiene datos como nombre, precio, cantidad en stock y categoría.
- **Categorías:** Organiza los productos en grupos y subgrupos.
- **Pedidos:** Registra los detalles de los pedidos realizados por los usuarios.



b) Modelo relacional de la Base de Datos



3.3. Game

En esta fase del proyecto se realizarán los Sprints necesarios para implementar las funcionalidades planificadas. La estrategia utilizada para desarrollar el producto final incluye una fase de análisis y diseño utilizando los modelos propios de la metodología UWE . Posteriormente, se procederá con el desarrollo de las interfaces y funcionalidades del sistema basadas en los modelos de presentación y navegación.

Para organizar y estructurar adecuadamente la información, se utilizarán los tres modelos principales de UWE :

- Modelo Lógico – Conceptual
- Modelo de Navegación
- Modelo de presentación

3.3.1. Primera Iteración (Sprint 1)

En esta etapa se analizarán y desarrollarán los módulos fundamentales para el sistema, los cuales incluyen:

- Control de acceso al sistema
- Gestión de usuarios y roles
- Gestión de productos y su inventario
- Gestión de proveedores
- Gestión de órdenes de compra
- Gestión de notas de recepción de productos.

A continuación, se detalla la planificación de las fases de implementación con su respectiva duración y estado de avance:

ID	Tareas	Tipo	Días de Trabajo	Estado
1.1	Análisis de requerimientos para la autenticación de usuarios y gestión de roles.	Análisis	1	Terminado
1.2	Análisis de requerimientos para la gestión de	Análisis	1	Terminado

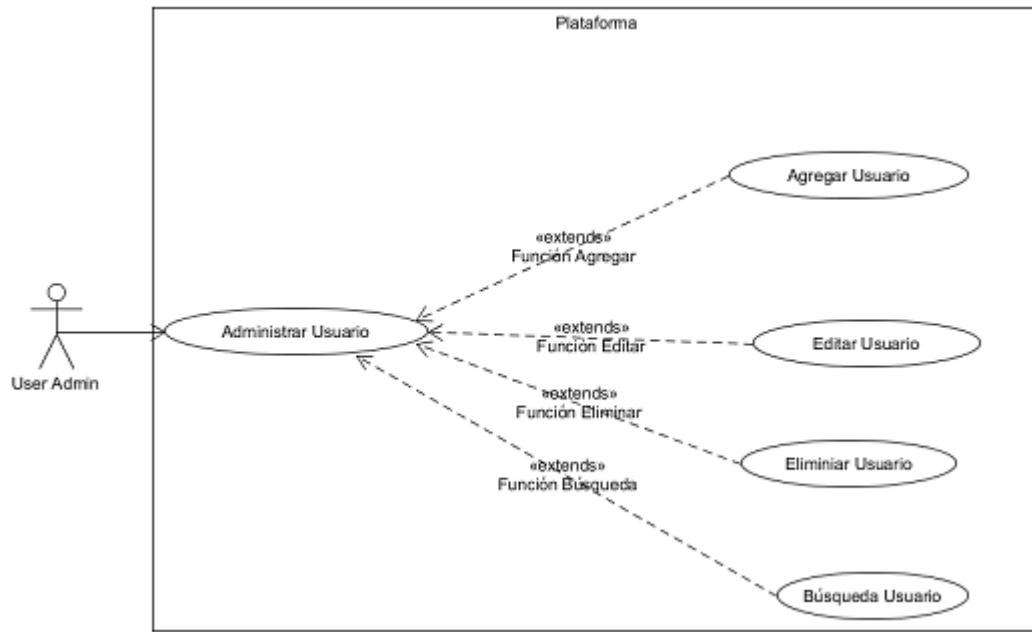
	productos e inventario			
1.3	Análisis de requerimientos para la gestión de proveedores.	Análisis	1	Terminado
1.4	Análisis de requerimientos para la gestión de órdenes de compra.	Análisis	1	Terminado
1.5	Análisis de requerimientos para notas de recepción (con y sin orden de compra)	Análisis	1	Terminado
1.6	Diseño del modelo lógico para la autenticación de usuarios y gestión de roles	Diseño	1	Terminado
1.7	Diseño del modelo lógico para la administración de productos e inventario	Diseño	1	Terminado
1.8	Diseño del modelo lógico para la gestión de proveedores.	Diseño	1	Terminado
1.9	Diseño del modelo lógico para la gestión de órdenes de compra.	Diseño	2	Terminado
1.10	Diseño del modelo lógico para notas de recepción (con y sin orden de compra)	Diseño	3	Terminado
1.11	Diseño de la interfaz de autenticación de usuarios	Diseño	2	Terminado
1.12	Diseño de la interfaz para la gestión de productos e inventario	Diseño	1	Terminado
1.13	Diseño de la interfaz para la gestión de proveedores.	Diseño		Terminado
1.14	Diseño de la interfaz para la gestión de órdenes de compra.	Diseño	2	Terminado
1.15	Diseño de la interfaz para	Diseño	2	Terminado

	notas de recepción (con y sin orden de compra)			
1.16	Desarrollo del módulo de autenticación de usuarios y gestión de roles.	Desarrollo	4	Terminado
1.17	Desarrollo del módulo de gestión de productos e inventario	Desarrollo	5	Terminado
1.18	Desarrollo del módulo de gestión de proveedores	Desarrollo	4	Terminado
1.19	Desarrollo del módulo de gestión de órdenes de compra	Desarrollo	4	Terminado
1.20	Desarrollo del módulo de gestión de órdenes de compra	Desarrollo	4	Terminado

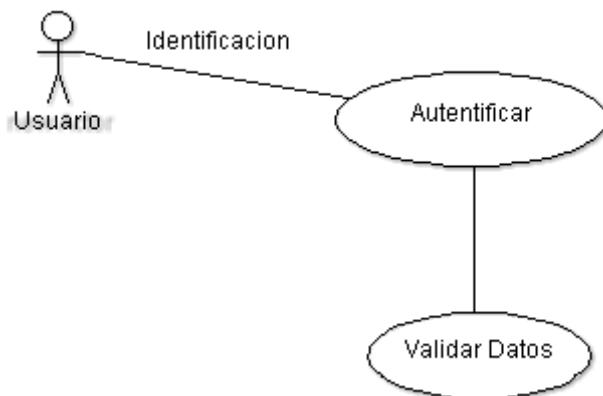
a) Desarrollo de Sprint

- **Modelo Lógico-Conceptual**

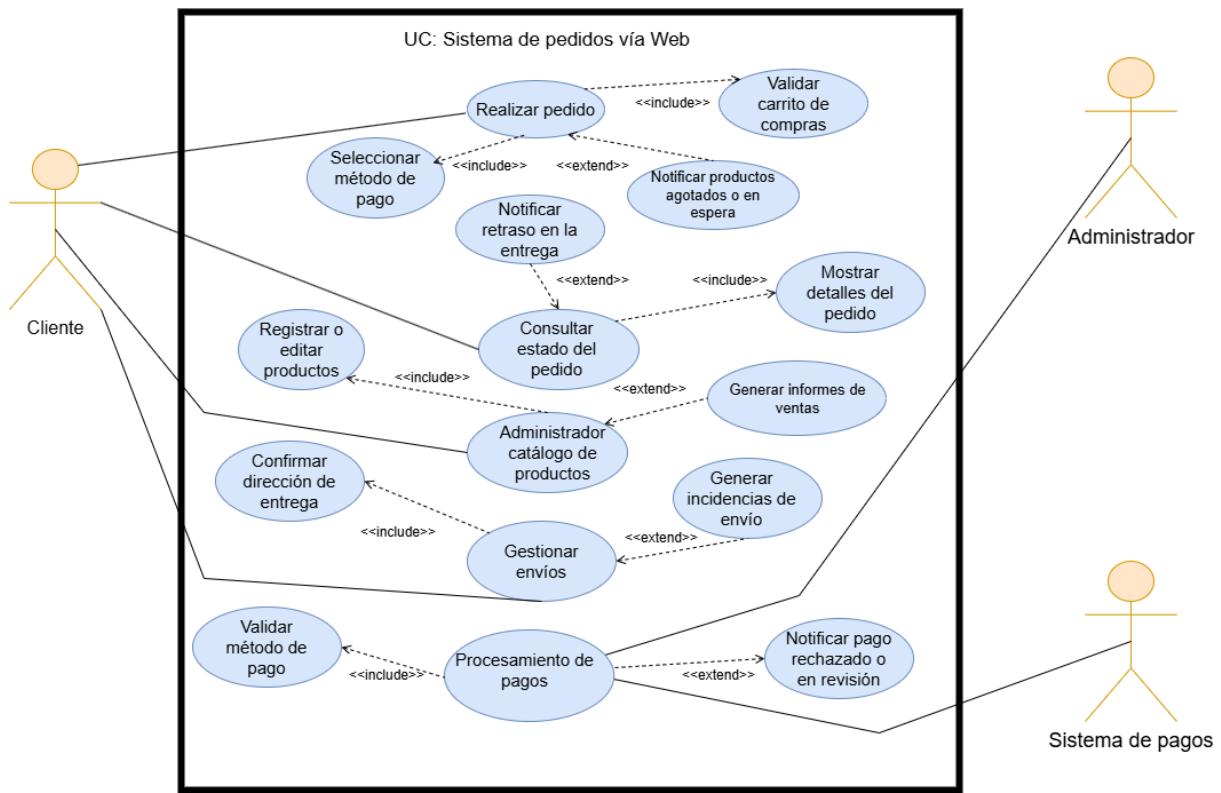
Para la gestión de usuarios y roles dentro del sistema, se muestra el flujo correspondiente a la creación de usuarios y asignación de roles según sus permisos.



El diagrama de autenticación de usuarios detalla la interacción del usuario al momento de ingresar al sistema.



Para la creación de nuevas órdenes de compra se muestra la siguiente interacción de datos dentro de un formulario como se muestra en la figura.

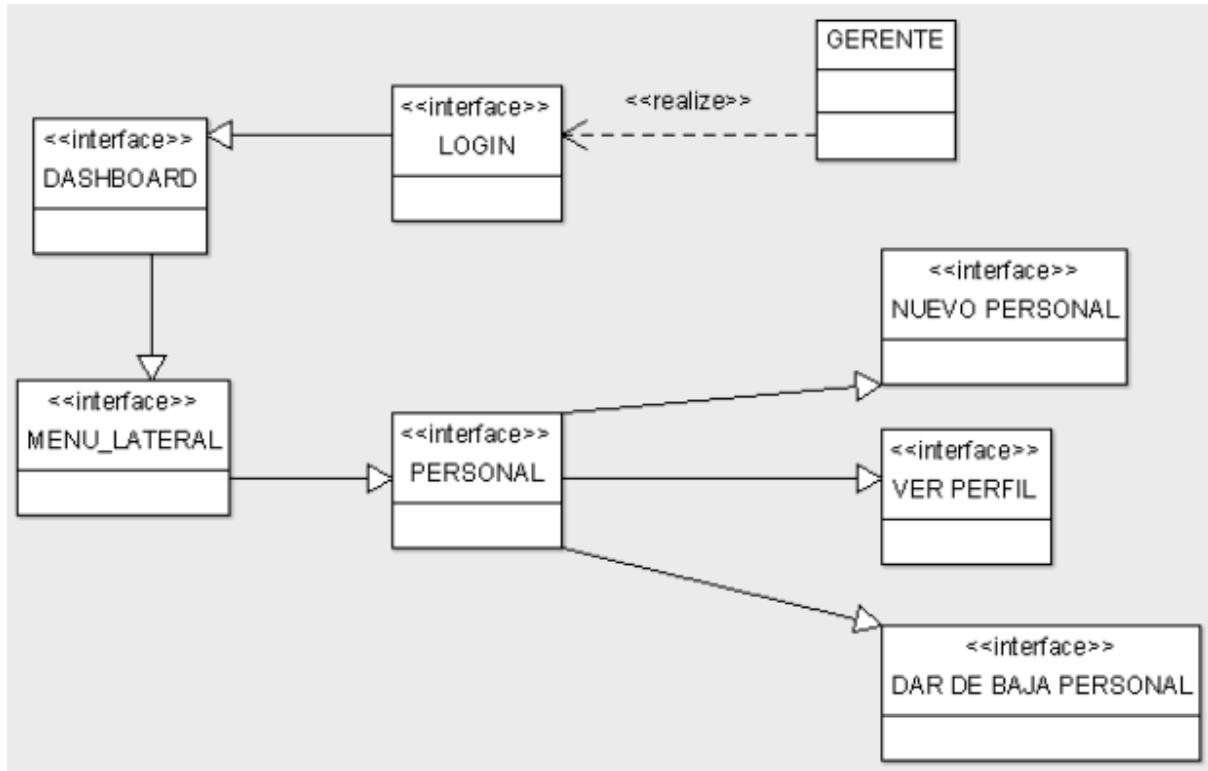


Las órdenes de compra pueden asumir distintos estados, que son esenciales para la gestión y auditoría del sistema.

