

SOLUCIÓN E-COMMERCE PARA LA COMERCIALIZACIÓN DE LOS
PRODUCTOS QUE OFRECE LA EMPRESA ASOPROCAFENPA.

Autor

JHEFFERSON ANDRES WALTEROS SOTABAN

Director

EDGAR ALBORNOZ ESPINEL

Magister en ciencias de la computación

INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERIAS Y ARQUITECTURAS



UNIVERSIDAD DE PAMPLONA
PAMPLONA, 30 noviembre del 2022

AGRADECIMIENTOS

Quiero agradecer primeramente a Dios, que me ha brindado fortaleza en toda mi trayectoria académica, así mismo les agradezco a mis padres Felipe Walteros y Ana Silvia Sotaban que siempre me han brindado su apoyo incondicional para poder cumplir todos mis objetivos personales y académicos. Ellos son los que con su cariño me han impulsado siempre a perseguir mis metas y nunca abandonarlas frente a las adversidades. También son los que me han brindado el soporte material y económico para poder concentrarme en los estudios y nunca abandonarlos.

En Segundo lugar, agradezco muy profundamente a mi tutor Edgar Albornoz Espinel por su dedicación y paciencia, sin sus palabras y correcciones precisas no hubiese podido lograr llegar a esta instancia tan anhelada. Gracias por su guía y todos sus consejos, los llevaré grabados para siempre en la memoria en mi futuro profesional.

RESUMEN

Los sistemas de comercio electrónico facilitan la interacción entre compradores y vendedores, actualmente las personas utilizan diferentes tipos de comercio electrónico para vender sus productos de manera eficiente, podemos encontrar tiendas como mercado libre o Amazon, que ofrecen una gran variedad de opciones para sus usuarios y compradores, también grandes y pequeñas empresas crean sus propias tiendas de comercio electrónico de acuerdo con sus objetivos de marketing y, por último, encontramos pequeños empresarios o revendedores que utilizan herramientas como Facebook Marketplace o Instagram Shop para llegar a clientes potenciales.

El objetivo principal de este trabajo de grado es desarrollar un sitio web E-Commerce para la comercialización de los productos de la empresa ASOPROCAFENPA de forma online, que se realizan tradicionalmente de casa a casa y por inventarios. Este proyecto describe cuándo el volumen de ventas es grande y un solo asesor no puede hacer manualmente estos procesos y mantenerlos de manera óptima.

Para la realización de este trabajo, se inició con la recopilación de la información y comprender el proceso manual que realizan los asesores, también se realizó, estudios de las herramientas que nos ofrece el mercado para identificar los aspectos puntuales con la necesidad de ver las características de apoyo para el desarrollo de software como estrategia direccional.

Para lograr los objetivos del proyecto, se desarrolló un comercio electrónico (E-Commerce) para el control de las ventas, con el framework de angular que utiliza como lenguaje de programación Typescript. A través de la metodología de Scrum que ayudo en la implementación del sistema como marco de trabajo y satisfacción del cliente.

Palabras clave: *E-Commerce, comercio electrónico, compras en línea, AngularJS.*

ABSTRACT

Electronic commerce systems facilitate the interaction between buyers and sellers, currently people use different types of electronic commerce to sell their products efficiently, we can find stores such as free market or Amazon, which offer a wide variety of options for their users and buyers, also large and small companies create their own e-commerce stores according to their marketing objectives and finally, we find small entrepreneurs or resellers who use tools like Facebook Marketplace or Instagram Shop to reach potential customers.

The main objective of this degree work is to develop an E-Commerce website for the commercialization of the ASOPROCAFENPA company's products online, which are traditionally carried out from house to house and by inventories. This project describes when the volume of sales is large and a single advisor cannot manually do these processes and maintain them optimally.

To carry out this work, it began with the collection of information and understanding the manual process carried out by the advisors, studies of the tools that the market offers us were also carried out to identify specific aspects with the need to see the characteristics support for software development as a directional strategy.

To achieve the objectives of the project, an electronic commerce (E-Commerce) was developed to control the sale, with the Angular framework that uses Typescript as programming language. Through the Scrum methodology, I help in the implementation of the system as a framework and customer satisfaction.

Keywords: E-Commerce, electronic commerce, online shopping, AngularJS.

Tabla de contenido

AGRADECIMIENTOS.....	2
RESUMEN	3
ABSTRACT	4
1. INTRODUCCION	10
CAPITULO I: EL PROBLEMA	12
1.1. Planteamiento del problema.....	12
1.2. Justificación	13
1.3. Delimitaciones	14
1.3.1. Objetivo general	14
1.3.2. Objetivos específicos.....	14
CAPITULO 2: MARCO TEORICO	15
2.1.1. La teoría de la Innovación tecnológica.....	15
2.1.2. La teoría de Schumpeter	15
2.1.3. E-Commerce.....	16
2.1.4. Diferencias del E-commerce y los negocios tradicionales.....	17
2.1.4.1. Las características físicas.....	17
2.1.4.2. Búsqueda y visualización de productos.	18
2.1.4.3. Estrategias de marketing.....	18
2.1.4.4. Estructura de personal	19
2.1.5. Posicionamiento	19
2.1.6. ¿Cómo posicionar para E-commerce?	20
2.1.7. Arquitectura de Software.....	21
2.1.7.1 Definición de arquitectura de software	22
2.1.7.2. Ciclo de desarrollo de la arquitectura	22
2.1.7.3. Beneficios de la arquitectura	25
2.1.8. Arquitectura Cliente/Servidor.....	25
2.1.9. Metodología SCRUM.....	26
2.1.9.1. ¿Qué es Scrum?.....	26
2.1.9.2. Roles SCRUM	27
2.1.9.3. Ventajas de SCRUM.....	28
2.2.3. Herramientas de desarrollo	28
2.2.3.1. Tecnologías para desarrollo web	28
2.2.3.2. Framework	29

2.2.3.3. Tabla frameworks para desarrollo web.	29
Framework Frontend	29
Framework Backend.....	30
2.2.3.4. Base de datos	31
MongoDB	32
SQLite	32
2.2.3.5. Angular Framework.....	32
2.2.3.6. Route Guards Angular - CanActivate.....	33
2.2.3.7. ¿Por qué utilizar el framework de Angular?	34
2.2.3.8. Node JS	35
2.2.3.9. ¿Por qué utilizar en el proyecto Node JS?	35
2.2.3.10. MongoDB.....	36
2.2.3.11. ¿Por qué utilizar MongoDB en el proyecto?	36
2.2.3.12. Seguridad API mediante Json Web Token (JWT)	38
2.2.3.13. Diseño Web Responsive	38
2.2.3.14. Lenguaje unificado de modelado (UML)	39
CAPÍTULO 3: ESTADO DEL ARTE	41
3.1. REGIONAL:.....	41
3.2. NACIONAL:	41
3.3. INTERNACIONAL:	42
CAPÍTULO 4: METODOLOGÍA.....	43
4.1. GESTIÓN DE PROYECTO.....	43
4.1.1 Planificación del proyecto	43
4.1.2 Ejecución y Seguimiento.	43
4.1.3. Finalización	44
4.2 Metodología de desarrollo	44
4.2.1 Recolección de requisitos.....	44
4.2.2 Gestión de backlog	44
4.2.3 Sprint Planning Meeting	45
4.2.4 Ejecución de sprint.	45
4.2.5 Inspección e iteración.	45
CAPITULO 5: DESARROLLO DE LA PROPUESTA	46
5.1 Planificación del proyecto	46
5.1.1 Análisis técnico y de requerimientos para la empresa ASOPROCAFENPA.....	46

5.1.1.2	49
5.1.2. Diagramas de casos de uso	50
5.1.2. Estudio del framework Angular para desarrollo del E-Commerce.....	54
5.1.3. Desarrollo del E-Commerce	67
• Disposición General	68
• Listado de los productos	72
• Detalle del producto.....	73
• Carrito de compras.....	74
• Perfil usuario	75
• Contacto.....	76
• Dashboard.....	78
• Gestor de usuarios	79
• Productos	80
• Cupones	81
• Descuentos.....	82
• Mensajes	83
• Ventas	84
• Configuraciones.....	85
CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES.....	92
6.1. Conclusiones.....	92
BIBLIOGRAFÍAS.....	94

INDICE DE TABLAS

<i>Tabla 1. Frameworks para desarrollo web</i>	<i>30</i>
<i>Tabla 2. frameworks para desarrollo backend.</i>	<i>31</i>
<i>Tabla 3. Gestores de bases de datos.....</i>	<i>32</i>
<i>Tabla 4. Fuente propia</i>	<i>48</i>
<i>Tabla 5. Fuente propia</i>	<i>49</i>
<i>Tabla 6. Fuente propia</i>	<i>50</i>
<i>Tabla 7. Fuente propia</i>	<i>51</i>
<i>Tabla 8. Fuente propia.</i>	<i>52</i>
<i>Tabla 9. Fuente propia.</i>	<i>53</i>
<i>Tabla 10. Conceptos importantes Angular</i>	<i>56</i>
<i>Tabla 11. Principales Directivas en Angular</i>	<i>60</i>

INDICE DE ILUSTRACIONES

<i>Ilustración 1. Actividades asociadas al ciclo de desarrollo de la arquitectura (a la izquierda) y su mapeo dentro de las actividades técnicas del desarrollo de sistemas (a la derecha).</i>	23
<i>Ilustración 2. Modelo Cliente/Servidor</i>	26
<i>Ilustración 3. Roles en SCRUM</i>	27
<i>Ilustración 4. Arquitectura AngularJS</i>	33
<i>Ilustración 5. Caso de uso registrarse</i>	50
<i>Ilustración 6. Caso de uso iniciar sesión</i>	51
<i>Ilustración 7. Caso de uso Agregar al carrito</i>	52
<i>Ilustración 8. Caso de uso pagar producto</i>	53
<i>Ilustración 9. Logotipo a todo color Angular</i>	55
<i>Ilustración 10. Uso de Factory y Service</i>	65
<i>Ilustración 11. Routing</i>	66
<i>Ilustración 12. Estructura de app-routing.module.ts</i>	67
<i>Ilustración 13. Landing page Createx – Home</i>	68
<i>Ilustración 14. Disposición General – Menú de navegación.</i>	69
<i>Ilustración 15. Página principal (Home)</i>	70
<i>Ilustración 16. Inicio de sesión y registro</i>	71
<i>Ilustración 17. Inicio de sesión y registro</i>	72
<i>Ilustración 18. Detalle del producto</i>	73
<i>Ilustración 19. Carrito de compras</i>	74
<i>Ilustración 20. Perfil Usuario</i>	75
<i>Ilustración 21. Modulo contacto</i>	76
<i>Ilustración 22. Modulo Iniciar sesión</i>	77
<i>Ilustración 23. Dashboard Administrativo</i>	78
<i>Ilustración 24. Gestor de usuarios</i>	79
<i>Ilustración 25. Productos</i>	80
<i>Ilustración 26. Cupones</i>	81
<i>Ilustración 27. Descuentos</i>	82
<i>Ilustración 28. Modulo Mensajes</i>	83
<i>Ilustración 29. Modulo ventas</i>	84
<i>Ilustración 30. Configuraciones</i>	85
<i>Ilustración 31. Productos</i>	86
<i>Ilustración 32. Modelo base de datos</i>	87
<i>Ilustración 33. Prueba de rendimiento e-commerce</i>	88
<i>Ilustración 34. Prueba de accesibilidad e-commerce</i>	89
<i>Ilustración 35. Buenas practicas</i>	90
<i>Ilustración 36. Prueba SEO</i>	91

1. INTRODUCCION

Este proyecto de grado en modalidad práctica empresarial en la empresa ASOPROCAFENPA, donde se detectaron las necesidades requeridas por la marca Café Baquiano que genera alta demanda operativa para su comercialización. Tiene como objetivo optimizar procesos operativos que se hacen manuales, llevándolos a través de una tienda online a un proceso automatizado con el uso de herramientas a un nivel de desarrollo de software sofisticadas para resolver la problemática, en donde pueden ofrecer productos a clientes potenciales y usuarios nuevos.

El desarrollo del presente proyecto de grado con el nombre “Solución E-Commerce para la comercialización de los productos que ofrece la empresa ASOPROCAFENPA.”, consta de cinco capítulos que se muestran a continuación para comprender los contenidos del proyecto.

Capítulo 1: “EL PROBLEMA”, Aborda las necesidades de la empresa, identificando los problemas de expansión de su producto en el mercado, la justificación y las metas a alcanzar.

Capítulo 2: “MARCO TEORICO”, Contiene una recopilación de información que brinda contexto para el desarrollo del proyecto, teórica que es necesaria para comprender un problema similar al caso y, por lo tanto, propone una solución al problema.

Capítulo 3: “ESTADO DEL ARTE”, en este capítulo se demuestra los documentos, fuentes bibliográficas, opiniones, investigaciones y ensayos existentes relacionados con el proyecto.

Capítulo 4: “METODOLOGÍA”, se determina la estructura, planificación y control del de las etapas del proceso de desarrollo del proyecto

Capítulo 5: “DESARROLLO DE LA PROPUESTA”, muestra el desarrollo del E-Commerce bajo la metodología seleccionada.

Capítulo 6: "CONCLUSIONES Y RECOMENDACIONES" Presenta conclusiones aprendidas y recomendaciones para su consideración.

