

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN



**“SISTEMA WEB DE TIENDA EN LINEA, GESTIÓN
DE INVENTARIOS Y PAGOS”**

***APLICADO EN LA TIENDA DE ROPA “BLESSING” DE
ZACAPA, GUATEMALA***

LUIS ADEMAR QUINTO ORELLANA

GUATEMALA, DICIEMBRE 2024

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

FACULTAD DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**“SISTEMA WEB DE TIENDA EN LINEA, GESTIÓN DE INVENTARIOS Y
PAGOS”**

*APLICADO EN LA TIENDA DE ROPA “BLESSING” DE
ZACAPA, GUATEMALA*

TRABAJO DE GRADUACIÓN PRESENTADO POR:

LUIS ADEMAR QUINTO ORELLANA

previo a optar al Grado Académico de

1966

**LICENCIATURA EN INGENIERÍA EN SISTEMAS DE
INFORMACIÓN Y CIENCIAS DE LA COMPUTACIÓN**

y el Título Profesional de

INGENIERO EN SISTEMAS DE INFORMACIÓN

Y CIENCIAS DE LA COMPUTACIÓN

GUATEMALA, DICIEMBRE 2024

**AUTORIDADES DE LA FACULTAD Y ASESOR DEL
TRABAJO DE GRADUACIÓN**

DECANO DE LA FACULTAD:

ING. JORGE ALBERTO ARIAS TOBAR

SECRETARIO DE LA FACULTAD:

ING. M.A. HUGO ADALBERTO HERNÁNDEZ SANTIZO

ASESOR:

ING. WILLIAM ESTUARDO ESCOBAR ARGUETA

REGLAMENTO TRABAJO DE GRADUACIÓN

Artículo 8º: RESPONSABILIDAD

Solamente el autor es responsable de los conceptos expresados en el trabajo de tesis. Su aprobación en manera alguna implica responsabilidad de para la Universidad.

ÍNDICE

INTRODUCCION.....	1
1. CAPITULO I – MARCO CONCEPTUAL.....	5
1.1 Antecedentes	5
1.2 Justificación.....	6
1.3 Planteamiento del problema	6
1.3.1 Determinación del problema	7
1.3.2 Planteamiento del problema	7
1.3.3 Definición del problema	7
1.4 Objetivos del proyecto.....	7
1.4.1 Objetivo general	7
1.4.2 Objetivos específicos.....	7
1.5 Delimitación, alcances y límites.....	8
1.5.1 Delimitación geográfica	8
1.5.2 Delimitación institucional	8
1.5.3 Delimitación personal.....	8
1.5.4 Delimitación temática.....	8
1.5.5 Alcances y límites.....	9
1.6 Viabilidades	9
1.6.1 Viabilidad operativa	9
1.6.2 Viabilidad financiera	9
1.6.3 Viabilidad legal	10
1.7 Preguntas	12
1.7.1 Pregunta general	12
1.7.2 Preguntas auxiliares.....	12
CAPITULO II – MARCO TEÓRICO.....	13
2.1 Boutique..	13
2.1.2 Ventas	13
2.1.3 E-commerce.....	13
2.1.4 Inventario.....	15
2.1.5 Métodos para el control de inventarios	15
2.1.6 Método por promedio.....	15

2.1.7 Método PEPS	15
2.1.8 Sistema web.....	16
2.1.9 Pago en línea	16
2.1.10 Catálogo de productos	16
2.1.11 Carrito de compras	17
2.1.12 Antecedentes y actualidad de las leyes de protección de datos.....	17
2.2 Desarrollo web.....	18
2.2.1 <i>HTML (HyperText Markup Language)</i>	18
2.2.2 <i>CSS (Cascading Style Sheets)</i>	18
2.2.3 <i>JavaScript</i>	18
2.2.4 Servidor web.....	19
2.2.5 <i>Django</i>	19
2.3 <i>Python</i>	20
2.4 Base de datos	22
2.4.1 <i>MySQL</i>	22
CAPITULO III - MARCO METODOLÓGICO	25
3.1 Método general.....	25
3.2 Métodos auxiliares	25
3.2.1 Experimental	25
3.2.2 Análisis.....	25
3.2.3 Síntesis.....	25
3.3 Tipo de investigación	25
3.4 Prototipo	26
3.5 Metodología del desarrollo de la programación.....	26
4.1 Descripción del proyecto.....	27
4.2 Identificación de usuarios.....	27
4.3 Modelado del negocio	28
4.4 Metodología de desarrollo de software	28
4.5 Fases del proyecto	29
4.6 Características y artefactos	29
CAPITULO V -DISEÑO DE LA APLICACIÓN O SOLUCIÓN	31
5.1 Arquitectura del Sistema	31

5.2 Diagramas UML	32
5.3 Diagramas de caso de uso.....	32
5.3.1 Diagrama de caso de uso para la gestión de inventarios	32
5.3.2 Diagrama de caso de uso para el proceso de compras.....	33
5.3.3 Diagrama de caso de uso para la gestión de pagos en línea	34
5.3.4 Diagrama de caso de uso para agregar productos	35
5.3.5 Diagrama de caso de uso para la administración del sistema.....	36
5.4 Diagrama de actividades.....	36
5.4.1 Diagrama de actividades para la gestión de inventarios.....	36
5.4.2 Diagrama de actividades para el proceso de compras	37
5.4.3 Diagrama de actividades para la gestión de pagos en línea.....	38
5.4.4 Diagrama de actividades para agregar productos	39
5.4.5 Diagrama de actividades para la administración del sistema	40
5.5 Diagrama de secuencias	40
5.5.1 Diagrama de secuencias para la gestión de inventarios	41
5.5.2 Diagrama de secuencias para el proceso de compras	42
5.5.3 Diagrama de secuencias para la gestión de pagos en línea	43
5.5.4 Diagrama de secuencias para agregar productos.....	44
5.5.5 Diagrama de secuencias para la administración del sistema.....	45
5.6 Diagrama de clases	46
5.6.1 Diagrama de clases para la gestión de inventarios	46
5.6.2 Diagrama de clases para el proceso de compras	47
5.6.3 Diagrama de clases para la gestión de pagos en línea	48
5.6.4 Diagrama de clases para agregar productos	49
5.6.5 Diagrama de clases para la administración del sistema.....	50
5.7 Diagrama de entidad relación.....	50
5.7.1 Diagrama de entidad relación para la gestión de inventarios	51
5.7.2 Diagrama de entidad relación para el proceso de compras	52
5.7.3 Diagrama de entidad relación para la gestión de pagos en línea.....	53
5.7.4 Diagrama de entidad relación para agregar productos	54
6.1 Interfaz de usuario	55
6.2 Pruebas	56

6.3 Código	61
GLOSARIO	74
REFERENCIAS	77

ÍNDICE DE TABLAS

Tabla 1. Presupuesto de equipamiento informático	11
Tabla 2. Presupuesto de costos de desarrollo.....	12
Tabla 3. Presupuesto de tecnología y costos de desarrollo.....	12

ÍNDICE DE ILUSTRACIONES

Figura #1. Diagrama caso de uso para la gestión de inventarios.....	32
Figura #2. Diagrama caso de uso para la para el proceso de compras	33
Figura #3. Diagrama caso de uso para la para la gestión de pagos en línea.....	34
Figura #4. Diagrama caso de uso para la para la agregar productos	35
Figura #5. Diagrama caso de uso para la administración del sistema	36
Figura #6. Diagrama de actividades para la gestión de inventarios	36
Figura #7. Diagrama de actividades para la para el proceso de compras.....	37
Figura #8. Diagrama de actividades para la para la gestión de pagos en línea	38
Figura #9. Diagrama de actividades para la para la agregar productos	39
Figura #10. Diagrama de actividades para la administración del sistema.....	40
Figura #11. Diagrama de secuencias para la gestión de inventarios	41
Figura #12. Diagrama de secuencias para la para el proceso de compras.....	42
Figura #13. Diagrama de secuencias para la para la gestión de pagos en línea	43
Figura #14. Diagrama de secuencias para la para la agregar productos.....	44
Figura #15. Diagrama de secuencias para la administración del sistema.....	45
Figura #16. Diagrama de clases para la gestión de inventarios.....	46
Figura #17. Diagrama de clases para la para el proceso de compras	47
Figura #18. Diagrama de clases para la para la gestión de pagos en línea.....	48
Figura #19. Diagrama de clases para la para la agregar productos	49
Figura #20. Diagrama de clases para la administración del sistema	50
Figura #21. Diagrama de entidad relación para la gestión de inventarios.....	51
Figura #22. Diagrama de entidad relación para la para el proceso de compras	52
Figura #23. Diagrama de entidad relación para la para la gestión de pagos en línea.....	53
Figura #24. Diagrama de entidad relación para la para la agregar productos	54
Figura #25. Diagrama de entidad relación para la administración del sistema	54

INTRODUCCION

En el contexto actual de la revolución digital, el comercio electrónico ha emergido como un motor clave en la transformación de los hábitos de consumo y la operativa empresarial. Esta evolución es especialmente notable en la industria de la moda y la venta de ropa, donde la conveniencia y la accesibilidad son cada vez más determinantes en la experiencia de compra de los clientes. Este proyecto se centra en el desarrollo de un sistema web de venta de ropa en línea para la tienda "*Blessing*", ubicada en Zacapa, Guatemala.

La escogencia de este tema se debe a la creciente necesidad de pequeñas y medianas empresas de adaptarse a las tendencias del mercado digital. "*Blessing*" enfrenta el desafío de mantenerse competitiva en un entorno donde los consumidores buscan cada vez más la comodidad de las compras en línea. Además, la falta de presencia en línea puede significar una desventaja considerable frente a competidores que ya han adoptado el comercio electrónico. Un sistema web eficiente permitirá a "*Blessing*" no solo mantenerse relevante, sino también expandir su presencia más allá de su ubicación física, alcanzando a clientes en todo Guatemala.

El objetivo de este proyecto es mejorar la experiencia de compra de los clientes de "*Blessing*" mediante la implementación de una plataforma de comercio electrónico robusta y personalizada. Esta plataforma permitirá a los clientes navegar y comprar productos con facilidad, al tiempo que brinda a la tienda herramientas para gestionar sus operaciones de manera más efectiva. Se espera optimizar los procesos internos de la tienda, desde la gestión de inventarios hasta las opciones de pago, ofreciendo una solución integral que aumente la eficiencia operativa y la satisfacción del cliente.

La metodología empleada para este proyecto es la Metodología Ágil, conocida por su flexibilidad y capacidad de adaptación a cambios. Este enfoque implica ciclos iterativos de desarrollo, donde se planifica, ejecuta, evalúa y revisa continuamente cada etapa del proyecto. Esta metodología permite responder rápidamente a nuevos requisitos y desafíos, asegurando una mayor probabilidad de éxito y alineación con las necesidades del cliente. Además, se utilizarán métodos auxiliares como el método experimental para probar y validar soluciones, y el método de análisis para descomponer y entender mejor los componentes del problema.

Este proyecto no solo representa una oportunidad para mejorar la experiencia de compra en "*Blessing*", sino que también contribuye al desarrollo económico y tecnológico de la región de Zacapa. Con una visión centrada en la innovación y la excelencia, se posicionará como un referente en el ámbito del comercio electrónico en Guatemala, anticipándose a las demandas futuras del mercado y estableciendo un nuevo estándar de calidad y servicio en la industria de la moda en línea.

El proyecto toma como referencia antecedentes relevantes que evidencian la importancia del comercio electrónico en la industria de la moda. Ejemplos como Poshmark, Depop y ThredUp han transformado la experiencia de compra mediante la venta en línea, al aprovechar la demanda de los consumidores por transacciones rápidas y eficientes. Estas plataformas han demostrado el valor de una gestión eficiente de inventarios y opciones de pago integradas, lo que permite que el presente proyecto aproveche sus lecciones aprendidas para optimizar los procesos en "*Blessing*".

La viabilidad de este proyecto se sustenta en diversos aspectos clave, que van desde la operatividad y la infraestructura tecnológica hasta la viabilidad financiera y legal. En términos de viabilidad operativa, la empresa cuenta con un entorno físico adecuado que incluye instalaciones espaciales y seguras, así como un parque informático equipado con tecnología actualizada. Además, se dispone de sistemas de comunicación eficientes que facilitan la colaboración interna y externa del equipo.

Desde el punto de vista financiero, la empresa cuenta con una sólida situación respaldada por una adecuada planificación presupuestaria. Se dispone de capital inicial suficiente para cubrir los costos de desarrollo, implementación y operación del proyecto a corto plazo, además de proyecciones de ingresos que permiten una planificación financiera efectiva. Por otro lado, la viabilidad legal del proyecto se asegura mediante un exhaustivo análisis legal que garantiza el cumplimiento de todas las leyes y regulaciones aplicables en el ámbito del comercio electrónico y la protección de datos.

En términos de costos, se ha realizado un detallado presupuesto que incluye tanto los costos de equipamiento informático como los costos de desarrollo del sistema, esto asegura una planificación financiera sólida y transparente, así como una gestión eficiente de los recursos disponibles.

Además de mejorar la experiencia de compra de los clientes de "Blessing", este proyecto tiene un impacto directo en el desarrollo económico y tecnológico de Zacapa, posicionando a la tienda como un referente local en el comercio electrónico, y anticipándose a las demandas futuras del mercado de moda en línea.

El marco metodológico del proyecto establece las directrices y estrategias que guiarán el proceso de investigación y desarrollo del sistema web para la optimización de la gestión de inventarios y pagos en la tienda "*Blessing*". Este marco se fundamenta en la definición del método general y los métodos auxiliares utilizados en la investigación, así como en la determinación del tipo de investigación y la metodología de desarrollo de la programación. La selección adecuada de estos elementos es fundamental para garantizar la calidad y la efectividad del proceso de desarrollo del sistema web.

En el marco teórico del proyecto, se exploran diferentes conceptos y tecnologías relacionadas con el desarrollo de sistemas web y la gestión de inventarios y pagos. Se analizan términos como boutique, *e-commerce*, inventario, método *PEPS*, sistema web, pago en línea, desarrollo web, *HTML*, *CSS*, *JavaScript*, servidor web, *Django*, *Python* y base de datos *MySQL*. Estos conceptos y tecnologías proporcionan la base teórica necesaria para el desarrollo e implementación del sistema web en la tienda "*Blessing*".

1. CAPITULO I – MARCO CONCEPTUAL

Es un apartado del proyecto de investigación en el cual se detallan las bases conceptuales de la investigación, es decir, se ofrece al lector el conjunto de definiciones clave para entender el resto del proyecto. (Etecé, concepto, 2023)

1.1 Antecedentes

Pueden llamarse antecedentes a cualquier tipo de elementos que vengan antes de algo tomado en consideración, o sea, que la anteceden en el tiempo, en el espacio o en un orden específico. De hecho, la palabra antecedentes proviene del latín antecedentes, conformado por el prefijo ante- (“antes”) y el verbo *cedere* (“andar”, “marchar”). Es traducible como “lo que ocurre antes”. (Editorial Etece , 2020)

Poshmark: Es una aplicación de comercio social que permite a los usuarios comprar y vender ropa nueva y usada. Una de sus principales ventajas es su comunidad activa y comprometida, lo que facilita las ventas rápidas y la interacción directa entre vendedores y compradores. Además, Poshmark ofrece herramientas integradas de envío y protección al comprador. Sin embargo, una desventaja es que *Poshmark* toma una comisión del 20% en las ventas superiores a \$15, lo cual puede ser significativo para los vendedores. Además, la competencia puede ser alta debido a la gran cantidad de usuarios y listados. (poshmark, s.f.)

Depop: Es una plataforma de comercio electrónico y aplicación móvil enfocada en la venta de moda, especialmente ropa retro y de segunda mano. Sus ventajas incluyen una interfaz fácil de usar y una fuerte comunidad de moda sostenible, lo que atrae a compradores interesados en estilos únicos y ecológicos. *Depop* también permite a los usuarios seguir a sus vendedores favoritos y ver sus nuevas publicaciones fácilmente. Por otro lado, una desventaja es la tarifa de transacción del 10%, que puede reducir las ganancias de los vendedores. Además, algunos usuarios reportan problemas con el fraude y la autenticidad de los productos. (Depop, s.f.)

ThredUp: es una tienda en línea y aplicación que se especializa en la venta de ropa de segunda mano de alta calidad. Entre sus ventajas se encuentra el hecho de que *ThredUp* se encarga de la logística, desde la recepción de las prendas hasta su listado y envío, lo que facilita el proceso para los vendedores. También ofrece precios atractivos para los compradores que buscan ropa asequible y de buena calidad. Sin embargo, una desventaja es que los vendedores reciben un porcentaje relativamente bajo del precio de venta, ya que *ThredUp* maneja la mayor parte del proceso. (Thredup, s.f.)

1.2 Justificación

El proyecto de un sistema web de venta de ropa en línea surge como respuesta directa a los desafíos identificados en la operación actual de la tienda "Blessing", ofreciendo soluciones integrales que beneficiarán tanto al negocio como a sus clientes.

Uno de los problemas más evidentes es la falta de una plataforma digital que permita a la tienda expandir su presencia y alcanzar nuevos mercados más allá de Zacapa. Al implementar una tienda en línea, "Blessing" podrá llegar a una audiencia más amplia, no solo local, sino también a nivel nacional, facilitando las compras a clientes que no pueden acceder físicamente a la tienda. Esto permitirá diversificar su base de clientes y mejorar su posicionamiento en el competitivo mercado de la moda.

El sistema también resolverá el control inadecuado de inventarios, uno de los problemas críticos actuales. A través de una gestión en tiempo real, sincronizada tanto con el inventario físico como con el canal en línea, se reducirá significativamente el riesgo de errores en el procesamiento de pedidos. Esto proporcionará a los clientes información precisa sobre la disponibilidad de productos, lo que reducirá cancelaciones por falta de stock y mejorará la confianza en la plataforma.

Desde la perspectiva del cliente, la plataforma ofrecerá una experiencia de compra más ágil y personalizada. Los clientes podrán realizar sus compras en cualquier momento, desde cualquier lugar, lo que les brindará mayor comodidad. Además, el acceso a información actualizada sobre productos, promociones y ofertas permitirá una toma de decisiones más informada. De esta manera, se espera mejorar la satisfacción del cliente, lo que podría traducirse en una mayor fidelización.

Otro aspecto clave es el impacto positivo que este sistema tendrá en la operativa de la tienda. La automatización y optimización de procesos internos como la actualización del inventario, el seguimiento de pedidos y la gestión de clientes contribuirán a reducir los costos operativos. Además, la digitalización de la tienda permitirá a los empleados centrarse en tareas más estratégicas, mejorando la eficiencia global del negocio.

1.3 Planteamiento del problema

En la situación actual de una tienda de ropa, se identifica la necesidad de mejorar el control y registro de clientes, así como el control de inventarios, la gestión eficiente del inventario es fundamental para garantizar la disponibilidad de productos en el momento adecuado.

La falta de un sistema adecuado para el control de inventarios puede resultar en problemas como el agotamiento de existencias, exceso de inventario o dificultades para identificar productos populares y en demanda. Además, la falta de sincronización entre el inventario en línea y físico puede llevar a errores en el procesamiento de pedidos y a una experiencia de usuario insatisfactoria.

1.3.1 Determinación del problema

El desafío radica en la incapacidad de aprovechar las oportunidades de ventas en línea debido a la carencia de una plataforma web dedicada. Esta limitación obstaculiza el progreso y la satisfacción del cliente en la empresa, impidiendo una expansión efectiva del negocio.

1.3.2 Planteamiento del problema

La falta de acceso a un medio de compra virtual limita la capacidad de la empresa para alcanzar a un importante grupo de clientes potenciales que valoran la conveniencia de realizar sus adquisiciones en línea. Esta carencia no solo afecta el crecimiento de las ventas, sino que también pone en riesgo la visibilidad y la competitividad del negocio en el entorno digital, donde cada vez más consumidores llevan a cabo sus transacciones.

1.3.3 Definición del problema

¿Cómo puede una plataforma web de tienda en línea mejorar el acceso a los productos, aumentar las ventas y garantizar una experiencia satisfactoria para los clientes de una tienda específica?

1.4 Objetivos del proyecto

Los objetivos de un proyecto son lo que esperas haber logrado al terminar tu proyecto. Entre ellos se incluyen los entregables y los activos, u objetivos más intangibles como aumentar la productividad o la motivación. Los objetivos de tu proyecto deben ser alcanzables, de duración limitada y específicos, de manera que puedas medirlos al finalizar tu proyecto. (Martins, 2024)

1.4.1 Objetivo general

Facilitar la gestión de inventarios y ventas en línea, permitiendo un control eficiente de existencias y ofreciendo a los clientes un acceso fácil y directo a los productos a través de una plataforma web.

1.4.2 Objetivos específicos

- Presentar un catálogo digital que permita a los clientes seleccionar y comprar productos mediante un carrito de compras y un sistema de pago en línea.

- Administrar el inventario con un módulo que controle las existencias máximas y mínimas.
- Implementar un sistema de facturación electrónica que optimice los registros contables y el control de inventarios.

1.5 Delimitación, alcances y límites

La delimitación del problema, que forma parte del planteamiento del problema, es la etapa preliminar de la investigación en la que se realiza un recorte o un enfoque del tema a investigar, de manera tal de abordar un aspecto específico y evitar las aproximaciones panorámicas, generales y demasiado abarcativas, ya que resultarían ambiguas o imposibles en términos prácticos. (Editorial Etecé, 2024)

1.5.1 Delimitación geográfica

La implementación del sistema web de la tienda en línea estará dirigida específicamente a usuarios ubicados en áreas urbanas de Guatemala, inicialmente enfocándose en la ciudad de Zacapa.

1.5.2 Delimitación institucional

El proyecto se llevará a cabo en colaboración con la empresa propietaria de la tienda física ubicada en Zacapa, con el objetivo de integrar la experiencia de compra en línea con la operación existente del negocio.

1.5.3 Delimitación personal

La información necesaria para el desarrollo del sistema se obtendrá a través de reuniones con el equipo de gestión de la tienda física y con el personal técnico encargado del desarrollo del sistema web.

1.5.4 Delimitación temática

Para el desarrollo del sistema web de tienda en línea, se ha optado por una combinación de tecnologías que se consideran óptimas para alcanzar los objetivos planteados. Se elige *python* como lenguaje de programación principal debido a su versatilidad, facilidad de uso y gran cantidad de bibliotecas disponibles. Para la creación del *backend* de la aplicación, se ha elegido *Django*, un *framework* de desarrollo web en *Python* que proporciona una estructura sólida y escalable. Además, se ha seleccionado *MySQL* como el sistema de gestión de base de datos relacional, debido a su rendimiento robusto, escalabilidad y amplia compatibilidad con *Django*.

1.5.5 Alcances y límites

Los alcances del proyecto del sistema web de la tienda en línea delinean claramente qué se puede anticipar y qué aspectos se abordarán en la investigación y desarrollo. Por otro lado, las limitaciones definen qué aspectos quedarán fuera del alcance del proyecto, estableciendo los límites o fronteras hasta donde se extienden las metas y objetivos de la investigación, sin hacer referencia a las dificultades de realización. Es esencial comprender tanto los alcances como las limitaciones al planificar y ejecutar un proyecto, ya que esto permite una gestión eficiente de recursos y una evaluación realista de los resultados esperados.

1.6 Viabilidades

Viabilidad se refiere a la capacidad de una idea, proyecto o acción para tener éxito o ser efectivo en circunstancias específicas. Implica evaluar si es factible, práctico y posible llevar a cabo una determinada iniciativa, considerando diversos aspectos como recursos disponibles, costos, beneficios, riesgos y condiciones externas. Se determina mediante un análisis detallado y sistemático que permite tomar decisiones informadas sobre la viabilidad y la realización de un proyecto, plan o acción, asegurando así una dirección estratégica clara y una asignación eficiente de recursos.

1.6.1 Viabilidad operativa

La empresa cuenta con un entorno físico adecuado que incluye instalaciones espaciosas y seguras para albergar al equipo de trabajo y llevar a cabo las operaciones relacionadas con el proyecto. Además, se dispone de un parque informático equipado con computadoras actualizadas y software necesario para el desarrollo y la gestión del sistema. En cuanto a los sistemas de comunicación, se cuenta con líneas telefónicas, servicios de correo electrónico y plataformas de mensajería instantánea que facilitan la comunicación interna y externa del equipo. Estos recursos son esenciales para garantizar la eficiencia y la efectividad en la ejecución del proyecto.

1.6.2 Viabilidad financiera

La empresa cuenta con una sólida situación financiera respaldada por una adecuada planificación presupuestaria. Se dispone de capital inicial suficiente para cubrir los costos de desarrollo, implementación y operación del proyecto a corto plazo. Además, se ha realizado un análisis detallado de costos y proyecciones de ingresos, lo que permite una planificación financiera efectiva y la identificación de posibles fuentes de financiamiento adicionales si fuera necesario para expandir el proyecto.