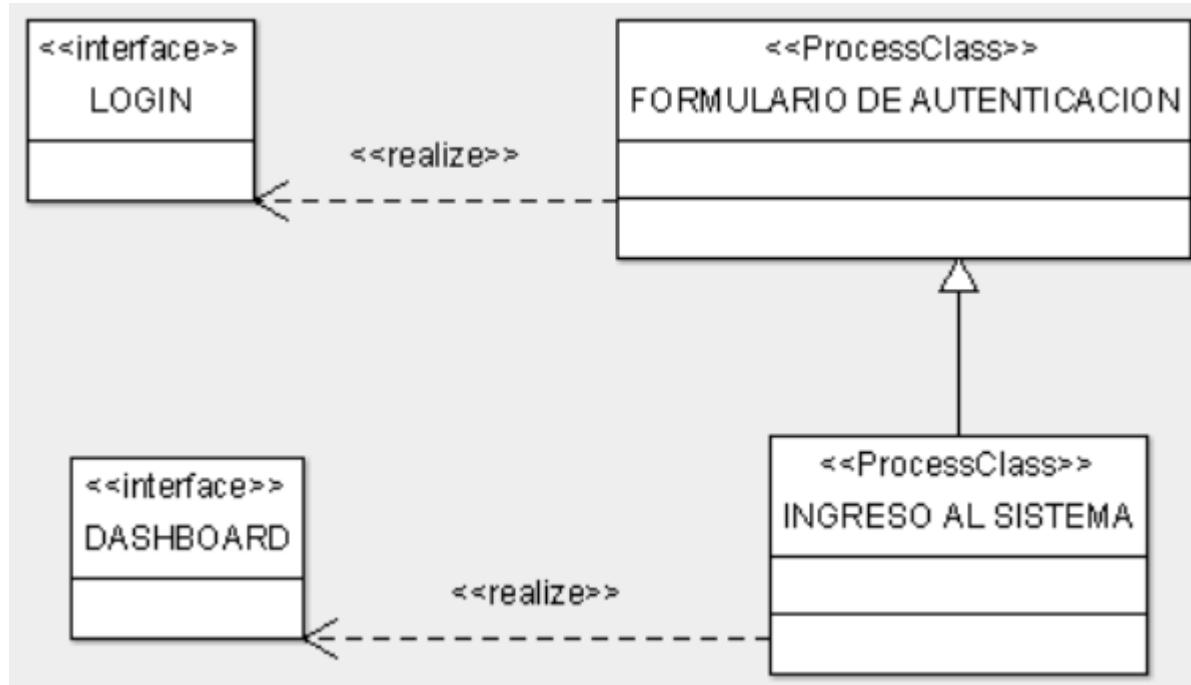
Tipo de Orden	Descripción
Borrador	Órdenes no acabadas o enviadas para validación.
Aprobadas	Órdenes validadas en estado aprobado.
Rechazadas	Órdenes validadas en estado rechazado.
Incompletas	Órdenes incompletas en su elaboración.
Pendientes	Órdenes pendientes de validación.
Ejecutadas	Órdenes ya recibidas, es decir, aquellas que tienen una nota de recepción asociada.
Eliminadas	Órdenes que han sido descartadas o desusadas.

- **Modelo Navegacional**

Para la gestión de usuarios y roles , se utilizará un modelo de navegación que permita una experiencia de usuario fluida y sencilla. Esto garantizará que la interacción con el sistema sea intuitiva para el usuario final.

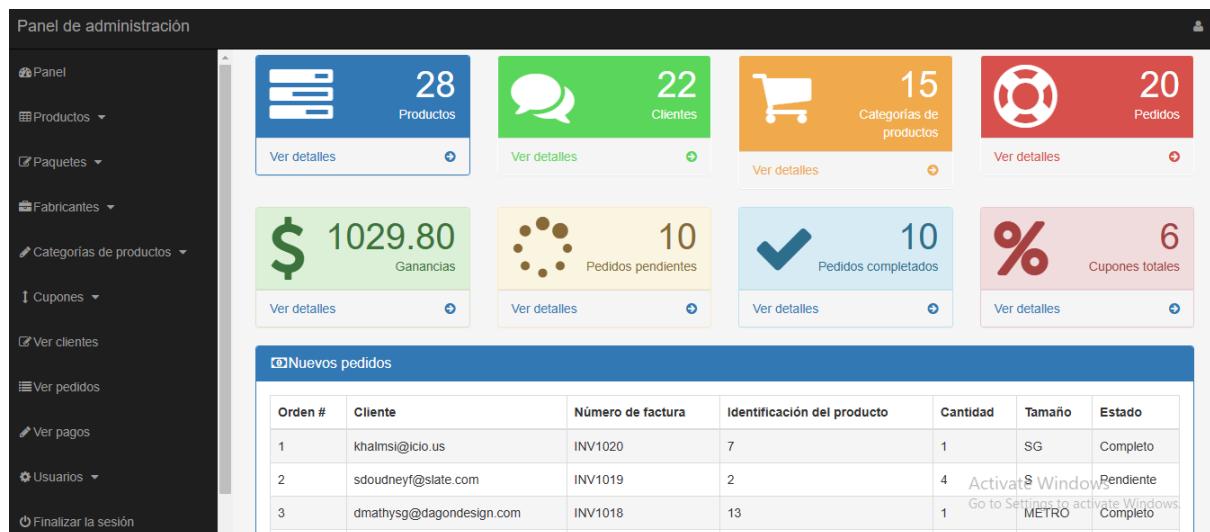


Se presenta el modelo de autenticación para el ingreso de nuevos usuarios al sistema, garantizando la seguridad y validación de los mismos antes de acceder a las funcionalidades.



● Modelo de Presentación

En el modelo de presentación se observa las capturas de pantalla para cada proceso que se realizó anteriormente.



El Panel de Administración del sistema de gestión de ventas e inventario es una interfaz centralizada que permite al administrador supervisar y controlar diversos aspectos clave del funcionamiento del sistema. Es el punto de partida para la gestión eficiente de las operaciones, proporcionando una visión clara y accesible de las métricas más importantes. Entre sus secciones se encuentran:

1. **Productos:** Muestra un resumen de todos los productos registrados, incluyendo detalles como la cantidad disponible en inventario, permitiendo al administrador hacer ajustes según sea necesario.
2. **Clientes:** Ofrece información sobre el número total de clientes registrados, con un seguimiento de su actividad reciente, facilitando la gestión de relaciones y la personalización de servicios.
3. **Categorías de Productos:** Muestra el número total de categorías creadas para organizar los productos, lo que ayuda a mantener un catálogo ordenado y de fácil acceso.
4. **Pedidos:** Proporciona información detallada sobre el estado y número total de pedidos, diferenciando entre los que están pendientes y los completados, lo que facilita la gestión de las entregas.
5. **Ganancias:** Muestra un desglose de las ganancias generadas en un período determinado, lo que permite al administrador realizar un seguimiento de los ingresos y analizar tendencias.
6. **Cupones Totales:** Indica la cantidad de cupones activos e inactivos, facilitando su control y la aplicación de descuentos.
7. **Nuevos Pedidos:** Ofrece una lista de los pedidos recientes que requieren atención inmediata, lo que permite al administrador priorizar y gestionar los pedidos de forma efectiva.

Este panel proporciona al administrador una herramienta completa para gestionar el inventario, las relaciones con los clientes, los pedidos y las finanzas del sistema, asegurando un control integral y una operación eficiente.

La creación de productos permite al administrador agregar nuevos artículos al catálogo de manera eficiente. A través de un formulario intuitivo, se ingresan detalles esenciales como nombre, descripción, precio, categoría, entre otros. Una vez completada la información, el producto se registra automáticamente en el sistema, actualizando el inventario y organizando los artículos por categorías para facilitar su acceso. Además, el administrador

puede editar o eliminar productos en cualquier momento, manteniendo el catálogo siempre actualizado y alineado con las necesidades del negocio.

Insertar productos

Título del producto	<input type="text"/>
URL del producto	<input type="text"/>
Ejemplo de URL de producto: camiseta azul marino	
Seleccione un fabricante	<input type="text"/> Seleccione un fabricante
Categoría de producto	<input type="text"/> Seleccione una categoría de producto
Precio del producto	<input type="text"/>
Precio de venta del producto	<input type="text"/>
Palabras clave del producto	<input type="text"/>
Pestañas de productos	<input type="text"/> Descripción del Producto <input type="text"/> Características del producto

Las categorías también tendrán su propia gestión, para realizar los cambios necesarios sin tener que afectar el funcionamiento del sistema. Permite al administrador organizar los productos de manera eficiente, facilitando la navegación y mejorando la experiencia de compra para los usuarios. El administrador puede crear, editar o eliminar categorías de productos, asignando a cada artículo la categoría que mejor se ajuste a su tipo, como "Camisetas", "Pantalones", "Accesorios", etc. Al agregar un producto, el sistema permite seleccionar de un listado de categorías existentes, asegurando que el catálogo esté bien estructurado. Esta funcionalidad también permite ordenar y filtrar productos rápidamente, tanto para los clientes en la tienda como para el administrador al gestionar el inventario.

Ver categorías de productos			
#	Nombre	Borrar	Editar
1	Suéter		
2	Camisetas		
3	Partes superiores		
4	Camisas		
5	Blusas		
6	Pantalones		
7	Vaqueros		
8	Bermudas		
9	Abrigos		
10	Cárdigans		
11	Ropa interior		
12	Chaquetas		

Activate Windows
Go to Settings to activate Windows.

Para la gestión de órdenes de compra en el sistema permite al administrador supervisar y controlar todo el proceso de los pedidos realizados por los usuarios. Cuando un cliente realiza una compra, el sistema genera automáticamente una orden que incluye detalles como los productos adquiridos, la cantidad, el precio total, la dirección de envío y el estado del pedido. El administrador puede consultar el estado de cada pedido (pendiente, terminado), actualizar la información en caso de cambios, y gestionar la logística de envío. Además, tiene la capacidad de gestionar pedidos recientes, priorizando aquellos que requieran atención inmediata. Este flujo de trabajo asegura que cada compra se procese de manera eficiente y que los clientes reciban sus productos en tiempo y forma.