Finalmente, el anexo correspondiente al proyecto de investigación

CAPITULO I: EL PROBLEMA

1.1. Planteamiento del problema

Colombia es un país que produce alrededor de 14 millones de sacos (60 kg de café cada uno) al año, y un año como el 2019 ha logrado 14,7 millones de sacos, e incluso se considera que se ha tocado techo de ventas en los últimos dos años. La historia de la producción nacional de café lleva 25 años sin registros, según la Fundación Nacional de Cafeteros (Cafeteros, 2020), algunos expertos en el tema han documentado ventas nacionales de café por 9 mil millones de pesos, movilizand o activamente a más de 600 cafeteros en el interior del país la economía de la ciudad.

A pesar de ello, la calidad excepcional del café colombiano y la enorme capacidad de producción y procesamiento artesanal e industrial del producto, los pequeños empresarios del sector han mostrado muchas veces dificultad para sostener su negocio (Aguiar et al., 2015), la cual persiste, debido a que la zona, aunque muy agrícola, y el hecho de que la mayoría de las familias del sector se dedican al cultivo y comercialización de café, cuando tienen que hacer una última tarea, no cumplen con los requisitos, por lo que en muchos casos el rubro de utilidad no es importante, por lo que es necesario identificar formas alternativas de comercialización de este producto para optimizar el nivel de ventas e ingresos, acordó vincular su proceso de marketing con la implementación de la estrategia de comercio electrónico.

La empresa ASOPROCAFENPA la cual es productora y vendedora de café ha venido presentando problemas al momento de tomar los pedidos de sus clientes, debido a que estos se realizan vía telefónica, constantemente este medio se satura, falla y las llamadas no se concretan, por consiguiente, varias ventas potenciales han fracasado, pues los compradores

al no poder contactarlos o continuar en la llamada deciden desistir de la compra. Al mismo problema se ven enfrentados los clientes nuevos que desean obtener más información sobre la empresa, sus procesos y los productos que venden. ASOPROCAFENPA al no tener abasto de personal para responder preguntas que suelen ser frecuentes no puede dar una calidad máxima de atención al usuario provocando que muchas preguntas queden sin resolver y algunos clientes pierdan el interés en conocerlos y adquirir sus productos.

De igual forma la información de pedidos y datos del cliente para realizar los envíos quedan guardados en papel y lápiz, lo cual suele generar conflictos y en varias ocasiones estos datos se pierden.

1.2. Justificación

En Colombia, el uso del comercio electrónico crece año tras año, cada vez es más común encontrar cualquier tipo de negocio en la web, la invitación es migrar las ventas online ya que las personas de hoy en día pasa más tiempo navegando en internet, por esto se debe aprovechar de esta oportunidad. Muchos usuarios que navegan por internet sin intención de comprar, pueden cambiar de opinión si le ofrece lo correcto, por lo tanto, las ventas en línea crece cada año y hace que las personas realicen compras desde la comodidad de su hogar.

Sin embargo, en la empresa ASOPROCAFENPA el comercio electrónico no es abordado en ninguno de sus procesos dejándola en clara desventaja ante su competencia. Actualmente las tiendas virtuales son una tendencia que impulsa el crecimiento en la comercialización de café, lo que permite analizar e integrar campos y segmentos de mercado más grandes de una manera más eficiente, eficaz, atractiva y responsable (Cabrera, 2012).

Este proyecto se desarrolla debido a la necesidad que presenta la empresa ASOPROCAFENPA y los efectos negativos a los que se ha tenido que someter por no poder

concretar ventas lo cual repercute en sus ingresos y sostenimiento. La implementación de este proyecto soluciona los problemas de visibilidad en el mercado que tiene esta empresa, así como el de automatizar los procesos de venta permitiendo que cualquier persona con acceso a internet pueda entrar al sitio web, conocer e interesarse por sus productos, resolver dudas y ahí mismo gestionar la compra del producto de su interés sin necesidad de recurrir a llamadas telefónicas que pueden hacer tedioso el proceso y en ocasiones terminar en una compra fallida.

Este proyecto aporta gran valor a la empresa ASOPROCAFENPA porque les abre camino a las nuevas tecnologías, las cuales ejecutadas de la manera correcta son grandes herramientas.

1.3. Delimitaciones

1.3.1. Objetivo general

Implementar un sitio web E-Commerce para la comercialización de los productos de la empresa ASOPROCAFENPA

1.3.2. Objetivos específicos

- Realizar el análisis técnico y de requerimientos para la empresa ASOPROCAFENPA.
- Estudiar el frameworks de Angular para el desarrollo del sitio web E-Commerce de la empresa ASOPROCAFENPA.
- Analizar, diseñar e implementar el sitio web E-Commerce para la empresa ASOPROCAFENPA.

CAPITULO 2: MARCO TEORICO

2.1.1. La teoría de la Innovación tecnológica

El presente marco teórico hace referencia a un constructo llamado La teoría de la Innovación tecnológica, la cual se refiere al desarrollo de nuevas opciones para suprimir necesidades mediante la creación y uso de diferentes medios y herramientas.

La innovación tecnológica brinda la oportunidad de desplazar prácticas engorrosas y ya obsoletas por la introducción de nuevos métodos significativos y mejores a los anteriores que encaminan al ahorro de recursos, y permiten llegar a un proceso de crecimiento y desarrollo alto para la persona o empresa que lo ponga en práctica.

Desde una empresa o como persona se pueden poner en práctica diferentes innovaciones tecnológicas a los métodos de trabajo, no existe límite y en este caso más es mejor siempre y cuando cada necesidad vaya siendo suprimida satisfactoriamente.

2.1.2. La teoría de Schumpeter

La teoría de Schumpeter se relaciona con la teoría de la innovación tecnológica, en esta el autor afirma que todo desarrollo económico se ve influenciado por el acto de nuevas innovaciones que conllevan a procesos dinámicos, los cuales pueden ser radicales o incrementales. En esta teoría Schumpeter afirma que las innovaciones ocurridas en las empresas transforman los modelos de negocio y esto encamina a nuevas oportunidades de crecimiento.

Schumpeter reconoce lo importante que es para una empresa el poder reconocer y aprovechar las oportunidades que se consideran competitivas en el mercado mediante una innovación radical.

En la actualidad la innovación tecnológica se ve en los diferentes procesos empresariales, lo que antes era una opción hoy poco a poco se convierte en una obligación, y quien no vaya de la mano con el auge que generan las nuevas tecnologías estará perdiéndose de diferentes oportunidades de crecimiento, como explorar nuevos mercados, llegar a más personas y crear reconocimiento a sus productos o servicios.

La teoría de la innovación tecnológica y la teoría de Schumpeter le aportan a este proyecto la visión clara de tomar falencias y transformarlas radicalmente en un proceso positivo e innovador para el crecimiento de la empresa ASOPROCAFENPA. Al identificar una necesidad esta se convertirá en un método a la espera de respuestas las cuales pueden ser abordadas de diferentes maneras.

Estas teorías aportan al proyecto aquí mencionado un canal para identificar la necesidad consentida por la empresa ASOPROCAFENPA, y transformarla en una acción innovadora que suprima esta necesidad, en este caso el realizar un sitio web basado en las nuevas tecnologías para llevar a cabo el propósito y misión de la empresa, así como crear nuevos recursos que generen mayores aportes y beneficios a la misma y una pasarela de pago que se acople a las nuevas tendencias en tecnología.

2.1.3. E-Commerce

El comercio electrónico (E-commerce) es la compra y venta de bienes y servicios, o la transmisión de fondos y datos, a través de una red electrónica, principalmente Internet. (SearchCIO, 2014) Hoy en día, Internet ha cambiado la forma en que hacemos negocios, ya sea encontrando nuevas fuentes de ingresos, adquiriendo nuevos clientes o administrando la cadena de suministro de una empresa. (Reynolds, 2004) El comercio electrónico es la

corriente principal, lo que permite a las empresas vender productos y servicios a los consumidores a nivel mundial.

La influencia de la web en la economía mundial es verdaderamente asombrosa. El mundo de los negocios sabe que la web es una de las mejores formas para que las empresas, como los fabricantes, vendan sus productos directamente al público, los minoristas tradicionales expandan sus tiendas a ubicaciones geográficas ilimitadas y los empresarios establezcan un nuevo negocio. económicamente (Reynolds, 2004)

2.1.4. Diferencias del E-commerce y los negocios tradicionales

2.1.4.1. Las características físicas

Tiendas minoristas tradicionales: una tienda física, visibilidad de productos, área de cobertura limitada, anaquel limitado, restricciones de horario, alquiler de tiendas, propiedad y gerentes de tienda.

Venta minorista en línea: exhibición de productos a través de imágenes, texto, etc. El área de cobertura depende de la capacidad de distribución. La ubicación geográfica ya no es un límite cuando se coopera con empresas de entrega. Los productos se pueden enviar a domicilio. Internet hizo que el lugar fuera menos importante y permite que muchos compradores y vendedores pasen por alto a los intermediarios tradicionales. Las tiendas en línea pueden estar abiertas 24/7; las personas pueden comunicarse según lo permitan sus horarios. (Smartamarketing, 2012) Además, se debe considerar el costo del personal de operación y servicio al cliente de la tienda.

2.1.4.2. Búsqueda y visualización de productos.

Venta minorista tradicional: el lugar adecuado para colocar los artículos es importante para la tienda minorista. Generalmente, los productos se muestran de acuerdo con el comportamiento de compra del cliente. Por ejemplo, la mercadería de marca de alto perfil y la mercadería promocional se colocarán en un lugar luminoso donde los clientes puedan llegar fácilmente a los estantes.

Venta minorista en línea: el cliente puede encontrar productos satisfactorios por palabra clave de búsqueda, categoría y rango de precios. Las imágenes y la descripción de los detalles del producto pueden proporcionar a los clientes una variedad de soporte para la decisión de compra, como la lista de productos de venta, imágenes de múltiples ángulos que muestran la apariencia de los productos, presentación de video en la mercancía, una variedad de tipos (como precio, inventario, venta, clasificación, etc.), recomendación personalizada, categoría y visualización claras, recomendaciones de productos relacionados, función de paridad de productos, combinación de productos básicos, etc. (Zhang y Wang, 2009)

2.1.4.3. Estrategias de marketing

Venta minorista tradicional: DM (marketing directo), publicidad (como exteriores, periódicos, televisión, etc.), mensajes de texto, ambiente de tienda y lugar de trabajo con sonido, luz, olor y otros efectos que pueden usarse para estimular la compra, cara a cara. Recomendar la comunicación a través del personal de ventas.

Venta minorista en línea: SEO / SEM (optimización de motores de búsqueda / marketing de motores de búsqueda), EDM (marketing por correo electrónico), alianza de red, publicidad en el portal, cooperación SNS. Los vendedores en línea pueden usar la

información de compra del cliente para la comercialización. En el mundo digital, la información del cliente es fácil y económica de recopilar, almacenar y analizar. Los gerentes pueden realizar un seguimiento de los resultados de marketing a medida que se implementan los planes y recibir informes detallados. Sin embargo, convertir grandes bases de datos en conocimiento significativo para guiar decisiones estratégicas es un gran desafío. (Smartamarketing, 2012)

2.1.4.4. Estructura de personal

Venta minorista tradicional: es importante contar con una gran cantidad de experiencia minorista en recursos humanos, la cantidad de personal depende del tamaño de las operaciones de la tienda.

Venta minorista en línea: el personal de TI y el personal de operaciones de red son muy importantes. Los especialistas en marketing deben comprender la tecnología para aprovechar su poder. No tienen que desarrollar personalmente las tecnologías, pero necesitan saber lo suficiente para seleccionar los proveedores adecuados y dirigir a los profesionales de la tecnología. (Smartamarketing, 2012)

2.1.5. Posicionamiento

El posicionamiento es una estrategia que tiene como objetivo hacer que una marca ocupe una posición distinta, en relación con las marcas competidoras, en la mente del cliente.

Las empresas aplican esta estrategia enfatizando las características distintivas de su marca (qué es, qué hace y cómo, etc.) o pueden tratar de crear una imagen adecuada (económica o premium, utilitaria o lujosa, de nivel de entrada o alta). -fin, etc). Una vez que

se posiciona una marca, es muy difícil reposicionarla sin destruir su credibilidad. (Wikipedia, 2015)

El posicionamiento ayuda a establecer la identidad del producto o servicio ante los ojos del comprador. La estrategia de posicionamiento de una empresa se ve afectada por una serie de variables relacionadas con las motivaciones y requisitos de los clientes, así como por las acciones de sus competidores.

2.1.6. ¿Cómo posicionar para E-commerce?

Análisis de tendencias de la industria

El análisis de tendencias de la industria mostrará una descripción general de las industrias con la situación de desarrollo actual. El análisis de tendencias de la industria se puede realizar mediante preguntas como: ¿cuál es el valor actual de esta industria? ¿Cómo es esta industria, está creciendo o no? ¿Quiénes son los principales competidores o empresas en esta industria? ¿Cómo es la participación de mercado como el tamaño del mercado, la participación y los datos de segmentación? (Universidad Estatal de Michigan, 2015)

Las empresas que estén listas para incursionar en un campo de comercio electrónico primero deben despejar los siguientes problemas:

- (1) ¿Este mercado existe o no?
- (2) ¿Es lo suficientemente grande?
- (3) ¿Cómo es la tasa de crecimiento anual en el número?

2.1.7. Arquitectura de Software

El software está presente en una amplia gama de objetos que nos rodean, desde los teléfonos y otros dispositivos que llevamos con nosotros casi permanentemente, hasta los sistemas que controlan el funcionamiento de varias organizaciones o los que operan sondas robóticas para explorar otros planetas. Uno de los factores clave para el éxito de un sistema es su buen diseño, en particular el llamado diseño de arquitectura de software.