1.6.3 Viabilidad legal

La empresa ha realizado un exhaustivo análisis legal para garantizar el cumplimiento de todas las leyes y regulaciones aplicables en el ámbito del comercio electrónico, protección de datos y propiedad intelectual. Se cuenta con asesoramiento legal especializado y se han implementado políticas internas para asegurar el cumplimiento continuo de las normativas legales. Además, se han llevado a cabo auditorías legales para identificar posibles riesgos y se han establecido medidas preventivas para mitigarlos. La empresa también está comprometida con la ética empresarial y el respeto a los derechos de los consumidores, lo que contribuye a su reputación y confianza en el mercado.

Tabla 1. Presupuesto de equipamiento informático

Costos fijos en equipos y servicios necesarios para el uso y el seguimiento del proyecto.

Cantidad	<i>Equipo</i>	<i>Precio Unitario</i>	<i>Costo</i>
2	Equipos de Cómputo marca Dell, 8GB RAM, CORE I5 8 generación, almacenamiento 256GB de estado sólido.	Q6,000.00	Q12,000.00
	Cableado estructurado y regletas.	Q1000.00	Q 1,000.00
3	UPS de 500VA	Q 400.00	Q 1,200.00
	Total		Q 14,200.00

Costos de los diferentes equipos y servicios para la realización del proyecto.

Tabla 2. Presupuesto de servicios

Costos fijos en servicios necesarios para el uso y el seguimiento del proyecto.

<i>Cantidad</i>	<i>Servicio</i>	<i>Precio Unitario</i>	<i>Costo</i>
1	Año de contratación de servidor web.	Q2400.00	Q 2,400.00
1	Año de contratación de Dominio con dirección de la empresa.	Q 300.00	Q 300.00
1	Año de contratación de Internet Corporativo(Fibra óptica).	Q3600.00	Q 3,600.00
1	Año de servicios de facturación	Q 600.00	Q 600.00
Total			Q 6,900.00

Costos de los diferentes servicios para la realización del proyecto.

Tabla 3. Presupuesto de costos de desarrollo

Costos fijos de Recurso Humano para la creación del proyecto del Sistema de Gestión de Inventarios con Sistema de Pago Seguro en Línea

<i>Actividad Laboral</i>	<i>Horas</i>	<i>Costo Hora/Unidad</i>	<i>Costo Total</i>
Análisis de necesidades.	20	Q40.00	Q 800.00
Diseño del sistema propuesto.	20	Q40.00	Q 800.00
Diseño y desarrollo de la base de datos del sistema.	40	Q55.00	Q 2,200.00
Codificación del software.	70	Q55.00	Q 3,800.00
Revisión y pruebas del sistema y base de datos.	20	Q55.00	Q 1,100.00
Total, estimado del proyecto.			Q 8,700.00

En esta tabla se presenta el tipo de actividad laboral necesaria para realizar el desarrollo el sistema, las horas requeridas, el precio por hora, el costo total derivado de la cantidad de horas y el precio promedio por hora. Este último se calcula utilizando datos promediados obtenidos del sitio web <https://gt.computrabajo.com/salarios/ingeniero-en-sistemas>.

Tabla 4. Presupuesto de tecnología y costos de desarrollo

Costos totales del proyecto.

<i>Recurso</i>	<i>Costo</i>
Recursos Materiales	Q14,200.00
Servicios	Q 6,900.00
Recurso Humano	Q 8,700.00
Total	Q29,800.00

1.7 Preguntas

La declaración explícita de los objetivos de investigación indica con precisión lo que el investigador intenta descubrir a través de la recopilación y análisis de datos. Esta declaración obliga al investigador a ser claro y específico sobre el propósito de su investigación, lo que proporciona una guía clara para el diseño y la ejecución del estudio." (Saunders, Lewis & Thornhill, 2019).

1.7.1 Pregunta general

¿De qué manera podría beneficiar un sistema de gestión de inventarios con sistema de pago seguro en línea a la eficiencia y rentabilidad de nuestro negocio?

1.7.2 Preguntas auxiliares

- ¿Cuáles son las características clave que deberíamos incluir en la interfaz de usuario para asegurar una navegación intuitiva y fácil búsqueda de productos?
- ¿Cómo integraremos un sistema de gestión de inventario para mantener un control eficiente del *stock* y procesar los pedidos de manera oportuna?
- ¿Cómo implementaremos un sistema de facturación en línea que permita a los clientes realizar compras con su respectiva factura?

CAPITULO II – MARCO TEÓRICO

Es una sección de un documento académico, investigativo o técnico en la que se presenta y discute la teoría relevante, los conceptos fundamentales, los estudios previos y cualquier otro conocimiento teórico pertinente relacionado con el tema de investigación o estudio. Esta sección proporciona el contexto teórico necesario para comprender la base conceptual y las ideas clave que fundamentan el trabajo realizado en el documento.

2.1 Boutique

La palabra boutique, de origen francés, se ha popularizado para hacer referencia a pequeños negocios o tiendas en los cuales, por lo general, se venden productos de alto refinamiento y exclusividad, accesibles por lo tanto para un reducido número de personas. Las boutiques no son nunca lugares donde uno puede encontrar productos masivos o baratos sino más bien elementos de diseño y de creatividad particular. (Bembibre, 2010)

2.1.2 Ventas

Se refieren al proceso de intercambio de bienes o servicios por dinero u otros medios de pago. Es un componente fundamental de cualquier negocio o actividad comercial, ya que representa la principal fuente de ingresos para las empresas. El proceso de ventas implica la identificación de clientes potenciales, la presentación y promoción de productos o servicios, la negociación de condiciones de compra y, finalmente, la conclusión de la transacción mediante la entrega del producto o la prestación del servicio y el pago correspondiente.

2.1.3 E-commerce

Un *e-commerce* o comercio electrónico hace referencia a una tienda virtual que se dedica a la compraventa de productos en Internet y que utiliza este medio para realizar transacciones y estar en contacto con sus consumidores no solo a través de su sitio web, sino también de sus redes sociales. Este modelo de negocio es fundamental para todo tipo de negocio de cualquier tamaño, pero especialmente para aquellas empresas pequeñas que tienen dificultades a la hora de asentarse por la vía tradicional. Esta versión en línea de su negocio permite acceder a otros mercados y crecer en ellos de forma más sencilla. (Meijomil, 2023)

Estudio de comercio electrónico!

Con el objetivo de fortalecer el crecimiento económico de las micro, pequeñas y medianas empresas, la Cámara de Comercio de Guatemala (CCG), dio a conocer hoy el 4to. Estudio de Comercio Electrónico, orientado a la actualización de sus socios, no socios y público en general.

El estudio divulgado en conferencia de prensa pone énfasis en la importancia de fortalecer las habilidades digitales de estas unidades de negocios y se realizó con el patrocinio de Cargo Expreso, Banco Industrial, Kebi, Pragma, Publimóvil y Más Digital en colaboración con Mercaplan.

Finalmente, el licenciado Juan José Castillo, gerente de la Unidad de Comercio electrónico de la CCG, resumió el 4º. Estudio de Comercio Electrónico con los siguientes datos y estadísticas:

- El 84% de los guatemaltecos entrevistados en la muestra han usado Internet en los últimos 7 días para compras en línea.
- El 28% de los guatemaltecos han comprado en línea en los últimos 3 meses.
- En 2022, el 75% de los compradores online están en la región central (Guatemala, Sacatepéquez y Chimaltenango). En 2021 era el 70%.
- La compra online se ha convertido en algo más cotidiano para los usuarios del comercio electrónico, ya que 7 de cada 10 personas realizan al menos una compra al mes, es decir un 70% de los entrevistados, mientras que en 2021 era el 64%.
- 8 de cada 10 empresas alcanzaron entre 1 y 6 mil productos vendidos, con celulares y teléfonos encabezando la lista, luego productos de cosmética y cuidado personal para llegar a un tercer lugar de electrónica, audio, video y TV.
- En promedio 6 de cada 10 empresas han obtenido un incremento del 10% al 30% en sus ventas en línea de junio 2021 a junio 2022.
- Solo 2 de cada 10 empresas realizaron ventas en el exterior, lo que significó un 13% en promedio de su facturación.
- En 2022, las ventas internacionales representaron un USD \$261 millones y el 19% de crecimiento.
- Para las ventas en tiendas nacionales, en 2022 se representó en USD \$535 millones, un crecimiento del 36%.
- La red social más utilizada para las compras online es Facebook Messenger con un 81%, seguida de WhatsApp con un 50%.
- En los hábitos de compra, los pagos de servicios se posicionan en el 1º. Lugar en 2022, desplazando al rubro de alimentos y bebidas que fue en 2020 y 2021, durante la pandemia. (Alvarez, 2022).

2.1.4 Inventario

Inventario es un elemento crucial en la gestión empresarial, abarcando una amplia gama de bienes tangibles que una empresa posee en un momento determinado. Este conjunto de activos incluye desde productos terminados listos para su venta hasta materias primas y suministros esenciales para el proceso de producción. El inventario también engloba productos en proceso, aquellos que se encuentran en diferentes etapas de fabricación, pero aún no están listos para su distribución final estos activos representan el valor de una empresa.

2.1.5 Métodos para el control de inventarios

Los métodos para el control de inventarios son políticas y procedimientos para contar con las cantidades óptimas de *stock*, que están basadas en modelos determinísticos y de probabilidad. A través de estos sistemas de control de inventarios, tu empresa puede: Calcular los niveles de inventario óptimos, determinar los puntos de reorden, cuándo hay que reabastecer y de qué tamaño deben ser los pedidos y conservar una inversión mínima, asegurando el aprovisionamiento para no perder ventas por falta de *stock*.

2.1.6 Método por promedio

Con este método lo que se hace es determinar un promedio del costo de las distintas unidades compradas, sumando los valores existentes en el inventario con los valores de las nuevas compras, para luego dividirlo entre el número de unidades existentes en el inventario incluyendo tanto los inicialmente existentes, como los de la nueva compra. (Gerencie.com, 2022)

2.1.7 Método PEPS

El método PEPS es un método para el control de inventarios que consiste en identificar los primeros artículos en entrar al almacén, para que sean los primeros en ser vendidos o utilizados en la producción. También conocido como *FIFO* por sus siglas en inglés (*first inputs, first outputs*), PEPS es un método que sigue un orden cronológico. Así se evita, por ejemplo, que productos perecederos caduquen, ya que tendrán un movimiento constante y oportuno. De hecho, este segundo método para el control de inventarios PEPS es utilizado principalmente en los negocios de alimentos y en la industria textil, sujeta por lo general a las modas cíclicas o temporales.

Una de las ventajas del método PEPS es que permite ofrecer precios competitivos en momentos de alta inflación. Así, cuando los precios se elevan, el negocio puede ganar mercado al fijar un precio de venta menor al de la competencia, debido a que la mercancía fue adquirida con antelación. (Hernández, s.f.)

2.1.8 Sistema web

Un sistema web es una aplicación o plataforma de software que opera a través de un navegador web y utiliza tecnologías basadas en la web para funcionar. Estos sistemas pueden ofrecer una amplia gama de funcionalidades y servicios, desde aplicaciones simples como *blogs* y sitios web de comercio electrónico hasta sistemas complejos como sistemas de gestión empresarial (*ERP*), sistemas de gestión de contenido (*CMS*), sistemas de gestión de relaciones con clientes (*CRM*) y mucho más.

2.1.9 Pago en línea

El pago en línea, también conocido como pago electrónico o pago digital, es un método de realizar transacciones financieras a través de internet. Permite a los usuarios realizar compras, transferencias de dinero, pagos de facturas y otras transacciones financieras utilizando dispositivos conectados a la red, como computadoras, teléfonos inteligentes o tabletas.

El proceso de pago en línea generalmente implica lo siguiente:

Selección de productos o servicios: El usuario selecciona los productos o servicios que desea comprar en un sitio web o aplicación.

Ingreso de información de pago: El usuario proporciona los detalles de su método de pago, que pueden incluir información de tarjeta de crédito o débito, información de cuenta bancaria, o el uso de plataformas de pago en línea como *PayPal*, *Stripe*, *cybersource* o servicios de billetera digital como *Apple Pay* o *Google Pay*.

Autorización y procesamiento: Una vez que se ingresa la información de pago, se verifica la autenticidad de la transacción y se procesa el pago. Esto puede implicar la comunicación con el banco emisor de la tarjeta o el proveedor de la plataforma de pago para obtener la aprobación.

Confirmación: Una vez que se completa la transacción, se muestra una confirmación al usuario y se le proporciona un recibo electrónico.

2.1.10 Catálogo de productos

Es un documento que reúne toda, o casi toda, la información esencial de sus productos, como las dimensiones del producto, el precio, el material. (Memon, 2022)

2.1.11 Carrito de compras

El carrito de compras es esencialmente un software que se encarga de mantener un registro de los productos seleccionados por el cliente, la cantidad de cada uno, su precio y el total de la compra. Además, también puede permitir a los clientes actualizar la cantidad de un producto en el carrito, eliminar productos o aplicar cupones o descuentos. (Yepes, 2023)

2.1.12 Antecedentes y actualidad de las leyes de protección de datos

Iniciativa 4090 - Iniciativa que dispone aprobar la Ley de Protección de Datos Personales

Esta iniciativa fue presentada a la Dirección Legislativa el 6 de agosto de 2009 por los siguientes diputados: Gustavo Ernesto Blanco Segura, Herbert Leonel Barragán Ochoa, José Alejandro Arévalo Alburez, Marco Antonio Solares Pérez y Mariano Rayo Muñoz. Posteriormente se presentó en el Pleno el 20 de agosto del mismo año y se asignó a la Comisión de Economía y Comercio Exterior. Recibió un dictamen favorable de la Comisión el 14 de diciembre del mismo año y se discutió en su primer debate el 2 de marzo de 2010. El 15 de julio de 2010 se discutió en segundo debate.

Iniciativa 5921 - Iniciativa que dispone aprobar la Ley de Protección de Datos Personales

Esta iniciativa fue presentada a la Dirección Legislativa el 09 de junio de 2021 por los diputados Aníbal Estuardo Samayoa Alvarado y Douglas Rivero Mérida. Posteriormente se presentó en el Pleno el 6 de octubre del mismo año y se asignó a las Comisiones de Gobernación y la Comisión de Educación, Ciencia y Tecnología. Recibió un dictamen favorable con modificaciones de la Comisión correspondiente el 6 de diciembre de 2021.

Al momento de la publicación de este artículo, aún no se ha discutido en primer debate ni se encontró más información de su estado actual.

Iniciativa 6103 - Iniciativa que dispone aprobar la Ley Integral de Protección de Datos Personales en Poder de Terceros.

Esta iniciativa fue presentada a la Dirección Legislativa el 23 de junio de 2022 por el diputado Oscar Stuardo Chinchilla Guzmán. Posteriormente se presentó en el Pleno el 2 de agosto del mismo año y se asignó a la Comisión de Reformas al Sector Justicia. Recibió un dictamen favorable de la Comisión correspondiente el 17 de enero de 2023 y se discutió en su primer debate el 25 de enero del presente año. El reciente 1 de febrero del 2023 se discutió en segundo debate. Al momento de la publicación de este artículo, esta iniciativa es la que más tracción y avances ha tenido. (Pando, 2023)

2.2 Desarrollo web

Es el proceso de creación y mantenimiento de sitios web y se conforma de una amplia variedad de acciones que van desde la creación de códigos y diseño, hasta la administración de los contenidos y del servidor. El desarrollo web funciona mediante el uso de lenguajes de programación para crear el código que hace funcionar las páginas.

Algunos de los lenguajes para el desarrollo web más comunes y utilizados son el lenguaje de marcado de hipertexto (*HTML*), las hojas de estilo en cascada (*CSS*), *PHP*, *JavaScript*, y *MySQL*. Además, los desarrolladores web requieren del uso constante de bases de datos como *Microsoft SQL Server* y *Oracle*, y de servidores web como *Unix* y *Linux* para realizar sus tareas diarias. (Ortega K. , 2023)

2.2.1 *HTML (HyperText Markup Language)*

HTML son las siglas en inglés de *HyperText Markup Language*, que significa Lenguaje de Marcado de HiperTexto. Se llama así al lenguaje de programación empleado en la elaboración de páginas web, y que sirve como estándar de referencia para la codificación y estructuración de las mismas, a través de un código del mismo nombre (*HTML*). (Editorial Etecé, 2023)

2.2.2 *CSS (Cascading Style Sheets)*

CSS son las siglas en inglés de *Cascading Style Sheets*, que significa «hojas de estilo en cascada». Es un lenguaje que se usa para estilizar elementos escritos en un lenguaje de marcado como *HTML*. Fue desarrollado por W3C (*World Wide Web Consortium*) en 1996 por una razón muy sencilla. *HTML* no fue diseñado para tener etiquetas que ayuden a formatear la página. Está hecho solo para escribir el marcado para el sitio. (B, Hostinger, 2023)

2.2.3 *JavaScript*

JavaScript es un lenguaje de programación ligero que los desarrolladores web suelen utilizar para crear interacciones más dinámicas al desarrollar páginas web, aplicaciones, servidores e incluso juegos. Los desarrolladores suelen utilizar *JavaScript* junto con *HTML* y *CSS*. El lenguaje de programación funciona bien con *CSS* a la hora de dar formato a los elementos *HTML*.

Las implementaciones de *JavaScript* en la web, las aplicaciones móviles y el desarrollo de juegos hacen que valga la pena aprender este lenguaje de programación. Puedes hacerlo a través de plataformas de aprendizaje como *BitDegree* o explorando plantillas y aplicaciones gratuitas de *JavaScript* en plataformas de alojamiento de código como *GitHub*. (B, Hostinger, 2023)

2.2.4 Servidor web

Es un *software* que se ejecuta en un servidor y se encarga de recibir, procesar y responder a las solicitudes de los clientes, generalmente navegadores web, que desean acceder a recursos en internet, como páginas web, imágenes, archivos, entre otros.

El servidor web utiliza el protocolo *HTTP (Hypertext Transfer Protocol)* o su versión segura *HTTPS (HTTP Secure)* para comunicarse con los clientes. Cuando un usuario ingresa la dirección de un sitio web en su navegador, el navegador envía una solicitud al servidor web correspondiente. El servidor web entonces procesa la solicitud, accede a los archivos solicitados y los envía de vuelta al navegador del usuario para que sean visualizados.

El servidor web es esencial para la publicación y el acceso a contenido en la web. Sin él, los sitios web no estarían disponibles para su visualización en los navegadores de los usuarios. Algunos de los servidores web más populares son *Apache HTTP Server*, *Nginx* y *Microsoft Internet Information Services (IIS)*. Estos servidores web son utilizados por millones de sitios web en todo el mundo para servir contenido de manera confiable y eficiente. (Gillis)

2.2.5 Django

Django es un *framework* de desarrollo para *Python* que se emplea para la creación de páginas web. Se trata de una herramienta de código abierto y gratuita que cuenta con una comunidad amplia y que comparte recursos constantemente. Además, *Django* también cuenta con funciones de pago que pueden facilitar más el trabajo de los desarrolladores.

Django es una herramienta que se puede usar para el desarrollo *full-stack* de aplicaciones y páginas web, así como para el desarrollo de servidores. Está considerado como el mejor *framework* para el desarrollo de aplicaciones web con *Python* y es uno de los marcos de desarrollo más demandados por los programadores que trabajan con este lenguaje en el desarrollo web.

Creado por Adrian Holovaty y Simon Willison mientras estaban trabajando en *PHP* y necesitaban algo que les facilitase actualizar una web de manera rápida para cumplir con los plazos. En ese momento se pasaron a *Python* y, en 2005, lanzaron *Django* para que la programación fuese aún más sencilla. *Django* se destaca por su enfoque en la seguridad y escalabilidad, una de sus características más atractivas es su capacidad para reducir el tiempo de desarrollo, gracias a herramientas integradas como el ORM (Object-Relational Mapping), que simplifica la interacción con bases de datos, y su robusto sistema de autenticación de usuarios.

Estas son algunas de las características por las que *Django* es uno de los *frameworks* más usados para el desarrollo web:

Completo: *Django* proporciona casi todo lo que los programadores necesitan y pueden querer usar. Se trata de una herramienta que sigue unos principios de diseño consistentes y que cuenta con una buena base de documentación para facilitar el trabajo de los desarrolladores.

Escalable: *Django* funciona por componentes sustituibles e intercambiables. Eso significa que se puede escalar con bastante facilidad. En este sentido, un ejemplo de uso de *Django* lo tenemos en *Instagram*, plataformas que lo han empleado para mejorar sus servidores gracias a la escalabilidad del entorno de desarrollo.

Versátil: Este entorno de desarrollo se ha empleado para la creación de todo tipo de páginas web. Desde sistemas que son puramente para administración de contenidos, como puede ser, por ejemplo, una wiki; hasta redes sociales o páginas webs de noticias.

Y es que, *Django* es compatible con cualquier *framework* que se emplea del lado del cliente, por lo que puede mandar contenidos en cualquier formato.

Seguro: *Django* facilita la detección y solución de posibles fallos en la seguridad en las páginas web del lado del servidor. Proporciona una administración segura de usuarios y contraseñas y evita errores que son comunes en el diseño y desarrollo *back-end*.

Portátil: *Django* se puede usar en cualquier sistema y plataforma y, además, está respaldado por muchos de los proveedores de hosting que, además, suelen proporcionar la documentación necesaria para implementarlo.

Mantenimiento: *Django* facilita el mantenimiento y la velocidad del mismo en el desarrollo web. Fomenta la creación de código reutilizable, lo cual simplifica, en gran medida, el trabajo de los desarrolladores web. (Fernández, 2022)

2.3 Python

Es un lenguaje de programación de alto nivel, interpretado y multipropósito, creado con el objetivo de ser simple, legible y poderoso. Fue concebido a finales de los años 80 por Guido van Rossum en los Países Bajos y desde entonces ha experimentado un crecimiento constante, convirtiéndose en uno de los lenguajes de programación más populares y versátiles del mundo.

En su diseño, *Python* busca priorizar la legibilidad del código, enfocándose en la claridad y la simplicidad sintáctica. Esto se logra mediante el uso de una sintaxis limpia y estructuras de datos intuitivas, lo que facilita la comprensión del código incluso para aquellos que son nuevos en la programación. Además, Python fomenta un estilo de programación limpio y ordenado, promoviendo buenas prácticas de desarrollo y facilitando la colaboración en proyectos de código abierto. (Duque)

Python es un lenguaje de programación multipropósito que se utiliza en una amplia variedad de campos y aplicaciones. Es especialmente popular en el desarrollo web, la ciencia de datos, la inteligencia artificial, el análisis de datos, la automatización de tareas, el desarrollo de juegos y la administración de sistemas, entre otros. Su flexibilidad y versatilidad lo convierten en una herramienta invaluable para desarrolladores de todos los niveles de experiencia, desde principiantes hasta expertos. Entre las funcionalidades de *python* tenemos:

Sintaxis clara y legible: Python utiliza una sintaxis limpia y fácil de leer, lo que facilita la escritura y comprensión del código.

Interpretado e interactivo: *Python* es un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea, facilitando la depuración y prueba de código. También es interactivo, lo que permite ejecutar comandos y expresiones directamente desde el intérprete de *Python*.

Orientado a objetos: *Python* soporta la programación orientada a objetos, lo que permite la creación de clases y objetos, encapsulación, herencia y polimorfismo.

Amplia biblioteca estándar: *Python* viene con una amplia biblioteca estándar que incluye módulos y paquetes para realizar una variedad de tareas, desde operaciones básicas de entrada/salida hasta desarrollo web, procesamiento de datos, *networking* y más.

Multiplataforma: *Python* es compatible con la mayoría de los sistemas operativos, incluyendo *Windows*, *macOS* y *Linux*, lo que lo hace altamente portable.

Comunidad activa: *Python* cuenta con una gran comunidad de desarrolladores y usuarios que contribuyen con paquetes, recursos y soporte técnico, lo que facilita la resolución de problemas y la obtención de ayuda cuando sea necesario.

Entre las ventajas de *python* tenemos:

Sintaxis clara y legible: La sintaxis de Python es simple y fácil de entender. Su estilo de codificación facilita la lectura y escritura de código, lo que acelera el desarrollo y facilita la colaboración entre equipos.

Gran comunidad y soporte: Python cuenta con una comunidad activa de desarrolladores que contribuyen con bibliotecas, tutoriales y resuelven dudas en foros. Esto proporciona un valioso respaldo y facilita el aprendizaje y la resolución de problemas.

Amplia biblioteca estándar: *Python* incluye una biblioteca estándar extensa que cubre diversas áreas, desde manipulación de archivos hasta desarrollo web. Esto les permite a los desarrolladores aprovechar herramientas existentes sin necesidad de reinventar la rueda.

Versatilidad y portabilidad: *Python* es un lenguaje multiplataforma, lo que significa que el código escrito en *Python* puede ejecutarse en diferentes sistemas operativos sin modificaciones significativas. Esto facilita el desarrollo de aplicaciones que pueden ser utilizadas en una variedad de entornos.