Ver pedidos									
#	Cliente	Factura	Producto	Cantidad	Tamaño	Fecha de pedido	Importe total	Estado	Acción
1	pstrode2@omniture.com	INV1001	Vestido de flores	2	METRO	15 de noviembre de 2023 00:00:00	\$49,99	Pendiente	
2	rivingstone4@home.pl	INV1002	Chaqueta deportiva	1	yo	10 de diciembre de 2023 00:00:00	\$89,99	Terminado	
3	jturninini6@yolasite.com	INV1003	Suéter con cuello en V	3	S	22/10/2023 00:00:00	\$29,99	Pendiente	
4	dholde0@trellian.com	INV1004	Vestido de flores	1	METRO	18/09/2022 00:00:00	\$19,99	Terminado	
5	mbreinl9@usda.gov	INV1005	Zapatillas para correr	2	SG	30/07/2021 00:00:00	\$54,99	Pendiente	
6	mlevee@themeforest.net	INV1006	Sudadera con capucha gráfica	4	yo	25 de enero de 2023 00:00:00	\$44,99	Terminado	
7	apibworth7@home.pl	INV1007	Falda de lunares	1	METRO	05/06/2022 00:00:00	\$64,99	Pendiente	

Activate Windows
Go to Settings to activate Windows.

b) Revisión Sprint 1

En esta etapa se revisó el cumplimiento de las tareas planificadas en el Sprint 1. A continuación, se detallan las pruebas de funcionalidad correspondientes a distintas operaciones dentro de tu sistema de gestión de ventas e inventario.

Las pruebas de funcionalidad del módulo de gestión de personal se muestran en la siguiente tabla.

Prueba 1.1	Operación: Registro al personal
Precondición : <ul style="list-style-type: none">● Conexión al servidor que contiene el sistema.● Inicio de sesión correcta con el rol de administrador.	
Datos/proceso : <ul style="list-style-type: none">● Acceda al menú de administración y seleccione la opción Usuarios .● Seleccione la opción Insertar Usuario .● Ingresar los datos del nuevo miembro del personal, como nombre, cargo, dirección de correo electrónico válida, identificación (CI) y establecer un rol específico para él dentro del sistema.● Especificar al menos un referente personal si es necesario.● Confirmar y guardar los datos.	
Resultados esperados : <p>El personal se registra exitosamente con los datos correctos y se asigna un rol específico. El sistema debe asegurar que la dirección de correo electrónico sea válida y que se proporcione al menos un referente personal.</p>	
Postcondición : <p>El nuevo miembro del personal se registra correctamente en el sistema y puede acceder a las funciones asignadas según su rol.</p>	
Resultados obtenidos : <p>El registro del personal se realizó exitosamente, y los datos fueron guardados correctamente en el sistema.</p>	

Para el registro de un nuevo producto, como se observa en la tabla, se realizó la siguiente prueba de funcionalidad.

Prueba 1.2	Operación: Registro de un nuevo producto
Precondición :	<ul style="list-style-type: none">● Conexión al servidor que contiene el sistema.● Inicio de sesión correcta con el rol de administrador.
Datos/proceso :	<ul style="list-style-type: none">● Seleccione en el menú de administración la opción Productos .● Hacer clic en la opción Insertar Producto .● Ingresar los detalles del producto (nombre, descripción, precio, categoría, cantidad en inventario, imagen, etc.).● Seleccione la categoría correspondiente del producto.● Confirmar la creación del producto.
Resultados esperados :	El producto se registra exitosamente en el sistema con todos los datos correctos, incluida la categoría y subcategoría, y se asocia correctamente al inventario.
Postcondición :	El producto se registra correctamente en la base de datos y aparece en el catálogo de productos.
Resultados obtenidos :	El producto se registró exitosamente y fue añadido al catálogo.

La prueba de funcionalidad para la realización de la creación de categorías se observa en la siguiente tabla.

Prueba 1.3	Operación: Creación de Categorías
Precondición :	<ul style="list-style-type: none">● Conexión al servidor que contiene el sistema.● Inicio de sesión correcta con el rol de administrador.

Datos/proceso :

- Acceda al menú de administración.
- Seleccione la opción Categorías de Productos .
- Crea una nueva categoría o subcategoría, asignando un nombre y, si es necesario, una descripción.
- Guardar los cambios.

Resultados esperados :

La categoría o subcategoría se crea correctamente y se asocia a los productos correspondientes.

Postcondición :

La categoría o subcategoría se refleja en el sistema y está disponible para asignar a los productos.

Resultados obtenidos :

La categoría se creó correctamente y está disponible para la asignación de productos.

3.4. Post-Game

3.4.1. Diseño de Interfaces Gráficas

El diseño de las interfaces gráficas en el sistema fue desarrollado iterativamente a lo largo de cada sprint, y las pruebas de funcionalidad se realizaron durante las revisiones de cada fase. Se corrigieron errores de ortografía y el orden de los componentes de las interfaces, los cuales fueron solucionados y optimizados de forma inmediata, garantizando una experiencia de usuario fluida y coherente. La disposición de los elementos en cada interfaz fue ajustada para asegurar que fueran intuitivos y eficientes, mejorando la usabilidad del sistema.

3.4.2. Pruebas de Stress

Las pruebas de Stress se realizaron para evaluar el desempeño del sistema bajo condiciones de carga máxima, simulando el uso simultáneo de 100 usuarios. Este tipo de prueba permite medir las capacidades del sistema y las condiciones en las que puede operar al realizar múltiples peticiones y procesos de forma continua.

Para las pruebas de Stress, se utilizó el software Webserver Stress Tools 8, permitiendo evaluar áreas clave del sistema como el dashboard, la gestión de productos, las órdenes de compra, y las funciones de inicio de sesión. Los resultados obtenidos proporcionaron una visión clara de cómo el sistema responde a la carga y qué tan bien maneja las solicitudes en tiempo real.

URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	Login	100	0	0.00	4,963	236
2	Personal	100	0	0.00	5,117	301
3	Órdenes de Compra	99	0	0.00	7,461	324

En la tabla se observa que el inicio de sesión no presentó errores, lo que indica que todos los 100 usuarios pudieron ingresar al sistema sin problemas, demostrando que el sistema soportó eficientemente la carga durante la prueba.

3.5. Calidad

El enfoque utilizado para evaluar la calidad del sistema es la Metodología WebQem , que proporciona una metodología cuantitativa y sistemática tanto en la fase operativa como en la etapa de desarrollo. Esta metodología se emplea para medir el cumplimiento de los factores de calidad establecidos por el estándar ISO 9126, que abarca usabilidad , funcionalidad , confiabilidad y eficiencia . En este contexto, se aplica a la evaluación de la calidad del Sistema de Gestión de Ventas e Inventario .

3.5.1 Fases de WebQem

a) Definición de las metas de evaluación y selección del perfil de usuario

En esta fase se definen los objetivos de la evaluación y se selecciona el perfil de usuario adecuado. Las metas que se buscan cumplir incluyen:

- **Metas de Evaluación :**

- i) Comprender la percepción de calidad de los usuarios finales sobre el sistema.
- ii) Asegurar que la calidad del sistema supere las expectativas de los usuarios finales.

- **Selección de perfil de usuario :**

El objetivo es identificar las expectativas y necesidades del usuario final para el

diseño e implementación de un caso de estudio que evalúe el cumplimiento de los requisitos de calidad de la aplicación web operativa.

b) Definición de los requisitos de calidad y/o costo

Esta fase implica establecer atributos de calidad de usuario cuantificables que representen un modelo de calidad apropiado para el perfil seleccionado. Los atributos incluyen aspectos como la usabilidad, eficiencia y confiabilidad del sistema.

c) Definición de criterios de preferencia elementales y procedimientos de medición

Para evaluar los atributos de calidad, se definen criterios de medición que especifican cómo medir los atributos cuantificables. Esto permite obtener un indicador elemental que refleja el grado de satisfacción de los requisitos del sistema. Los criterios de evaluación se basan en una escala que categoriza la calidad de cada atributo.

CARACTERISTICAS DE CALIDAD	MALA	REGULAR	BUENO	MUY BUENO
Funcionalidad	0-30%	31-50%	51-90%	91-100%
Confiabilidad	0-30%	31-50%	51-90%	91-100%
Portabilidad	0-30%	31-50%	51-90%	91-100%
Total	0-30%	31-50%	51-90%	91-100%

d) Definición de estructuras de agregación e implementación de la evaluación global

Para obtener una evaluación global del sistema, se utiliza un mecanismo de agregación que combina las preferencias elementales de los diferentes atributos de calidad. El puntaje final se calcula conforme al modelo de agregación previamente definido, reflejando la calidad global del sistema en base a los criterios establecidos.

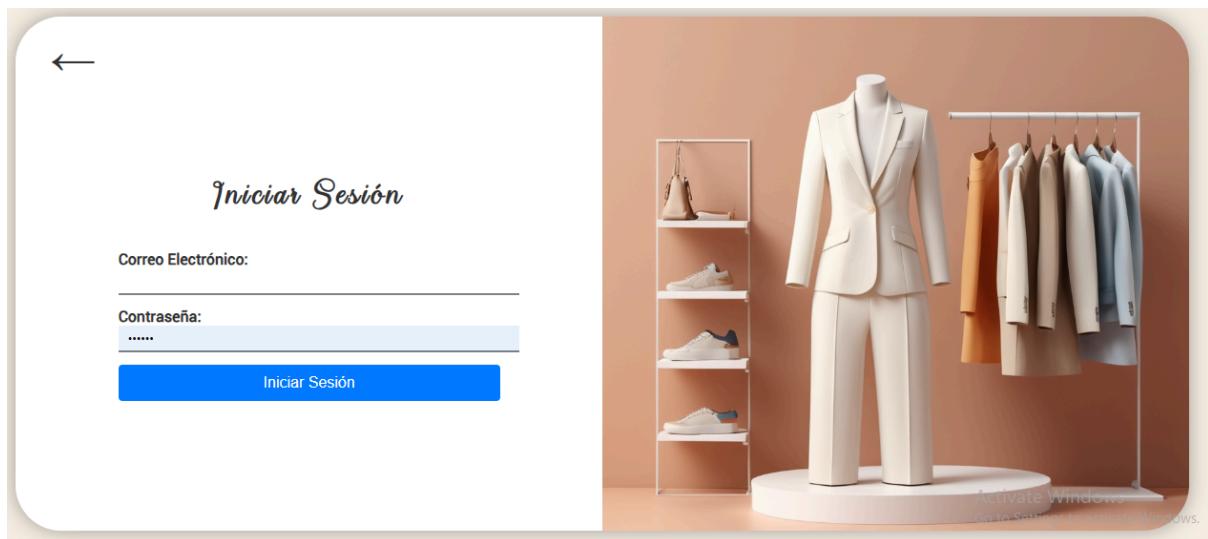
Este proceso de evaluación de calidad garantiza que el Sistema de Gestión de Ventas e Inventario cumple con los estándares de calidad necesarios para ofrecer una experiencia satisfactoria a los usuarios.

3. 6. Pantallas

Las pantallas son la interfaz principal del sistema de ventas, donde los usuarios interactúan con las diversas funcionalidades del sistema. Estas pantallas deben ser intuitivas, fáciles de usar y coherentes con el flujo de trabajo del sistema. En esta sección se detallan las pantallas principales, su propósito y los elementos clave que cada una debe contener.