El desarrollo de un sistema de software puede verse como un avance hacia una solución técnica a un problema u oportunidad. Durante la transición desde el comienzo del dominio del problema hasta el final del dominio de la solución, se realizan diferentes actividades técnicas como se describe a continuación, según (Cervantes Maceda, Castro Careaga, & Velasco-Elizondo, 2015)

- **Requerimientos.** Se refiere a identificar las necesidades de los clientes y otras partes interesadas en el sistema y generar una especificación suficientemente detallada de lo que debe hacer el sistema.
- **Diseño.** En esta fase, los requisitos se traducen en un diseño o modelo para construir el sistema. Se refiere esencialmente a tomar una decisión sobre la forma de abordar los requisitos identificados previamente. El resultado del diseño es identificar las partes del sistema que satisfarán esas necesidades y ayudarán a las personas que componen el equipo de desarrollo a construirlo simultáneamente.
- **Construcción.** Se refiere a la creación de sistemas mediante el desarrollo y hacer pruebas de las partes que componen el sistema para su posterior integración, es decir, conectando partes relacionadas entre sí. Las partes deben detallarse individualmente,

pero el diseño detallado de los componentes difiere del diseño estructural general del sistema completo, descrito en el punto anterior.

- **Pruebas.** Esta actividad se refiere a la prueba de un sistema o parte del mismo para verificar que se cumplan los requisitos previamente determinados y para identificar y corregir fallas.
- **Implantación.** En esta etapa se realiza la transición de un sistema de un entorno de desarrollo a un entorno que se ejecuta permanentemente y está disponible para los usuarios finales.

2.1.7.1 Definición de arquitectura de software

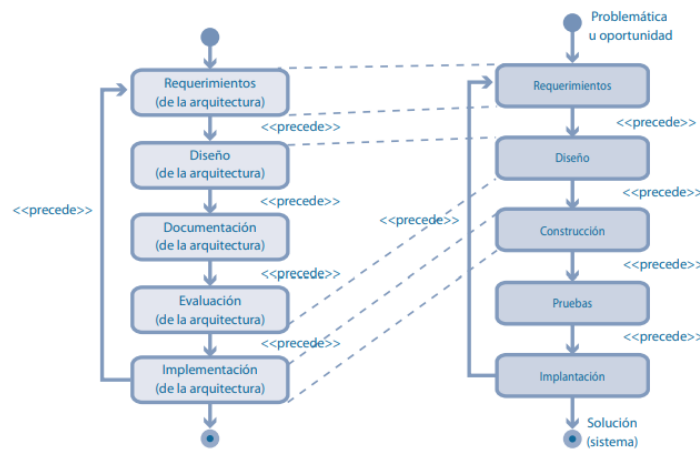
No existe una definición uniforme del concepto de arquitectura de software. Sin embargo, la siguiente definición general propuesta por Software Engineering Institute tiende ampliamente ser aceptada:

“La arquitectura de software de un sistema es el conjunto de estructuras necesarias para razonar sobre el sistema. Comprende elementos de software, relaciones entre ellos, y propiedades de ambos. (Bass, Clements y Kazman, 2012).”

2.1.7.2. Ciclo de desarrollo de la arquitectura

Como se mencionó anteriormente, una de las definiciones de arquitectura de software es definir la estructura que debe tener el software, por lo que podemos decir que el ciclo de desarrollo de la arquitectura de software contiene actividades específicas, como se describe a continuación y se integran en las actividades del desarrollo del sistema, como se muestra en la ilustración 1, según (Cervantes Maceda, Castro Careaga, & Velasco-Elizondo, 2015)

Ilustración 1. Actividades asociadas al ciclo de desarrollo de la arquitectura (a la izquierda) y su mapeo dentro de las actividades técnicas del desarrollo de sistemas (a la derecha).



Fuente: (Cervantes Maceda, Castro Careaga, & Velasco-Elizondo, 2015)

- **Requerimientos de la arquitectura**

La fase de requerimientos y la primera fase del ciclo de desarrollo de la arquitectura se basan en la captura, documentación y priorización de los requisitos que afectan a la arquitectura, donde los atributos de calidad juegan un papel importante en términos de requisitos. Los casos de uso y las restricciones también están relacionados con la arquitectura.

- **Diseño de la arquitectura**

La fase de diseño es probablemente la fase más compleja del ciclo de desarrollo de la arquitectura. En ella definen las estructuras que componen la arquitectura tomando decisiones de diseño. En esta etapa esta la creación estructural generalmente se basa en dos clases de soluciones abstractas probadas, llamadas

patrones de diseño y tácticas, también soluciones específicas como opciones tecnológicas, como frameworks.

- **Documentación de la arquitectura**

Una vez creado el diseño de la arquitectura, es necesario darlo a conocer a otros interesados como a los desarrolladores, agentes de implementación, líderes de proyectos o los propios clientes. La comunicación exitosa a menudo depende de la documentación adecuada del diseño. A pesar de que durante el proceso de diseño se realiza la documentación inicial que puede incluir un boceto de la estructura o la captura de las decisiones de diseño, la documentación formal implica representar su estructura mediante vistas.

- **Evaluación de la arquitectura**

Dado que la arquitectura de software juega un papel vital en el desarrollo, para identificar posibles riesgos es crucial que una vez se documente, evaluar el diseño. Las ventajas de evaluación es que representa una actividad que se puede hacer temprano, y el costo de corregir los defectos encontrados a través de él es mucho menor que el costo de corregirlos después de construir el sistema.

- **Implementación de la arquitectura**

Una vez establecida la arquitectura, se establece el sistema. En esta etapa, es importante evitar que sucedan desviaciones del diseño definido por el arquitecto.

2.1.7.3. Beneficios de la arquitectura

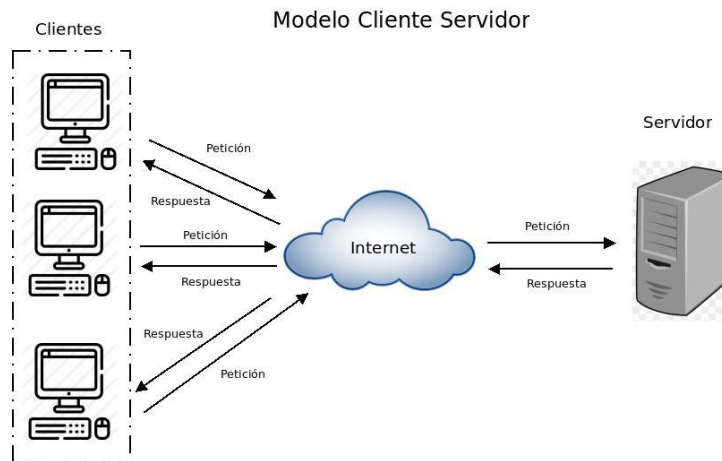
Como se mencionó al comienzo de este ítem, el desarrollo de sistemas a menudo enfrenta las siguientes limitaciones, la relación entre tiempo, costo y calidad. A continuación, se describen diversos beneficios que mejoran estas limitaciones, según (Cervantes Maceda, Castro Careaga, & Velasco-Elizondo, 2015)

- Aumentar la calidad de los sistemas: la arquitectura permite satisfacer los atributos de calidad de un sistema y disminuye la cantidad de defectos del mismo.
- Mejorar tiempos de entrega de proyectos: La estructura arquitectónica contribuye directamente a una estimación más precisa del tiempo requerido para el desarrollo.
- Reducir costos de desarrollo: Un buen diseño ayuda a reducir la necesidad de reelaboración y facilita el mantenimiento, lo que también reduce los gastos.

2.1.8. Arquitectura Cliente/Servidor

El proyecto se diseñará de acuerdo con un patrón llamado arquitectura cliente/servidor, que consiste en una serie de servicios de red que serán alojados en un servidor, quien escucha constantemente al cliente y es el responsable de solicitar y recibir una respuesta concreta al recurso solicitado, para lo cual los métodos a menudo se denominan API REST.

Ilustración 2. Modelo Cliente/Servidor



Fuente: (Schiaffarino, 2019)

Además, se utilizará una base de datos relacional, que es un tipo de base de datos Almacene y proporcione acceso a puntos de datos relacionados. Las bases de datos relacionales se basan en el modelo relacional, un sistema intuitivo y Una forma sencilla de representar datos en una tabla (Herrera, 2022).

2.1.9. Metodología SCRUM

2.1.9.1. ¿Qué es Scrum?

Scrum es una de las metodologías de desarrollo ágil más populares en la actualidad. Básicamente son entregas parciales del producto final, entregadas en orden de prioridad según los clientes lo requieran, se caracteriza por un progreso gradual en la entrega de software de forma colaborativa, funcional, flexible y Adaptable. Este método de trabajo permite una mejor distribución del tiempo y evita un proyecto muy extenso.

Con Scrum, estos grandes proyectos se pueden dividir en una serie de tareas y, de esta manera, el trabajo se vuelve más ágil. Al comienzo de cada ciclo se define tareas a lo largo

del período de desarrollo y entrega de resultados específicos al final. Además, propone una división de roles entre los equipos de trabajo fomentando la colaboración.

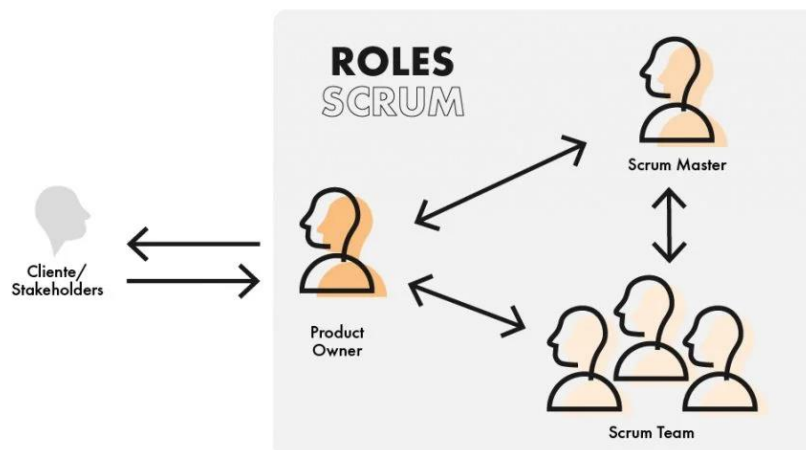
2.1.9.2. Roles SCRUM

Product owner: Representantes de accionistas y clientes que utilizan el software. Se enfoca en la parte comercial y es responsable del ROI del proyecto. Transfiere la visión del proyecto al equipo, formaliza los beneficios en la historia del Product Backlog y priorízalos regularmente.

Scrum master: Es quien lidera el equipo, a su vez los guía a seguir las reglas y procesos de la metodología. También es responsable de reducir los obstáculos del proyecto y trabajar con el Product Owner para maximizar el ROI.

Team: es el grupo de profesionales que tienen los conocimientos técnicos necesarios para el desarrollo de los entregables del proyecto.

Ilustración 3. Roles en SCRUM



Fuente: (López, 2022)

2.1.9.3. Ventajas de SCRUM

Scrum ofrece varias ventajas, la más destacada de las cuales son las siguientes:

- Establecer horas de trabajo para las tareas y planifica el proyecto en fases.
- Previene la distracción constante de nuevas tareas y busca el enfoque en el plan de ciclo actual.
- Mejorar la comunicación dentro de los equipos de trabajo.
- Delimita roles y asigna tareas a los miembros del equipo.

2.2.3. Herramientas de desarrollo

2.2.3.1. Tecnologías para desarrollo web

HTML (HyperText Markup Language) es la tecnología encargada de dar la estructura del sitio web, con esta se crearán los enlaces, los espacios adecuados para cada texto, imagen etc y tiene gran responsabilidad en el tema SEO, que sería la intención con que los buscadores presentan el sitio web entre los primeros resultados.

CSS (cascading style sheet) Tecnología encargada de dar los estilos al sitio web con el propósito de que su visión sea cómoda, agradable y adaptable a diferentes dispositivos, de aquí se implementará el color para la web, su tipografía, fuente, tamaño y cualquier tema relacionado en cómo se verá el sitio web para el usuario final.

JavaScript es un lenguaje de programación interpretado y ejecutado por el navegador, o sea desde el lado del cliente. Es el lenguaje más demandado y querido por desarrolladores al momento de crear sitios web, su versatilidad permite dar interactividad y dinamismo a los mismos. Desde crear nuevos enlaces, dar nuevos estilos, enviar mensajes al

usuario, crear alertas y mucho más es posible con JavaScript por eso es ideal en el desarrollo web.

2.2.3.2. Framework

Un Framework es un marco o esquema de trabajo que los programadores suelen utilizar para desarrollar software. El uso de un marco le permite simplificar el proceso de desarrollo, ya que evita la duplicación de código, garantiza las buenas prácticas y la coherencia del código.

Entonces, un Framework es un conjunto de herramientas y módulos que se pueden reutilizar en varios proyectos. Uno de los frameworks más famosos y utilizados es .NET Framework de Microsoft para la web.

2.2.3.3. Tabla frameworks para desarrollo web.

Framework Frontend

A continuación, se listan los 6 mejores frameworks utilizados para el desarrollo web (Tápanes, 2022)

Nombre	Código abierto	Licencia	Tipo de aplicación	Curva de Aprendizaje	DOM Virtual	Creador
Angular	Si	MIT	Una página (SPA)	Media	No	Google
React	Si	MIT	Una página (SPA)	Media	Si	Facebook
Vue	Si	MIT	Una página (SPA)	Leve	Si	Vue Core Team
Ember	Si	MIT	Varias páginas	Media	No	Ember Core Team

Qooxdoo	Si	MIT	Aplicaciones enriquecidas (RIA)	Alta	No	Qooxdoo Core Team
Dojo	Si	BSD	Aplicaciones enriquecidas (RIA)	Alta	Si	JS Fundación

Tabla 1. Frameworks para desarrollo web

Framework Backend

La siguiente tabla contiene los mejores Framework de trabajo backend para el desarrollo web. (Clark, 2021)

Marco de Trabajo	Lenguaje de Programación	Casos de Usos Famosos
Django	Python	Instagram Pinterest Coursera
Laravel	PHP	Deltanet Travel Neighborhood Lender MyRank
Ruby on Rails	Ruby	ZendDesk Shopify GitHub
ExpressJS	NodeJS	MySpace GeekList Storify
CakePHP	PHP	Mapme Educationunlimited Followmy Tv
Flask	Python	Red Hat Rackspace Reddit

Asp .NET	C#	Microsoft Godaddy Ancestry
Spring Boot	Java	Trivago Via Varejo Intuit
Koa	NodeJS	—
Phoenix	Elixir	Financial Times Fox 10 ABC15

Tabla 2. frameworks para desarrollo backend.