Desarrollo rápido de prototipos: La facilidad de escritura y la sintaxis concisa hacen que Python sea ideal para el desarrollo rápido de prototipos. Esto les permite a los desarrolladores probar y experimentar con ideas antes de comprometerse con implementaciones más extensas.

Integración fácil con otros lenguajes: Python puede integrarse fácilmente con otros lenguajes, como C y C++. Esto le permite a los desarrolladores aprovechar código existente y obtener un rendimiento óptimo cuando sea necesario. (Maldonado, 2024)

2.4 Base de datos

En un proyecto de software, la base de datos es una parte fundamental del sistema, ya que se utiliza para almacenar, organizar y gestionar los datos del negocio o de la aplicación en cuestión. La base de datos debe estar diseñada de manera eficiente y segura, para garantizar la integridad y disponibilidad de los datos y su acceso por parte de los usuarios autorizados. (Connolly y Begg, 2014).

2.4.1 MySQL

MySQL es un sistema gestor de bases de datos (*SGBD*, *DBMS* por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros *SGBD* del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo. *MySQL* está disponible para múltiples plataformas, la seleccionada para los ejemplos de este libro es *GNU/Linux*. (Luis Alberto Casilla Santillan)

Modelo de datos entidad-relación (E-R)

Está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades. Las entidades se describen en una base de datos mediante un conjunto de atributos. Por ejemplo, los atributos número-cuenta y saldo describen una cuenta particular de un banco y pueden ser atributos del conjunto de entidades cuenta.

La estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama ER, que consta de los siguientes componentes:

- Rectángulos, que representan conjuntos de entidades.
- Elipses, que representan atributos.
- Rombos, que representan relaciones entre conjuntos de entidades.
- Líneas, que unen los atributos con los conjuntos de entidades y los conjuntos de entidades con las relaciones. (Silberschatz, 2022)

Características de *MySQL*

Al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar *MySQL* en alguno de sus proyectos porque al estar ampliamente extendido cuenta además con una ingente comunidad que ofrece soporte a otros usuarios. Pero estas no son las únicas características como veremos a continuación:

Arquitectura Cliente y Servidor: *MySQL* basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.

Compatibilidad con *SQL*: *SQL* es un lenguaje generalizado dentro de la industria. Al ser un estándar *MySQL* ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a *MySQL*.

Vistas: Desde la versión 5.0 de *MySQL* se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos *SQL*. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.

Procedimientos almacenados. *MySQL* posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.

Desencadenantes. *MySQL* permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.

Transacciones. Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas.

Historia y versiones de *MySQL*

La historia de *MySQL* comienza cuando Michael Widenius, un programador de software, buscaba una solución sencilla que permitiera dar salida a sus proyectos sin recurrir a soluciones de terceros. Al comienzo *MySQL* carecía de elementos considerados esenciales como la integridad referencial y transacciones pero a pesar de ello atrajo rápidamente a desarrolladores web. (Robledano, 2019)

CAPITULO III - MARCO METODOLÓGICO

Es un apartado de los proyectos de investigación en el que se detallan los métodos de análisis empleados para abordar el objeto de estudio. Dicho de otro modo, en este apartado se explica cuáles fueron los procedimientos de investigación aplicados y de qué manera fueron escogidos entre los diferentes abordajes metodológicos que existen. (Etecé, concepto, 2023)

3.1 Método general

Para este proyecto, se empleará el método participativo. Se identificará un problema relacionado con la gestión de inventarios y el proceso de pago en línea, y se trabajará en conjunto con los interesados para proponer una solución que satisfaga sus necesidades.

3.2 Métodos auxiliares

Además del método participativo, se utilizarán métodos auxiliares para complementar la investigación y obtener información adicional sobre variables específicas del proyecto, estos métodos incluirán encuestas y entrevistas para recolectar datos cuantitativos y cualitativos, así como revisiones de documentación existente y análisis de casos de estudio relacionados.

3.2.1 Experimental

Se llevarán a cabo pruebas para identificar posibles obstáculos en el proceso de gestión de inventarios y pago en línea, utilizando diferentes escenarios de prueba que simulen las interacciones de los usuarios y el funcionamiento del sistema.

3.2.2 Análisis

Se realizará un análisis exhaustivo de los procesos actuales de gestión de inventarios y pago en línea para identificar áreas de mejora, evaluando la eficiencia de cada etapa del proceso y recogiendo datos sobre su rendimiento.

3.2.3 Síntesis

Se elaborará un documento que resuma las principales ideas y hallazgos del proyecto, proporcionando una visión general clara y concisa.

3.3 Tipo de investigación

El proyecto se enmarca dentro de la investigación aplicada, que busca abordar problemas prácticos y aplicar soluciones concretas en un entorno real. Se apoyará en los resultados de la investigación básica para fundamentar las decisiones y acciones tomadas durante el desarrollo del sistema.

3.4 Prototipo

El sistema de gestión de inventarios con sistema de pago seguro en línea tiene como objetivo principal mejorar la eficiencia operativa y la seguridad de las transacciones en línea para la empresa. Se espera que el sistema permita una gestión más eficiente del inventario, así como una experiencia de compra segura y conveniente para los clientes. Además, se evaluará la viabilidad de expandir el uso del sistema a otros procesos de la empresa en el futuro.

3.5 Metodología del desarrollo de la programación

Para el desarrollo del proyecto, se aplicará la metodología ágil *Scrum* en conjunto con las tecnologías seleccionadas (*Python*, *Django* y *MySQL*) y las medidas de seguridad web (*SSL/TLS*). *Scrum* se adaptará para gestionar el desarrollo iterativo e incremental del sistema de gestión de inventarios y pago en línea.

Este enfoque garantizará la creación de un sistema robusto, escalable y seguro que cumpla con los objetivos del proyecto y satisfaga las necesidades de la empresa y sus clientes.

CAPÍTULO IV – ANÁLISIS DE LA APLICACIÓN O SOLUCIÓN

Es una sección fundamental en cualquier proyecto de desarrollo de software que tiene como objetivo analizar detalladamente la solución propuesta, desglosando sus características principales, los usuarios involucrados y el contexto en el cual se desarrollará. Este capítulo ofrece una visión profunda del sistema, identificando sus componentes esenciales, los flujos de trabajo y los actores clave que interactuarán con la aplicación.

Se exploran aspectos como la descripción del proyecto, donde se contextualiza la razón de ser del sistema y se detallan los problemas que pretende resolver. También se realiza una identificación de usuarios, identificando los roles principales que usarán el sistema y sus responsabilidades. Además, incluye un modelado del negocio, que describe cómo el sistema se integra en el contexto comercial o funcional para satisfacer las necesidades del cliente o la empresa.

4.1 Descripción del proyecto

El Sistema Web de Tienda en Línea está desarrollado para la tienda de ropa "Blessing", una empresa localizada en Zacapa, Guatemala. Este proyecto tiene como objetivo optimizar dos aspectos fundamentales del negocio: la gestión de inventarios y los pagos en línea. Ambos procesos son vitales para garantizar una experiencia eficiente y agradable para los clientes, al mismo tiempo que mejoran la capacidad operativa de la tienda. La solución permitirá que los administradores gestionen el inventario de productos en tiempo real, asegurando que los niveles de stock se mantengan actualizados tanto en la tienda física como en la plataforma digital. De igual manera, el sistema ofrecerá a los clientes la posibilidad de acceder al catálogo en línea, seleccionar productos, añadirlos al carrito y efectuar pagos mediante pasarelas de pago seguras, ampliando la experiencia de compra de manera sencilla e intuitiva.

4.2 Identificación de usuarios

El sistema web cuenta con tres tipos principales de usuarios:

Administrador de la tienda: Este usuario tiene el control total del sistema, permitiéndole gestionar el inventario, actualizar el catálogo, crear productos, manejar pedidos y supervisar la administración general del sistema, asegurando que la tienda funcione correctamente.

Clientes: Los usuarios finales del sistema son los clientes, quienes tienen acceso a navegar por el catálogo de productos, añadir productos al carrito, realizar compras y pagar de manera segura a través de distintas opciones de pago.

Soporte técnico: Personal encargado de garantizar el buen funcionamiento del sistema, realizando tareas de mantenimiento, actualizaciones y resolución de problemas técnicos, asegurando que la plataforma esté disponible y operativa en todo momento.

4.3 Modelado del negocio

La tienda "Blessing" se basa en un modelo de negocio minorista enfocado en la venta de ropa a nivel local y, mediante la implementación del sistema web, busca expandirse a nivel nacional. El sistema web permitirá que los clientes puedan comprar productos desde cualquier lugar en Guatemala, brindando mayor alcance y optimizando la operatividad de la tienda. Los componentes clave del negocio que se verán beneficiados por el sistema incluyen:

Gestión de inventarios: La planificación y el control de los niveles de stock se mejorarán significativamente, con actualizaciones en tiempo real que garantizan que la información del inventario sea precisa tanto en la tienda física como en línea.

Venta en línea: La integración del catálogo digital permitirá a los clientes visualizar productos disponibles, ver detalles, seleccionar opciones, agregar al carrito y realizar compras con total seguridad.

Logística: La cadena de suministro y los procesos de entrega de productos se optimizarán, permitiendo a la tienda satisfacer las demandas de los clientes con mayor eficiencia y rapidez.

4.4 Metodología de desarrollo de software

El desarrollo del sistema se llevará a cabo utilizando la metodología Ágil Scrum, que ha demostrado ser altamente eficaz en proyectos de software gracias a su enfoque en la adaptabilidad y la entrega de valor continuo. Scrum se basa en ciclos cortos e iterativos denominados *sprints*, los cuales permiten realizar entregas frecuentes de partes funcionales del sistema, ajustando el desarrollo según los **comentarios de los usuarios**. Este enfoque incrementa la calidad del producto final y asegura que se alineen los resultados con las necesidades del negocio. Características de Scrum:

- **Sprints:** Iteraciones de desarrollo que durarán entre dos y cuatro semanas, con entregas de funcionalidades al final de cada ciclo.
- **Product Backlog:** Una lista priorizada de tareas y funcionalidades que necesita el sistema, la cual será gestionada y adaptada a lo largo del proyecto según las prioridades del negocio.

- ***Sprint Review***: Al finalizar cada sprint, se presentará una versión funcional del sistema para su evaluación y retroalimentación.
- ***Sprint Retrospective***: Reflexión sobre el proceso y ajustes necesarios para mejorar el siguiente ciclo de trabajo.

4.5 Fases del proyecto

El desarrollo del sistema se llevará a cabo en las siguientes fases:

- **Análisis de requerimientos**: Se identificarán las necesidades específicas de la tienda "Blessing" y sus clientes, incluyendo los flujos de trabajo para la gestión de inventarios, el carrito de compras, y los métodos de pago.
- **Diseño del sistema**: Se crearán los diagramas de flujo, arquitectura y modelos de datos para garantizar que el sistema sea escalable y seguro. Se determinará la estructura del sistema utilizando tecnologías como *Django* para el backend y MySQL para la base de datos.
- **Desarrollo**: Cada módulo del sistema se desarrollará de forma iterativa durante los sprints, implementando funcionalidades como la creación de productos, la gestión de inventario, y la integración de pasarelas de pago.
- **Pruebas**: Se llevarán a cabo pruebas de funcionalidad y de usuario para asegurar que el sistema cumpla con los requerimientos, identificando y corrigiendo errores antes del lanzamiento.
- **Implementación**: El sistema se desplegará en un servidor web, y se capacitará al equipo de "Blessing" en su uso.
- **Mantenimiento y mejora continua**: Después de la implementación, se realizará un seguimiento del sistema para aplicar mejoras y actualizaciones según las necesidades emergentes.

4.6 Características y artefactos

Características del Sistema:

- **Gestión de inventarios en tiempo real**: Se refiere a la capacidad del sistema para monitorear, actualizar y reflejar de manera automática y continua los niveles de stock de productos en la tienda, tanto en el entorno físico como en la tienda en línea. Esto significa que cada vez que se realiza una venta, ya sea en la tienda física o a través del sitio web, el inventario se ajusta instantáneamente en todos los canales. De este modo, la tienda tiene información actualizada sobre la cantidad de productos disponibles en cualquier momento.

- **Carrito de compras y pagos en línea:** El carrito de compras es una funcionalidad esencial en una tienda en línea, diseñada para permitir a los clientes seleccionar productos, revisarlos y proceder a la compra de manera sencilla y organizada. Este carrito almacena temporalmente los productos que el usuario desea comprar, ofreciendo opciones para modificar la cantidad, eliminar artículos o seguir comprando antes de finalizar la transacción.
- **Catálogo digital:** Presentación visual de los productos con detalles sobre precios y disponibilidad.
- **Sistema de facturación electrónica:** Generación automática de facturas para cada compra realizada, cumpliendo con las normativas locales.

Artefactos generados:

- ***Product Backlog*:** Lista priorizada de funcionalidades y tareas del sistema.
- **Diagramas UML:** Modelado de casos de uso, clases y secuencia para el desarrollo del sistema.
- **Documentación técnica:** Descripción detallada de la arquitectura del sistema, base de datos y API utilizadas.

CAPITULO V -DISEÑO DE LA APLICACIÓN O SOLUCIÓN

En este capítulo se presentará una descripción detallada del diseño de la aplicación propuesta, que busca optimizar la gestión de inventarios y pagos en línea para la tienda de ropa "Blessing". La solución está diseñada para ofrecer una experiencia de usuario intuitiva y eficiente, facilitando la navegación a través de un entorno web responsivo y atractivo.

5.1 Arquitectura del Sistema

Describe la combinación de capas tecnológicas que interactúan para soportar el flujo de datos, la lógica de negocio, la gestión de usuarios, y las transacciones en línea.

Cliente (front-end): La interfaz gráfica de usuario está desarrollada utilizando tecnologías como HTML, CSS, JavaScript y Bootstrap, brindando una experiencia fluida y responsiva. El cliente puede navegar por los productos, realizar búsquedas, agregar artículos al carrito y completar el proceso de compra.

Servidor de Aplicaciones: El *backend* está implementado con *Django* y *Python*, manejando la lógica de negocio, autenticación de usuarios, gestión de sesiones y permisos. El sistema gestiona las compras, el inventario, los reportes, y el procesamiento de pagos en línea.

Capa de Persistencia (Base de Datos): La base de datos utiliza *MySQL* para almacenar datos estructurados sobre productos, usuarios, categorías y transacciones. Se asegura la integridad de los datos y se realiza una gestión segura de la información.

Capa de Pagos y Servicios Externos: Para el procesamiento de pagos en línea se integran las *APIs* de *Stripe* y *Cybersource*, proporcionando una plataforma segura para realizar transacciones con tarjetas de crédito o débito. Se utiliza un sistema de *webhooks* para recibir notificaciones sobre el estado de las transacciones.

Capa de Gestión de Inventario: El sistema de inventario, basado en *Django ORM* y *Python*, monitorea en tiempo real la disponibilidad de productos. Al producirse una venta o ingreso de nuevos productos, el inventario se actualiza automáticamente. Además, el sistema emite alertas cuando el stock está bajo, permitiendo al administrador tomar decisiones rápidas y efectivas sobre el reabastecimiento.

Servidor Web: Son utilizados para manejar eficientemente las solicitudes HTTP y distribuir la carga entre el cliente y el servidor.

Infraestructura de *Hosting*: El sistema se aloja en la nube, utilizando plataformas como *digitalocean*, lo que permite escalar los recursos de servidor según las necesidades de tráfico. La infraestructura en la nube garantiza disponibilidad y flexibilidad, alojando tanto la aplicación como la base de datos, además de proveer almacenamiento para medios como imágenes de productos.

5.2 Diagramas UML

Un diagrama UML es una representación visual que ayuda a entender y comunicar cómo funciona un sistema de software. Estos diagramas pueden mostrar diferentes aspectos, como la estructura, el comportamiento o las interacciones de los componentes del sistema. Es como un mapa que te permite ver y planificar el diseño de un software de manera clara y organizada.

5.3 Diagramas de caso de uso

Un diagrama de casos de uso se utiliza para mostrar cómo los usuarios interactúan con un sistema. Representa las diferentes maneras en que los usuarios pueden utilizar el software, destacando las funciones que el sistema debe ofrecer. Es como un guion que describe las acciones que pueden realizar los actores (que son los usuarios o sistemas externos) y cómo el sistema responde a esas acciones.

5.3.1 Diagrama de caso de uso para la gestión de inventarios

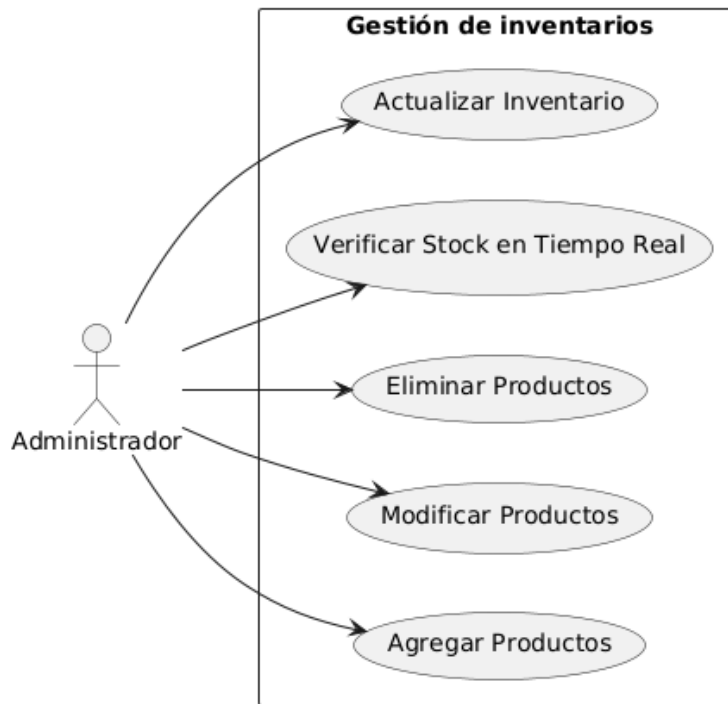


Figura #1. Diagrama caso de uso para la gestión de inventarios.

Fuente: Elaboración propia.

Este diagrama de casos de uso UML muestra las funciones de gestión de inventarios disponibles para un administrador. Incluye cinco acciones principales: actualizar inventario, verificar stock en tiempo real, eliminar, modificar y agregar productos. Todas estas operaciones están contenidas dentro del sistema de "Gestión de inventarios". El diagrama proporciona una visión general clara de las capacidades del administrador para manejar el inventario de productos.

5.3.2 Diagrama de caso de uso para el proceso de compras

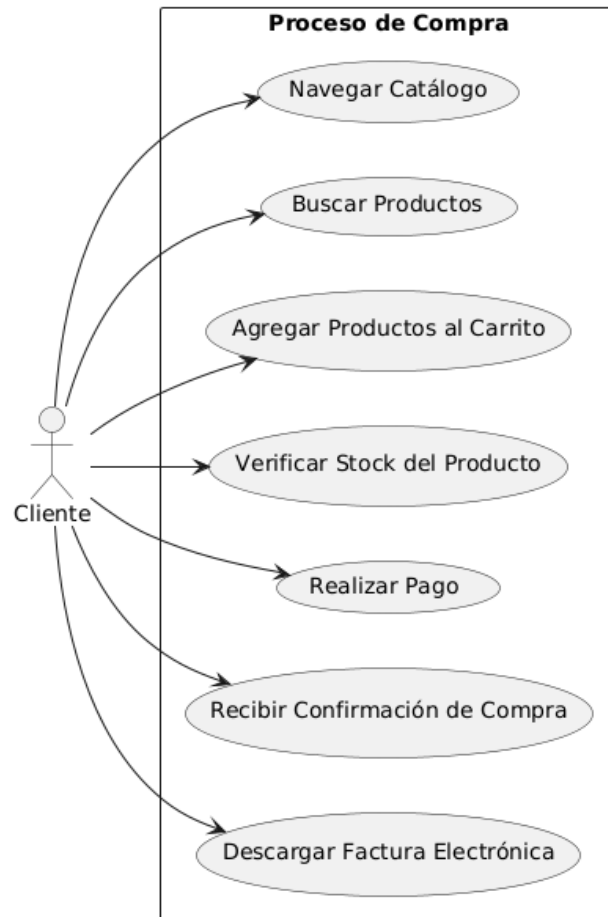


Figura #2. Diagrama caso de uso para la para el proceso de compras

Fuente: Elaboración propia.

Este diagrama de casos de uso UML ilustra el proceso de compra desde la perspectiva del cliente. Muestra siete acciones principales que un cliente puede realizar: navegar el catálogo, buscar productos, agregar productos al carrito, verificar el stock, realizar el pago, recibir confirmación de compra y descargar la factura electrónica. El diagrama proporciona una visión clara y concisa de las etapas típicas en una experiencia de compra online, desde la exploración inicial hasta la finalización de la transacción.

5.3.3 Diagrama de caso de uso para la gestión de pagos en línea

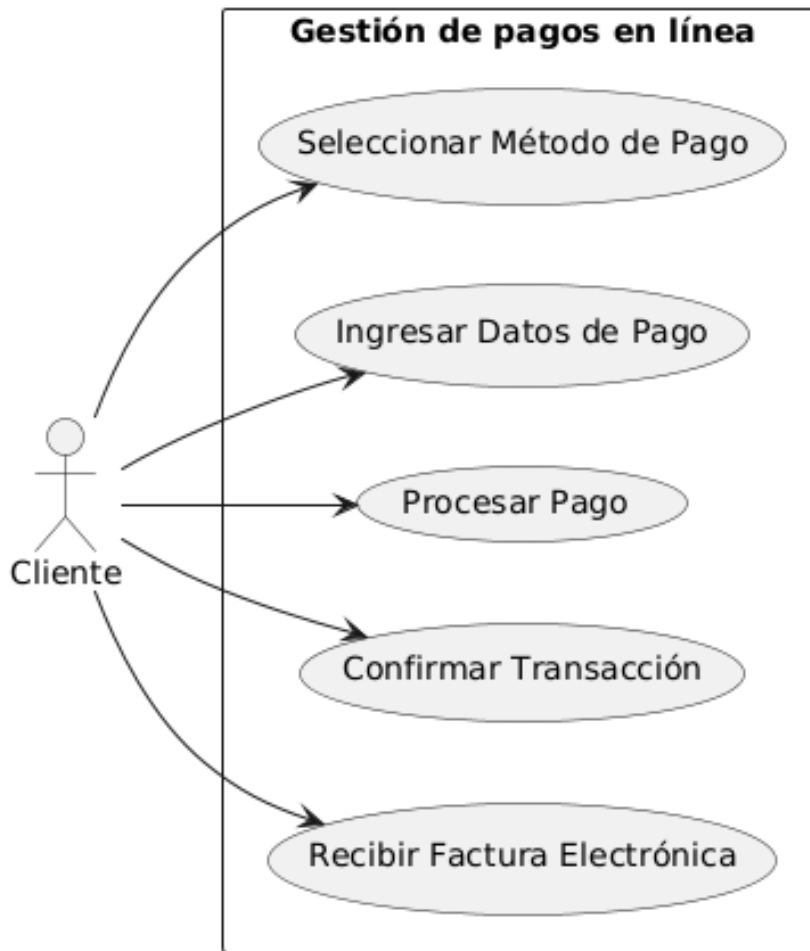


Figura #3. Diagrama caso de uso para la para la gestión de pagos en línea Fuente: Elaboración propia.

Este diagrama es útil para entender las diferentes etapas del proceso de pago en línea y las interacciones específicas que el cliente debe llevar a cabo para completar su compra. Facilita la identificación de requisitos y ayuda a los desarrolladores a diseñar un sistema que cumpla con las necesidades del usuario.