3.6.1. Pantallas Principales del Sistema

1. Pantalla de Inicio de Sesión

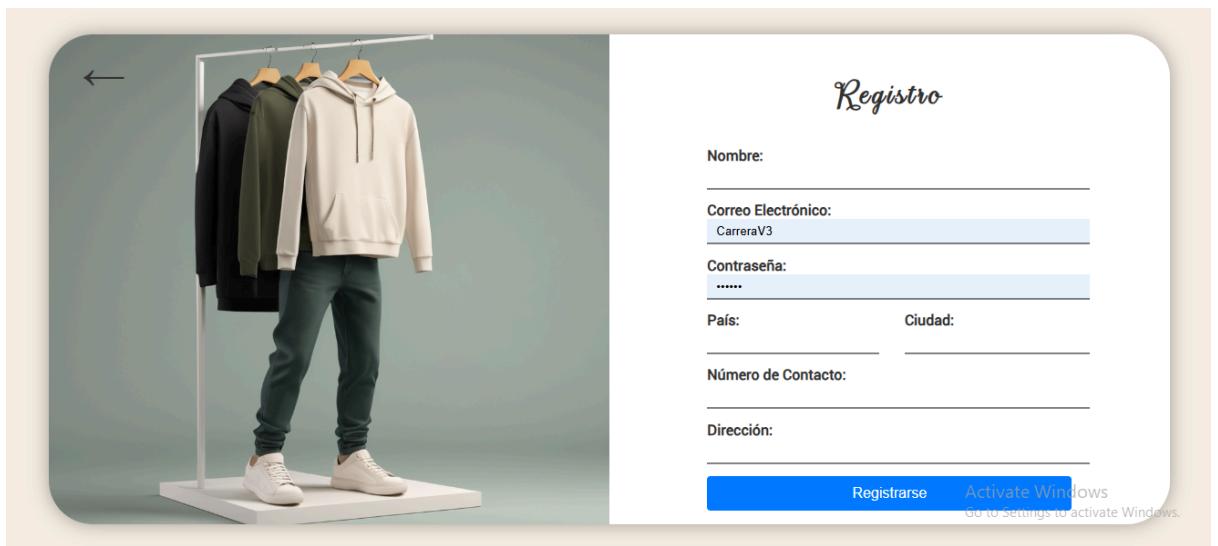


Su objetivo es permitir que los usuarios ingresen al sistema con sus credenciales.

Elementos clave:

- Campo de Correo para el ingreso del usuario.
- Campo de contraseña para la autenticación.
- Botón Iniciar sesión.

2. Pantalla de Registro de Usuario



Permit que los nuevos usuarios se registren en el sistema.

Elementos clave:

- Campos para ingresar Nombre, Correo, Contraseña, País, Ciudad, Número de contacto, Dirección.
- Botón de Registrarse.

3. Pantalla del Catálogo de Productos



Su objetivo es mostrar los productos disponibles para la venta y permitir que el usuario los agregue al carrito.

Elementos clave:

- Lista de productos con imagen, nombre, precio y descripción corta.
- Botón para añadir al carrito.
- Filtros para ordenar por categoría o precio.
- Paginación para mostrar más productos si es necesario.

4. Pantalla de Panel de Administración (para administradores)

The screenshot shows the Admin Dashboard interface. On the left is a sidebar with navigation links: Panel, Productos, Paquetes, Fabricantes, Categorías de productos, Cupones, Ver clientes, Ver pedidos, Ver pagos, Usuarios, and Finalizar la sesión. The main area has a header 'Panel de administración' and a user icon. Below the header are four cards: 'Productos' (28), 'Cientes' (22), 'Categorías de productos' (15), and 'Pedidos' (20). Below these are four more cards: 'Ganancias' (\$ 1029.80), 'Pedidos pendientes' (10), 'Pedidos completados' (10), and 'Cupones totales' (6). At the bottom is a table titled 'Nuevos pedidos' with columns: Orden #, Cliente, Número de factura, Identificación del producto, Cantidad, Tamaño, and Estado. The table contains three rows of data. A watermark for 'Activate Windows' is visible in the background.

Orden #	Cliente	Número de factura	Identificación del producto	Cantidad	Tamaño	Estado
1	khalmsi@icio.us	INV1020	7	1	SG	Completo
2	sdoudneyf@slate.com	INV1019	2	4	Activate Windows Go to Settings to activate Windows METRO	Pendiente
3	dmathysg@dagondesign.com	INV1018	13	1	METRO	Completo

Proporciona a los administradores una vista general para gestionar productos, usuarios y pedidos.

Elementos clave:

- Secciones para gestionar productos, gestionar pedidos y gestionar usuarios.
- Listados de productos, pedidos y usuarios con opciones para editar, eliminar o agregar nuevos elementos.

3.7. Pruebas del Sistema

El sistema de ventas debe pasar por un proceso riguroso de pruebas para garantizar su calidad, funcionalidad, seguridad y desempeño. En esta sección se describen los tipos de pruebas que se aplicarán, sus objetivos y la metodología utilizada para cada una.

3.7.1 Pruebas de Caja Negra

En estas pruebas, el evaluador se centra en los resultados de las entradas sin tener conocimiento del código interno del sistema. Su objetivo es verificar que el sistema responda correctamente a los datos de entrada y produzca las salidas esperadas.

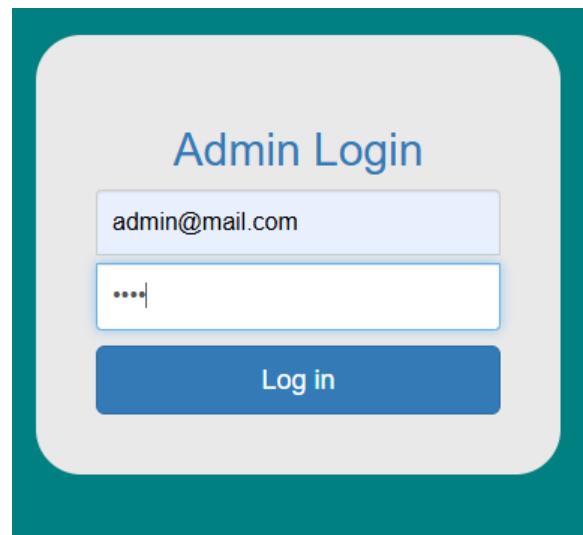
- Caso de Prueba: Inicio de Sesión

Del lado del usuario:

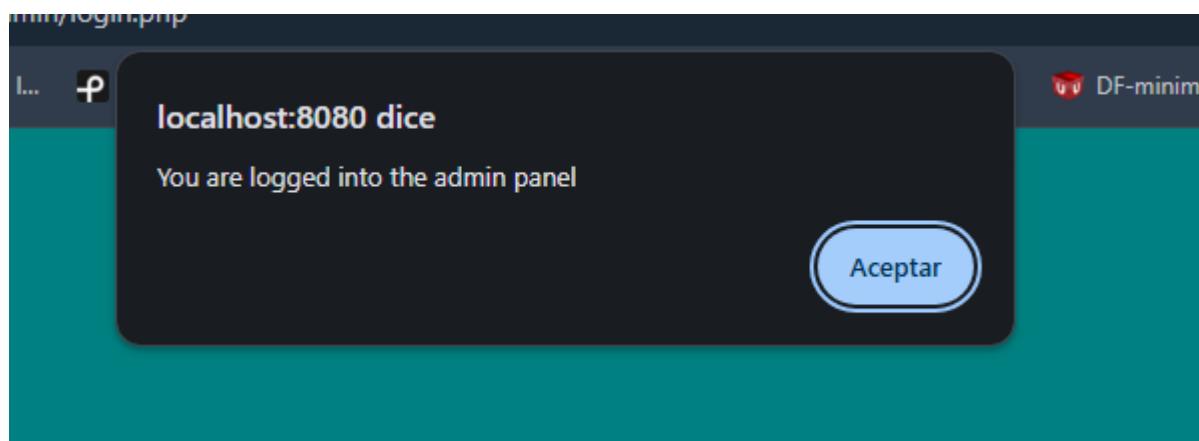


Del lado del administrador

Inicio de sesión con un correo electrónico y contraseña válidos.



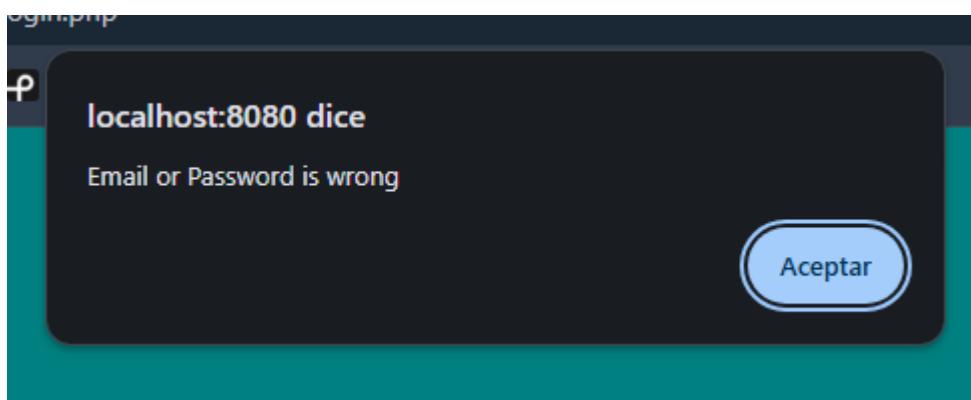
Validación de datos de inicio de sesión del administrador



Redirección exitosa al panel principal tras el inicio de sesión correcto.

A screenshot of the 'Panel de administración' (Admin Dashboard). The left sidebar contains navigation links like 'Panel', 'Productos', 'Paquetes', 'Fabricantes', 'Categorías de productos', 'Cupones', 'Ver clientes', 'Ver pedidos', 'Ver pagos', 'Usuarios', and 'Finalizar la sesión'. The main area shows various statistics in colored boxes: 28 Productos, 22 Clientes, 15 Categorías de productos, 20 Pedidos, 1029.80 Ganancias, 10 Pedidos pendientes, 10 Pedidos completados, and 6 Cupones totales. Below this is a table titled 'Nuevos pedidos' with columns: Orden #, Cliente, Número de factura, Identificación del producto, Cantidad, Tamaño, and Estado. Three new orders are listed: 1 (khalmsi@cio.us, INV1020, 7, SG, Completo), 2 (sdoudneyf@slate.com, INV1019, 2, S, Pendiente), and 3 (dmathysg@dagondesign.com, INV1018, 13, METRO, Completo).

Mensaje de error al ingresar credenciales incorrectas.



3.7.2 Pruebas de Caja Blanca

- **Definición:** Estas pruebas se enfocan en el análisis del código fuente para verificar su lógica y estructura.
- **Objetivo:** Garantizar que las rutas de código se ejecuten correctamente y que no existan errores en la lógica interna.
- **Ejemplo de análisis:**
 - Validación de los bucles para manejar pedidos.
 - Pruebas de condiciones y ramas lógicas en la autenticación.
 - Cobertura de funciones críticas, como el cálculo del total del carrito o la validación del stock.
- **Método:** Se utiliza una herramienta de análisis de código estático para identificar errores, verificar la cobertura y evaluar la calidad del código.

Análisis del flujo de datos

```

session_start();

use Illuminate\Database\Capsule\Manager as DB;

require_once __DIR__ . '/../vendor/autoload.php';
require_once __DIR__ . '/../database/db_connector.php';

if (!isset($_SESSION['admin_email'])) {
    echo "<script>window.open('login.php', '_self')</script>";
    exit;
}

```

1. Verificación de sesión:

- **Línea:** if (!isset(\$_SESSION['admin_email'])) { ... }
- **Condición:** Comprueba si `$_SESSION['admin_email']` está configurado.
- **Casos de prueba:**
 1. `$_SESSION['admin_email']` está configurado → Permitir acceso al panel.
 2. `$_SESSION['admin_email']` no está configurado → Redirigir a `login.php`.

2. Consulta de administrador en la base de datos:

```

// Obtener datos del Administrador

$adminSession = $_SESSION['admin_email'];
$admin = DB::table('admins')->where('admin_email', $adminSession)->first();

if (!$admin){
    echo "<script>window.open('login.php', '_self')</script>";
    exit;
}

```

- **Línea:** `$admin = DB::table('admins')->where('admin_email', $adminSession)->first();`
- **Condición:** Verifica si el administrador existe en la base de datos.
- **Casos de prueba:**
 1. El email de sesión (`$adminSession`) coincide con un registro en la tabla `admins` → Permitir acceso.