2.2.3.4. Base de datos

Es un software que controla la organización, el almacenamiento, la recuperación, la seguridad y la integridad de los datos en las bases de datos, acepta solicitudes de datos de las aplicaciones y ordena al sistema operativo que envíe los datos apropiados.

Una base de datos es una colección de datos interrelacionados a través de los cuales se pueden registrar y tener un significado implícito.

Un Sistema de Gestión de Base de Datos (DBMS) o DGBA (Database Management System) es un conjunto de programas invisibles que gestionan la información contenida en una base de datos. Un administrador de base de datos o administrador de datos administra todos los accesos a una base de datos, ya que su objetivo es actuar como una interfaz entre esta, los usuarios y las aplicaciones. (Darias Pérez, 2021)

A continuación, se muestran los cinco principales gestores de bases de datos más utilizados en la actualidad junto con sus características (Maldonado, 2022):

Nombre	Características	Tipo
MySQL	• Arquitectura cliente/servidor • Compatibilidad con SQL • Procedimientos almacenados	SQL
Microsoft SQL Server	• Visualización en dispositivos móviles • Cifrado de datos y cumplimiento normativo • Facilidad de uso	SQL
PostgreSQL	• Soporte para múltiples tipos de datos • Objeto/relacional • Soporte para gran variedad de lenguajes	SQL
MongoDB	• Gran sintaxis de consultas • Indexación y replicación • Balanceo de carga	NoSQL
SQLite	• Archivo de datos único • Sin servidor • Migración de información	SQL

Tabla 3. Gestores de bases de datos.

2.2.3.5. Angular Framework

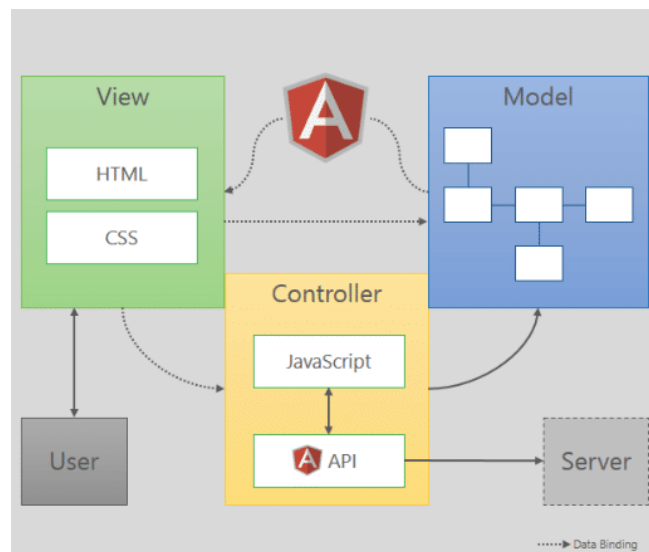
Angular es un framework de código abierto desarrollado por Google (Misko Hevery, Adam Abrons), para facilitar el diseño y desarrollo de aplicaciones web de estilo Single Page Application (SPA) y Progressive Web App (PWA) para que sean eficientes e igualmente sofisticadas.

Angular separa completamente el frontend y el backend de la aplicación, evitando escribir código redundante y establece un orden gracias a su arquitectura MVC (Modelo –

Vista - Controlador), garantizando un desarrollo rápido, a la vez que permite actualizaciones y modificaciones del mismo.

El lenguaje de programación principal de Angular es Typescript, por lo que toda la sintaxis y la forma en que se desarrolla el código es igual, lo que aumenta la consistencia y coherencia de la información, por ejemplo, permitir la incorporación de nuevos desarrolladores, si es necesario, ya que pueden continuar con el proyecto sin alta complejidad. (Deyimar A, 2022)

Ilustración 4. Arquitectura AngularJS



Fuente: (Kothari, 2022)

2.2.3.6. Route Guards Angular - CanActivate

Siempre que construimos una aplicación web, hay algunas rutas (direcciones o páginas) a las que solo pueden acceder los administradores de la aplicación. Por ejemplo, los usuarios no pueden ingresar una lista de usuarios para eliminarlos. Para eso están las Guardias Angulares.

En Angular los guards trabaja junto con una función llamada CanActivate. Esta función debe ser utilizada en la URL que se quiere controlar. La función en la URL es la que hace la solicitud al Guards y dependiendo de lo que devuelva, la ruta se puede activar o mostrar, por eso se llama CanActivate.

2.2.3.7. ¿Por qué utilizar el framework de Angular?

La elección del framework Angular para el desarrollo del E-commerce, se debe al conjunto de ventajas que ofrece al permitir el uso de desarrollo web estructurado que integra lógica y procesamiento de datos con usuarios finales, lo que conduce a una excelente calidad del producto, así como también una alta mejora de buenas prácticas de codificación (BPC). Entre sus ventajas que se diferencia de otros framework son:

- Un framework basado en componentes para crear aplicaciones web eficientes, sofisticadas y escalables.
- Un conjunto de bibliotecas integradas que soporta una amplia diversidad de funcionalidades, la gestión de formularios, incluido el enrutamiento, la comunicación cliente-servidor entre otras.
- Una colección de herramientas de desarrollo para ayudarlo a desarrollar, compilar, testear y actualizar el código. (*Angular*, s. f.)

2.2.3.8. Node JS

Node.js es un entorno de tiempo de ejecución de JavaScript diseñado para crear Aplicaciones web escalables del lado del servidor, Además, utiliza una arquitectura dirigida por eventos y se basa en un motor Google V8 (OpenJS Foundation, s.f.).

Es una máquina virtual extremadamente rápida y alta eficacia escrita por personas como Lars Bak, especializado en VM (máquina virtual). No olvidemos que V8 se actualiza a menudo y es uno de los intérpretes más ligeros que posiblemente existan en la actualidad para cualquier lenguaje dinámico (Urquiaga, 2018)

Además, las capacidades de I/O (entrada/salida) de Node.js son muy ligeras y extremadamente potentes, que permite a los desarrolladores aprovechar al máximo las I/O del sistema. Node.JS admite los protocolos TCP, DNS y HTTP, Fue creado por Ryan Dahl en 2009, Su desarrollo fue patrocinado por Joyent Corporation, que también firmó el contrato Dahl (Urquiaga, 2018).

2.2.3.9. ¿Por qué utilizar en el proyecto Node JS?

Node.js nació con la idea de poder ejecutar este lenguaje fuera del entorno del navegador y fue desarrollada por los fundadores de JavaScript.

Por lo tanto node.js no solo te permite crear sitios web interactivos, también los puede hacer más ágiles y capaces de trabajar con otros lenguajes de secuencia como Python. Esto permite a los desarrolladores usarlo principalmente en aplicaciones networking en grandes proyectos que buscan ser rápidos o en proyectos que los procesos sean ágiles, por ejemplo, APIS, desarrollo de aplicaciones Web usando Ajax, Push Messaging y lo más importante IoT (Gacelaweb, 2020)

2.2.3.10. MongoDB

MongoDB es una base de datos orientada a documentos. Esto significa que en lugar de guardar datos en registros, guarda datos en documentos. Estos documentos se almacenan en BSON, que es la representación binaria de JSON.

Una de las diferencias más importantes con una base de datos relacional es que no tiene que seguir un esquema. Los documentos de la misma colección (un concepto similar a una tabla en una base de datos relacional) pueden tener diferentes esquemas. (Rubenfa, 2014)

2.2.3.11. ¿Por qué utilizar MongoDB en el proyecto?

MongoDB nos ofrece modelamiento de datos el cual tiene una arquitectura MEAN (Mongo Express Angular Node.js) Dado que los datos se almacenan en una estructura similar a JSON, esto significa que el flujo de datos dentro de la aplicación no realiza cambios significativos en la estructura de datos. Mencionare 3 razones para usar MongoDB en el presente proyecto.

Escalabilidad, Como base de datos distribuida, MongoDB no solo puede escalar verticalmente (CPU y RAM), sino también escalar horizontalmente (creando más nodos).

Flexibilidad, El uso de una base de datos basada en documentos Schema Less como MongoDB permite que su base de datos crezca con su aplicación sin tener que ejecutar un script cada vez que desee agregar un nuevo campo a su aplicación para crear un campo con registros de valor predeterminado. En MongoDB, es normal que los documentos de una colección no tengan exactamente los mismos campos.

Alta disponibilidad, MongoDB le permite tener clústeres distribuidos. Esto mejora la velocidad de las consultas al reducir la latencia que existe entre el clúster de la base de datos y el servicio que ejecuta la consulta. (ramirez, 2019)

2.2.3.12. Real Time con Socket.io

Con la aparición de HTML5 y nuevas funciones, ha surgido la tecnología WebSockets basada en el protocolo ws. Fue creado para mantener una conexión constante entre el cliente y el servidor, haciendo que la comunicación sea mucho más fluida que las tradicionales llamadas http. Esto pinta un escenario ideal para aplicaciones de chat, videojuegos, carritos de compras en línea o visualización de información en tiempo real.

A la hora de implementar WebSockets no es tan complejo por el lado del cliente, con solo mirar la documentación podemos hacerlo rápido, pero la parte del servidor puede variar mucho dependiendo del lenguaje que usemos.

Así nació Socket.io que es una biblioteca de código abierto, en un principio fue diseñado para usarse con servidores codificados en NodeJS, pero la razón para llevarlo a otros lenguajes de programación como Python, Go, Java, entre otros, fue gracias a su éxito.

Socket.io tiene una comunidad muy amplia que nos ayuda a crear nuestras páginas web con conexiones real time entre clientes y servidor, facilitándonos las vidas proporcionándonos un mismo código tanto por el cliente como por el servidor.

2.2.3.12. Seguridad API mediante Json Web Token (JWT)

Según una investigación (Sergio, 2017), JSON Web Token (JWT) es un estándar abierto (RFC-7519) basado en JSON para crear un token que sirva para enviar datos entre aplicaciones o servicios y garantizar que sean válidos y seguros.

Cuando se usan JWT, el caso más común es usarlos en páginas web o app móviles para autenticar a los usuarios. Primero, un usuario ingresaría su información de inicio de sesión y crearía un JWT antes de enviarlo a la aplicación. Cada vez que la aplicación realiza una solicitud, envía el JWT que utiliza el servidor para verificar que el usuario está autenticado y quién dice ser.

Los JWT tienen muchos otros usos además de transferir datos entre servicios. Cualquier dato que deba transferirse puede validarse enviándolo a través de un JWT. Por ejemplo, un servicio de envío de correo electrónico podría solicitar datos a través de un JWT con el contenido del correo electrónico o cualquier otra información necesaria para enviar el correo electrónico. De esta forma, estamos seguros de que los datos no han sido alterados o perdidos.

2.2.3.13. Diseño Web Responsive

Con el crecimiento tecnológico vivido en la actualidad en cuanto a dispositivos como celulares, tablets, pc y una gran variedad de tipos de pantalla, se hace necesario crear una adaptabilidad del diseño web a estos nuevos mecanismos de comunicación y consumo de información que surgieron. Gracias a tecnologías como CSS que permiten crear mejores diseños y darle estilo a sitios web, surge la implementación de media queries que para el día de hoy es un gran aliado para la creación de sitios web responsivos, sitios web capaces de

adaptarse y verse de la mejor manera desde diferentes tipos de resolución de pantalla sin importar si el sitio web se ve desde un celular, tablet o pc, este tipo de herramienta permite cambiar colores, fondos, tipos y tamaños de letra, ubicación de elementos y demás dependiendo de la resolución de pantalla desde donde se encuentre navegando el usuario. El diseño web responsive hace referencia a temas de usabilidad desde el lado del usuario, el cómo este sitio web se adapta al ambiente del usuario e interactúa con él, esto conlleva a un sitio web más comprensivo, amigable, con mejor usabilidad y visión de su contenido. (Aaron Gustafson, 2015)

2.2.3.14. Lenguaje unificado de modelado (UML)

El lenguaje modelo UML es el estándar más utilizado para describir y estructurar software y estos sistemas deben documentarse con precisión en un lenguaje gráfico. Modelar un sistema requiere crear tres piezas separadas de UML especificar, construir y documentar las partes que modelan un sistema. UML fue diseñado para ser un lenguaje de modelado de propósito general para que pueda usarse para especificar la mayoría de los sistemas basados en objetos o componentes y para modelar aplicaciones de muchos dominios de aplicaciones diferentes y plataformas de objetos distribuidos. (Vallecillo, L. Fuentes y A., 2004).

El uso de UML para el diseño de sistemas trae una serie de beneficios, debido a que van de la mano de un conjunto de diagramas que admiten modelar sistemas mediante análisis de los requerimientos del sistema implementado, en UML los diagramas más comunes son:

- Diagrama de Clases
- Diagrama de Objetos
- Diagrama de Casos de Uso

- Diagrama de Estados
- Diagrama de Secuencias
- Diagrama de Actividades

CAPÍTULO 3: ESTADO DEL ARTE

A continuación, se realizará una síntesis de las investigaciones y/o plataformas que generaron aportes al presente proyecto.