5.3.4 Diagrama de caso de uso para agregar productos

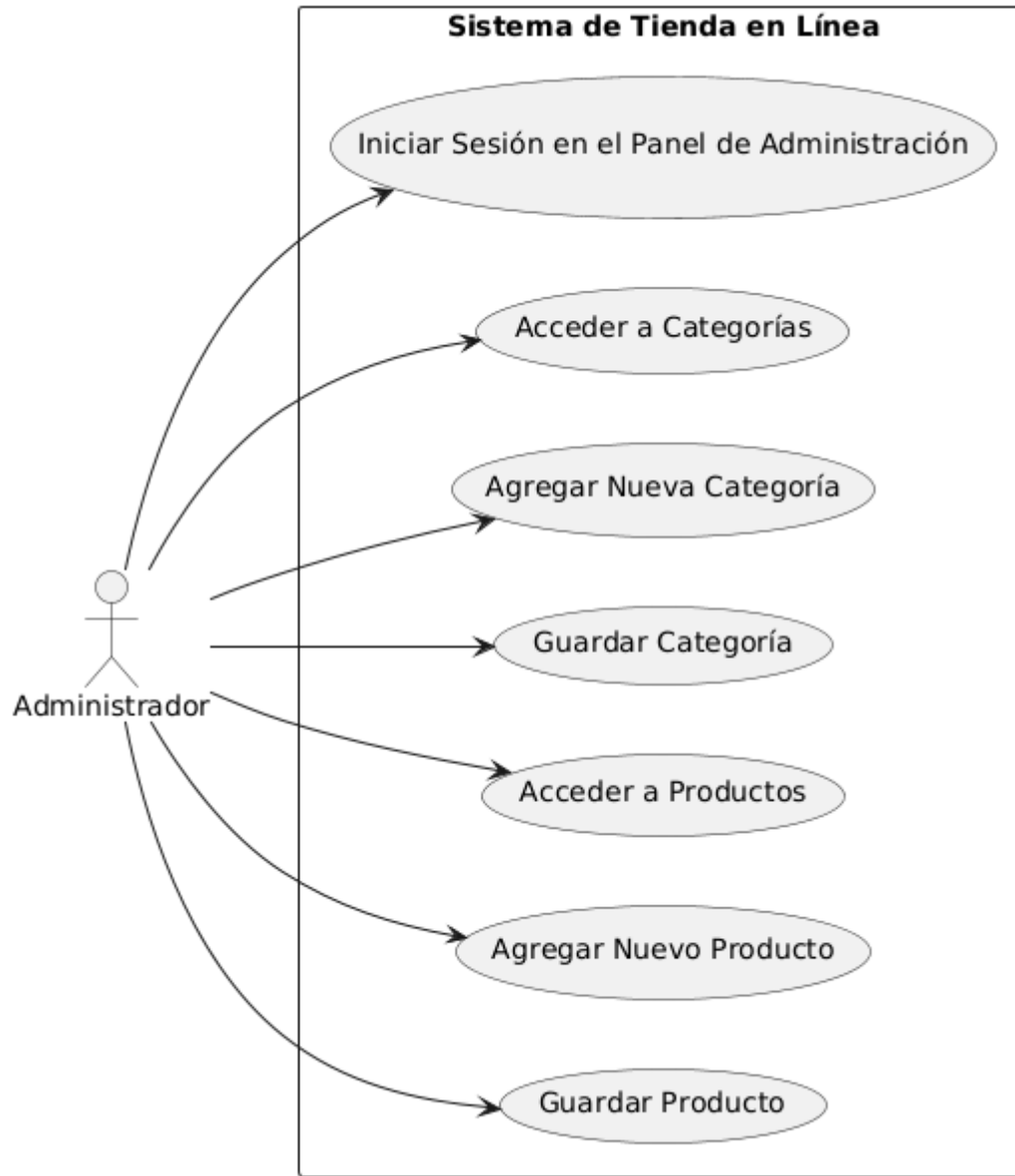


Figura #4. Diagrama caso de uso para la para la agregar productos

Fuente: Elaboración propia.

Este diagrama ilustra las interacciones entre el administrador y el sistema para agregar categorías y productos. Su relevancia radica en resaltar cómo se gestionan las entradas de información en el inventario, asegurando que el administrador tenga una herramienta eficiente para organizar los productos, lo que mejora la experiencia del cliente y la administración del inventario.

5.3.5 Diagrama de caso de uso para la administración del sistema



Figura #5. Diagrama caso de uso para la administración del sistema

Fuente: Elaboración propia.

5.4 Diagrama de actividades

Un diagrama de actividades es una representación visual que muestra cómo se lleva a cabo un proceso o una tarea, desglosando las acciones y decisiones que se toman a lo largo del camino.

5.4.1 Diagrama de actividades para la gestión de inventarios

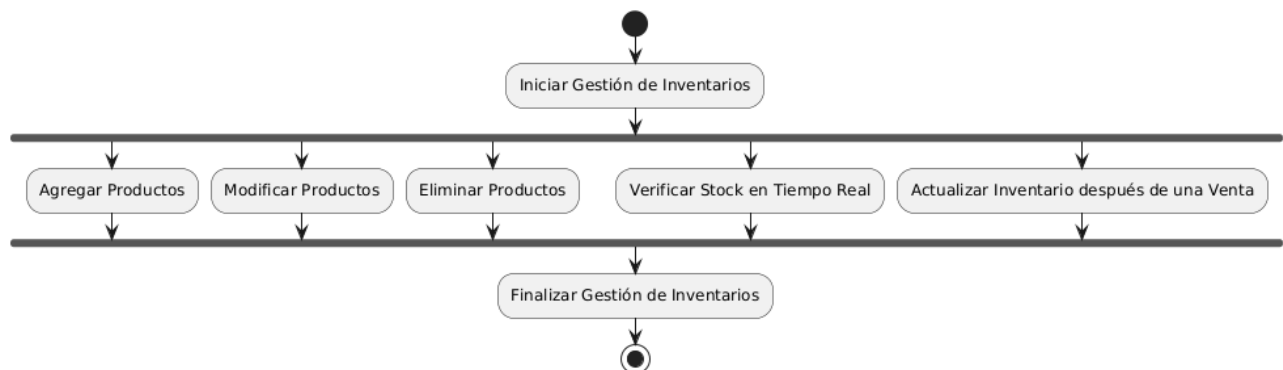


Figura #6. Diagrama de actividades para la gestión de inventarios.

Fuente: Elaboración propia.

Este diagrama de actividades proporciona una visión clara y organizada de los procesos implicados en la gestión de inventarios, facilitando la comprensión y el análisis de las operaciones

5.4.2 Diagrama de actividades para el proceso de compras

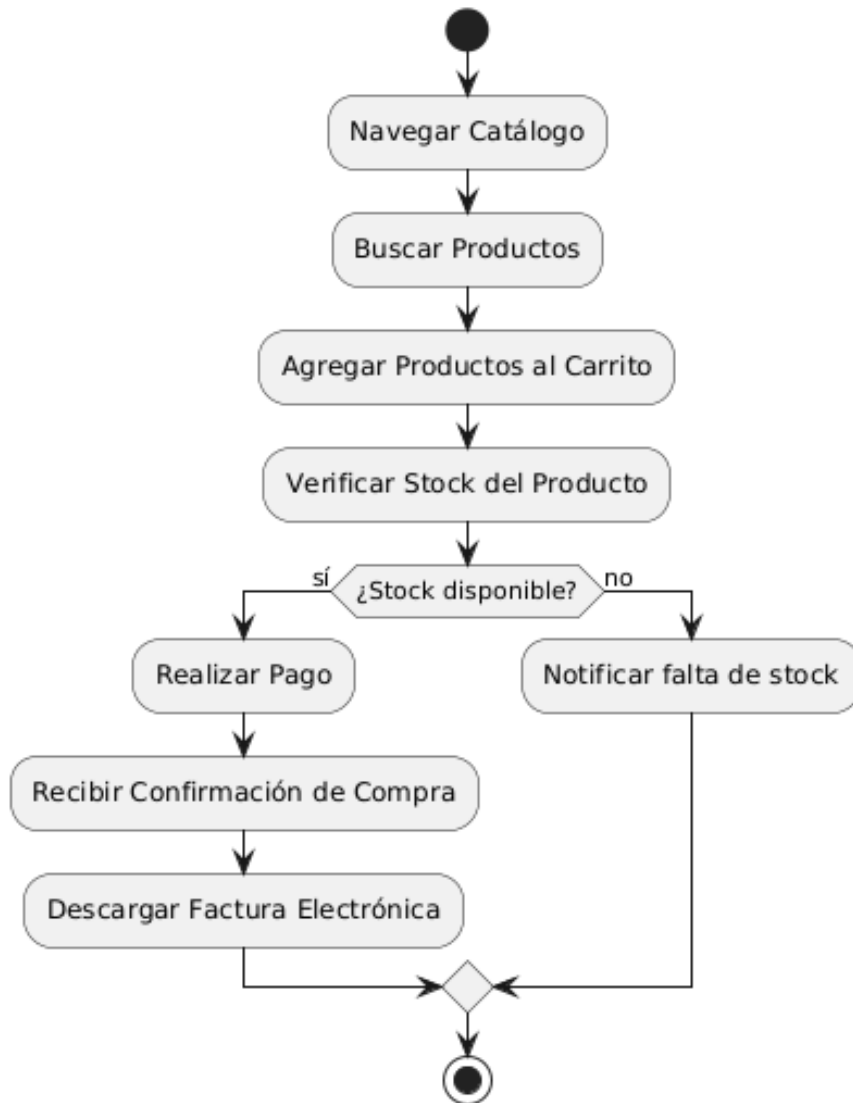


Figura #7. Diagrama de actividades para la para el proceso de compras

Fuente: Elaboración propia.

Este diagrama de actividades ilustra de manera detallada el flujo de acciones que un cliente sigue al realizar una compra en la tienda en línea. Comienza con la selección de productos y abarca todos los pasos necesarios hasta la confirmación del pedido.

5.4.3 Diagrama de actividades para la gestión de pagos en línea

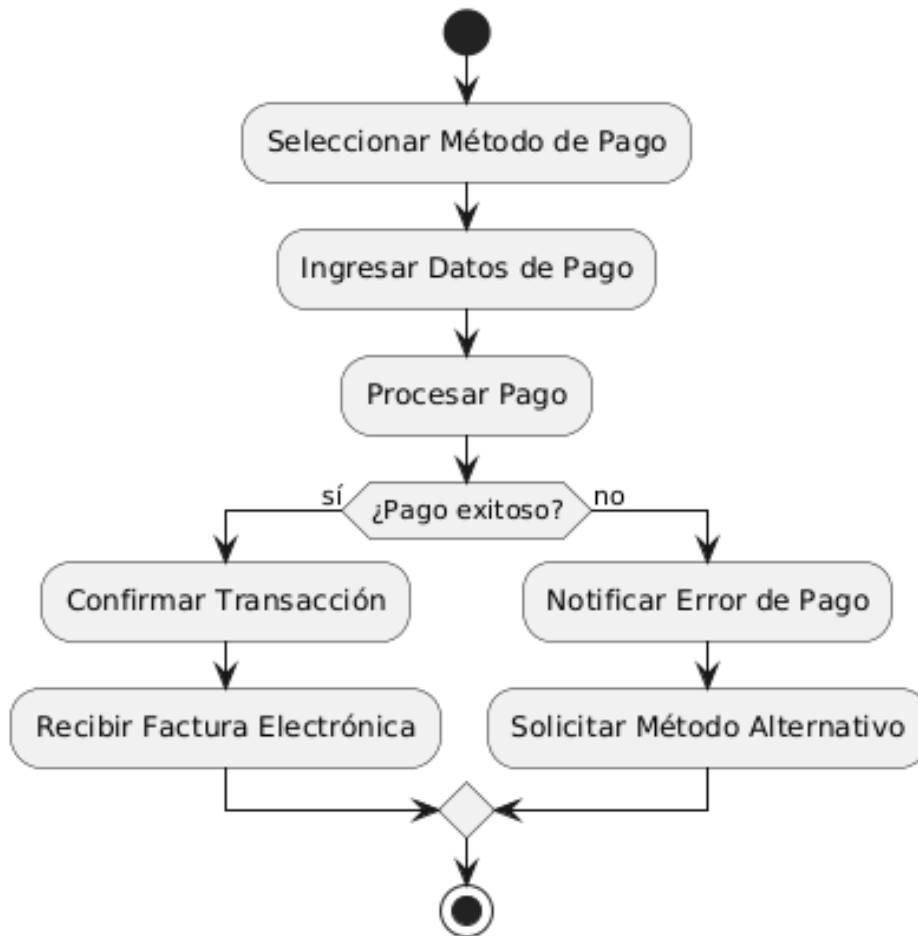


Figura #8. Diagrama de actividades para la para la gestión de pagos en línea Fuente: Elaboración propia.

Este diagrama de actividades describe el flujo de acciones involucradas en la gestión de pagos en línea. Desde la iniciación del pago por parte del cliente hasta la confirmación de la transacción, cada paso es crucial para garantizar que los pagos se procesen de manera segura y eficiente.

5.4.4 Diagrama de actividades para agregar productos

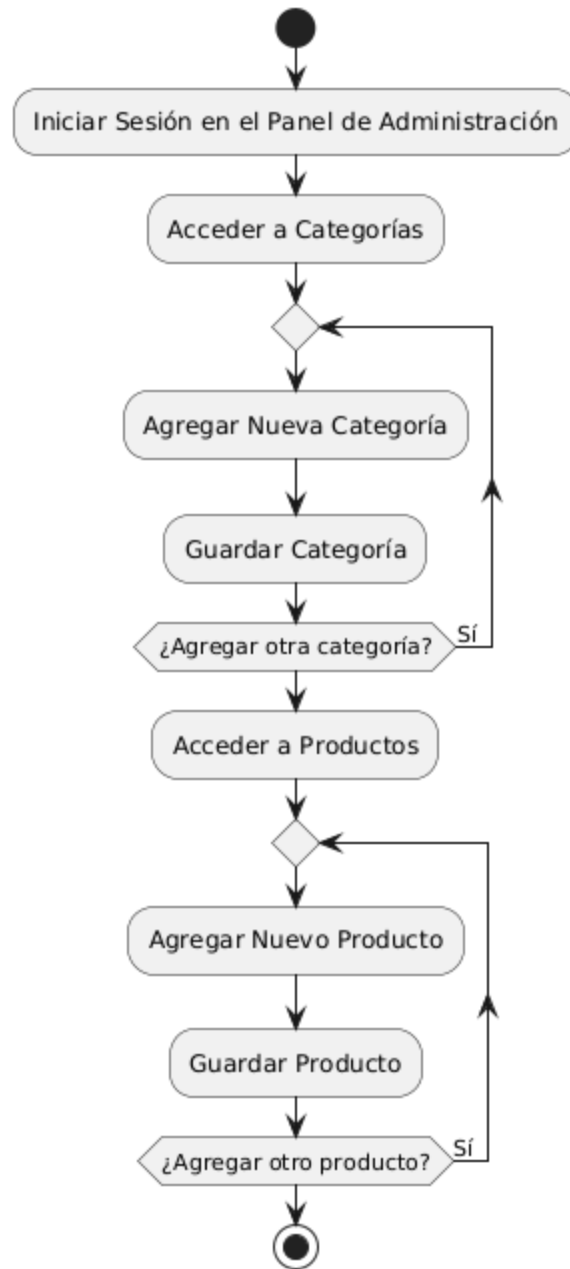


Figura #9. Diagrama de actividades para la para la agregar productos

Fuente: Elaboración propia.

Este diagrama representa de manera secuencial y detallada el flujo de actividades necesarias para gestionar productos en la tienda en línea. Su importancia radica en mostrar cada paso del proceso, desde la creación de un nuevo producto hasta su publicación en la tienda, lo que facilita la identificación de áreas de mejora y optimización en la gestión del inventario.

5.4.5 Diagrama de actividades para la administración del sistema

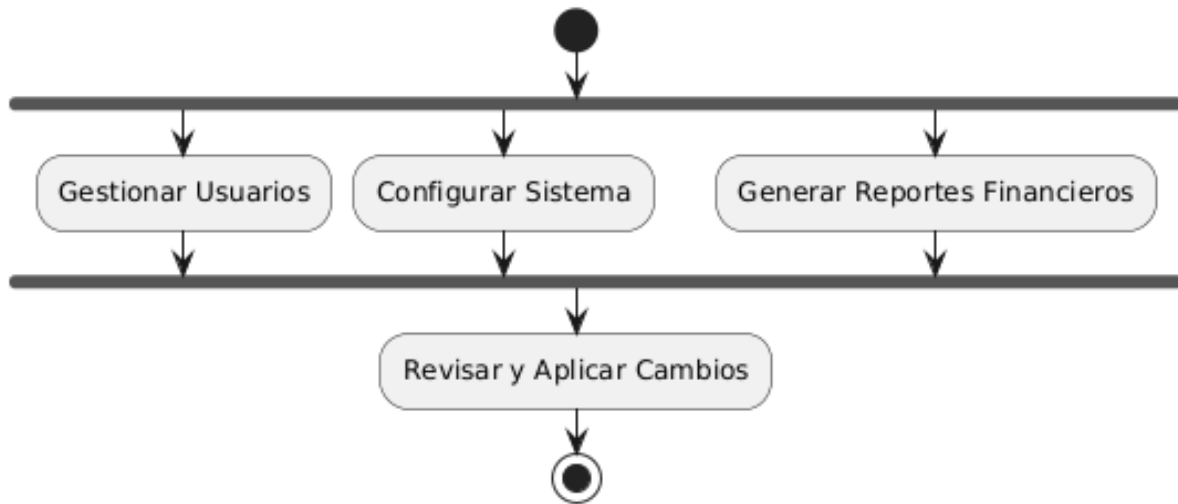


Figura #10. Diagrama de actividades para la administración del sistema

Fuente: Elaboración propia.

Este diagrama de actividades para la administración del sistema proporciona una representación clara y estructurada de las diversas funciones que un administrador puede realizar en el entorno de la tienda en línea. A través de la visualización de estas actividades, se logra identificar y comprender el flujo de tareas involucradas en la gestión de usuarios, configuraciones y generación de reportes.

5.5 Diagrama de secuencias

es una representación gráfica que muestra cómo los objetos en un sistema interactúan entre sí a lo largo del tiempo, destacando el orden en que se intercambian mensajes. Este tipo de diagrama se utiliza para modelar procesos específicos, permitiendo visualizar la secuencia de pasos que ocurren durante una interacción particular, como un proceso de compra o la gestión de un pedido.

5.5.1 Diagrama de secuencias para la gestión de inventarios

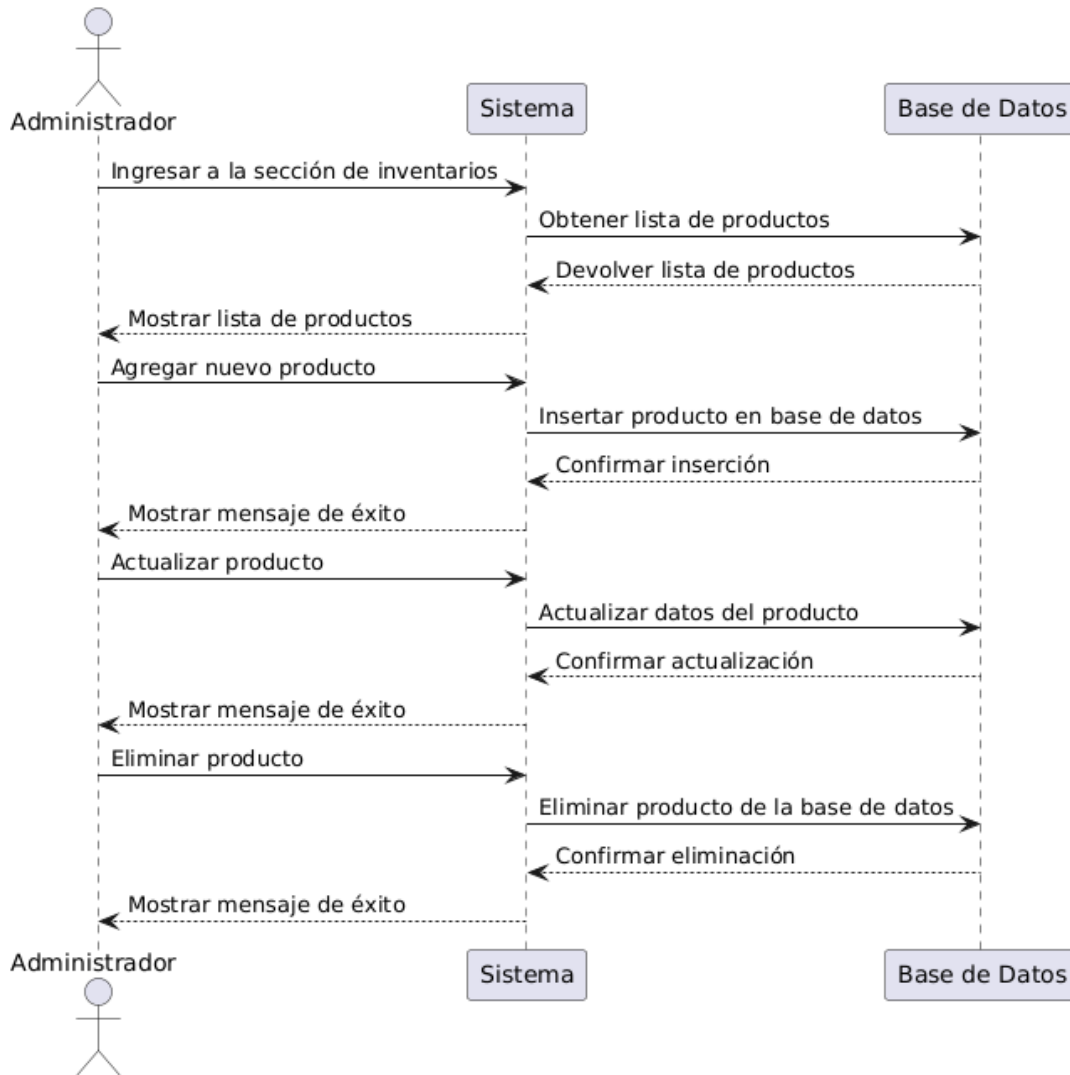


Figura #11. Diagrama de secuencias para la gestión de inventarios.

Fuente: Elaboración propia.

Este diagrama detalla el flujo de información entre el sistema y la base de datos durante la gestión de inventarios. Su relevancia radica en mostrar cómo se actualizan y consultan los datos de inventario, lo que permite mantener un control efectivo sobre los productos disponibles y evitar problemas de stock, garantizando una operación fluida de la tienda.

5.5.2 Diagrama de secuencias para el proceso de compras

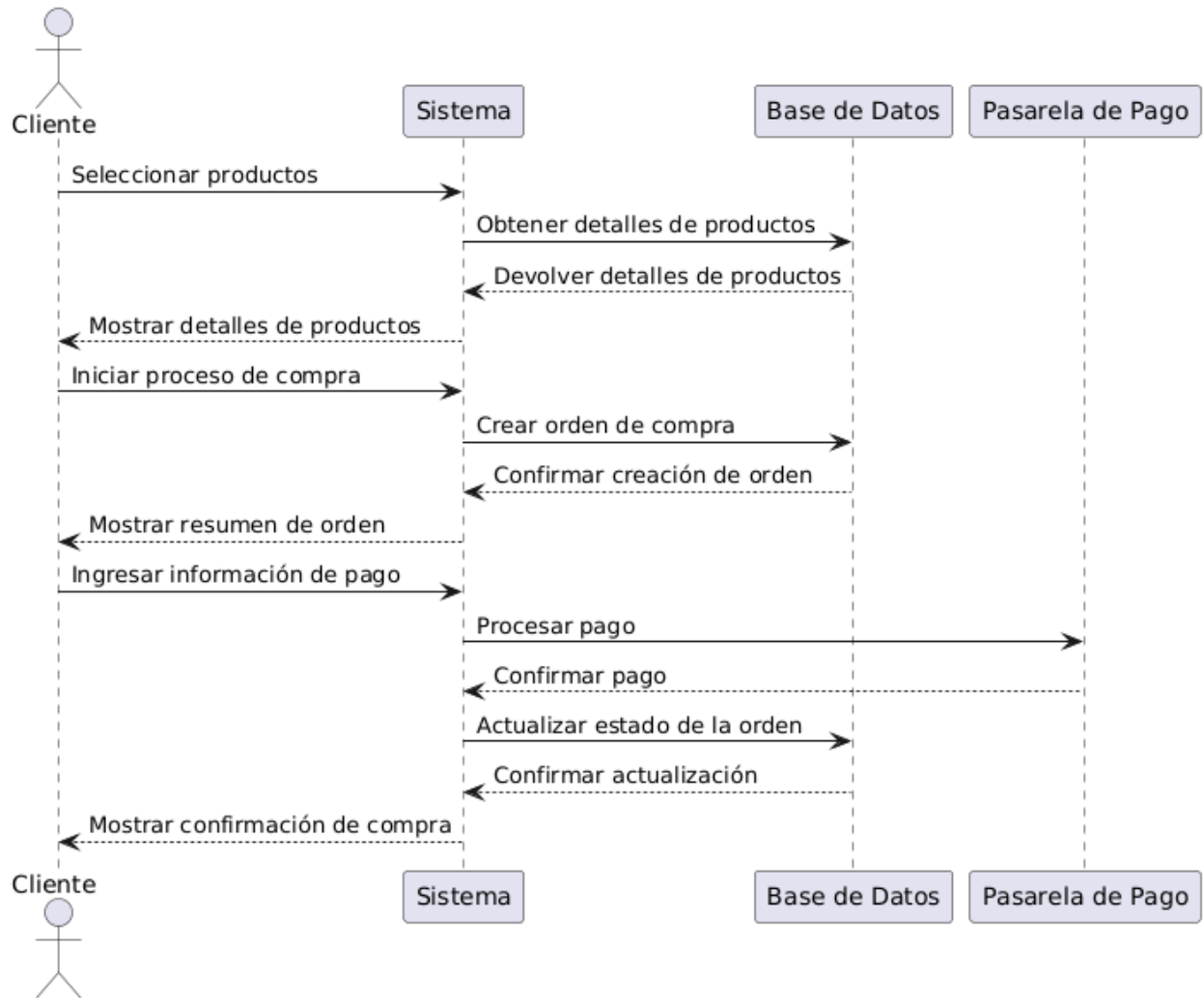


Figura #12. Diagrama de secuencias para la para el proceso de compras

Fuente: Elaboración propia.