2. El email no coincide con ningún registro → Redirigir a `login.php`.

3. Inclusión de rutas dinámicas:

```
<div id="wrapper">
    <?php include("includes/sidebar.php"); ?>

    <div id="page-wrapper">
        <div class="container-fluid">
            <?php
                $routes = [
                    'dashboard' => 'dashboard.php',
                    'insert_product' => 'insert_product.php',
                    'view_products' => 'view_products.php',
                    'delete_product' => 'delete_product.php',
                    'edit_product' => 'edit_product.php',
                    'insert_p_cat' => 'insert_p_cat.php',
                    'view_p_cats' => 'view_p_cats.php',
                    // Agrega aquí las demás rutas
                ];

                foreach ($routes as $key => $file) {
                    if (isset($_GET[$key])) {
                        include($file);
                        break;
                    }
                }
            ?>
        </div>
    </div>
</div>
```

- **Línea:** `foreach ($routes as $key => $file) { ... }`
- **Condición:** Comprueba si una clave de ruta (`$_GET[$key]`) está presente.
- **Casos de prueba:**
 1. `$_GET['dashboard']` está configurado → Incluir `dashboard.php`.
 2. `$_GET['insert_product']` está configurado → Incluir `insert_product.php`.
 3. `$_GET` no tiene ninguna clave válida → No se incluye ningún archivo.

Casos de prueba (simulación)

Aquí están los casos diseñados para probar las condiciones anteriores:

Caso de prueba	Entrada (variables o estado)	Salida esperada
----------------	------------------------------	-----------------

Sesión activa	<code>\$_SESSION['admin_email']</code> configurado	Continúa con la ejecución del script.
Sesión inactiva	<code>\$_SESSION['admin_email']</code> no configurado	Redirige a <code>login.php</code> .
Email administrador válido	<code>\$adminSession</code> coincide con un registro	Continúa con la ejecución del script.
Email administrador inválido	<code>\$adminSession</code> no coincide con ningún registro	Redirige a <code>login.php</code> .
Ruta válida	<code>\$_GET['dashboard'] = true</code>	Incluir el archivo <code>dashboard.php</code> .
Ruta inválida	Ninguna clave de <code>\$_GET</code> coincide con <code>\$routes</code>	No se incluye ningún archivo, solo carga la plantilla base.

3.7.3. Estimación de Esfuerzo y Tiempo de Desarrollo usando el Modelo COCOMO

Para realizar la estimación del esfuerzo y el tiempo de desarrollo de un proyecto de software, se ha utilizado el modelo COCOMO (Constructive Cost Model). Este modelo fue creado por Barry Boehm en 1981 y se basa en la cantidad de líneas de código (LOC) para hacer las estimaciones de esfuerzo y tiempo. En este caso, se ha trabajado con un proyecto que tiene un total de 2300 líneas de código (LOC) aproximadamente, y se utilizará el modelo COCOMO Básico para realizar la estimación.

El modelo COCOMO es conocido por su simplicidad y por ser fácil de aplicar. Sin embargo, no toma en cuenta todos los factores posibles que pueden influir en un proyecto de software,

como la experiencia del equipo de desarrollo, los requisitos de hardware o las expectativas del cliente.

Fórmulas del Modelo COCOMO Básico

La fórmula básica del modelo COCOMO es:

$$E = a * (KLOC)^b$$

Donde:

- **E**: Esfuerzo estimado en persona-meses.
- **KLOC**: Número de miles de líneas de código (en este caso, 2300 líneas = 2.3 KLOC).
- **a y b**: Constantes dependientes del tipo de proyecto.

Para este caso, el proyecto se considera de tipo **Semidetallado**, lo que nos lleva a utilizar los siguientes valores para **a** y **b**:

- **a** = 3.0
- **b** = 1.12

La fórmula de tiempo de desarrollo es:

$$TDEV = 3.0 * (E)^{0.33}$$

Donde:

- **TDEV**: Tiempo de desarrollo estimado en meses.
- **E**: Esfuerzo estimado en persona-meses.

Finalmente, el número de personas en el equipo de desarrollo se calcula de la siguiente manera:

$$P = E / TDEV$$

Donde:

- **P**: Número de personas en el equipo de desarrollo.

Cálculos y Resultados

1. Estimación del Esfuerzo (E):

Sustituyendo los valores en la fórmula de esfuerzo:

$$E = 3.0 * (2.3)^{1.12}$$

$$E \approx 3.0 * 2.581 \approx 7.74 \text{ persona-meses}$$

Esto nos da un esfuerzo estimado de 7.74 persona-meses.

2. Estimación del Tiempo de Desarrollo (TDEV):

Usamos la fórmula del tiempo de desarrollo:

$$TDEV = 3.0 * (7.74)^{0.33}$$

$$TDEV \approx 3.0 * 1.987 \approx 5.96 \text{ meses}$$

Esto nos da un tiempo de desarrollo estimado de 5.96 meses.

3. Estimación del Número de Personas (P):

Usando la fórmula de número de personas en el equipo:

$$P = E / TDEV$$

$$P = 7.74 / 5.96 \approx 1.30 \text{ personas}$$

Esto indica que se necesitaría un equipo de 1.30 personas para completar el proyecto dentro del tiempo estimado.

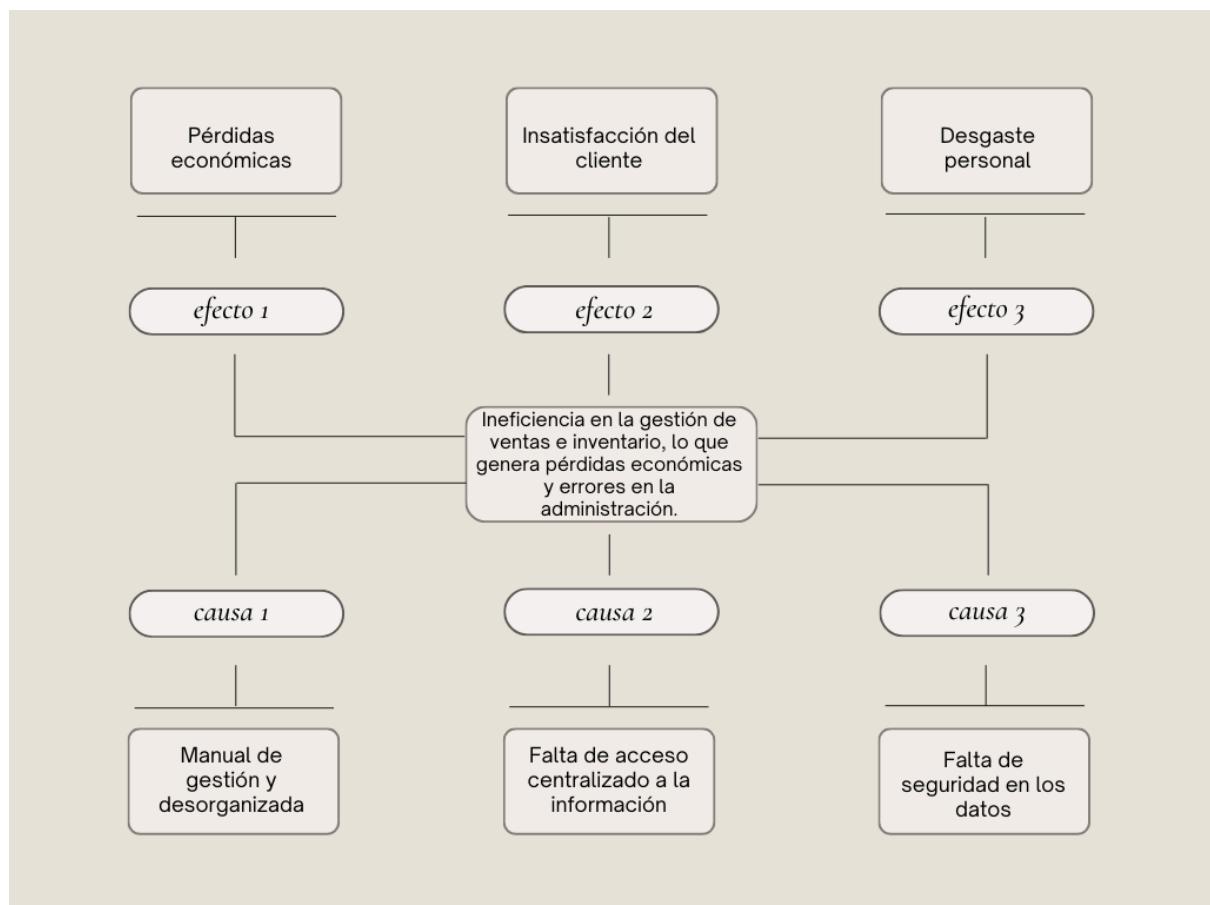
Resumen de la Estimación

- Esfuerzo estimado: 7.74 persona-meses
- Tiempo de desarrollo estimado: 5.96 meses
- Número de personas en el equipo: 1.30 personas

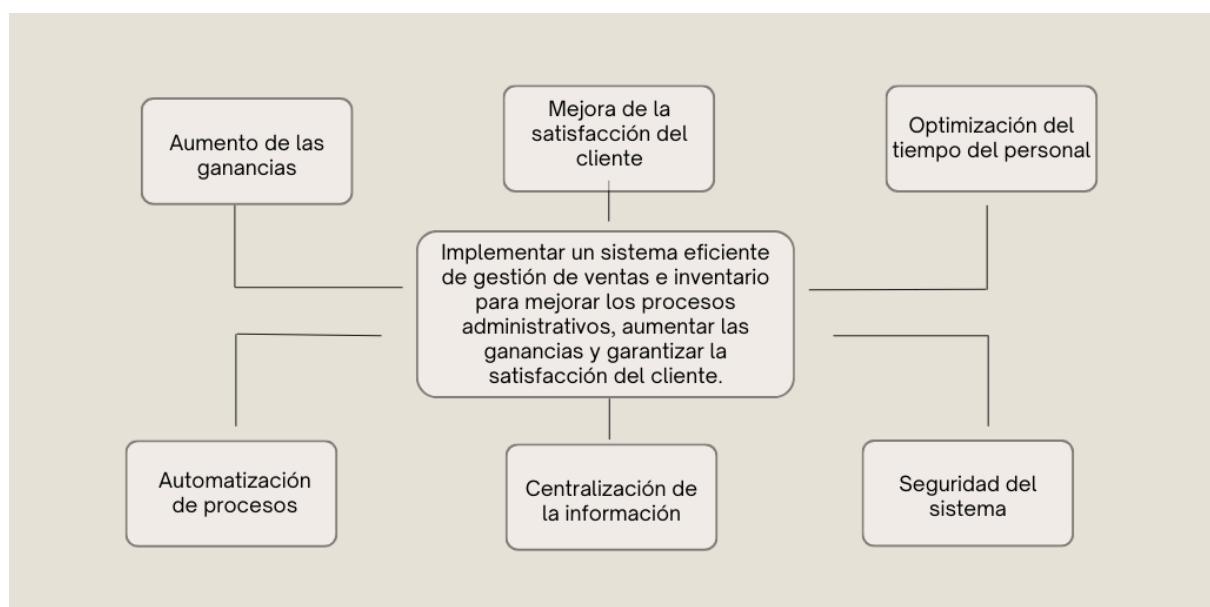
Este cálculo se ha realizado utilizando el modelo COCOMO Básico. Cabe señalar que esta estimación no toma en cuenta factores adicionales como la complejidad del software, la experiencia del equipo de desarrollo o los requisitos del cliente. Para un cálculo más detallado y preciso, se podrían considerar estos factores adicionales utilizando el modelo.