3.1. REGIONAL:

En Casanare, MinTIC socializó su oferta de transformación digital, conectividad y apropiación, MinTIC, Casanare, Colombia, 2022.

El objetivo de este artículo es promover el servicio de internet gratuito para incentivar el uso del proyecto Vende Digital.

Este artículo contribuyó a este proyecto porque promueve la participación de comerciantes y empresarios en las nuevas tecnologías e implementaciones de negocio, demuestra cómo a través de estos medios digitales se es posible aumentar las ventas, exposición de productos y generar mayores ganancias.

3.2. NACIONAL:

Los medios de pago y su impacto en el comportamiento del comercio electrónico en Colombia, Diana Chipaquetua, Luis Felipe Infante, Bogotá, Colombia, Facultad de administración, finanzas y ciencias económicas, Ean Universidad, 2019.

Este artículo tiene como objetivo dar a conocer las ventajas que generan los pagos electrónicos implementados en los sitios web en Colombia.

El estudio de este artículo fue un valioso aporte para el proyecto porque menciona las necesidades actuales que viven las empresas en desarrollo y cómo ese mismo desarrollo va generando mayores necesidades que deben ser abarcadas por la misma, para este caso las

crecientes ventas generaron la necesidad de implementar un comercio electrónico adaptándose así a las nuevas tecnologías y su evolución.

3.3. INTERNACIONAL:

Sistemas de pago para comercio electrónico, Sandra Mireya Monreal Mendoza, Zacatecas, México, Maestría en Ingeniería de Software, CIMAT, 2012.

Este artículo tiene como objetivo mostrar diferentes pasos para implementar comercios electrónicos en sitios web y cómo estos pueden generar mayor valor al negocio de la empresa.

Este artículo de (Monreal Mendoza, 2012) ayudó a establecer bases en el proyecto para implementar correctamente un modelo de negocio basado en el comercio electrónico, ventas y pagos desde un sitio web, de igual manera muestra datos de cómo este tipo de implementaciones generan innovación en los sitios web y presenta un aval positivo por parte de los clientes, quienes se ven mayormente beneficiados. Demuestra también cómo los sitios web generan mayor valor de exposición en los productos y servicios comercializados por las empresas.

CAPÍTULO 4: METODOLOGÍA

Los proyectos de desarrollo de software requieren elegir una metodología adecuada basada en los requisitos del proyecto, el tamaño del equipo, el presupuesto y tiempo de vida del mismo. Actualmente existen muchas metodologías de desarrollo de software, elegir la metodología adecuada depende del propósito del proyecto y otras variables.

4.1. GESTIÓN DE PROYECTO

Al momento de ejecutar un proyecto es importante definir el inicio y el final, ya que la gestión de proyectos recomienda dividirlo en pequeñas tareas, fases o etapas para que se alcancen satisfactoriamente las metas deseadas, de esta manera teniendo en cuenta las buenas prácticas en la gestión de proyectos. El presente trabajo de grado está dividido en las siguientes tres etapas principales.

4.1.1 Planificación del proyecto

En esta fase inicial, se hace el levantamiento de la información necesaria para comprender los requisitos, posteriormente proceder con el diseño y desarrollo del E-Commerce, sabiendo el PMV (Producto Mínimo Viable), recolectar los requerimientos, construir los inventarios del E-Commerce y las limitaciones para ejecutar el proyecto.

4.1.2 Ejecución y Seguimiento.

Después de obtener datos, recolectar los requerimientos y comprender las limitaciones, se inicia con el diseño, desarrollo e implementación del E-Commerce utilizando

la metodología Scrum en la que se realiza el proceso de desarrollo y seguimiento. En cada iteración se realiza el proceso de testing y monitoreo, ya que dentro de la metodología de scrum lo incluye.

4.1.3. Finalización

Esta es la fase final, donde el proyecto termina cumpliendo el alcance inicial y satisface a todas las partes involucradas, posteriormente se realiza un informe donde se especifica que el proyecto ha culminado satisfactoriamente.

4.2 Metodología de desarrollo

4.2.1 Recolección de requisitos.

La recolección de requisitos es fundamental para el desarrollo de proyectos ya que incluye el levantamiento de la información de los procesos de ventas recopilando la información del cliente y planificar qué desarrollar dentro del E-Commerce, también se observan, describen y analizan detalladamente las diferentes tecnologías y agentes que intervienen durante este proceso. Como resultado de esta fase se consiguen las historias de usuario y un diseño del aplicativo.

4.2.2 Gestión de backlog.

Una vez definido y conociendo los objetivos o requisitos priorizados, es decir, la importancia o relevancia de poder iniciar el desarrollo del E-Commerce, se dividen las tareas por orden de prioridad para cumplir con las historias de cada uno de los usuarios. Al final de esta fase, se crea un backlog organizado por prioridad y estimaciones en base a equipos de trabajo.

4.2.3 Sprint Planning Meeting.

Al terminar la gestión del backlog se inicia la fase Sprint Planning Meeting, donde se organiza y agrupando las tareas a realizar para resolver las historias de usuario. Organizandó esta información y entendiendo la capacidad de trabajo, se definen los tiempos de entrega y se irá actualizando en cada Sprint

4.2.4 Ejecución de sprint.

En esta fase comienza el proceso de desarrollo, comenzando con el primer sprint que son las historias de usuario más valiosos, desarrollando así la Arquitectura o la base del proyecto. Al iniciar el Spring se socializa las historias del usuario con el cliente y este tendrá que esperar al terminar el Spring. una vez completado, se entrega a producción y se continúa con el próximo sprint.

4.2.5 Inspección e iteración.

Después de completar cada sprint, el equipo de trabajo realiza una reunión de cierre para entregarlo al cliente. En este punto, el cliente firma un documento de aprobación y, en su caso, se identifican oportunidades de mejora la entrega.

Una vez realizado el entregable con el cliente, se realiza una reunión con el equipo de trabajo, se hace una retrospectiva el cual se mencionan los puntos positivos y negativos durante el sprint y se hace un compromiso frente a ellos para el próximo sprint.

CAPITULO 5: DESARROLLO DE LA PROPUESTA

5.1 Planificación del proyecto

Como primera etapa, se levanta la información para conocer el alcance del proyecto, en este caso el MVP (producto mínimo viable). La información recopilada en esta etapa da inicio a la ejecución del proyecto.

5.1.1 Análisis técnico y de requerimientos para la empresa ASOPROCAFENPA.

Presentar los requisitos o necesidades del cliente ayuda a analizar mejor y comprender en detalle lo que se quiere o necesita en el E-Commerce, esto es al comprender las necesidades del cliente, la forma en que actualmente está ejecutando el proceso y ejecutando cada requisito resultante de esas necesidades.

Para definir los requisitos se propuso un formato basado en los siguientes módulos o funciones principales:

- **Inicio de sesión:** Hace referencia al inicio de sesión en el sistema al administrador y acceso a la tienda de los clientes.
- **Usuarios:** Se refiere a la gestión de usuarios en el panel de administración.
- **Perfiles:** Es para crear y modificar perfiles en el panel de administración y en la aplicación en la tienda.
- **Catálogos:** Es la creación y gestión de categorías con las que trabajará la tienda.
- **Productos:** Este módulo se encarga de gestionar los productos, desde su creación hasta su modificación para hacerlos llamativos en la tienda.
- **Pedidos:** Es una función para recibir y gestionar los pedidos de los clientes.
- **Condiciones de envío:** Es el módulo encargado de gestionar los gastos de envío de los pedidos.

- **Formas de pago:** En este módulo, administra los métodos de pago que estarán habilitados en la tienda.

5.1.1.1 Requerimientos Funcionales.

N°	Descripción	Prioridad
Administración de usuarios		
RF_1	El administrador del sistema podrá crear, editar e inactivar usuarios.	
RF_2	El usuario deberá ingresar al panel administrativo con un usuario y contraseña.	
RF_3	El sistema debe permitir al usuario cambiar la contraseña.	
RF_4	El sistema debe permitir al usuario cambiar la contraseña.	
RF_5	El usuario podrá actualizar su información personal.	
RF_6	El usuario podrá acceder a la tienda sin necesidad de tener un usuario registrado	
Perfiles		
RF_7	El sistema debe permitir a los administradores crear, editar, eliminar y listar perfiles.	
RF_8	El sistema debe permitir al administrador asignar usuarios a los perfiles.	
Categorías		
RF_9	El administrador del sistema debe poder crear, editar, listar y eliminar categorías para ingresar los productos en la aplicación.	Alta
RF_10	El sistema debe mostrar en la tienda las categorías en estado activo.	Alta
Productos		
RF_11	El administrador del sistema debe poder crear registros de nuevos productos en la aplicación	Alta

RF_12	El administrador podrá visualizar todos los productos listados por categorías, además de visualizar las características de cada producto.	Media
RF_13	El cliente podrá navegar en la tienda por categorías o catálogos y visualizar los detalles de cada producto.	Alta
Pedidos		
RF_14	El cliente podrá agregar los productos al carrito de compras y realizar el pedido en la tienda.	Alta
RF_15	El cliente podrá realizar cambios y devoluciones de su pedido siempre y cuando este cumpla con las políticas establecidas en la tienda.	Alta
RF_16	El cliente podrá escoger entre las opciones de pago habilitadas para el para terminar el pedido y poder ser procesado, las modalidades de pago son definidas para cada cliente por el administrador.	Alta
RF_17	El administrador podrá ver los pagos que han realizado los clientes, así mismo observar los registros en el sistema.	Alta
RF_18	El cliente podrá ver desde cualquier dispositivo que se encuentre logueado su carrito si aún no ha completado la compra	Media

Tabla 4. Fuente propia

5.1.1.2 Requerimientos No funcionales.

Aspectos	Descripción	N°
Seguridad	El sistema debe permitir la gestión de privilegios de los usuarios.	RNF_01
	El sistema debe proveer las funciones de autenticación tanto para el panel administrativo como para la tienda.	RNF_02
	El usuario solo podrá ingresar a los módulos que tenga permisos en el panel administrativo.	RNF_03
Escalabilidad	El sistema debe tener un sistema de monitoreo de recursos.	RNF_04
	La arquitectura del sistema debe ser compatible con infraestructuras escalables en nube.	RNF_05
	El sistema debe poder escalarse por medio de plugins para ir incrementando sus funcionalidades sin necesidad de afectar las ya existentes.	RNF_06
Disponibilidad	El sistema debe estar en la capacidad de dar respuesta a un alto tráfico de usuarios.	RNF_07
	El sistema debe estar disponible 24/7.	RNF_08
Usabilidad	El sistema debe ser intuitivo y amigable para el usuario.	RNF_09
	La interfaz debe ser responsive y construida con los parámetros de usabilidad para garantizar la experiencia de usuario.	RNF_10

Tabla 5. Fuente propia

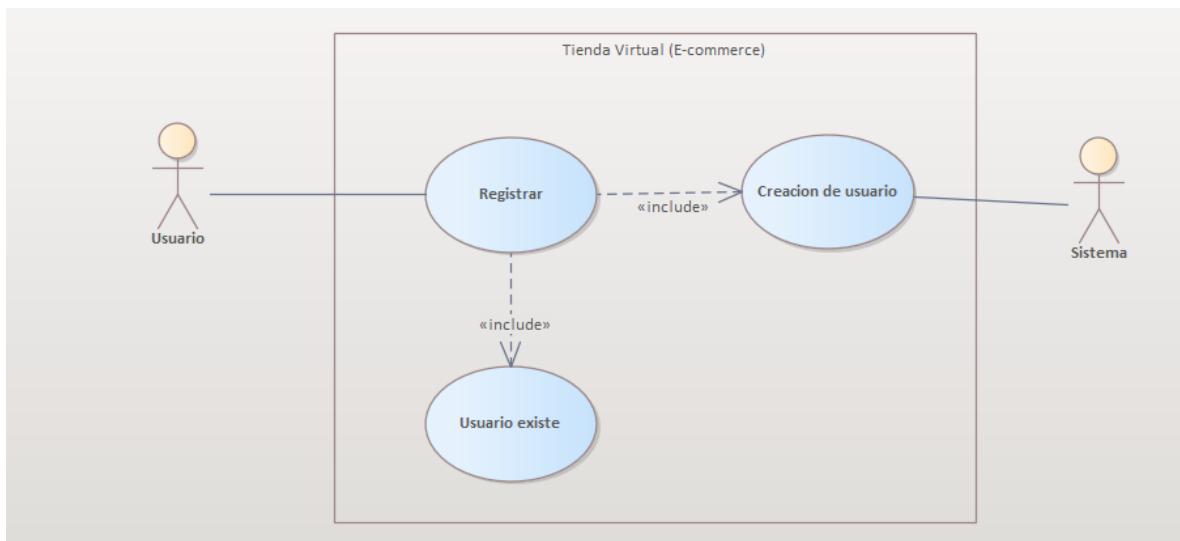
5.1.2. Diagramas de casos de uso

5.1.2.1. Casos de uso registrarse.

Nombre	Registrarse
Actores	Usuarios no autenticados
Descripción	Los clientes se registran en el sistema para comprar productos.
Precondiciones	No estar registrado con el mismo correo electrónico con anterioridad.
Flujo normal	1. Acceder al link del sistema en https://www.asoprocafenpa.com/ 2. Dar clic en "iniciar sesión" 3. Diligenciar el formulario de registro rápido 4. Dar clic en el botón "Registrarme"
Flujo alternativo	En caso que el correo ya este registrado: El sistema notifica que ya el usuario ya existe
Postcondiciones	Registra el usuario en el sistema