Este diagrama ilustra el proceso de compras desde la selección de productos hasta la confirmación del pedido. Su importancia radica en mostrar cómo interactúan los diferentes componentes del sistema para asegurar que la experiencia de compra sea eficiente y satisfactoria, optimizando la conversión de visitantes en clientes.

5.5.3 Diagrama de secuencias para la gestión de pagos en línea

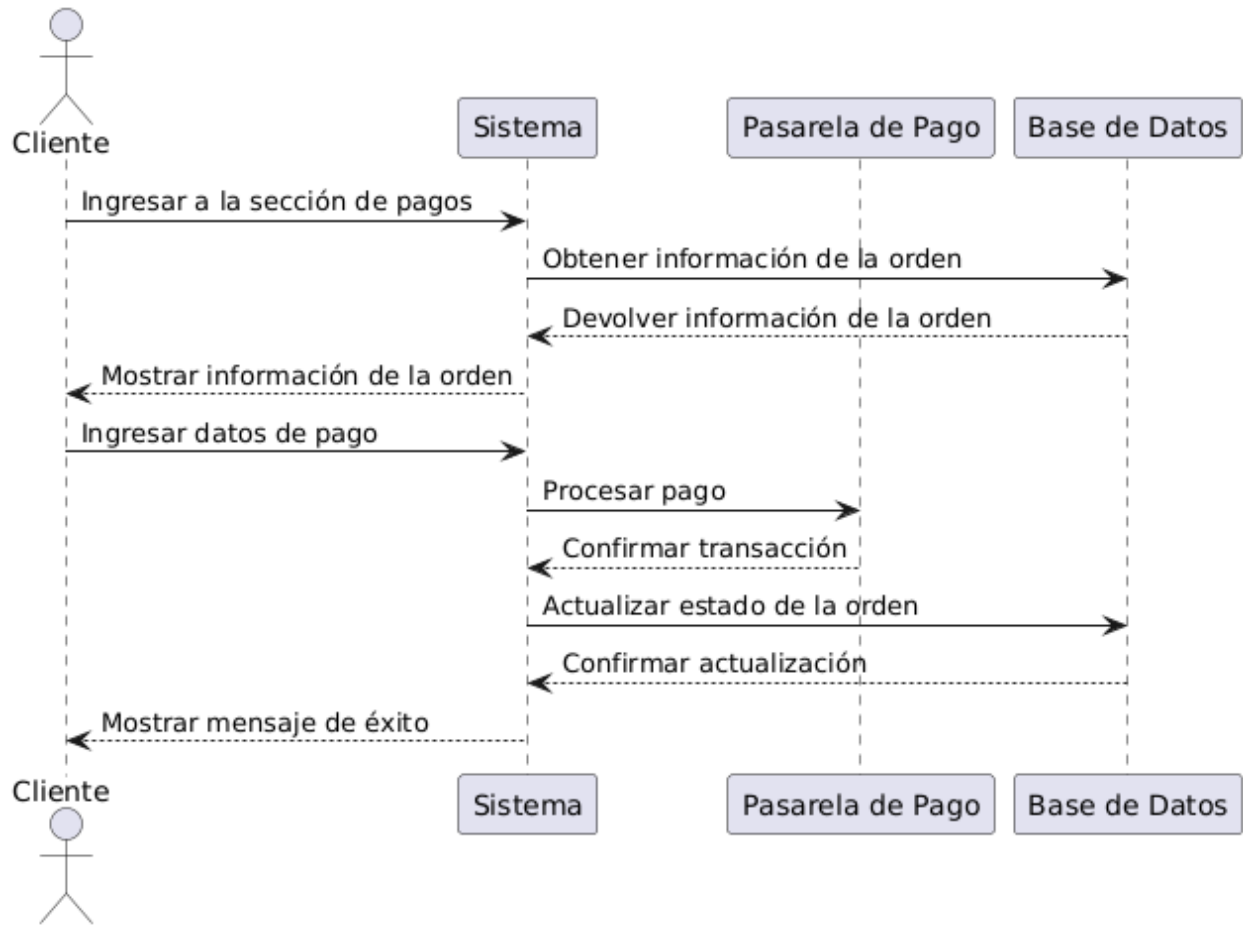


Figura #13. Diagrama de secuencias para la para la gestión de pagos en línea Fuente: Elaboración propia.

Este diagrama representa el flujo de información en el proceso de pagos en línea, desde que el cliente realiza el pago hasta que se confirma la transacción. Su relevancia se encuentra en asegurar que las transacciones sean seguras y eficientes, lo que es crucial para la confianza del cliente y el éxito del comercio electrónico.

5.5.4 Diagrama de secuencias para agregar productos

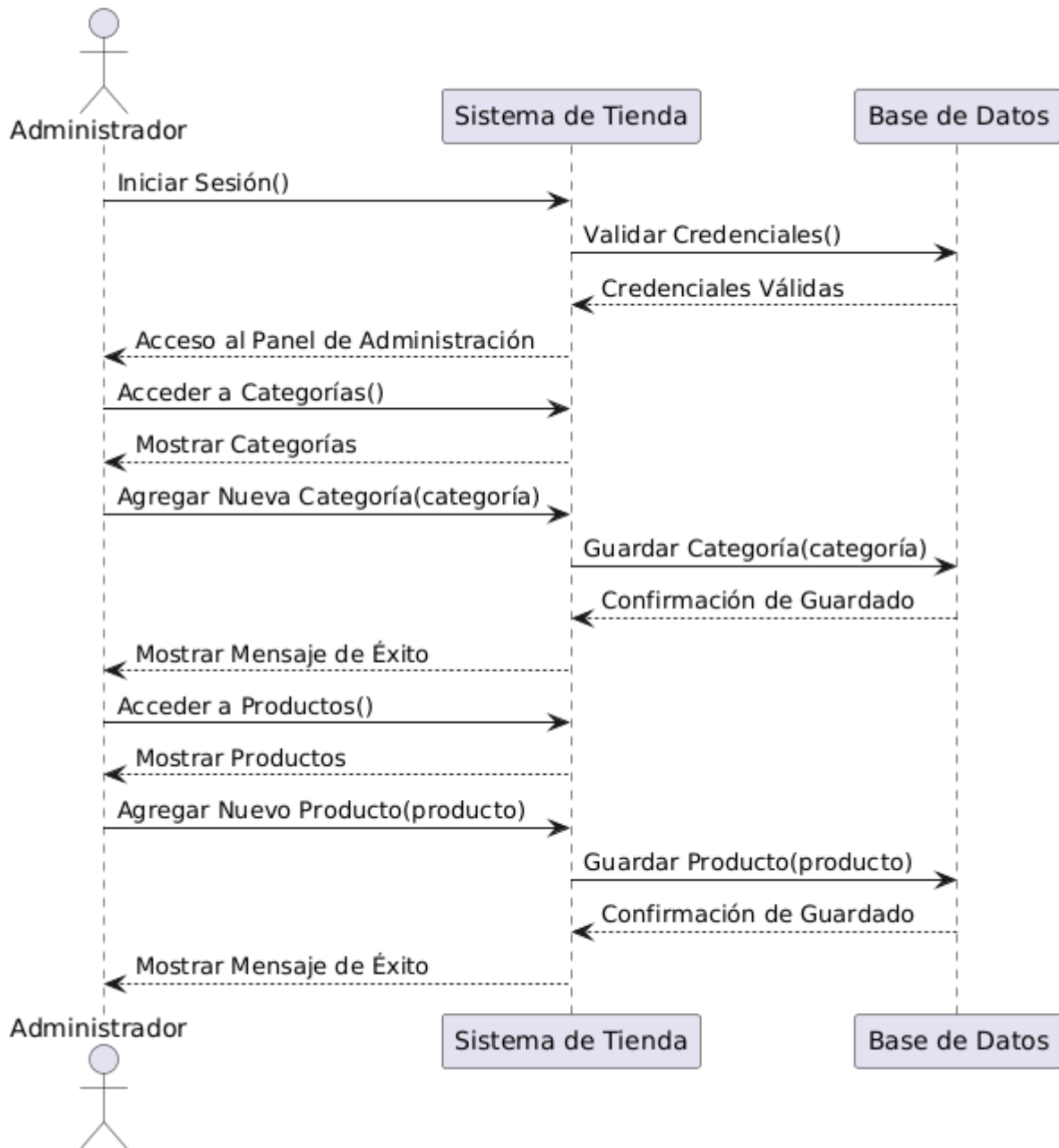


Figura #14. Diagrama de secuencias para la para la agregar productos

Fuente: Elaboración propia.

Este diagrama de secuencias proporciona una representación clara y estructurada de cómo el administrador interactúa con el sistema para agregar categorías y productos.

5.5.5 Diagrama de secuencias para la administración del sistema

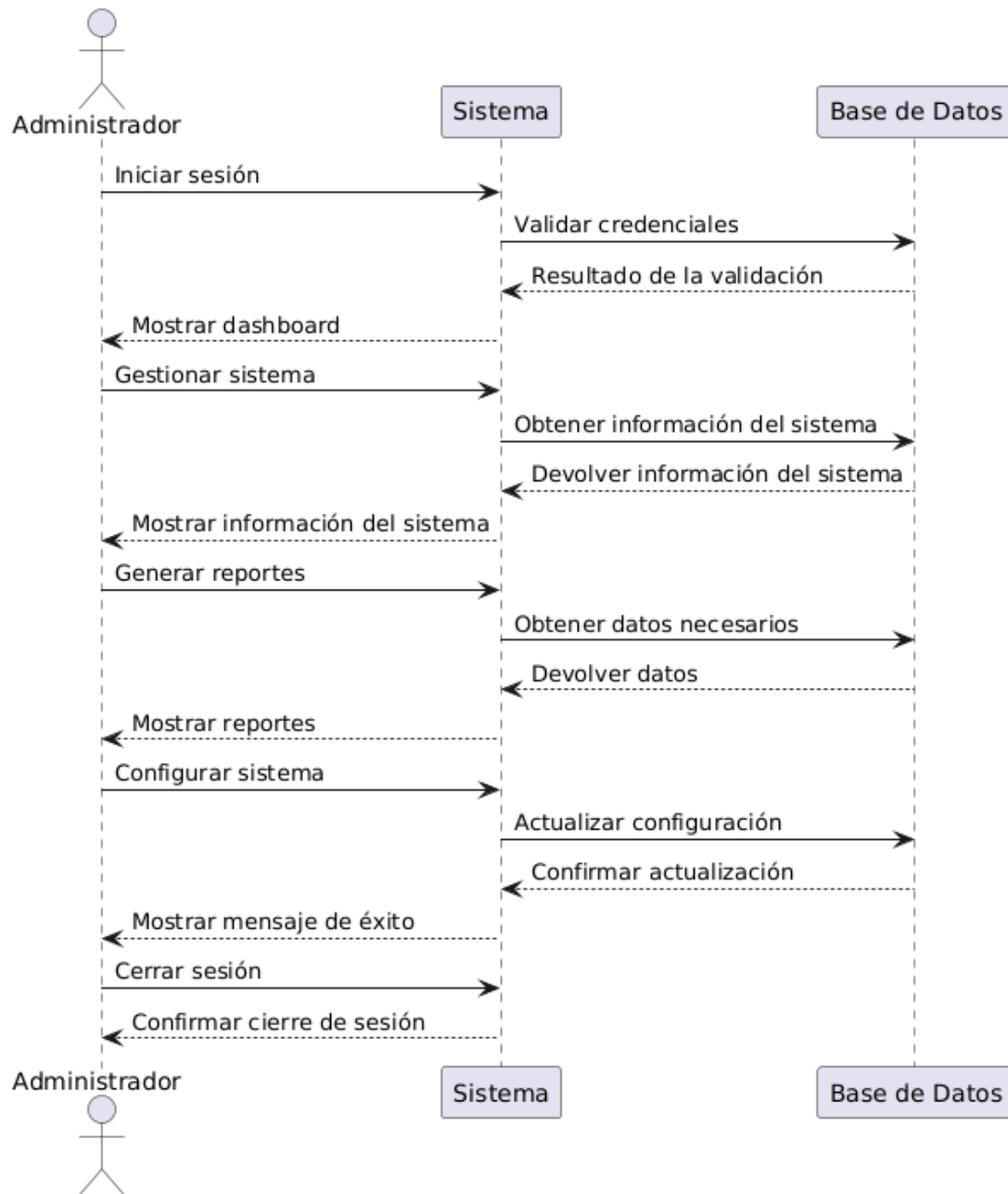


Figura #15. Diagrama de secuencias para la administración del sistema

Fuente: Elaboración propia.

Este diagrama proporciona una visión clara de las interacciones entre el administrador y el sistema durante la administración general. Destaca la importancia de las funciones administrativas, como la gestión del sistema y la generación de reportes, asegurando que el administrador tenga acceso a la información necesaria para tomar decisiones informadas y mantener la eficiencia operativa.

5.6 Diagrama de clases

Es una representación visual de la estructura de un sistema, que muestra las clases, sus atributos, métodos y las relaciones entre ellas. Es una herramienta fundamental en el diseño orientado a objetos, ya que permite a los desarrolladores y diseñadores entender cómo se organizan y se interrelacionan los diferentes componentes del sistema.

5.6.1 Diagrama de clases para la gestión de inventarios

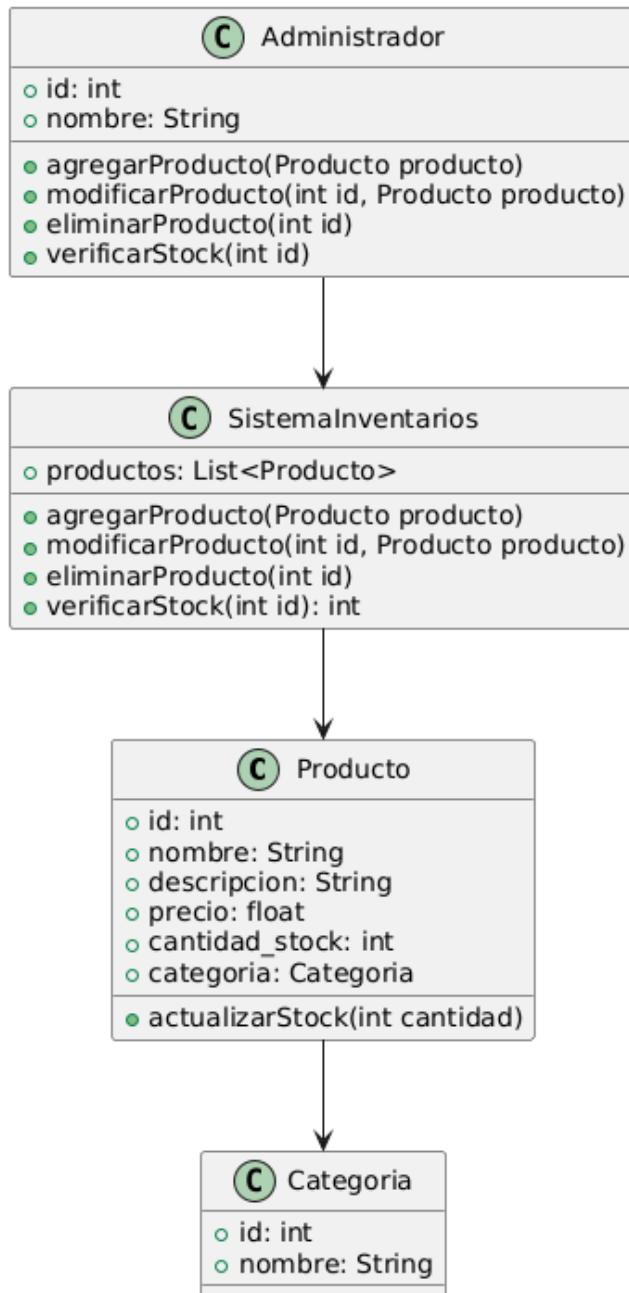


Figura #16. Diagrama de clases para la gestión de inventarios.

Fuente: Elaboración propia.

La representación de las clases involucradas en la gestión de inventarios permite entender cómo se organizan los datos relacionados con los productos y su stock. Este diagrama facilita la identificación de las relaciones entre los productos, categorías y el sistema de control de inventarios, asegurando que se mantenga la coherencia y la integridad de los datos en la aplicación.

5.6.2 Diagrama de clases para el proceso de compras

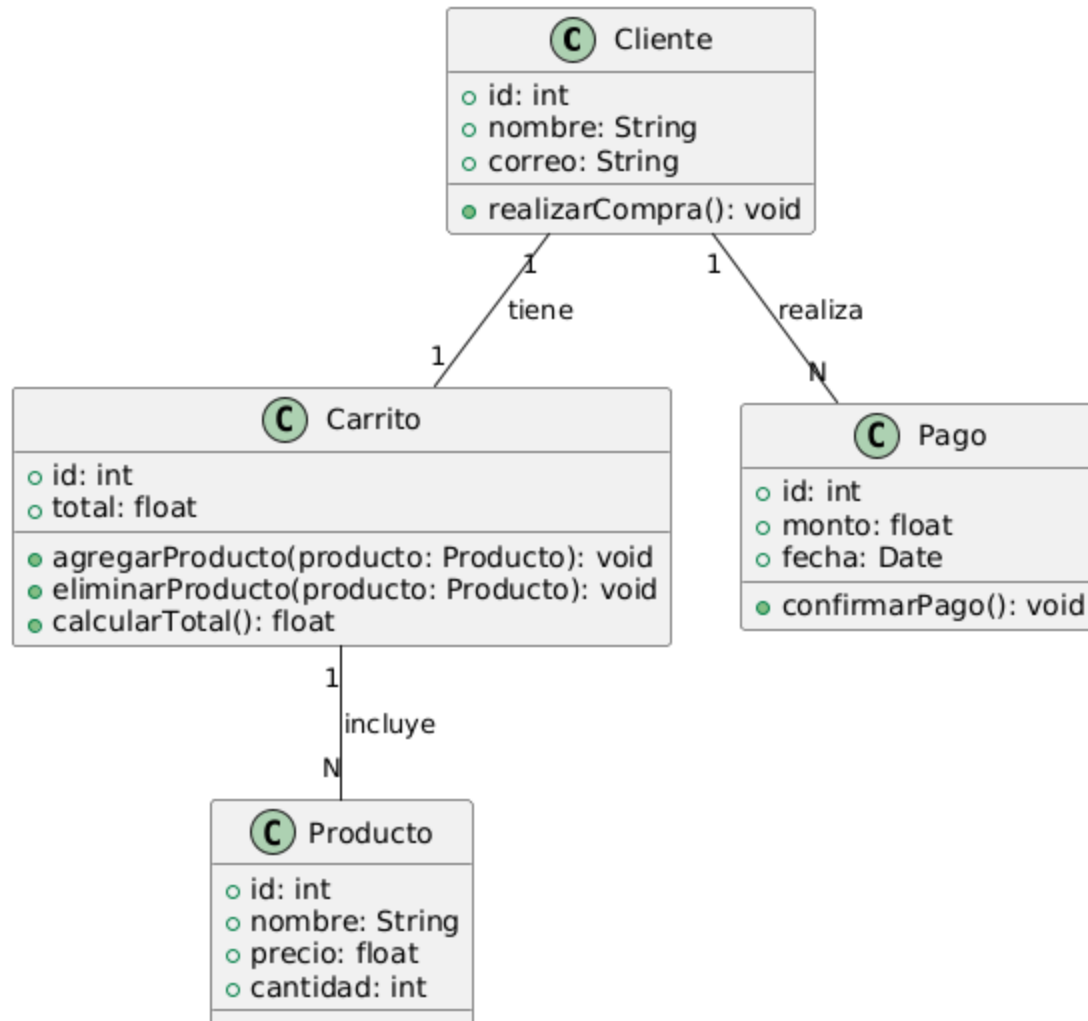


Figura #17. Diagrama de clases para el proceso de compras

Fuente: Elaboración propia.

El diagrama de clases que aborda el proceso de compras muestra las interacciones entre las entidades clave, como el cliente, el carrito de compras y los productos. Esta visualización ayuda a definir claramente cómo se gestionan los datos durante la experiencia de compra, asegurando una implementación efectiva de las funcionalidades requeridas.

5.6.3 Diagrama de clases para la gestión de pagos en línea

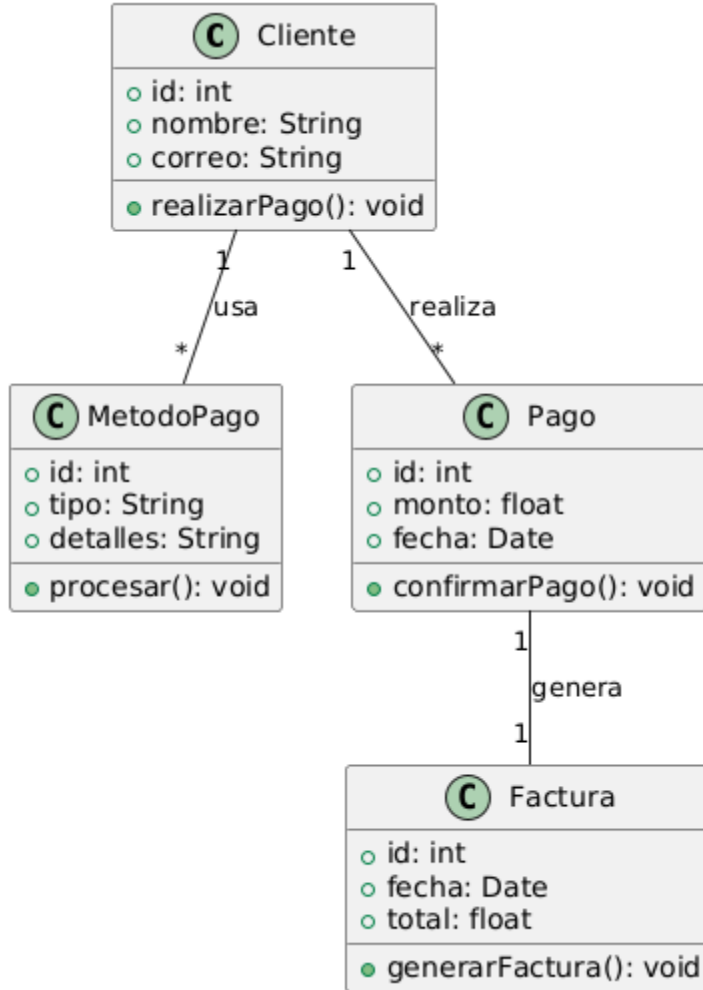


Figura #18. Diagrama de clases para la para la gestión de pagos en línea Fuente: Elaboración propia.

La estructura presentada en el diagrama de clases para la gestión de pagos en línea destaca las clases que manejan la información de transacciones y los métodos de pago. Al detallar las relaciones entre estas clases, se facilita la implementación de un sistema seguro y eficiente, garantizando una experiencia de usuario fluida en el proceso de pago.

5.6.4 Diagrama de clases para agregar productos

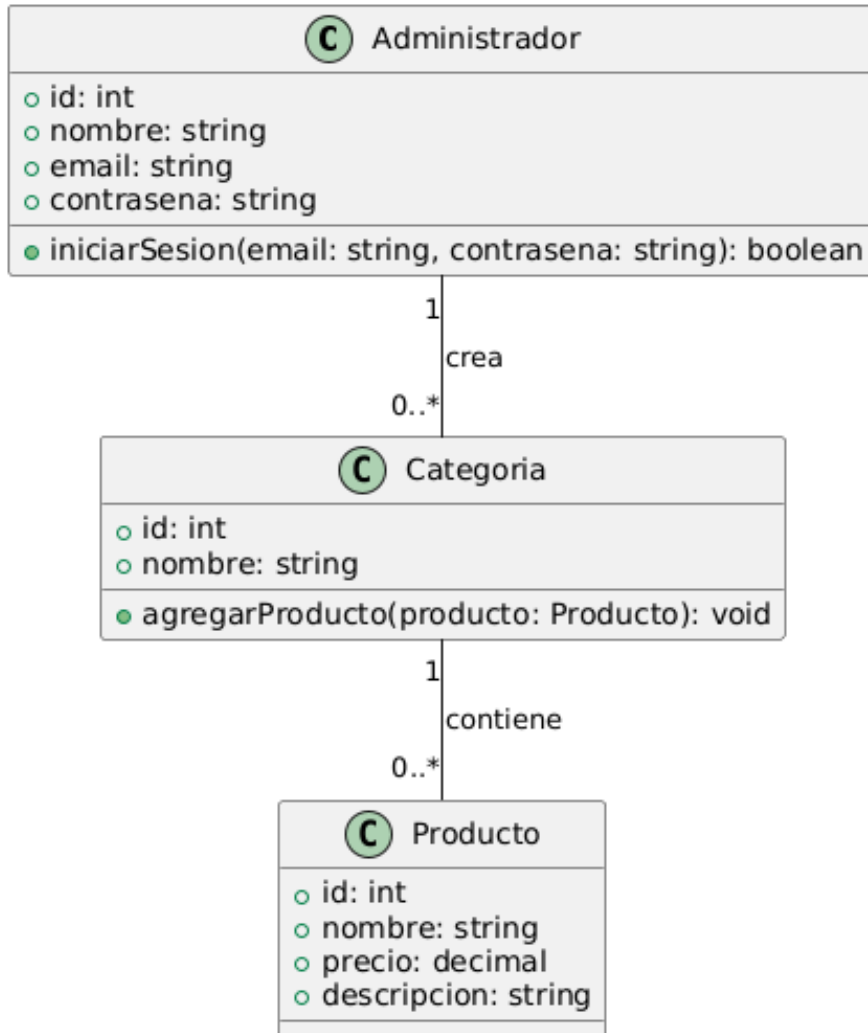


Figura #19. Diagrama de clases para la para la agregar productos

Fuente: Elaboración propia.