ANEXOS

ANEXO N. 1. Árbol de Problemas



ANEXO N. 2. Árbol de Objetivos



ANEXO N. 3. Diccionario de Datos

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
customers	customer_id	SERIAL	-	Identificador único del cliente	PRIMARY KEY	-	1
	customer_name	VARCHAR	255	Nombre del cliente	NOT NULL	-	Jane Doe
	customer_email	VARCHAR	255	Correo electrónico del cliente	UNIQUE, NOT NULL	-	jane.doe@example.com
	customer_pass	VARCHAR	255	Contraseña del cliente	NOT NULL	-	password456
	customer_country	TEXT	-	País del cliente	NOT NULL	-	Canadá
	customer_city	TEXT	-	Ciudad del cliente	NOT NULL	-	Toronto
	customer_contact	VARCHAR	255	Contacto del cliente	-	-	987654321
	customer_address	TEXT	-	Dirección del cliente	-	-	123 Main St
	customer_image	TEXT	-	Imagen del cliente	-	-	/images/customer1.jpg

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
admins	admin_id	SERIAL	-	Identificador único del administrador	PRIMARY KEY	-	1
	admin_name	VARCHAR	255	Nombre del administrador	NOT NULL	-	John Doe
	admin_email	VARCHAR	255	Correo electrónico del administrador	UNIQUE, NOT NULL	-	admin@example.com
	admin_pass	VARCHAR	255	Contraseña del administrador	NOT NULL	-	password123
	admin_image	TEXT	-	Imagen del administrador	NOT NULL	-	/images/admin1.jpg
	admin_contact	VARCHAR	255	Contacto del administrador	NOT NULL	-	123456789
	admin_country	TEXT	-	País del administrador	NOT NULL	-	USA
	admin_job	VARCHAR	255	Puesto de trabajo del administrador	NOT NULL	-	Manager

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
wishlist	wishlist_id	SERIAL	-	Identificador único de la lista de deseos	PRIMARY KEY	-	1
	customer_id	INT	-	Identificador del cliente	NOT NULL	customer_id	1
	product_id	INT	-	Identificador del producto	NOT NULL	product_id	101

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
payments	payment_id	SERIAL	-	Identificador único del pago	PRIMARY KEY	-	1
	invoice_no	VARCHAR	50	Número de factura relacionado	NOT NULL	(invoice_no)	INV12345

	amount	DECIMAL	10,2	Monto pagado	NOT NULL	-	150.00
	payment_mode	TEXT	-	Método de pago	NOT NULL	-	Tarjeta de crédito
	ref_no	VARCHAR	50	Número de referencia del pago	NOT NULL	-	REF123456
	code	VARCHAR	50	Código asociado al pago	NOT NULL	-	CODE123
	payment_date	TIMESTAMP	-	Fecha del pago	NOT NULL	-	##########

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
product_categories	p_cat_id	SERIAL	-	Identificador único de la categoría	PRIMARY KEY	-	10
	p_cat_title	TEXT	-	Título de la categoría	NOT NULL	-	Ropa
	p_cat_top	TEXT	-	Categoría destacada	-	-	Si

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
	order_id	SERIAL	-	Identificador único del pedido pendiente	PRIMARY KEY	-	1

pending_orders	customer_id	INT	-	Identificador del cliente relacionado	NOT NULL	(customer_id)	1
	invoice_no	VARCHAR	50	Número de factura	NOT NULL	(invoice_no)	INV12345
	product_id	INT	-	Identificador del producto relacionado	NOT NULL	(product_id)	101
	qty	INT	-	Cantidad de productos	NOT NULL	-	2
	size	TEXT	-	Tamaño del producto	NOT NULL	-	L
	order_status	TEXT	-	Estado del pedido	NOT NULL	-	Pendiente/Completado

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
products	product_id	SERIAL	-	Identificador único del producto	PRIMARY KEY	-	101

	p_cat_id	INT	-	Identificador de la categoría de producto	NOT NULL	(p_cat_id)	10
	manufacturer_id	INT	-	Identificador del fabricante	NOT NULL	(manufacturer_id)	1
	product_title	TEXT	-	Título del producto	NOT NULL	-	Camiseta
	product_url	TEXT	-	URL del producto	NOT NULL	-	/products/camiseta
	product_img1	TEXT	-	Imagen principal del producto	NOT NULL	-	/images/product1.jpg
	product_stock	INT	-	Cantidad en inventario del producto	NOT NULL	-	50
	product_price	DECIMAL	10,2	Precio del producto	NOT NULL	-	20.00
	product_psp_price	DECIMAL	10,2	Precio promocional del producto	-	-	15.00
	product_desc	TEXT	-	Descripción del producto	NOT NULL	-	Camiseta de algodón 100%
	product_features	TEXT	-	Características del producto	-	-	Ligera, cómoda
	product_keywords	TEXT	-	Palabras clave del producto	-	-	ropa, camiseta, algodón
	product_label	TEXT	-	Etiqueta del producto	NOT NULL	-	Nuevo
	status	VARCHAR	255	Estado del producto	-	-	Activo

	date	TIMESTAMP	-	Fecha de creación del producto	NOT NULL, DEFAULT CURRENT_TIMESTAMP	-	##########
	product_user_type	VARCHAR	50	Tipo de usuario para el producto	-	-	Adulto

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
customer_orders	order_id	SERIAL	-	Identificador único del pedido	PRIMARY KEY	-	1
	customer_id	INT	-	Identificador del cliente relacionado	NOT NULL	(customer_id)	1
	due_amount	DECIMAL	10,2	Cantidad adeudada	NOT NULL	-	150.00
	invoice_no	VARCHAR	50	Número de factura	UNIQUE, NOT NULL	-	INV12345

	qty	INT	-	Cantidad de productos	NOT NULL	-	3
	size	TEXT	-	Tamaño del producto	NOT NULL	-	M
	order_date	TIMESTAMP	-	Fecha del pedido	NOT NULL, DEFAULT CURRENT_TIMESTAMP	-	##########
	order_status	TEXT	-	Estado del pedido	NOT NULL	-	Pendiente

Tabla	Campo	Tipo de Dato	Longitud	Descripción	Restricciones	Relaciones	Ejemplo
manufacturers	manufacturer_id	SERIAL	-	Identificador único del fabricante	PRIMARY KEY	-	1
	manufacturer_title	TEXT	-	Nombre del fabricante	NOT NULL	-	Nike
	manufacturer_top	TEXT	-	Fabricante destacado	-	-	Sí
	manufacturer_image	TEXT	-	Imagen del fabricante	-	-	/images/manufacturer1.jpg

ANEXO N. 4. Calendarización

ANEXO N. 5. Tareas de Pruebas y Mantenimiento

TAREAS DE MANTENIMIENTO

1. Actualizar las dependencias y las librerías, como bibliotecas y frameworks para evitar problemas de compatibilidad:

- Revisión periódica de bibliotecas y frameworks para identificar versiones desactualizadas.
- Implementación de actualizaciones asegurándose de que no afecten la funcionalidad del sistema.
- Realización de pruebas de regresión tras cada actualización para verificar la compatibilidad.

2. Realizar pruebas de carga para escenarios de alto tráfico:

- Simulación de picos de tráfico.
- Identificación de cuellos de botella en el rendimiento.
- Generación de reportes para analizar los resultados y planificar mejoras.

3. Actualización de la documentación del código tras la implementación de modificaciones en el software:

- Uso de herramientas de documentación automatizada .
- Registro de cambios significativos en un sistema de control de versiones, como Git.
- Creación de diagramas actualizados que reflejen las modificaciones en el software.

4. Implementación de parches de seguridad para vulnerabilidades identificadas:

- Monitoreo continuo de alertas de seguridad de dependencias y frameworks.
- Aplicación de parches recomendados y ejecución de pruebas de penetración para garantizar que las vulnerabilidades están mitigadas.
- Configuración de herramientas de monitoreo para detectar nuevas amenazas.

5. Adicionar Scripts para crear backups de datos que se actualizan frecuentemente:

- Desarrollo de scripts automatizados que generen copias de seguridad incrementales y completas según la frecuencia establecida.

- Almacenamiento seguro de las copias en sistemas en la nube o servidores externos.
- Pruebas regulares de restauración para garantizar la integridad de los backups.

6. Monitoreo del rendimiento del sistema:

- Configuración de alertas para detectar tiempos de respuesta altos o errores frecuentes.

7. Optimización de la base de datos:

- Revisión periódica de índices y consultas SQL para mejorar el rendimiento.
- Limpieza de datos redundantes y actualización de los esquemas de la base de datos según las necesidades del negocio.

8. Auditoría del sistema y revisión de logs:

- Análisis de registros del sistema para detectar patrones sospechosos o errores frecuentes.
- Implementación de un sistema de análisis automatizado para enviar alertas ante anomalías.

9. Capacitación al equipo de soporte técnico:

- Actualización periódica de los conocimientos del equipo sobre nuevas funciones del sistema.
- Creación de guías rápidas y manuales de usuario para el personal operativo.

10. Revisión de la experiencia del usuario (UX):

- Análisis de comentarios de los clientes para identificar posibles mejoras.
- Pruebas de usabilidad para simplificar procesos y mejorar la interfaz gráfica.

11. Evaluación y actualización de la infraestructura del servidor:

- Revisión periódica de la capacidad del servidor para manejar el tráfico actual y proyectado.
- Implementación de soluciones escalables como balanceadores de carga o servicios en la nube.
- Actualización de hardware o software del servidor según sea necesario.

12. Control de integridad de datos:

- Implementación de herramientas que verifiquen la consistencia y exactitud de los datos almacenados.
- Automatización de procesos para detectar y corregir datos corruptos o inconsistencias.
- Generación de reportes que muestren el estado de la integridad de los datos en intervalos regulares.

TAREAS DE PRUEBAS

1. Ejecutar pruebas funcionales para verificar el cumplimiento de los requisitos del cliente:

- Verificar que todas las funciones del sistema cumplan con los requisitos establecidos.
- Probar casos de uso principales, como el proceso de compra, manejo de inventario y generación de facturas.

2. Realizar pruebas de integración para asegurar la interoperabilidad entre módulos:

- Evaluar la interacción entre módulos del sistema, así como la comunicación entre la base de datos y la interfaz de usuario.
- Comprobar la integración con servicios externos, como pasarelas de pago o sistemas de envío.

3. Ejecutar pruebas de carga y estrés para medir el rendimiento del sistema:

- Realizar pruebas de carga para evaluar la capacidad del sistema bajo diferentes niveles de tráfico.
- Ejecutar pruebas de estrés para determinar el punto de falla y el comportamiento del sistema bajo condiciones extremas.

4. Ejecutar pruebas de regresión para validar que las actualizaciones no afecten funcionalidades existentes:

- Repetir pruebas funcionales después de implementar cambios o actualizaciones para asegurar que no se hayan introducido errores.

5. Realizar pruebas de seguridad para identificar vulnerabilidades en el sistema:

- Ejecutar pruebas de penetración para identificar vulnerabilidades en el sistema.
- Evaluar la seguridad de los datos sensibles, como información de usuarios y transacciones.

6. Validación de informes y precisión de datos :

- Verificar que los informes generados contengan datos precisos y actualizados.
- Probar diferentes escenarios para garantizar la coherencia de los datos mostrados en los informes.

7. Realizar pruebas de usabilidad para garantizar una experiencia de usuario óptima y accesible :

- Evaluar la experiencia del usuario mediante pruebas con participantes reales.
- Identificar problemas de navegación, diseño o accesibilidad.

8. Pruebas de recuperación para verificar la capacidad del sistema para superar fallos inesperados :

- Simular fallos en el sistema, como cortes de energía o pérdida de conexión, y verificar la capacidad de recuperación.
- Probar la restauración de datos desde los backups.

9. Pruebas de aceptación del usuario (UAT):

- Involucrar a usuarios finales para validar que el sistema cumple con las expectativas y necesidades del negocio.
- Recopilar retroalimentación para posibles mejoras.

10. Documentación de casos de prueba y resultados obtenidos :

- Crear y mantener un registro detallado de los casos de prueba ejecutados, incluyendo pasos, datos utilizados y resultados esperados.
- Archivar los resultados de las pruebas para referencia futura y cumplimiento de auditorías.

11. Pruebas de datos en tiempo real para comprobar la actualización constante y la sincronización de información:

- Comprobar la actualización en tiempo real de datos como inventario, precios y transacciones.
- Asegurar la sincronización de datos entre diferentes módulos del sistema.