Tabla 6. Fuente propia

Ilustración 5. Caso de uso registrarse



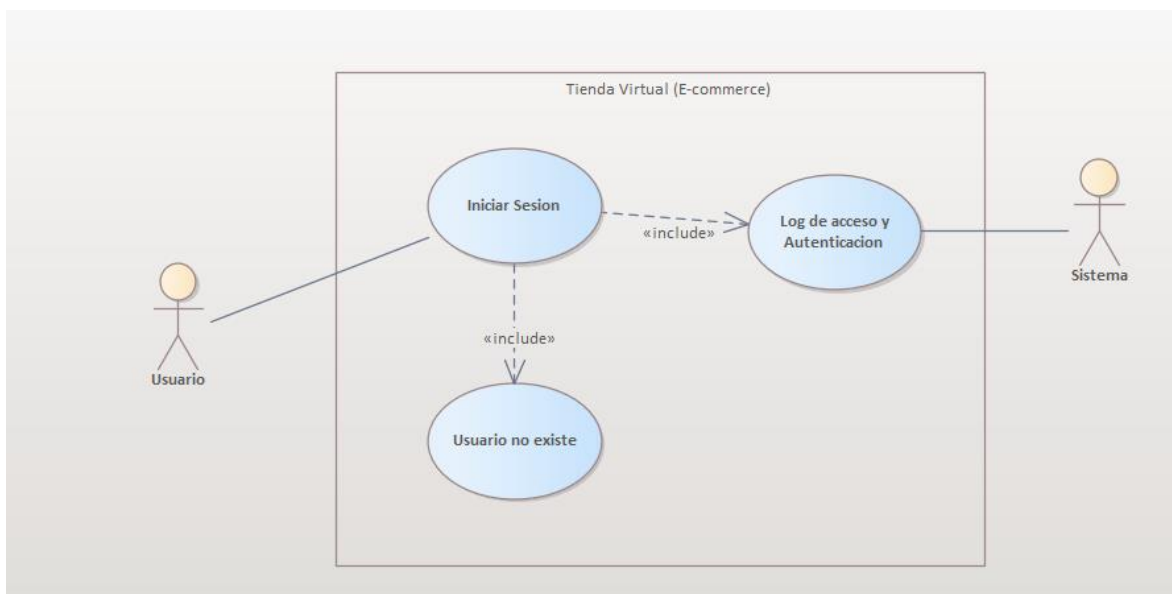
Fuente: Elaboración Propia

5.1.2.2. Caso de uso iniciar sesión.

Nombre	Iniciar sesión
Actores	Usuarios autenticados
Descripción	Los clientes se identifican en el sistema para una mejor experiencia de usuario.
Precondiciones	Estar registrado con el mismo correo electrónico con anterioridad.
Flujo normal	1. Acceder al link del sistema en https://www.asoprocafenpa.com/ 2. Dar clic en "iniciar sesión" 3. Diligenciar el formulario de iniciar sesión. 4. Dar clic en el botón "Iniciar sesión"
Flujo alternativo	En caso que el correo no esté registrado: El sistema notifica que el usuario no existe.
Postcondiciones	Guarda registro de acceso y autentica el usuario.

Tabla 7. Fuente propia

Ilustración 6. Caso de uso iniciar sesión



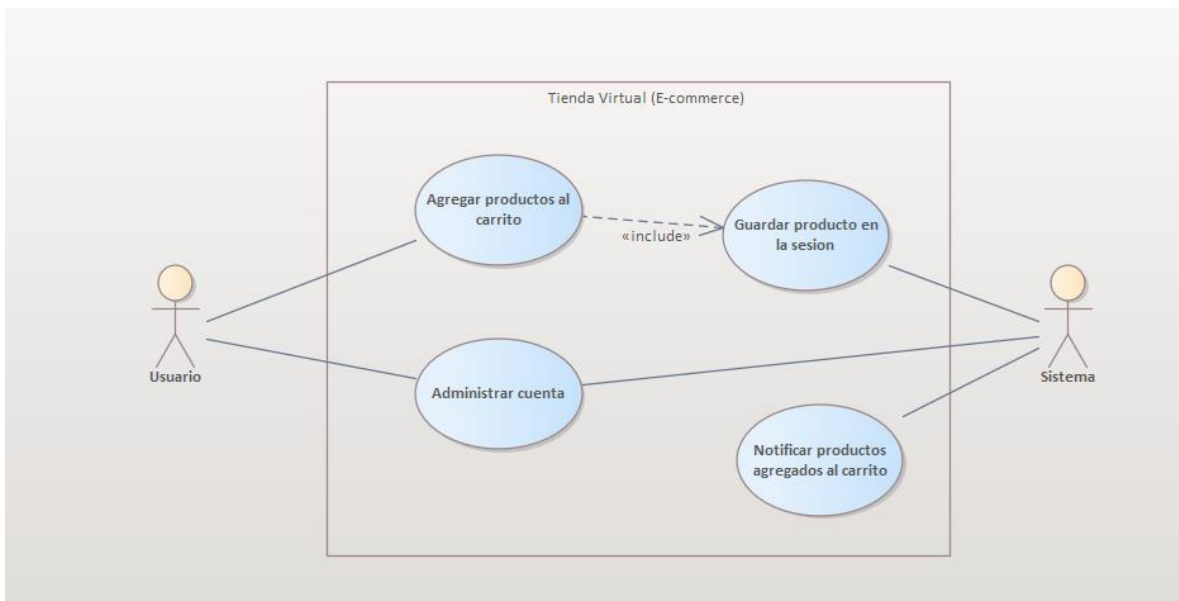
Fuente: Elaboración Propia

5.1.2.3. Caso de uso agregar al carrito.

Nombre	Agregar productos al carrito.
Actores	Usuarios autenticados
Descripción	Se agregan productos para no perderlos y posteriormente proceder al pago.
Precondiciones	Estar autenticado en el sistema
Flujo normal	1. Acceder al link del sistema en https://www.asoprocafenpa.com/ 2. Dar clic sobre el botón “ver productos” o “tienda” 3. Navegar por las categorías de la "Tienda Virtual" 3. Dar clic sobre el botón “agregar al carrito”
Flujo alternativo	Si el producto no está en el carrito: Notifica “se agregó el producto al carrito”.
Postcondiciones	El sistema asigna al carrito los productos añadidos.

Tabla 8. Fuente propia.

Ilustración 7. Caso de uso Agregar al carrito



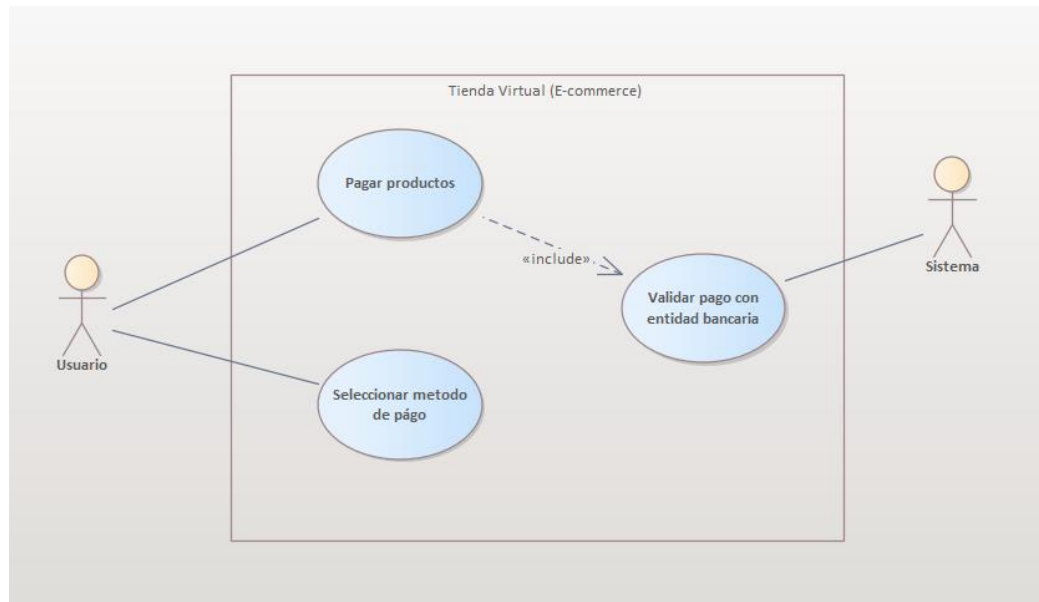
Fuente: Elaboración Propia

5.1.2.4. Caso de uso pagar productos.

Nombre	Pagar productos
Actores	Usuarios autenticados
Descripción	Se genera un pago y una factura al correo para despachar los productos ofrecidos por la empresa.
Precondiciones	• Estar autenticado en el sistema • Tener por lo menos un producto agregado en el carrito
Flujo normal	1. Acceder al link del sistema en https://www.asoprocafenpa.com/ 2. Dar clic sobre el botón “ver productos” o “tienda”. 3. Navegar por las categorías de la "Tienda Virtual”. 3. Dar clic sobre el botón “agregar al carrito. 4. Dar clic en el icono del carrito. 5. Dar clic en el botón “pagar ahora”. 6. Seleccionar método de pago. 7. Validar pago.
Flujo alternativo	Si el pago es rechazado o declinado: • El carrito no queda vacío. • Genera un registro de pago con el estado correspondiente.
Postcondiciones	El sistema genera un confirma el pago y genera la factura de compra al correo para así despachar el producto.

Tabla 9. Fuente propia.

Ilustración 8. Caso de uso pagar producto



Fuente: Elaboración Propia

5.1.2. Estudio del framework Angular para desarrollo del E-Commerce

5.1.2.1. Introducción

Angular es un framework de código abierto desarrollado por Google (Misko Hevery, Adam Abrons), para facilitar el diseño y desarrollo de aplicaciones web de estilo Single Page Application (SPA) y Progressive Web App (PWA) para que sean eficientes e igualmente sofisticadas.

El enlace de datos y la inyección de dependencia aumentan el impacto de un programa al reducir el número de líneas en su código, todo el procesamiento de comandos y datos tiene lugar dentro del navegador.

Angular es un framework de JavaScript del lado del cliente, no es un sistema de plantillas simple,

La razón se debe al "enlace de datos bidireccional", un enlace de datos bidireccional, que tiene y es funcionamiento automático. Las plantillas se compilan en el navegador y el pasos de ejecución como resultado es una vista en tiempo real. Los desarrolladores ya no necesitan que sincronice a menudo la vista con el modelo y viceversa.

Además, no hay necesidad de omitir a jQueryUI o Bootstrap. Angular es muy efectivo y funciona bien con bibliotecas de componentes de terceros. Ayuda a gestionar los llamados Solicitud de una sola página (SPA). (Angular, s. f.) Angular permite el uso de cualquier tipo de backend, incluidos Java, Python, Ruby, C#, etc.

Al usar Angular, es necesario usar una forma de comunicación de ida y vuelta. Con solicitud de servidor, HTTP XML o JSON.

5.1.2.2. Cronología de Angular

Angular se creó en 2009 como parte de un producto comercial llamado GetAngular. Poco después, Misko Hevery, uno de los ingenieros que crearon GetAngular, en pocas semanas logro recuperar una aplicación web de 17.000 líneas de código usando GetAngular que normalmente tarda 6 meses, este proyecto logro reducir a unas 1000líneas de código de una app, Google al ver esta gran hazaña hizo que patrocinara este proyecto. (AngularZone, 2015)

5.1.2.3. Logo Angular

Ilustración 9. Logotipo a todo color Angular



Fuente: (Angular, s. f.)

5.1.2.4. Conceptos Generales

La siguiente tabla se listarán los conceptos más importantes de Angular, según (AngularJS, 2016)