El diagrama de clases para agregar productos proporciona una visión clara de las entidades que intervienen en la incorporación de nuevos productos al sistema. Al ilustrar las relaciones entre las clases, este diagrama ayuda a los desarrolladores a entender los requisitos y asegurarse de que todas las funcionalidades necesarias estén adecuadamente implementadas.

5.6.5 Diagrama de clases para la administración del sistema

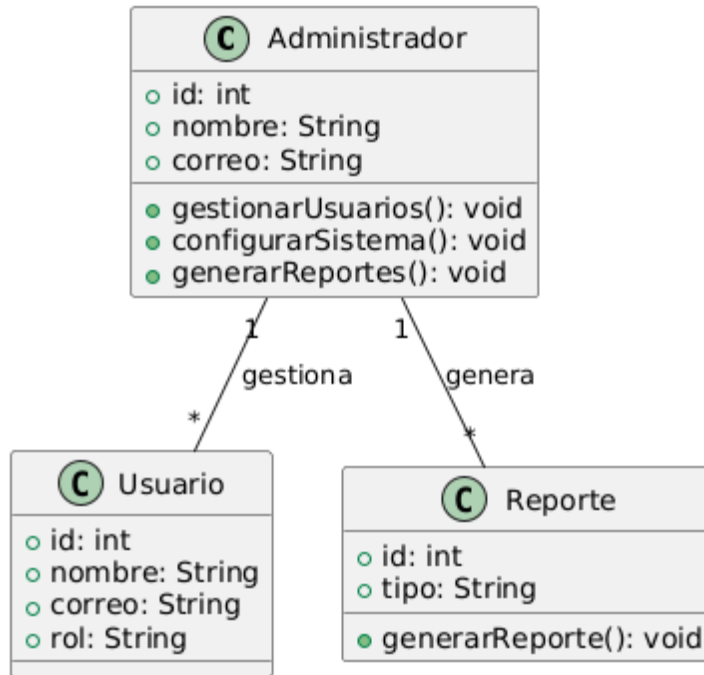


Figura #20. Diagrama de clases para la administración del sistema

Fuente: Elaboración propia.

La organización de las clases en el contexto de la administración del sistema permite visualizar cómo se gestionan las configuraciones y las funcionalidades administrativas. Esta representación es esencial para garantizar que todas las operaciones administrativas estén bien definidas, lo que contribuye a un mejor control y gestión del sistema en su conjunto.

5.7 Diagrama de entidad relación

Es una representación gráfica que ilustra las entidades de un sistema y las relaciones entre ellas. Este tipo de diagrama es fundamental en el diseño de bases de datos, ya que ayuda a visualizar cómo se organizan y conectan los datos en un sistema.

5.7.1 Diagrama de entidad relación para la gestión de inventarios

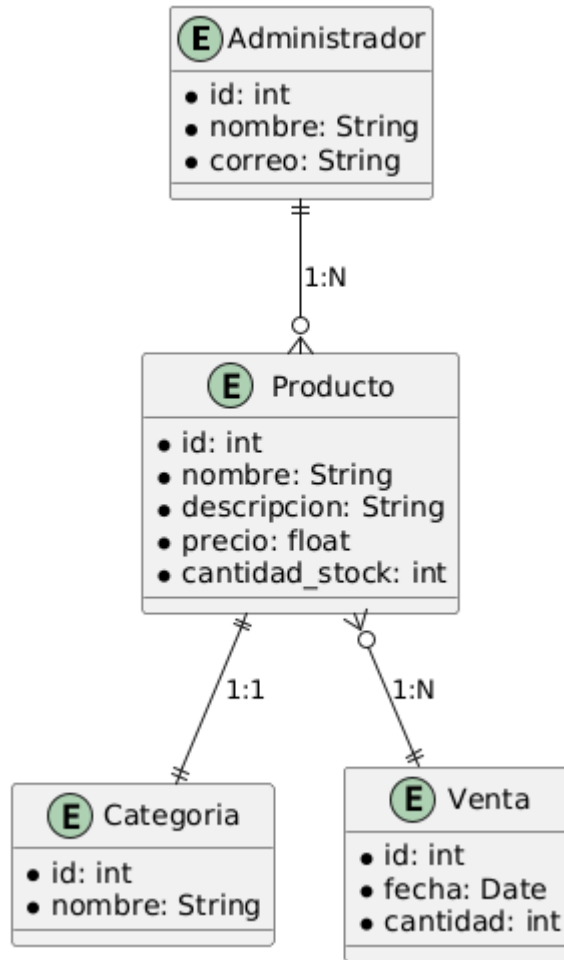


Figura #21. Diagrama de entidad relación para la gestión de inventarios.

Fuente: Elaboración propia.

La claridad en la relación entre las entidades en el diagrama de entidad-relación para la gestión de inventarios es crucial para el correcto manejo de los datos. Este diagrama asegura que se comprendan las dependencias entre los productos, sus categorías y el stock disponible, lo que es fundamental para una gestión eficiente del inventario.

5.7.2 Diagrama de entidad relación para el proceso de compras

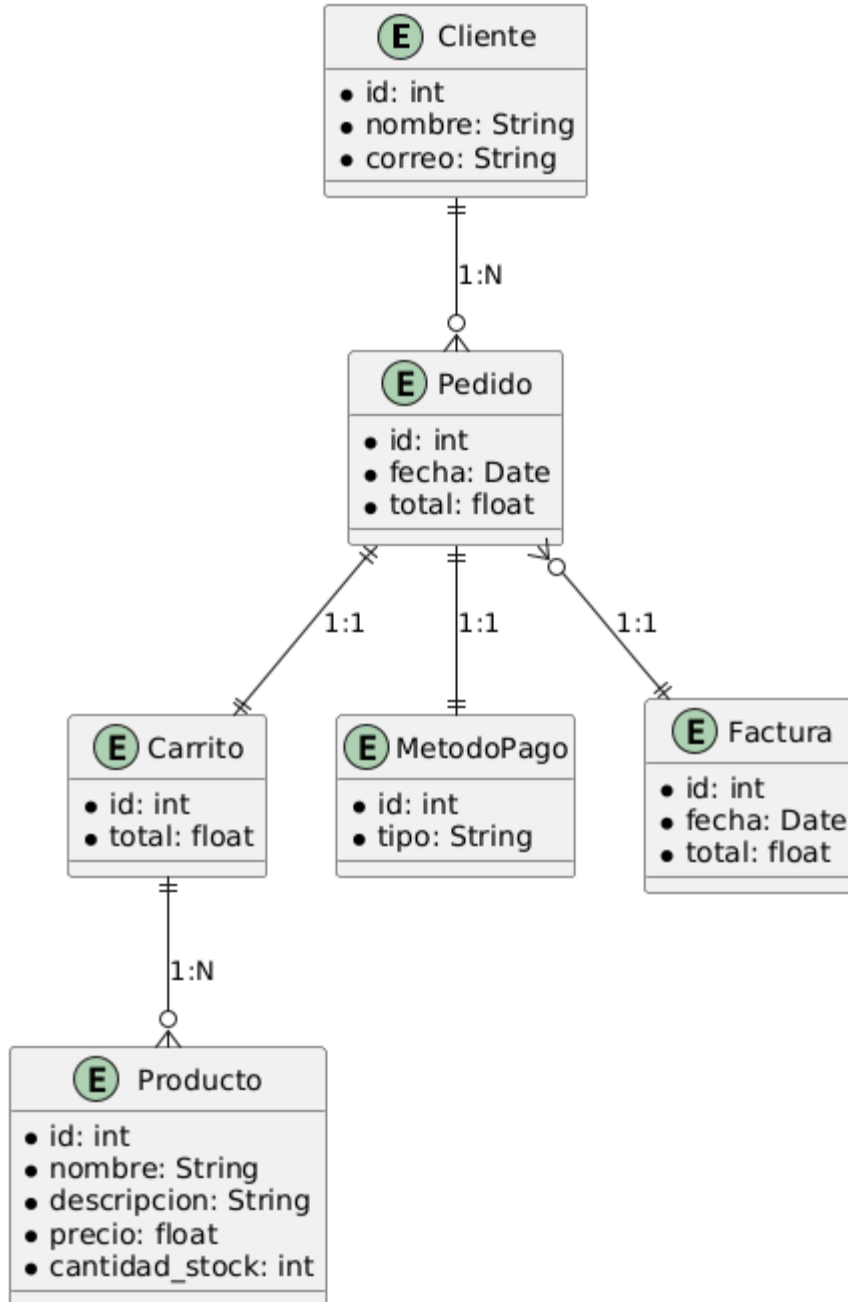


Figura #22. Diagrama de entidad relación para la para el proceso de compras

Fuente: Elaboración propia.

Este diagrama que se centra en el proceso de compras proporciona una visualización detallada de cómo se interrelacionan las entidades involucradas. Esta estructura permite asegurar que se mantengan los datos relevantes a lo largo del proceso de compra, mejorando la integridad y eficiencia del sistema.

5.7.3 Diagrama de entidad relación para la gestión de pagos en línea

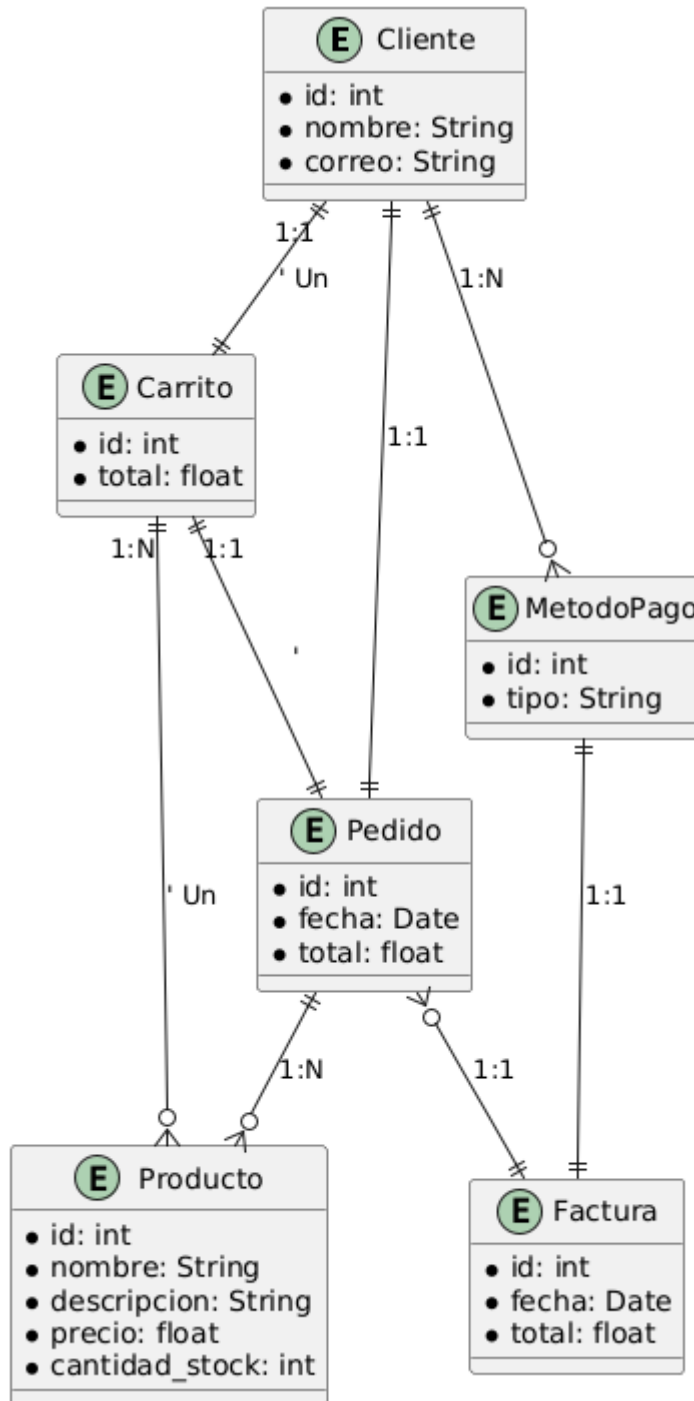


Figura #23. Diagrama de entidad relación para la para la gestión de pagos en línea Fuente: Elaboración propia.

En la representación de la gestión de pagos en línea, el diagrama de entidad-relación ilustra claramente las entidades que manejan las transacciones y su vinculación con el método de pago.

5.7.4 Diagrama de entidad relación para agregar productos

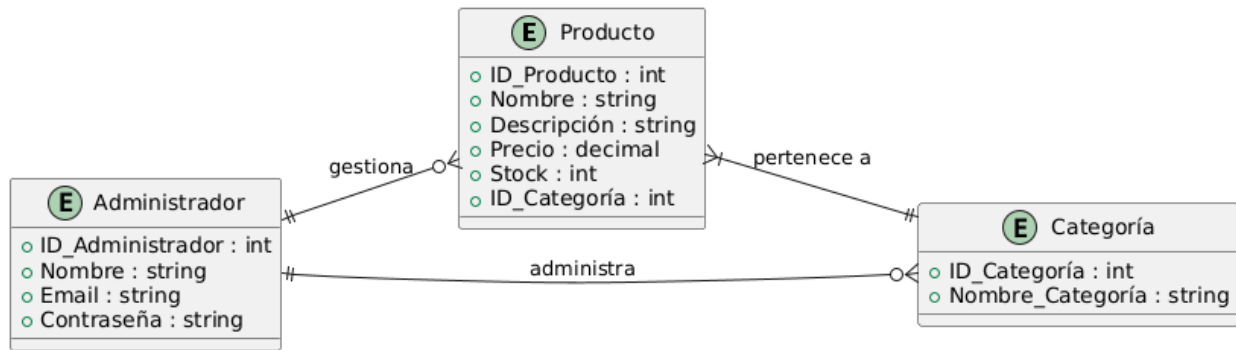


Figura #24. Diagrama de entidad relación para la para la agregar productos

Fuente: Elaboración propia.

El diagrama de entidad-relación para agregar productos establece de manera efectiva las conexiones entre las distintas entidades que intervienen en este proceso.

5.7.5 Diagrama de entidad relación para la administración del sistema

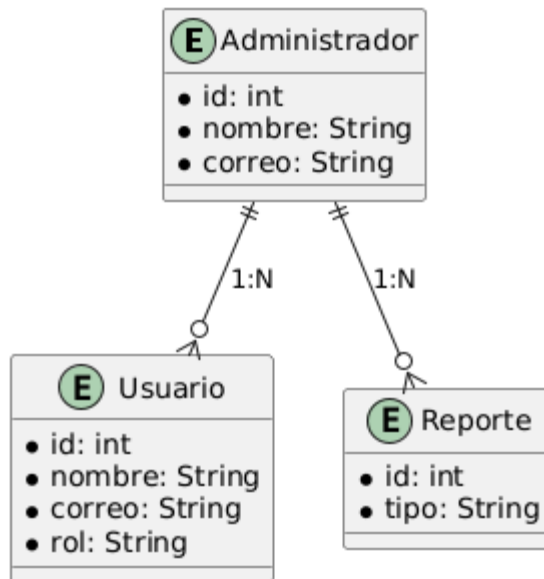


Figura #25. Diagrama de entidad relación para la administración del sistema

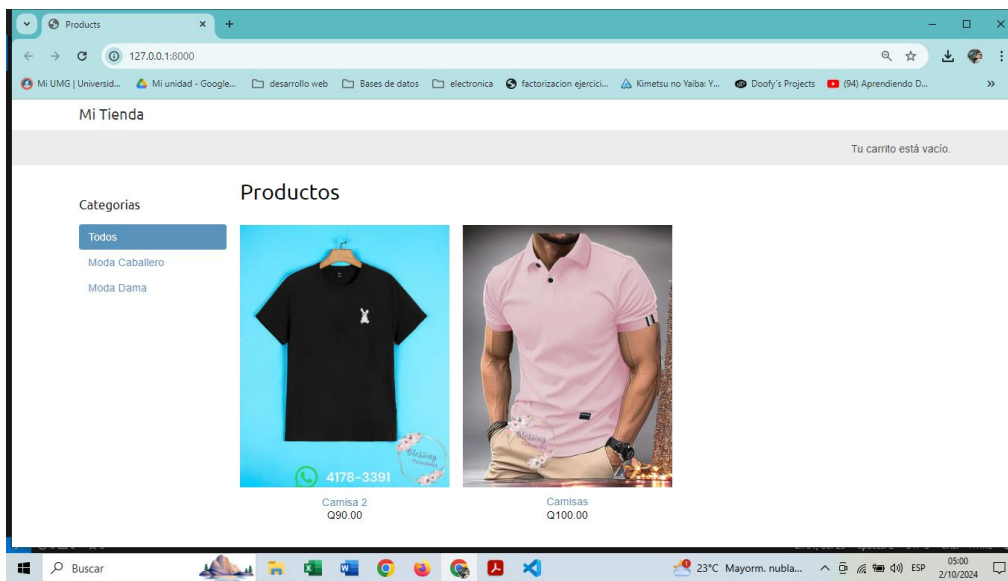
Fuente: Elaboración propia.

La estructura del diagrama de entidad-relación para la administración del sistema ofrece una visión integral de cómo se gestionan las distintas configuraciones y funcionalidades. Esta representación no solo facilita el diseño del sistema, sino que también asegura que las relaciones entre las entidades se manejen de manera efectiva, lo que es fundamental para un sistema administrativo robusto.

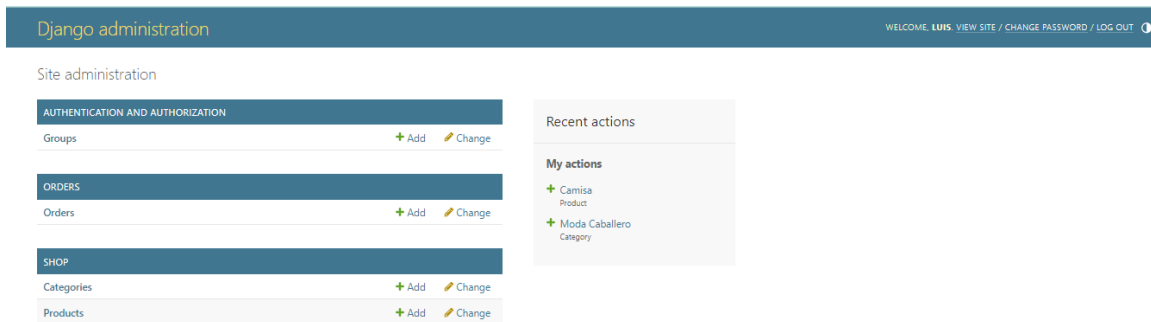
CAPÍTULO IV – ANÁLISIS DE LA APLICACIÓN O SOLUCIÓN

Se detallan los diversos aspectos de la aplicación propuesta, centrándose en la interfaz de usuario y las pruebas. En esta sección, se analiza cómo la interfaz de usuario ha sido diseñada para proporcionar una experiencia intuitiva y accesible, considerando la usabilidad y la estética. Se describen los elementos visuales, la navegación y la interacción del usuario con el sistema. Además, se aborda la importancia de las pruebas, describiendo los tipos de pruebas que se llevarán a cabo, como pruebas funcionales y de usabilidad para asegurar que la aplicación cumpla con los requisitos establecidos.

6.1 Interfaz de usuario

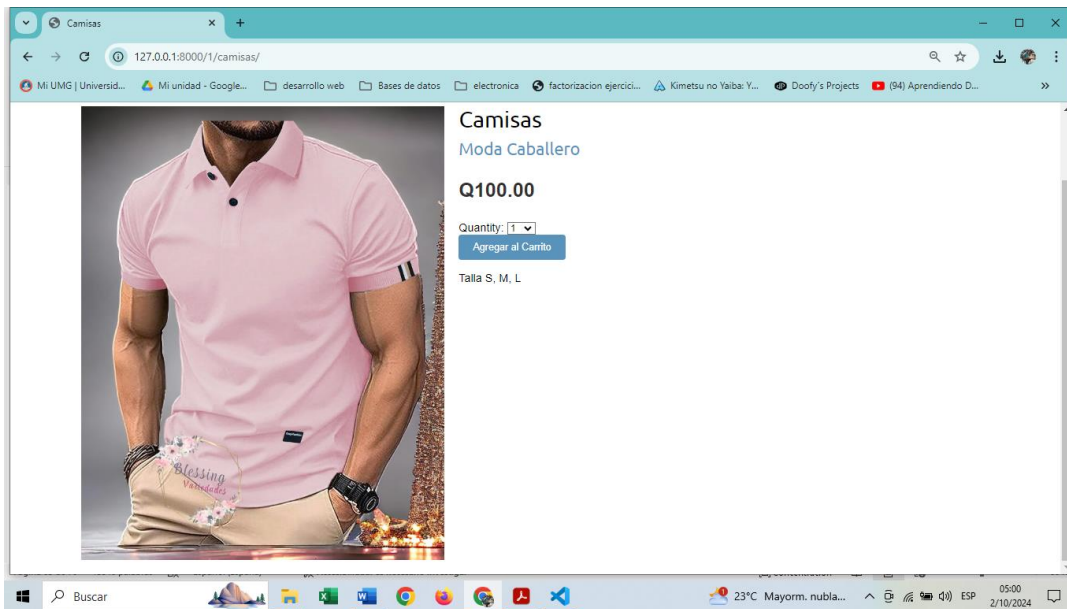


En la presente imagen se puede observar la interfaz de usuario

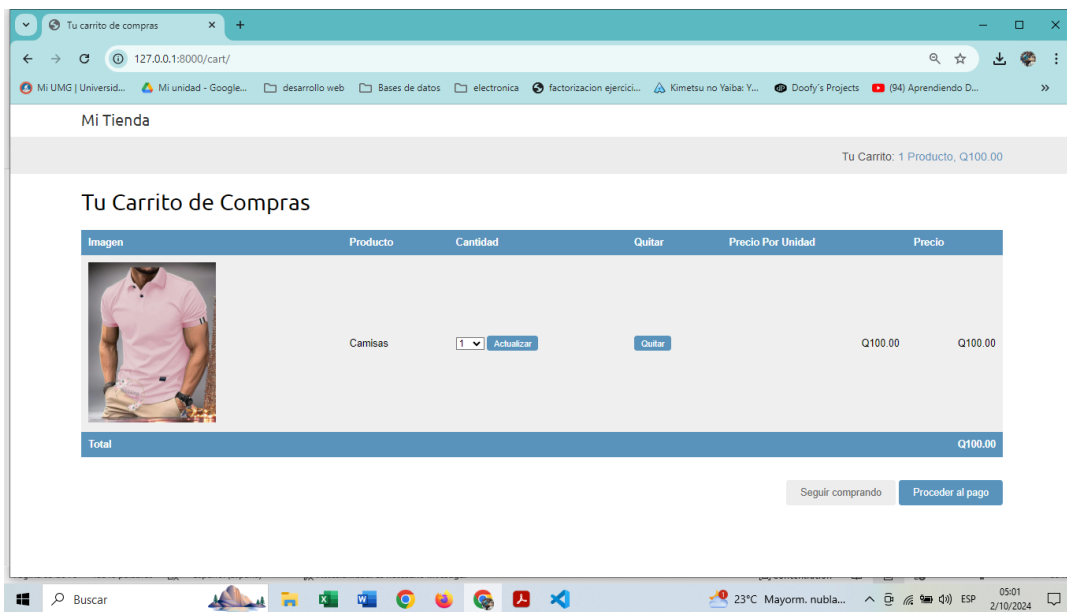


El panel de administración de Django ofrece una interfaz intuitiva para gestionar los modelos definidos en la aplicación, en este caso, los modelos Category y Product. Desde esta interfaz, los administradores pueden añadir, editar y eliminar categorías y productos de manera eficiente.