12. Revisión de errores y resolución :

- Registrar los errores detectados durante las pruebas.
- Confirmar la resolución de cada error ejecutando nuevamente los casos de prueba relacionados antes de avanzar con otras actividades.

ANEXO N.6. Reportes



Reporte de Productos Más Vendidos

Email: admin@mail.com
Reporte generado el: 17-01-2025 23:29:41

Categoría	Producto	Cantidad Vendida	Ingresos Totales
Sweater	Graphic Hoodie	4	199.96 USD
Shorts	Denim Jacket	2	159.98 USD
Skirts	Winter Coat	1	120.99 USD
Sweater	Graphic Sweater	2	99.98 USD
Pants	Formal Shirt	2	69.98 USD
Pants	V-Neck Sweater	2	69.98 USD
Cardigans	Sports Jacket	1	59.99 USD
Blouses	Floral Dress	1	45.99 USD
Dresses	Yoga Leggings	1	25.99 USD
T-shirts	Graphic Tee	1	19.99 USD

CHIRK S.R.L. | Reporte Generado Automáticamente



Reporte de Clientes con Mayor Actividad

Email: admin@mail.com
Reporte generado el: 17-01-2025 23:30:05

Cliente	Email	Total Compras (USD)	Total Órdenes
Muffin	mlevee@themeforest.net	199.96 USD	1
Belva	bwhifec@arlsteer.com	159.98 USD	1
Kilian	khalmsi@icio.us	120.99 USD	1
Caspar	cleng1@nytimes.com	99.98 USD	1
Brita	bdyeb@who.int	69.98 USD	1
Nixie	nslaughter8@sitemeter.com	69.98 USD	1
Rorie	rlivingstone4@home.pl	59.99 USD	1
Dael	dholde0@trellian.com	45.99 USD	1
Donnamarie	dmaffei3@weibo.com	25.99 USD	1
Dominique	dmathysg@dagondesign.com	19.99 USD	1



Reporte de Productos Pendientes por Enviar

Email: admin@mail.com

Reporte generado el: 17-01-2025 23:30:29

Producto	Pedidos Pendientes	Cantidad Pendiente
Denim Jeans	1	4
V-Neck Sweater	1	3
Leather Wallet	1	3
Plaid Scarf	1	3
Floral Dress	1	2
Running Shoes	1	2
Polka Dot Skirt	1	1
Casual T-Shirt	1	1
Bomber Jacket	1	1
Track Jacket	1	1

CHIRK S.R.L. | Reporte Generado Automáticamente



Reporte de Ventas por Categoría

Email: admin@mail.com

Reporte generado el: 17-01-2025 23:30:47

Categoría	Total de Productos	Ingresos Totales	Precio Promedio
Sweater	2	299.94 USD	49.99 USD
Shorts	1	159.98 USD	79.99 USD
Pants	2	139.96 USD	34.99 USD
Skirts	1	120.99 USD	120.99 USD
Cardigans	1	59.99 USD	59.99 USD
Blouses	1	45.99 USD	45.99 USD
Dresses	1	25.99 USD	25.99 USD
T-shirts	1	19.99 USD	19.99 USD



Reporte de Inventario por Fabricante

Email: admin@mail.com

Reporte generado el: 17-01-2025 23:31:03

Fabricante	Total de Productos	Precio Mínimo	Precio Máximo
Adidas	4	15.99 USD	69.99 USD
Calvin Klein	3	34.99 USD	59.99 USD
Gucci	3	19.99 USD	44.99 USD
H&M	3	39.99 USD	69.99 USD
Levis	4	25.99 USD	120.99 USD
Nike	3	19.99 USD	120.00 USD
Prada	3	49.99 USD	79.99 USD
Tommy Hilfiger	3	19.99 USD	89.99 USD
Zara	3	24.99 USD	45.99 USD



Reporte de Inventario por Fabricante

Email: admin@mail.com

Reporte generado el: 17-01-2025 23:31:19

Fabricante	Total de Productos	Precio Mínimo	Precio Máximo
Adidas	4	15.99 USD	69.99 USD
Calvin Klein	3	34.99 USD	59.99 USD
Gucci	3	19.99 USD	44.99 USD
H&M	3	39.99 USD	69.99 USD
Levis	4	25.99 USD	120.99 USD
Nike	3	19.99 USD	120.00 USD
Prada	3	49.99 USD	79.99 USD
Tommy Hilfiger	3	19.99 USD	89.99 USD
Zara	3	24.99 USD	45.99 USD