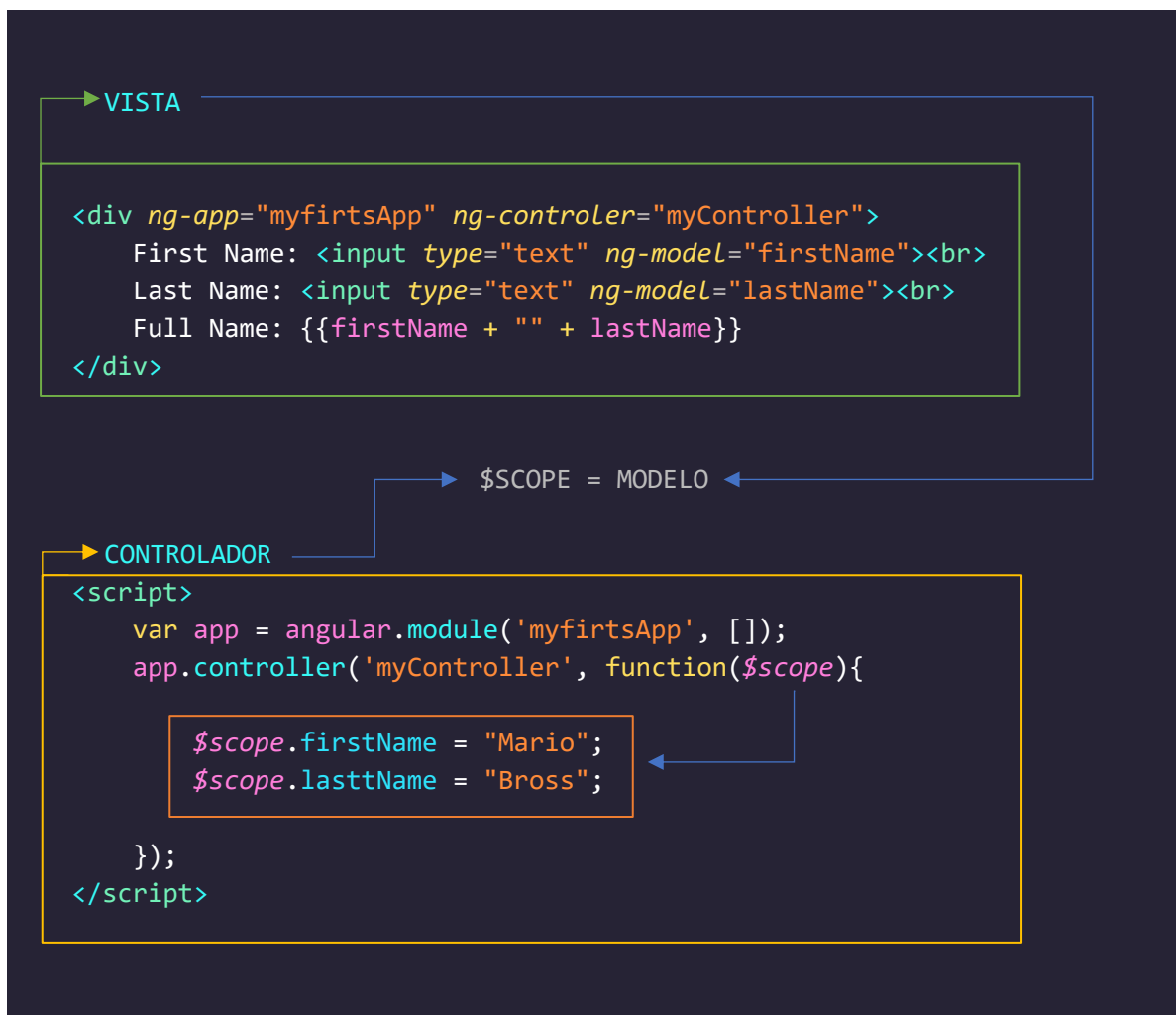
Concepto	Descripción
Template	HTML con marcado adicional
Directivas	HTML extendido con atributos y elementos personalizados
Model	Model Datos que se muestran al usuario en la vista
Scope	Aquí se almacena el modelo para que los controladores, directivas y expresiones puedan acceder a él
Expressions	Variables de acceso y funciones del ámbito
Compilador	Analiza la plantilla y crea una instancia, directivas y expresiones
Filtro	Formatea el valor de una expresión para la visualización al usuario
View	Lo que ve el usuario (DOM)
Data Binding	Datos de sincronización entre el modelo y la vista
Controller	Lógica de negocio detrás de las vistas
Dependency	Patrón de diseño de como los componentes se apoderan de sus dependencias
Injector	Contenedor de inyección de dependencias
Module	Contenedor para las diferentes partes de una aplicación, controladores, servicios, filtros, directivas que configura el inyector
Service	Lógica de negocio reutilizable independiente de vistas

Tabla 10. Conceptos importantes Angular

5.1.2.5. Características de AngularJS

5.1.2.5.1. \$scope

\$scope es uno de los puntos fundamentales de AngularJS, en el que se tiene Acceso a variables y métodos que vinculan controladores a vistas. me refiero a todo el controlador debe tener un alcance de \$scope para permitirle interactuar con la vista en la Arquitectura Modelo – Vista – Controlador. A continuación, se visualiza un ejemplo del \$scope funcionando entre la vista y el controlador.



5.1.2.5.2. Data Binding

"Data Link" uno de los puntos sin duda más fuertes de Angular

Data Binding vincula de alguna manera los datos de \$scope con lo que mostramos en HTML Automático, producido en ambos sentidos.

- One-way binding:

Los datos solo van desde \$scope a la parte de visualización (modelo a vista).

Se logra mediante el uso de sintaxis de doble llave.

```
{{datos}}
```

Si se cambian los datos del modelo (\$scope), la vista se actualizará automáticamente

- Two-way binding:

Los datos desde \$scope no solo fluyen hasta la vista, sino también viceversa implementado por la directiva ngModel

```
<input type="text" ng-model="misdatos">
```

Cuando el modelo es cambiado, los datos en el cuadro de texto también son afectados y cambian, Si el usuario cambia el valor del cuadro de texto, \$scope se actualiza automáticamente. A continuación, se mostrará como funciona los enlaces de bidireccionalmente de Angular.

```
<div ng-app>
  <input type="text" ng-model="valor">      <!-- Enlace Simple-->
  {{ valor}}      <!-- Enlace Simple-->
</div>
```

5.1.2.5.3. Directivas

Son etiquetas dentro del DOM (atributos, comentarios o clases CSS) que notifican al compilador de HTML angular (\$compile) un comportamiento adicional para el DOM o incluso puede modificar el DOM y su Hijos. Así es como se extiende HTML, lo que le permite agregar código.

- Tipos de Directivas

Las directivas se pueden basar en nombres de atributos, elementos, clases, entre otras, como lo son comentarios, que se pueden proporcionar nombres de atributos, etiquetas o nombre de tipos de atributos. A continuación, daré un ejemplo de tipos de directivas.

```
<my-dir></my-dir>
<span my-dir="exp"></span>
<!-- directive: my-dir exp -->
<span class="my-dir: exp;"></span>
```

Aunque se puede utilizar de maneras distintas, pero es preferible utilizar las directivas en el nombre de la etiqueta y atributos.

- Principales directivas de Angular.

La tabla 11 lista las directivas nativas de Angular, según (AngularJS, 2016).

Directiva	Descripción
<i>ngApp (ng-app)</i>	Permite auto arrancar una aplicación Angular, indica el elemento raíz, se coloca como atributo en la etiqueta que quieres que sea la raíz de la aplicación.
<i>ngController (ng-controller)</i>	Permite indicarle a la vista donde trabajará nuestro controlador y enlazar un \$scope, todo modelo que este dentro del ámbito de la directiva podrá ser accedido desde el controlador asignado
<i>ngModel (ng-model)</i>	Representa el modelo o dato, permite obtener la información ingresada por el usuario en algún elemento del formulario, basta con asociarle un modelo y éste podrá ser accedido tanto en el controlador como la vista mediante el nombre del modelo
<i>ngClick (ng-click)</i>	Trabaja relacionado al evento click, se le puede asociar alguna funcionalidad en cuanto el usuario haga click sobre algún elemento
<i>ngInit (ng-init)</i>	Permite evaluar una expresión en el scope donde se está trabajando.
<i>ngRepeat (ng-repeat)</i>	Permite iterar una colección de datos, generar un template por cada elemento de la colección y pintarlo en la vista, cada template o plantilla recibe su propio ámbito (\$scope).
<i>ngChange (ng-change)</i>	Detecta los cambios que se produzcan dentro de una etiqueta de entrada, inputs, checkbox.
<i>ngShow (ng-show) ngHide (ng-hide)</i>	Permiten mostrar y ocultar alguna parte de la vista.
<i>ngBind (ng-bind)</i>	Cumple la misma funcionalidad que las llaves {{ }}, sin embargo, ng-bind tiene una mejor performance en cuanto a tiempo

Tabla 11. Principales Directivas en Angular

El sistema de directivas personalizadas de Angular le permite crear nuevas directivas con menos líneas de código que la directiva original. Esto le ayuda a crear aplicaciones más modulares que se pueden estructurar de muchas maneras. A continuación, un ejemplo

```
<script>
  var app = angular.module('myfirtsModulo', []);
  app.controller('myController', function($scope){
    $scope.cliente = {
      nombre: "Mario",
      telefono: "3134587854"
    };
  });

  app.directive('miCliente', function(){
    return {
      templateUrl: 'cliente.html'
    }
  })
</script>
```

En la estructura vista estará representada de la siguiente manera

```
<body>
  <div ng-controller="myController">
    <div mi-cliente></div>
  </div>
</body>
```

5.1.2.5.4. Expresiones

Cualquier parte dentro de un documento HTML se puede mostrar datos. Pueden ser cadenas de texto, números, funciones o variables. La sintaxis de la expresión es: `{{Data}}`.

En Angular las expresiones validas son:

- Expresiones numéricas

```
<h1> Cantidad {{ 2 * 3 }} </h1>
```

- Expresiones de cadena de texto

```
<h1> Mi nombre es {{ "Mario" + "Bross" }}</h1>
```

- Expresiones con Arrays

```
<h1> Cliente {{ cliente[1] }}</h1>
```

- Expresiones con Objetos

```
<h1> Cliente {{ cliente.nombre }}</h1>
```

5.1.2.5.5. Modulos

Se necesitan contenedores separados para las diferentes partes de la aplicación. Esto ayuda a mantener el código fácilmente y escalarlo al dividir la funcionalidad y las características en partes separadas. Al hacer esto, el código es más manejable para trabajar y

más fácil de escalar cuando se agrega a otros proyectos. La declaración de un modulo se hace de la siguiente forma

```
angular.module('nombre_modulo', []);
```

Se debe hacer uso del método “module” nativo de Angular, se agrega un nombre que lo identificara, posteriormente se hace la declaración de las dependencias en los corchetes, si desea agregar más dependencias debe separarlos por comas, por ejemplo.

```
angular.module('nombre_modulo', ['dependencia1', 'dependencia2', 'depdN',]);
```

Se declara el módulo en el <head> con la etiqueta de <script>, dentro de la etiqueta debe importar el CDN de Angular para que funcione y debemos utilizar la directiva “ng-app” en la etiqueta <html>. A continuación, un ejemplo.

```
<html ng-app>
  <head>
    <title>My AngularJS App</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></script>
  </head>
  <body>
  </body>
</html>
```

5.1.2.5.6. Controladores

En general, los controladores se encargan de gestionar los eventos, permitiendo Desarrollar la lógica de las aplicaciones, además se enlaza la vista con el scope, teniendo en los datos un control total. El siguiente ejemplo es la declaración de un controlador.

```
var app = angular.module('nameApp', []);
app.controller('nameController', function($scope){
    $scope.cliente = "Mario";
});
```

Para el siguiente ejemplo se le asigna un nombre y agregamos las dependencias que necesitamos, el controlador siempre tiene asociado un \$scope, podemos agregarle otras dependencias de Angular. Para agregar más dependencias es importante separarlas por comas

```
var app = angular.module('nameApp', []);
app.controller('nameController', function($scope, depend1, depend2, depN){
    $scope.cliente = "Mario";
});
```

Estos controladores deben enlazarse a la vista mediante la directiva ng-controller

```
<body>
  <div ng-controller="nameController">
    <h1> cliente: {{cliente}}</h1>
  </div>
</body>
```

5.1.2.5.7. Servicios

Son objetos que nos brindan información, los servicios no interactúan con la página en sí, ellos interactúan con nuevos servicios, por ejemplo, un servidor que puede estar en otro hosting. Su característica principal es que solo hay una instancia, pero se puede usar varias veces.

Existen dos servicios principales service y factory

- Service: Después de que AngularJS crea una instancia de una clase, lo que se establece en ella es un servicio llamado “this” que es público y estará disponible para todos los controladores o directivas dispuestas a recibir el servicio.
- Factory: pasara una función para que devuelva el valor del servicio, este retorna selectivamente lo que pretendemos exponer. Esto le permite tener variables privadas que no podrá ser accedida desde los controladores.

Acontinucion un ejemplo de estos dos servicios

Ilustración 10. Uso de Factory y Service

```

1  var app = angular.module('miApp', []);
2  app.factory('miFirstFactory', function(){ /* creamos la factory*/
3      return {
4          datosFactory: [{
5              id: 0,
6              nombre: "mario",
7              edad: "12 años"
8          }]
9      }
10 });
11
12 app.service('miFirstService', function(){
13     this.datosServices = function(){ /* creamos el servicio*/
14         return [{
15             id: 0,
16             nombre: "mario",
17             edad: "12 años"
18         }]
19     }
20 })
21
22 /*Creamos el controlador e inyectamos tanto la factory, como el servicio */
23 app.controller("factoriController", function($scope, miFirstFactory, miFirstService){
24     /* devolvemos una factory con scope */
25     $scope.factorys = miFirstFactory.datosFactory;
26     /*devolvemos un servicio*/
27     $scope.miServices = miFirstService.datosServices();
28 })

```

Fuente: Elaboración Propia

Como se pudo observar anteriormente en el ejemplo los dos servicios hacen lo mismo, pero de distinta forma.

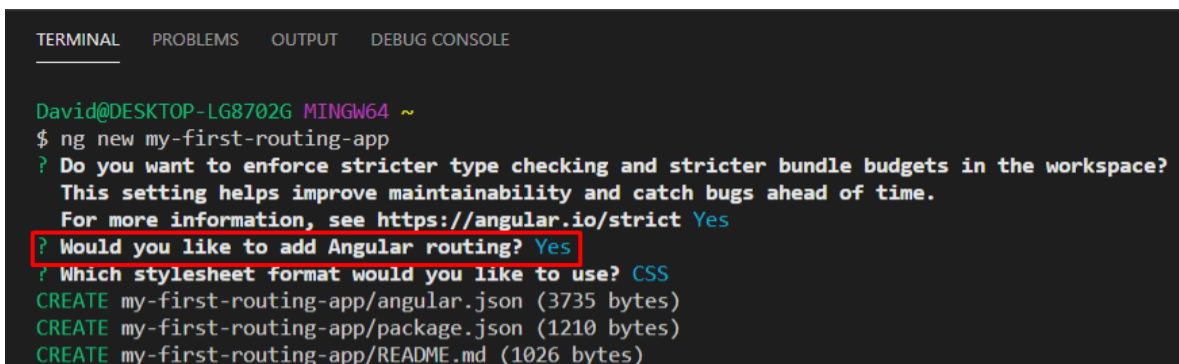
5.1.2.5.8. Routing

Para crear aplicaciones con muchas características o características complejas, el Framework Angular es el más adecuado. Por eso es importante poder acceder a diferentes partes de nuestra aplicación a través de enlaces web en los menús.