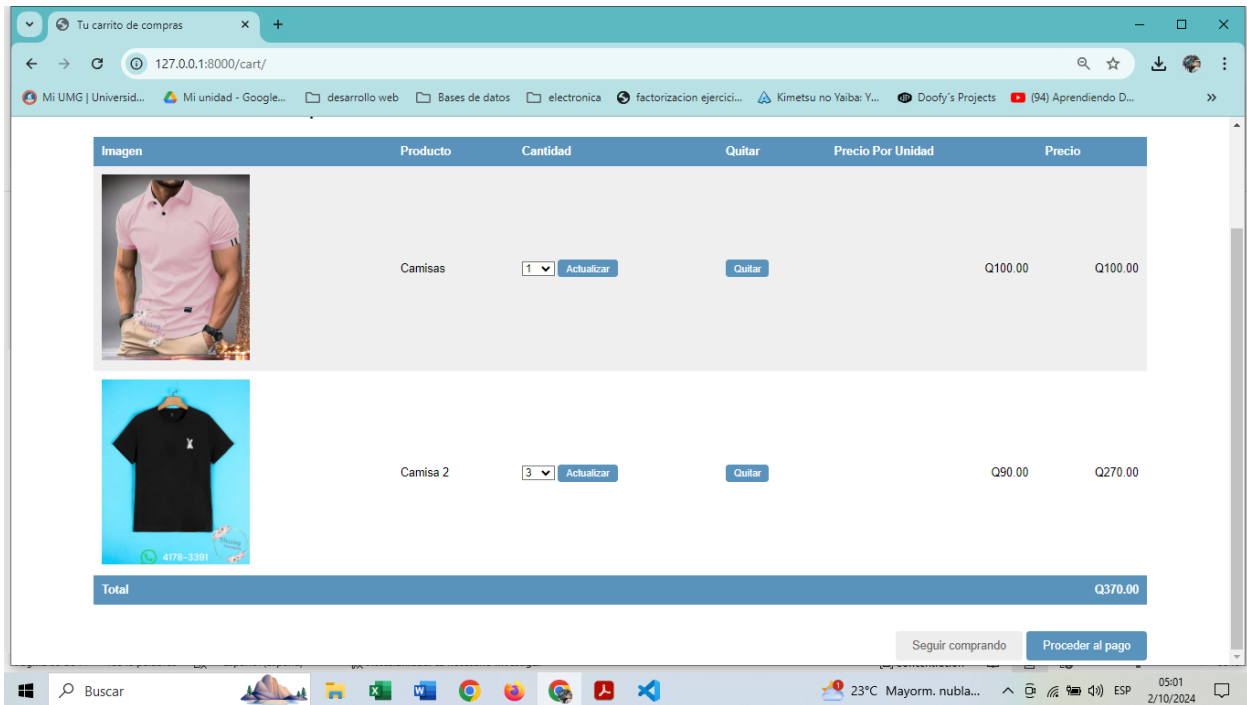
6.2 Pruebas



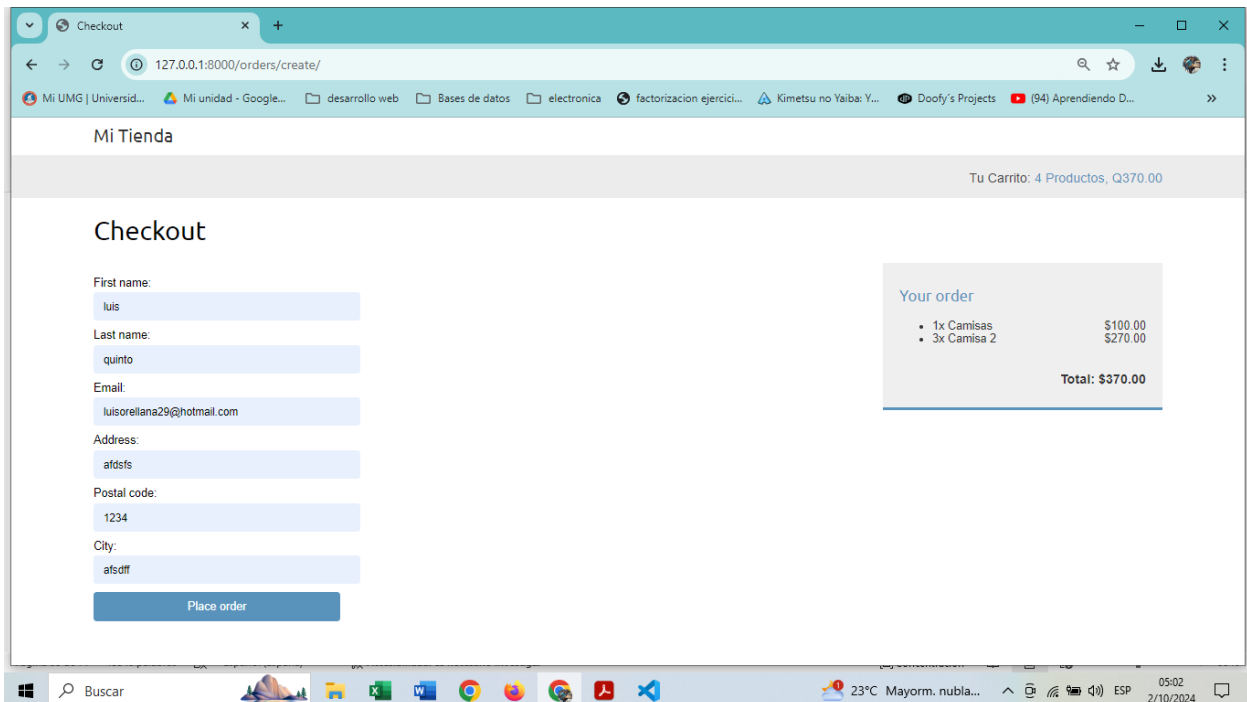
Esta es la vista de detalles donde encontramos el producto en una imagen mas amplia y se puede agregar al carrito



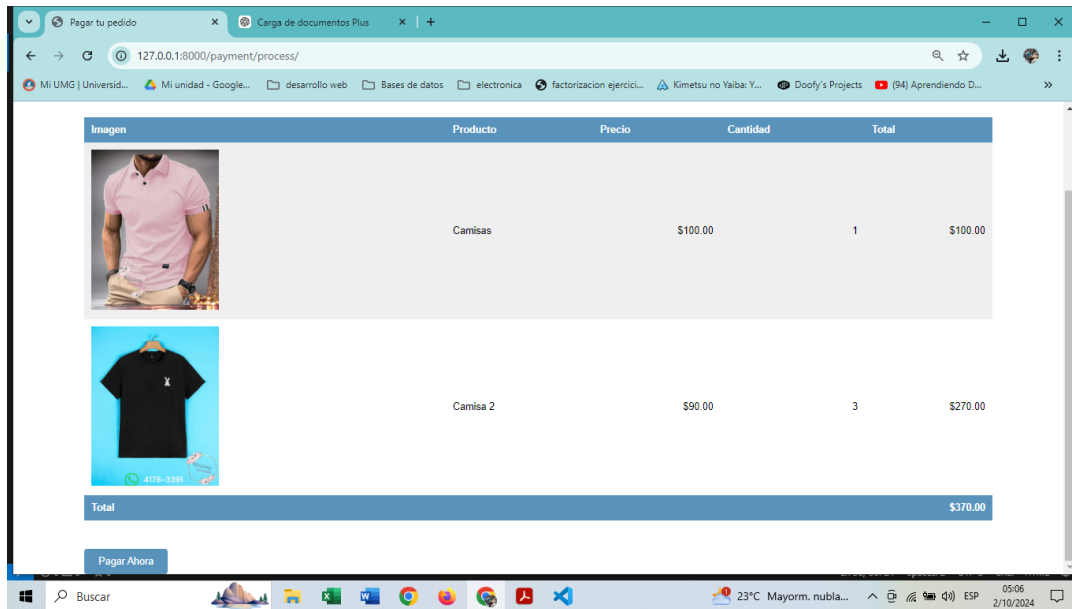
Agregando un producto dentro del carrito



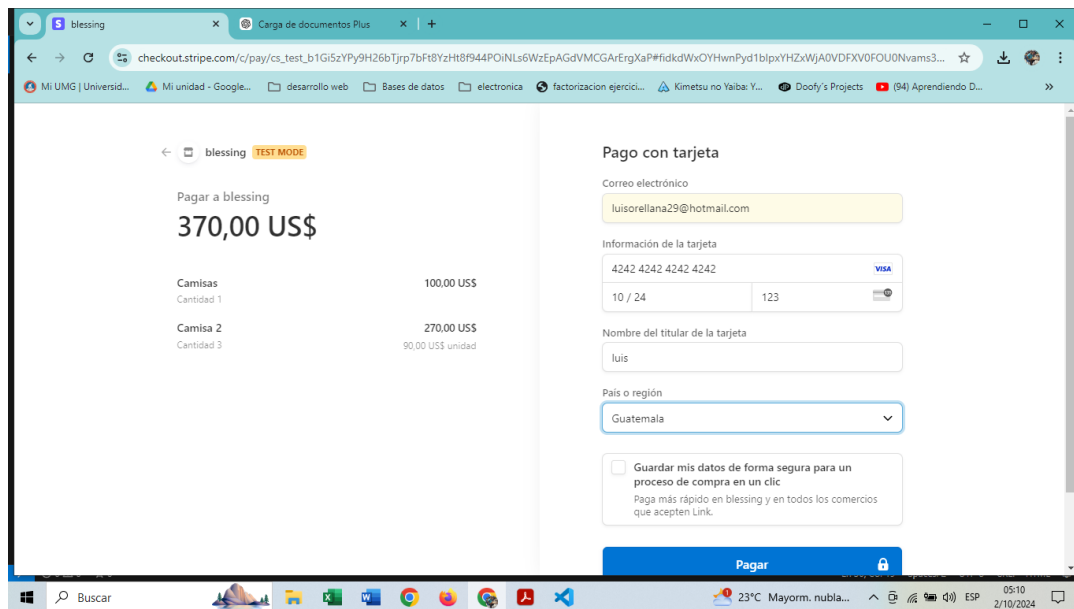
Se agrego otro producto y se aumentó la cantidad



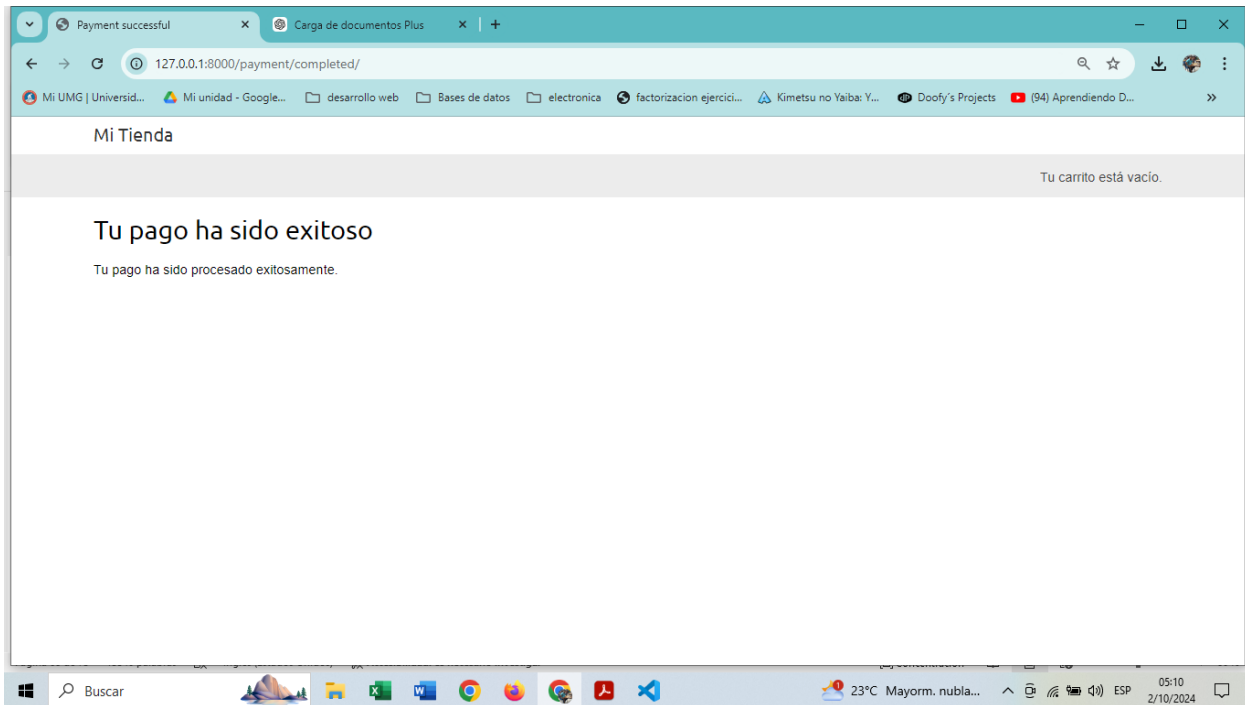
Vista del checkout donde se toma la información del usuario para la orden



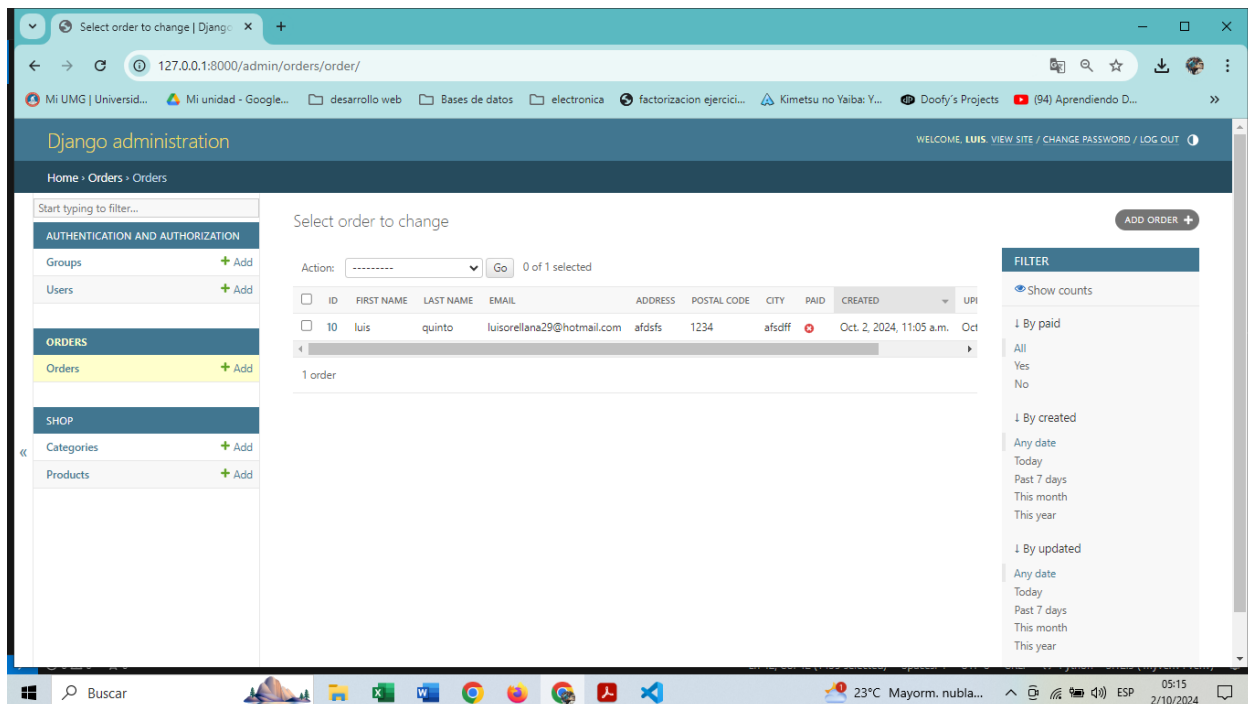
Vista de resumen de producto y acceder al pago



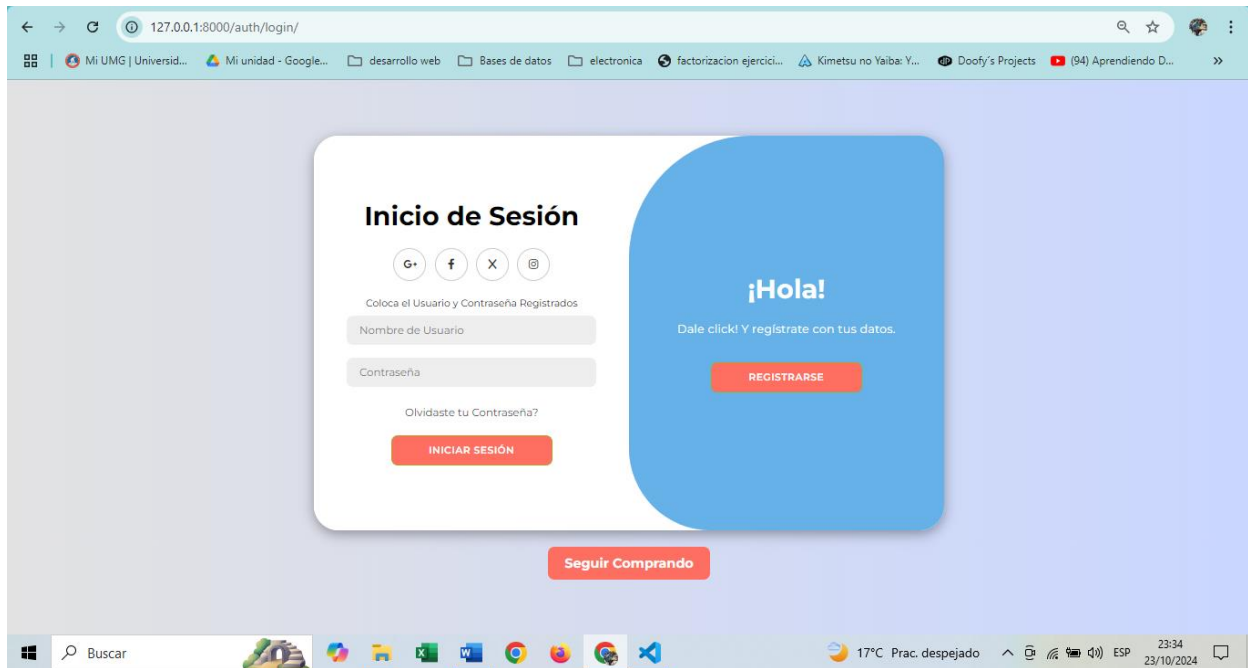
Vista de la pasarela de pago donde aparecen los producto y se ingresan los datos para cancelar el pedido



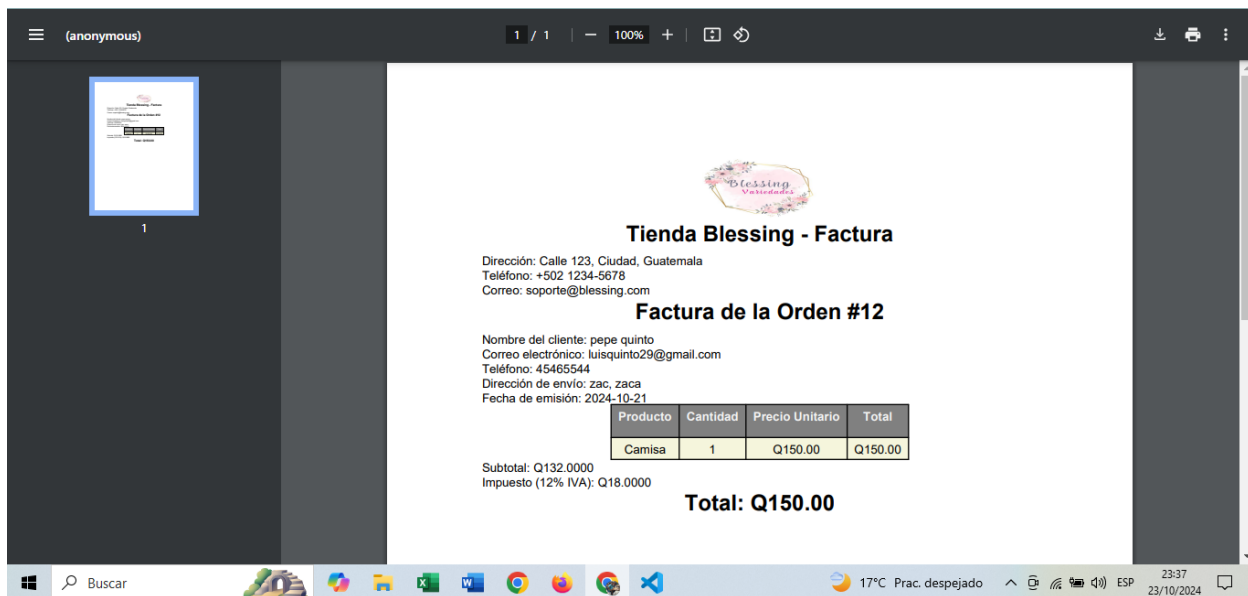
Vista de verificación del proceso de la transacción donde se confirma si se completo con éxito el pago del producto.



Vista de ordenes desde el administrador de django donde aparece los datos del usuario y estado de su compra



La imagen muestra una interfaz gráfica de la página de inicio de sesión de un sistema web. En el lado izquierdo, se encuentra el formulario de inicio de sesión con campos para ingresar el nombre de usuario y contraseña, así como opciones de autenticación mediante redes sociales como Google, Facebook, Twitter e Instagram.



La imagen muestra una factura generada por la tienda en línea Blessing. En la parte superior, se incluye la información de la tienda: dirección, teléfono, y correo electrónico. El título "Factura de la Orden #12" indica que esta factura corresponde a una orden específica.

6.3 Código

Conexion a base de datos

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'tienda2',
        'USER': 'root',
        'PASSWORD': 'Zacapaneco1',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

Esta configuración permite que un proyecto Django se conecte a una base de datos MySQL mediante el uso del motor de MySQL (`django.db.backends.mysql`). Los datos de conexión, como el nombre de la base de datos (`tienda2`), el usuario (`root`), la contraseña (`Zacapaneco1`), el host (`localhost`) y el puerto (`3306`), están especificados en el diccionario `DATABASES`. Cuando Django interactúa con la base de datos, como al crear tablas o realizar consultas, utiliza estos parámetros para establecer la conexión con MySQL. Es importante que las credenciales sean correctas y que el motor de MySQL esté en funcionamiento para que Django pueda operar sin problemas.