ANEXO N.7. Diagramas UML

Diagrama de clases

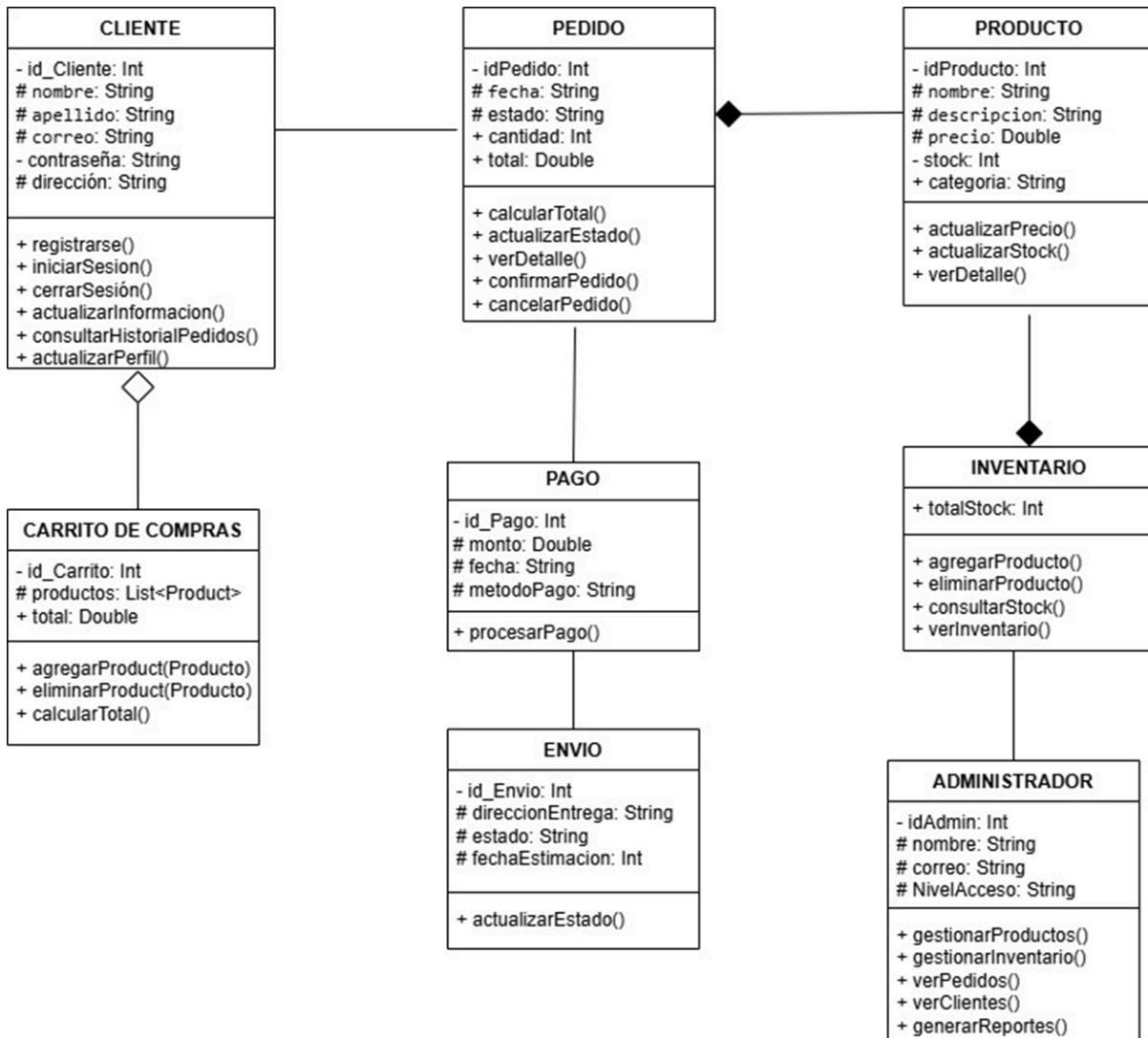


Diagrama de Objetos

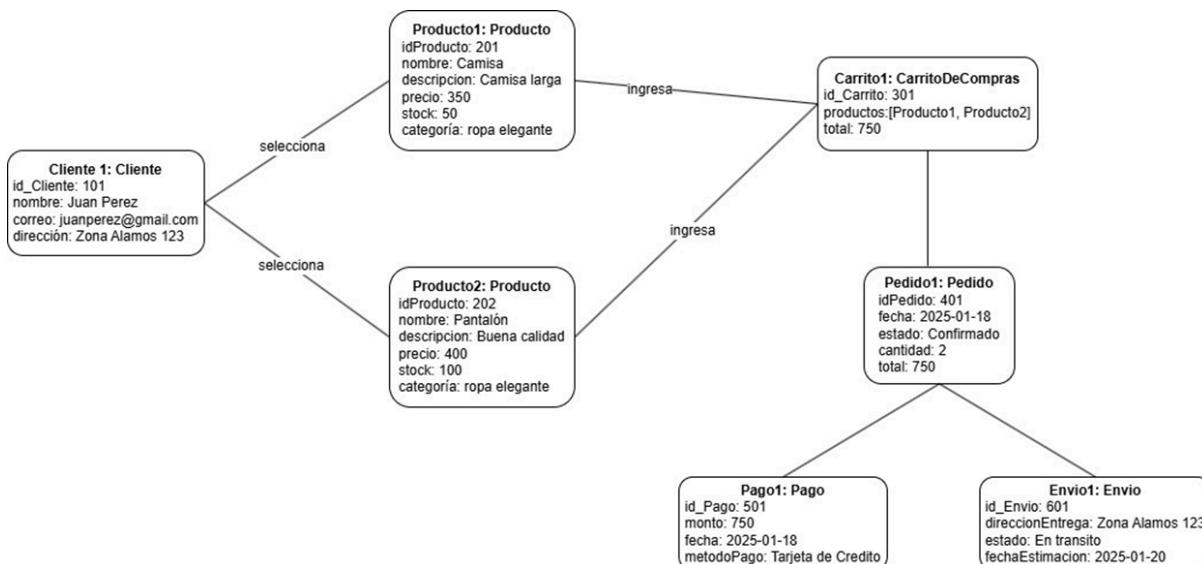


Diagrama de Componentes

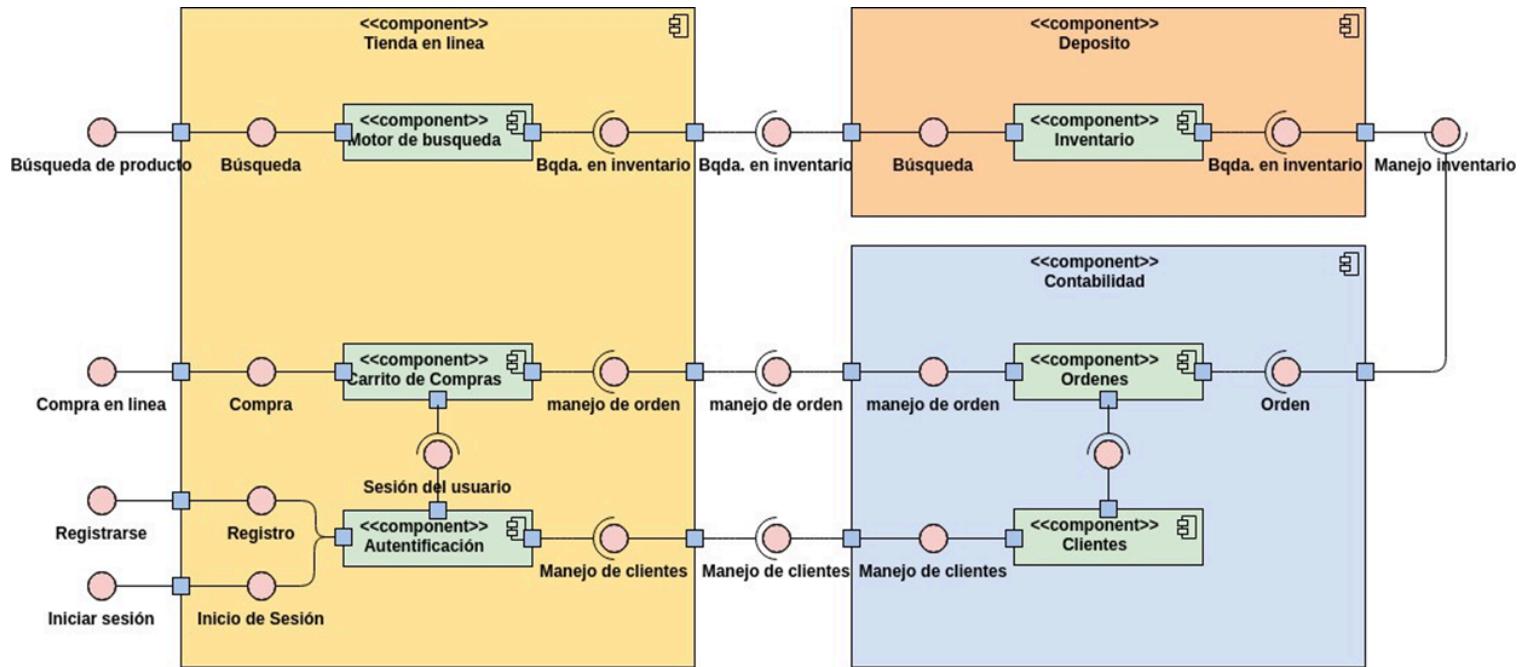
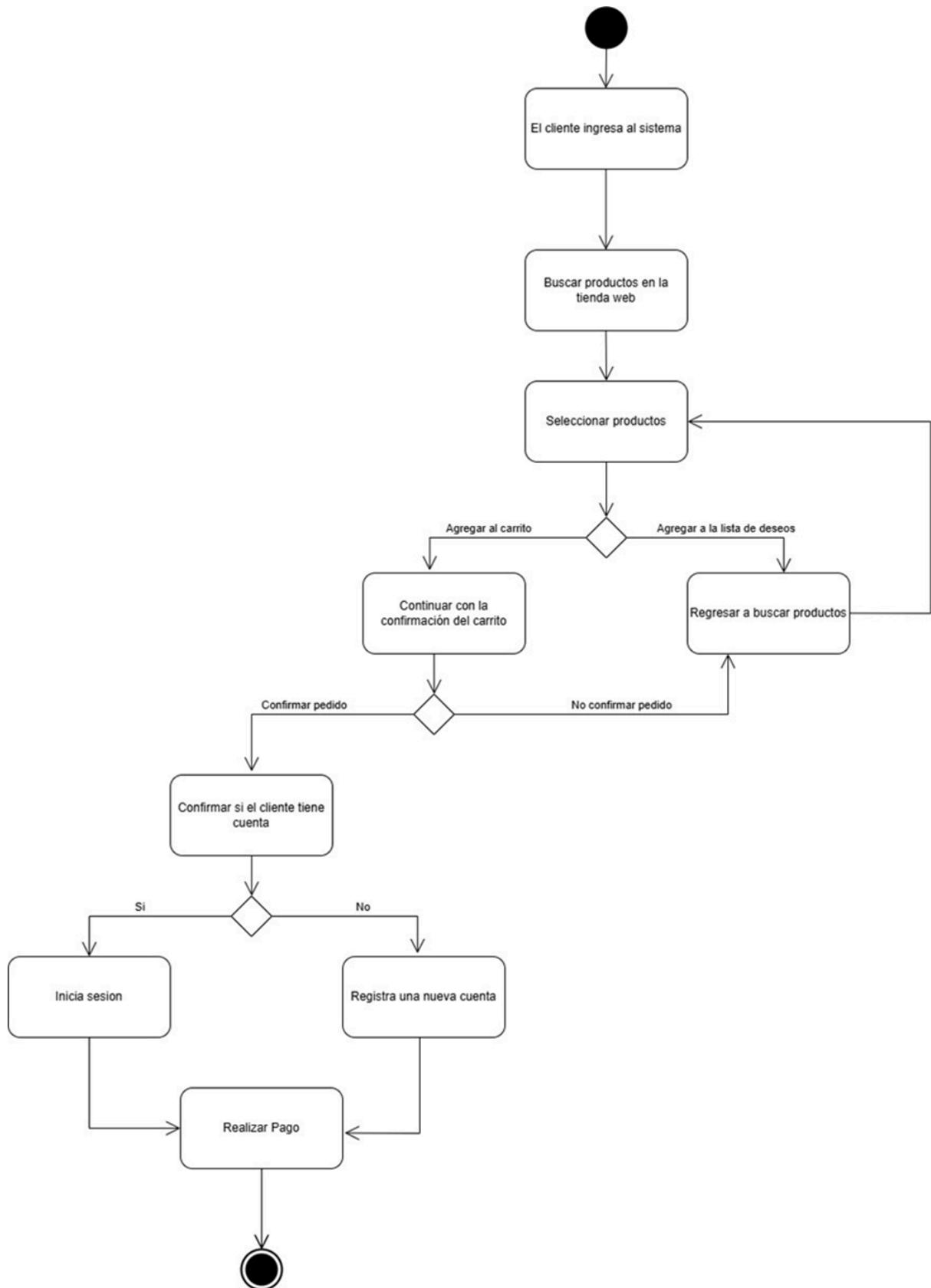


Diagrama de Actividades



Casos de Uso

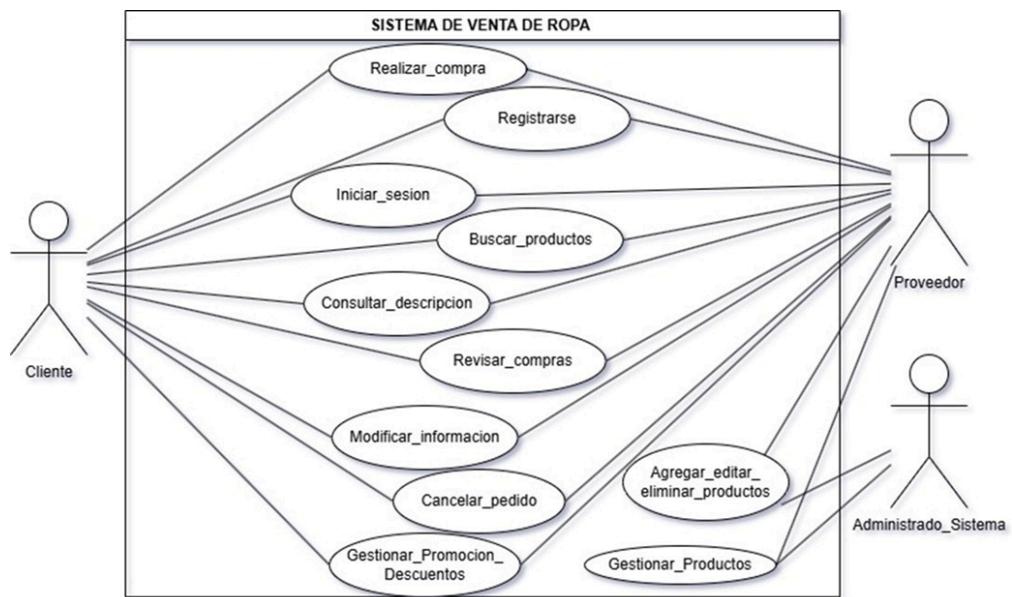


Diagrama de Interacción

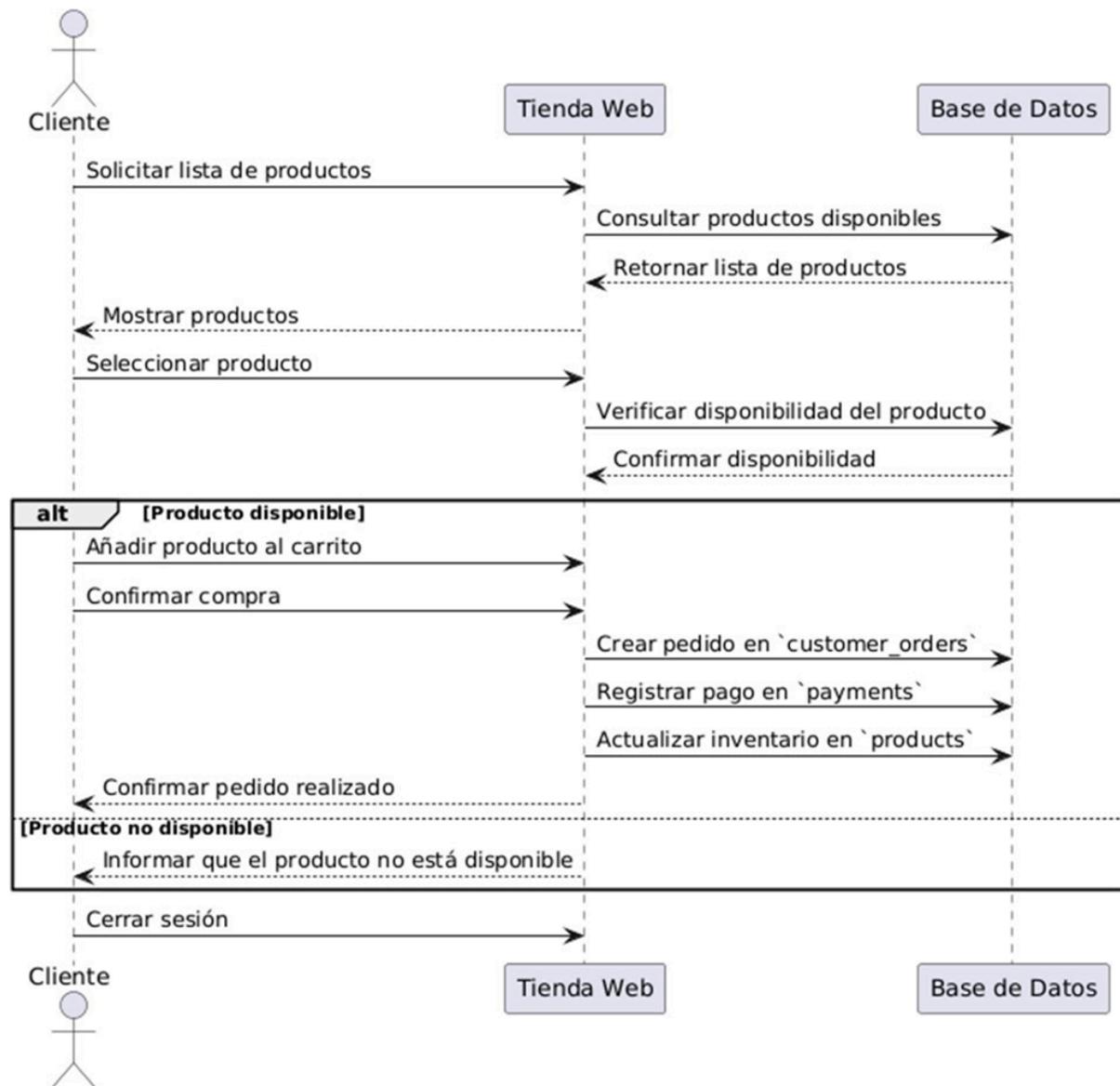


Diagrama de Secuencia