Aplicaciones instaladas

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'shop.apps.ShopConfig',
    'cart.apps.CartConfig',
    'orders.apps.OrdersConfig',
    'payment.apps.PaymentConfig',
    'authapp.apps.AuthappConfig',
]
```

El fragmento de código proporcionado representa la configuración de aplicaciones instaladas en un proyecto Django, especificando los componentes que se utilizarán en la aplicación web. En la lista `INSTALLED_APPS`, se incluyen las aplicaciones predeterminadas de Django, como `django.contrib.admin` para la interfaz de administración, `django.contrib.auth` para la gestión de usuarios y autenticación, y `django.contrib.sessions` para el manejo de sesiones de usuario.

Además de las aplicaciones predeterminadas, se incluyen varias aplicaciones personalizadas que forman parte del proyecto, tales como shop, cart, orders, payment y authapp. Cada una de estas aplicaciones tiene un propósito específico en el sistema:

shop: Maneja la gestión de productos y categorías, permitiendo la visualización y búsqueda de artículos en la tienda.

cart: Se encarga de la gestión del carrito de compras, permitiendo a los usuarios agregar productos y gestionar su selección antes de proceder al pago.

orders: Administra el proceso de pedidos, facilitando la creación y seguimiento de órdenes realizadas por los usuarios.

payment: Gestiona las transacciones de pago, integrando métodos de pago para permitir a los usuarios completar sus compras de manera segura.

authapp: Se ocupa de la autenticación y gestión de usuarios, permitiendo funcionalidades como el registro, inicio de sesión y gestión de cuentas.

Esta configuración modular permite una fácil escalabilidad y mantenimiento del proyecto, ya que cada aplicación se puede desarrollar y probar de forma independiente, contribuyendo a un sistema más organizado y eficiente. En resumen, la lista `INSTALLED_APPS` es fundamental para la arquitectura del proyecto, ya que define los componentes clave que interactuarán para ofrecer una experiencia completa y funcional al usuario.

Código plantilla base

```
{% load static %}
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}Mi Tienda{% endblock %}</title>
  <link href="{% static 'css/base.css' %}" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css" />
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
  </style>
</head>

<body>
  <div class="container">
    <div class="row">
      <div class="col-md-12">
        <div class="card">
          <div class="card-body">
            <div class="text-center">
              <h1>Mi Tienda</h1>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```
#header {
  background-color: #007bff;
  padding: 10px 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

#header > div {
  display: flex;
  align-items: center;
  justify-content: flex-end;
  flex-grow: 1;
}

.logo {
  font-size: 24px;
  color: white !important;
  text-decoration: none;
  transition: color 0.3s;
}

.logo:hover {
  color: #ffcc00;
}

.btn-login {
  display: inline-block;
  padding: 10px;
  font-size: 20px;
  color: white;
  background-color: #007bff;
  text-decoration: none;
  border-radius: 5px;
  transition: background-color 0.3s, transform 0.3s;
  margin-right: 5px;
}

.btn-login:hover {
  background-color: #0056b3;
  transform: scale(1.05);
}

#subheader {
  padding: 10px 20px;
```

```

        background-color: #f8f9fa;
    }

    .cart-icon {
        position: relative;
        display: inline-block;
        color: white;
        transition: color 0.3s;
        margin-left: 10px; /* Espacio adecuado entre el botón y el carrito */
    }

    .cart-count {
        position: absolute;
        top: -10px; /* Mueve el contador hacia arriba */
        right: 155px; /* Ajusta el espacio entre el ícono del carrito y el contador */
        background-color: red;
        color: white;
        border-radius: 160%;
        padding: 3px 6px;
        font-size: 10px;
        z-index: 1; /* Asegúrate de que esté en un nivel superior */
    }

    #content {
        padding: 20px;
    }

    .cart-info {
        margin-left: 7px;
    }

    .search-bar {
        padding: 8px;
        border-radius: 12px;
        border: 1px solid #ccc;
        margin-right: 0px;
        width: 300px; /* Ancho aumentado para la barra de búsqueda */
    }

    #search-container {
        flex-grow: 1; /* Permite que el contenedor de búsqueda ocupe más espacio */
        display: flex;
        justify-content: flex-start; /* Cambiado a flex-start para mover la barra a la izquierda
*/
    }

```

```

        .login-container {
            display: flex;
            align-items: center; /* Alinea verticalmente los elementos en el centro */
            justify-content: flex-start; /* Alinea los elementos hacia la izquierda */
            margin-right: -250px; /* Aumenta el margen derecho para más espacio */
            z-index: 2; /* Asegura que los elementos dentro de este contenedor estén por encima
*/
            position: relative; /* Asegura un contexto de posicionamiento */
        }
    </style>
</head>

<body>
    <div id="header">
        <a href="/" class="logo">BLESSING</a>

        <div id="search-container">
            <form method="GET" action="{% url 'shop:product_list' %}">
                <input type="text" name="q" placeholder="Buscar producto..." class="search-
bar">
                <button type="submit" style="border: none; background: transparent; cursor:
pointer;">
                    <i class="fas fa-search" style="font-size: 24px; color: #333;"></i>
                </button>
            </form>
        </div>

        <div class="login-container">
            <a href="{% url 'authapp:login' %}" class="btn-login">
                <i class="fas fa-circle-user"></i>
            </a>
            <a href="{% url 'authapp:login' %}" class="account-link" style="color: white;">Mi
Cuenta</a>
        </div>

        <div class="cart-icon">
            {% with total_items=cart|length %}
            {% if total_items > 0 %}
            <a href="{% url 'cart:cart_detail' %}" style="text-decoration: none; color: inherit;">
                <i class="fas fa-shopping-cart" style="color: white;"></i>
                <span class="cart-count">{{ total_items }}</span>
            </a>
            <a href="{% url 'cart:cart_detail' %}" style="text-decoration: none; color: white;">

```

```

        <span class="cart-info"> Producto{{ total_items|pluralize }}, Q{{
cart.get_total_price }}</span>
    </a>
    {% else %}
    <a href="{% url 'cart:cart_detail' %}" style="text-decoration: none; color: inherit;">
        <i class="fas fa-shopping-cart" style="color: white;"></i>
    </a>
    <span style="color: white; margin-left: 5px;">Carrito vacío</span>
    {% endif %}
    {% endwith %}
</div>
</div>
<div id="subheader">
    <!-- Aquí puedes incluir contenido adicional si es necesario -->
</div>
<div id="content">
    {% block content %}
    {% endblock %}
</div>
</body>

</html>

```

La interfaz de usuario de la tienda en línea "Blessing" se ha diseñado utilizando el framework Django, que facilita la construcción de aplicaciones web modulares y escalables. A continuación, se describe en detalle la estructura del código HTML y CSS que compone el encabezado y la funcionalidad de búsqueda y carrito de compras del sitio web.

Estructura HTML

La plantilla HTML está construida utilizando el motor de plantillas de Django, el cual permite la reutilización de componentes y la inserción dinámica de contenido. El código comienza cargando archivos estáticos y librerías externas necesarias para el estilo y los iconos, mediante la directiva `{% load static %}` de Django y la inclusión de la librería de iconos de Font Awesome. Dentro de la etiqueta `<head>`, se definen el conjunto de caracteres y la configuración responsiva, lo cual garantiza que la página sea accesible y se adapte a diferentes dispositivos.

El bloque `{% block title %}` permite personalizar el título de la página en cada una de las vistas del sitio web, garantizando consistencia y flexibilidad en la presentación del contenido.

En el cuerpo de la página, se define un encabezado (#header) que incluye los elementos principales para la navegación y la interacción del usuario:

Logo de la tienda: Un enlace con el texto "BLESSING" que dirige al usuario a la página principal. Este logo está estilizado mediante CSS para que su tamaño y color varíen al interactuar con él.

Barra de búsqueda: Se ha incorporado un formulario que permite a los usuarios buscar productos específicos en el catálogo. Este formulario envía una consulta GET a la vista `product_list`, utilizando el valor ingresado en el campo de texto como parámetro de búsqueda. El botón de búsqueda, representado por un ícono de lupa, está alineado junto al campo de texto para facilitar la interacción.

Enlace de inicio de sesión y cuenta: Un botón visible para los usuarios que no han iniciado sesión, el cual permite acceder a la página de autenticación (`authapp:login`). Además, se incluye un enlace a "Mi Cuenta", que facilita el acceso a la información personal del usuario una vez autenticado.

Carrito de compras: Este elemento utiliza una combinación de lógica de Django y CSS para mostrar el estado del carrito de compras. Si el carrito contiene productos, se muestra un contador dinámico que indica la cantidad de artículos y el precio total acumulado. Este bloque hace uso del filtro `pluralize` de Django para ajustar el texto en función del número de productos. En caso de que el carrito esté vacío, se muestra un mensaje indicando tal condición.

El código HTML incluye también un bloque de contenido (`{% block content %}`), el cual es dinámico y puede ser sustituido por el contenido específico de cada vista. Este bloque facilita la reutilización de la estructura de la página en distintas secciones del sitio web, como las páginas de detalles de productos, categorías o historial de pedidos.

Código Modelos

```
from django.db import models
from django.urls import reverse

class Category(models.Model):
    name = models.CharField(max_length=200)
    slug = models.SlugField(max_length=200,
                           unique=True)

    class Meta:
        ordering = ['name']
        indexes = [
            models.Index(fields=['name']),
        ]
```

```

        verbose_name = 'category'
        verbose_name_plural = 'categories'

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('shop:product_list_by_category',
                        args=[self.slug])

class Product(models.Model):
    category = models.ForeignKey(Category,
                                related_name='products',
                                on_delete=models.CASCADE)
    name = models.CharField(max_length=200)
    slug = models.SlugField(max_length=200)
    image = models.ImageField(upload_to='products/%Y/%m/%d',
                              blank=True)
    description = models.TextField(blank=True)
    price = models.DecimalField(max_digits=10,
                                decimal_places=2)
    available = models.BooleanField(default=True)
    created = models.DateTimeField(auto_now_add=True)
    updated = models.DateTimeField(auto_now=True)

    class Meta:
        ordering = ['name']
        indexes = [
            models.Index(fields=['id', 'slug']),
            models.Index(fields=['name']),
            models.Index(fields=['-created']),
        ]

    def __str__(self):
        return self.name

    def get_absolute_url(self):
        return reverse('shop:product_detail',
                        args=[self.id, self.slug])

```

Representa dos modelos de Django para una tienda en línea: **Category** y **Product**, ambos fundamentales para la estructura de datos de productos en un catálogo.

Category: Este modelo define las categorías de productos. Cada categoría tiene un nombre y un slug que actúa como una versión amigable de la URL. Además, establece ordenación por nombre (`ordering = ['name']`) y añade un índice para mejorar el rendimiento de las consultas basadas en el nombre. El método `get_absolute_url()` genera la URL que apunta a la lista de productos filtrada por categoría usando el *slug*.

Product: Este modelo representa los productos que pertenecen a una categoría específica. Cada producto tiene un nombre, slug, imagen, descripción, precio, estado de disponibilidad, y marcas de tiempo para su creación y última actualización. Se establece una relación de clave foránea con el modelo `category` mediante *foreignkey*, lo que permite acceder a los productos relacionados con una categoría. El modelo también establece índices para mejorar la eficiencia en la búsqueda y ordenación de los productos según distintos campos como el nombre, slug, y fecha de creación. Al igual que en `category`, el método `get_absolute_url()` devuelve la URL del detalle del producto.

Código vistas

```
from django.shortcuts import render, get_object_or_404
from cart.forms import CartAddProductForm
from .models import Category, Product

def product_list(request, category_slug=None):
    category = None
    categories = Category.objects.all()
    products = Product.objects.filter(available=True)

    # Lógica de búsqueda
    query = request.GET.get('q') # Obtén el término de búsqueda
    if query: # Si hay un término de búsqueda
        products = products.filter(name__icontains=query) # Filtra los productos por nombre

    if category_slug:
        category = get_object_or_404(Category, slug=category_slug)
        products = products.filter(category=category)

    return render(request,
                  'shop/product/list.html',
                  {'category': category,
                   'categories': categories,
                   'products': products,
                   'query': query}) # Pasa la consulta al contexto

def product_detail(request, id, slug):
```



```

product = get_object_or_404(Product,
                             id=id,
                             slug=slug,
                             available=True)
cart_product_form = CartAddProductForm()
return render(request,
              'shop/product/detail.html',
              {'product': product,
              'cart_product_form': cart_product_form})

```

El archivo *views.py* implementa la lógica de presentación para la gestión de productos en una tienda en línea utilizando el *framework Django*. En este contexto, se definen dos funciones principales: *product_list* y *product_detail*. La función *product_list* se encarga de recuperar y mostrar una lista de productos disponibles. Permite a los usuarios filtrar los productos por categorías y realizar búsquedas a través de un campo de búsqueda, proporcionando así una experiencia de navegación dinámica e intuitiva.

Al acceder a esta vista, se obtiene una lista de todas las categorías y productos que están marcados como disponibles. Si se proporciona un término de búsqueda, la lista de productos se filtra para mostrar únicamente aquellos que coinciden con el nombre ingresado. Además, si se incluye un *category_slug*, la función filtrará aún más los productos para mostrar solo aquellos que pertenecen a la categoría seleccionada.

Por otro lado, la función *product_detail* se ocupa de mostrar la información detallada de un producto específico. Esta vista se activa mediante la identificación del producto a través de su *id* y *slug*. Al acceder a esta vista, se recupera el producto correspondiente, junto con un formulario para agregarlo al carrito de compras. Esto permite a los usuarios interactuar con el producto y facilitar su adquisición.

Conclusiones

En el desarrollo del sistema web para la tienda "Blessing", se logró cumplir con los objetivos planteados de optimizar la gestión de inventarios y los pagos en línea, proporcionando una plataforma intuitiva y segura tanto para los administradores como para los usuarios finales. El uso de tecnologías como Django y MySQL permitió la implementación de un sistema robusto que ofrece una experiencia de usuario fluida y eficiente, capaz de gestionar de manera efectiva las transacciones y los niveles de stock en tiempo real. Además, la integración de pasarelas de pago seguras como Stripe y Cybersource garantiza la confiabilidad en el proceso de compra.

A lo largo del proyecto, se aplicó la metodología ágil Scrum, lo que permitió una entrega incremental de funcionalidades, facilitando la iteración y mejora continua basada en retroalimentación constante. La gestión de inventarios en tiempo real demostró ser un componente crucial para el éxito del sistema, ya que evita la falta de productos y mantiene la información actualizada para los usuarios. Por otro lado, la arquitectura del sistema fue diseñada para ser escalable y adaptable a futuros requerimientos o expansiones de la tienda, lo que asegura su sostenibilidad a largo plazo.

El análisis de viabilidad operativa y financiera indicó que el proyecto es factible dentro de los límites establecidos, lo cual garantiza su implementación y mantenimiento en el tiempo. Sin embargo, se identificaron áreas de mejora, principalmente en la usabilidad del sistema para garantizar una curva de aprendizaje más rápida para los administradores, así como la necesidad de una mayor automatización en las alertas de stock bajo.

Recomendaciones

Se recomienda implementar un sistema más avanzado de alertas automáticas que no solo avisen sobre el stock bajo, sino también que generen reportes periódicos para mejorar la toma de decisiones respecto al inventario.

Dado que el sistema ha sido diseñado para ser escalable, se sugiere considerar la expansión de la tienda en línea para incluir nuevas funcionalidades como la gestión de promociones y descuentos, así como la implementación de un sistema de lealtad para los clientes recurrentes.

Si bien las pasarelas de pago actuales son seguras, se recomienda explorar otras opciones como métodos de pago en cuotas o integraciones con wallets digitales locales, lo que ampliaría las opciones para los usuarios y mejoraría su experiencia.

Es esencial ofrecer una capacitación continua al personal encargado de administrar el sistema, con especial énfasis en la gestión de inventarios y el manejo del panel de administración, para asegurar que puedan aprovechar todas las funcionalidades del sistema.

Se recomienda establecer un plan de mantenimiento periódico para monitorear el rendimiento del sistema, identificar posibles vulnerabilidades, y realizar actualizaciones de seguridad que garanticen la estabilidad y seguridad de la plataforma en el tiempo.

GLOSARIO

Análisis: Proceso de examinar en detalle los componentes de algo para comprender su estructura, función y significado. En el contexto de investigación, se refiere a la descomposición de información compleja en partes más manejables para su estudio.

Comunidad: Grupo de personas que comparten intereses, objetivos, valores o características comunes y que interactúan entre sí. Puede referirse a una comunidad geográfica, una comunidad virtual en línea, o cualquier grupo unido por un interés común.

CSS: Siglas de "*Cascading Style Sheets*" (Hojas de Estilo en Cascada), es un lenguaje de diseño utilizado para describir la presentación visual de un documento escrito en *HTML o XML*. *CSS* controla la apariencia de los elementos en una página web, incluyendo colores, fuentes y disposición.

Desarrollo web: Proceso de creación y mantenimiento de sitios y aplicaciones web. Incluye aspectos como el diseño web, el desarrollo de contenido, la codificación del lado del cliente y del servidor, y la configuración de seguridad en redes.

Django: Framework de desarrollo web de alto nivel, escrito en *Python*, que fomenta el desarrollo rápido y el diseño limpio y pragmático. Es conocido por su "baterías incluidas", es decir, su amplio conjunto de herramientas y funcionalidades listas para usar.

Escalabilidad: Capacidad de un sistema, red o proceso para manejar una creciente cantidad de trabajo, o su potencial para ser ampliado para acomodar ese crecimiento. En el contexto de software, se refiere a la capacidad de una aplicación de mantener su rendimiento y eficiencia a medida que aumenta la carga de trabajo.

Experimental: Relativo a la realización de pruebas o experimentos para investigar y comprobar hipótesis, identificar obstáculos y evaluar la eficacia de soluciones propuestas. Se usa comúnmente en ciencias para descubrir nuevos conocimientos y validar teorías.

FIFO: Acrónimo de "First In, First Out" (Primero en Entrar, Primero en Salir), es un método de valoración de inventarios donde los primeros artículos en ser adquiridos o producidos son los primeros en ser utilizados o vendidos. Es usado para mantener la frescura de los productos y una adecuada gestión de stock.

HTML: Siglas de "*HyperText Markup Language*" (Lenguaje de Marcado de Hipertexto), es el estándar para crear páginas web y aplicaciones web. *HTML* describe la estructura de una página web utilizando una serie de elementos y atributos.

Inventario: Conjunto de bienes y productos, tangibles e intangibles, que posee una empresa en un momento determinado.

Investigación aplicada: Tipo de investigación que busca resolver problemas prácticos con soluciones concretas.

JavaScript: Lenguaje de programación utilizado principalmente para crear y controlar contenido dinámico en sitios web. Permite agregar interactividad a las páginas web, como formularios dinámicos, animaciones y otros elementos interactivos.

Marco teórico: Sección de un trabajo de investigación que presenta y discute las teorías, modelos y otros trabajos previos relevantes al tema en estudio. Proporciona el contexto conceptual y antecedentes necesarios para entender la investigación y su importancia.

Metodológico: Relativo a la metodología, es decir, al conjunto de métodos, técnicas y procedimientos utilizados para abordar un problema o realizar una investigación. Un enfoque metodológico adecuado asegura la validez y confiabilidad de los resultados obtenidos.

MySQL: Sistema de gestión de bases de datos relacional, basado en *SQL* (*Structured Query Language*). Es ampliamente utilizado en aplicaciones web y por empresas para gestionar y organizar grandes volúmenes de datos. *MySQL* permite almacenar, modificar y administrar datos de manera eficiente y segura.

Pago en línea: Proceso de realizar transacciones financieras a través de Internet. Incluye la compra de bienes y servicios mediante tarjetas de crédito, débito, transferencias bancarias y otros métodos electrónicos. Es una parte esencial del comercio electrónico.

Participativo: Que implica la participación activa de varias partes interesadas. En un contexto organizacional o comunitario, un enfoque participativo busca involucrar a todos los actores relevantes en el proceso de toma de decisiones para asegurar que se consideren diferentes perspectivas y necesidades.

PEPS: Siglas de "Primeras en Entrar, Primeras en Salir", es un método de valoración de inventarios donde los primeros productos en ser adquiridos o producidos son los primeros en ser utilizados o vendidos. Es equivalente a FIFO y se usa para gestionar el inventario de manera eficiente.

Prototipo: Versión inicial o modelo de un sistema, producto o aplicación, creado para ser evaluado y probado antes de su producción final. Los prototipos permiten identificar errores, recoger retroalimentación y mejorar el diseño antes de la implementación completa.

Python: Lenguaje de programación de alto nivel y multipropósito, conocido por su simplicidad y legibilidad. Es utilizado en diversas áreas, desde desarrollo web hasta análisis de datos y aprendizaje automático. *Python* facilita el desarrollo rápido y eficiente de aplicaciones.

Relacional: Tipo de base de datos que organiza la información en tablas relacionadas entre sí mediante claves. Las bases de datos relacionales permiten almacenar grandes volúmenes de datos de manera estructurada y son gestionadas mediante sistemas como *MySQL* y *PostgreSQL*.

Scrum: Metodología ágil para la gestión de proyectos, especialmente en el desarrollo de software, se basa en ciclos de trabajo cortos llamados *sprints* y promueve la colaboración entre equipos, la adaptación continua y la entrega incremental de productos funcionales.

Servidor web: Software que procesa y responde a las solicitudes de los clientes (generalmente navegadores web) para acceder a recursos en Internet. Un servidor web almacena, procesa y entrega páginas web a los usuarios, permitiendo la navegación en la web.

Síntesis: Proceso de combinar diferentes elementos o ideas para formar un todo coherente. En el contexto académico, la síntesis implica resumir y unificar información de diversas fuentes para crear una visión general o conclusiones sobre un tema.

Sistema web: Plataforma en línea diseñada para cumplir funciones específicas, como la gestión de contenido, comercio electrónico o servicios en la nube. Los sistemas web permiten a los usuarios interactuar y realizar tareas a través de navegadores de internet.

REFERENCIAS

- B., G. (2023). ¿Qué es CSS? *Hostinger*. <https://www.hostinger.es/tutoriales/que-es-css>
- B., G. (2023). Introducción básica a JavaScript. *Hostinger*.
<https://www.hostinger.es/tutoriales/que-es-javascript-introduccion-basica/>
- Bembibre, C. (2010). Boutique. *Significado*. <https://significado.com/boutique/>
- Depop. (s.f.). <https://news.depop.com/who-we-are/about>
- Duque, R. G. (s.f.). *Python para todos*.
<https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20para%20todos.pdf>
- Editorial Etece. (2020). Antecedentes. *Concepto*. <https://concepto.de/antecedentes/>
- Editorial Etecé. (2023). HTML. *Concepto*. <https://concepto.de/html/>
- Editorial Etecé. (2024). Delimitación del problema. *Concepto*. <https://concepto.de/delimitacion-del-problema/>
- Etecé. (2020). Entrevista. *Concepto*. <https://concepto.de/entrevista/>
- Etecé, E. (2023). Marco metodológico. *Concepto*. <https://concepto.de/marco-metodologico/>
- Etecé, E. (2023). Marco conceptual. *Concepto*. <https://concepto.de/marco-conceptual/>
- Fernández, E. (2022). ¿Qué es Django? *Tokio School*.
<https://www.tokioschool.com/noticias/que-es-django/>
- Gaona Espitia, J. L. (2014). *Repository*.
<https://repository.udistrital.edu.co/bitstream/handle/11349/4378/GaonaEspitiaJhonnyLeonardo2016.pdf?sequence=1&isAllowed=y>
- Gerencie.com. (2022). Método del promedio ponderado. *Gerencie*.
<https://www.gerencie.com/metodo-del-promedio-ponderado.html>
- Gillis, A. S. (s.f.). *ComputerWeekly*. <https://www.computerweekly.com/es/definicion/Servidor-web>
- Hamilton, T. (2024). Pilot testing. *Guru99*. <https://www.guru99.com/es/pilot-testing.html>
- Hernández, J. (s.f.). *Bind*. <https://bind.com.mx/blog/control-de-inventarios/tecnicas-y-metodos-para-el-control-de-inventarios>
- Jesús. (2003). ¿Qué es Django? *Dongee*. <https://www.dongee.com/tutoriales/que-es-django-en-programacion/>
- Loredo, A. (2018). Técnicas de observación directa e indirecta. *Prezi*.
<https://prezi.com/p/vksogmoh79-k/tecnicas-de-observacion-directa-e-indirecta/>

- Lozada Torres, E. F., & Ruiz Bermeo, L. A. (2018). *Dspace*.
<https://dspace.uniandes.edu.ec/handle/123456789/8522>
- Luis Alberto Casilla Santillan, O. p. (s.f.). *Base de datos en MySQL*.
<http://ual.dyndns.org/biblioteca/Bases%20de%20Datos%20Avanzado/Pdf/05%20Bases%20de%20datos%20en%20MySQL.pdf>
- Maldonado, R. (2024). Ventajas y desventajas de Python. *KeepCoding*.
<https://keepcoding.io/blog/ventajas-y-desventajas-de-python/>
- Martins, J. (2024). Objetivos del proyecto. *Asana*. <https://asana.com/es/resources/how-project-objectives>
- Meijomil, S. (2023). ¿Qué es un e-commerce? *InboundCycle*.
<https://www.inboundcycle.com/blog-de-inbound-marketing/que-es-un-ecommerce-tipos-y-como-crear-uno>
- Memon, M. (2022). Catálogo de productos. *Visme*. <https://visme.co/blog/es/catalogo-de-productos/>
- Mendez, B. (2015). El concepto de control en la investigación. *Prezi*.
<https://prezi.com/uepay6smb5rk/el-concepto-de-control-en-la-investigacion/>
- Narvaez, M. (s.f.). Trabajo de campo. *QuestionPro*.
<https://www.questionpro.com/blog/es/trabajo-de-campo/>
- Ortega, C. (2023). Procesamiento de datos en investigación. *QuestionPro*.
<https://www.questionpro.com/blog/es/procesamiento-de-datos-de-investigacion/>
- Ortega, K. (2023). ¿Qué es el desarrollo de páginas web? *World Campus*.
<https://worldcampus.saintleo.edu/noticias/que-es-el-desarrollo-de-paginas-web-que-hace-un-desarrollador-web>
- Pando, P. (2023). La situación actual del desarrollo de una ley de datos. *LinkedIn*.
<https://es.linkedin.com/pulse/la-situaci%C3%B3n-actual-del-desarrollo-de-una-ley-datos-paula>
- Parra, A. (s.f.). Recolección de datos para investigación. *QuestionPro*.
<https://www.questionpro.com/blog/es/recoleccion-de-datos-para-investigacion/>
- Posada, J. (2021). Cómo identificar los recursos necesarios para el éxito de un proyecto. *Visual Planning*. <https://www.visual-planning.com/es/blog/como-identificar-los-recursos-necesarios-para-el-exito-de-un-proyecto>
- Poshmark. (s.f.). <https://poshmark.com/about>
- Robledano, A. (2019). ¿Qué es MySQL? *OpenWebinars*. <https://openwebinars.net/blog/que-es-mysql/>

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2022). *Fundamentos de base de datos*. McGraw-Hill Interamericana. <https://doi.org/10.1036/0073523321>

ThredUp. (s.f.). <https://www.thredup.com/about>

Yepes, J. E. (2023). ¿Qué es un carrito de compras? *Dazzet*. <https://dazzet.co/que-es/carrito-de-compras/>