El routing, esta baso en la navegación URL's (rutas), este servicio los podemos integrar en nuestras aplicaciones web.

Cuando estamos creando un proyecto en Angular, nos pregunta el asistente de Angular CLI si queremos agregar routing a nuestro aplicativo, si la respuesta es Y (yes), se agrega automáticamente este servicio.

Ilustración 11. Routing



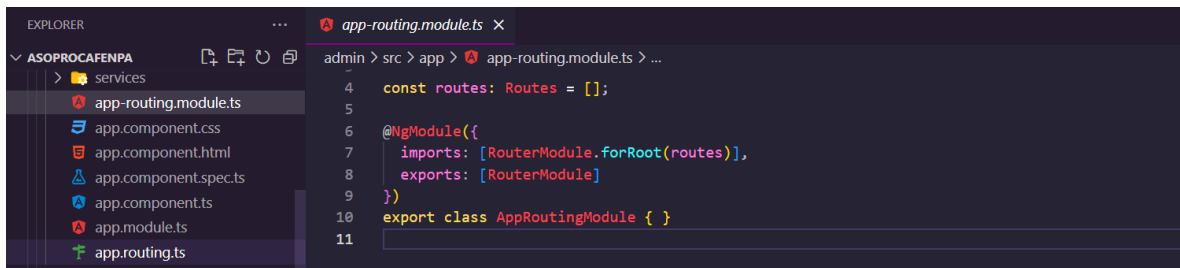
```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

David@DESKTOP-LG8702G MINGW64 ~
$ ng new my-first-routing-app
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE my-first-routing-app/angular.json (3735 bytes)
CREATE my-first-routing-app/package.json (1210 bytes)
CREATE my-first-routing-app/README.md (1026 bytes)
```

Fuente: (Java desde 0, 2020)

Si abrimos nuestro proyecto y buscamos el archivo, podemos ver que se agregó app-routing.module.ts y es simplemente un módulo con un solo export. Este archivo nos permitirá importarlo en el módulo principal para así trabajar con el Routing.

Ilustración 12. Estructura de app-routing.module.ts



Fuente: Elaboración Propia

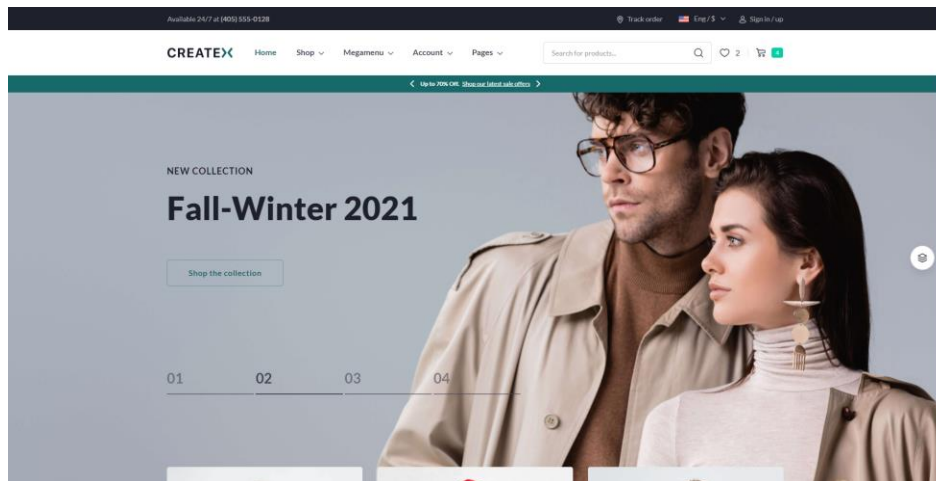
5.1.3. Desarrollo del E-Commerce

5.1.3.1. Diseño

Para el presente proyecto se tomó como base una plantilla comprada en Createx Studio, esta es una página que crea prediseños web y nos ahorra muchas horas de trabajo en la codificación de la estructura básica de una página web estática, estas son páginas que su contenido no cambia y su información se mantiene constante y paralizada. Es conveniente utilizarlas para paginas informativas, blogs o landing pages que se mostrara siempre igual para todos los usuarios.

Posteriormente, tomando la base anterior se convierte la página estática a una página web dinámica. Las páginas web dinámicas también conocidas como aplicaciones web, son páginas que su información se va actualizando con la interacción del usuario. Sus datos provienen de una base de datos, que actualizan y alteran de acuerdo con la información ingresada por cada usuario. Estos aplicativos varían significativamente según el usuario que los crea. Ejemplo, un sistema bancario, LinkedIn o un comercio electrónico (E-Commerce).

Ilustración 13. Landing page Createx – Home



Fuente: (Createx Studio, s. f.)

5.1.3.2. Software Frontend

El comercio electrónico (E-Commerce), está dividido en dos aplicativos, el primero es donde el cliente final va a interactuar con nuestro E-Commerce y el segundo aplicativo es donde se va a gestionar el E-Commerce por los administradores.

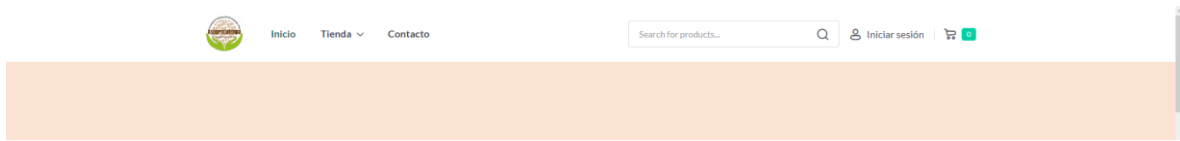
5.1.3.2.1. Cliente

Este proyecto tiene numerosas vistas y componentes atractivos que tienen que ser amigables con el cliente final. También incorpora una funcionalidad intuitiva a través de varios módulos de esta aplicación. Los siguientes son algunos de los componentes incluidos:

- **Disposición General**

En cualquier lugar de la página web, los usuarios verán el menú, el buscador general, el botón de inicio de sesión/registro, el carrito de compras, información de la empresa y un pie de página con los requisitos de la empresa y otras disposiciones generales.

Ilustración 14. Disposición General – Menú de navegación.



Fuente: Elaboración propia

Ilustración 15. Disposición General – Pie de página.



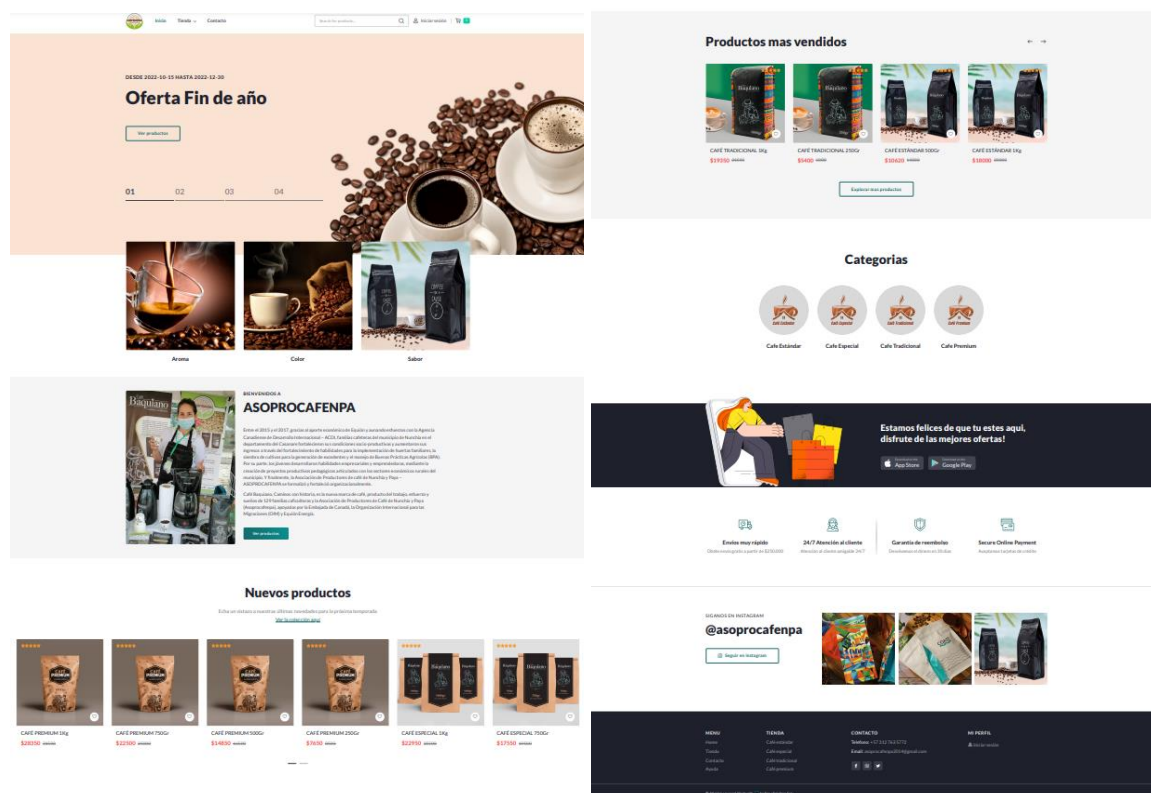
Fuente: Elaboración propia

- **Página principal (Inicio)**

Página principal del e-commerce es la más importante ya que es la primera impresión del usuario final, aquí se incluye la información más relevante de la empresa, se crearán banners de ofertas, avisos, entre otros.

La página de inicio debe ser una sección que guste mucho a los usuarios, tanto en función y diseño, porque es muy importante que los clientes finales sepan que es un sitio en el que puedes comprar e informarse. Este módulo debe tener imágenes y texto deslizante convincente, tanto para promocionar productos, dar a conocer la oferta, etc. Una Sección de Información comercial como datos de contacto y demás.

Ilustración 15. Página principal (Home)



Fuente: Elaboración propia

- **Inicio de sesión y registro**

El presente módulo está diseñado para dar a los clientes finales la opción de registrarse e iniciar sesión esto es necesario en el momento de la compra, para estos módulos están diseñados en una ventana y se puede acceder desde cualquier lugar del e-commerce.

Ilustración 16. Inicio de sesión y registro

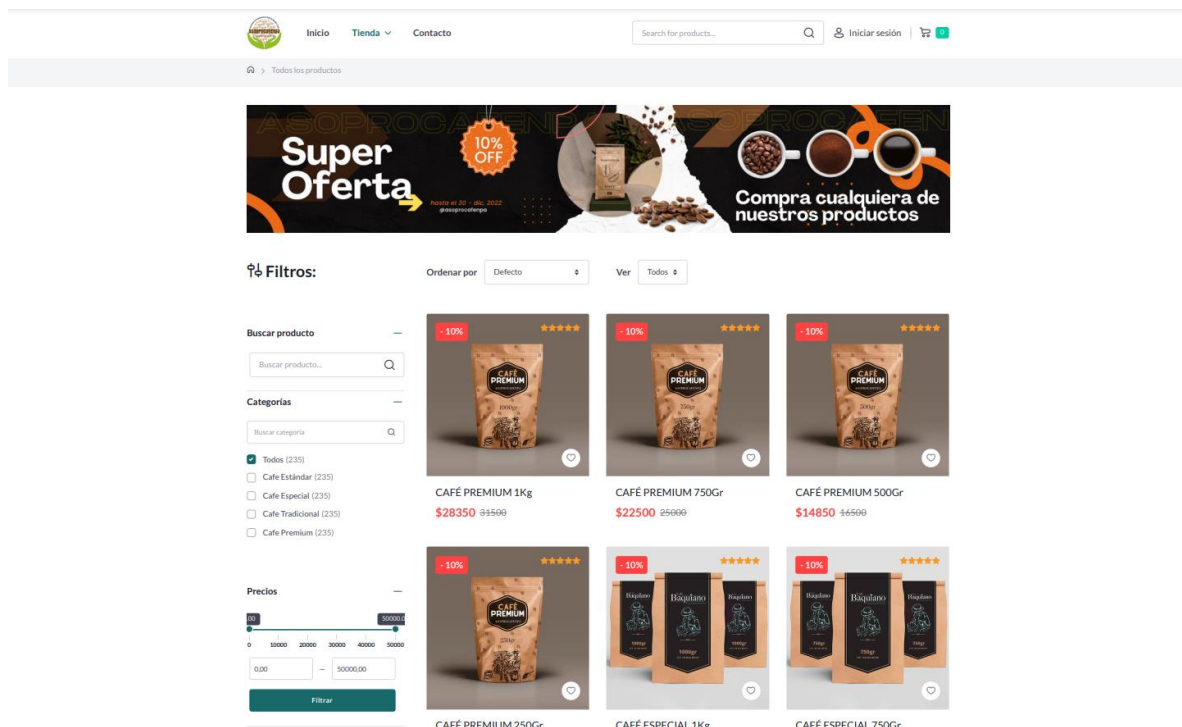
The screenshot displays a web interface for user authentication. At the top, there is a navigation bar with a logo, links for 'Inicio', 'Tienda', and 'Contacto', a search bar, and links for 'Iniciar sesión' and a shopping cart. Below the navigation bar is a 'Checkout' button. The main content area is divided into two columns. The left column is titled 'Iniciar Sesión' and includes a sub-header 'Inicie sesión en su cuenta utilizando las credenciales proporcionados durante el registro.' It features input fields for 'Correo electrónico' and 'Contraseña', a 'Recuérdame' checkbox, a link for 'Olvidó su contraseña?', and an 'Iniciar sesión' button. The right column is titled 'Registro rápido' with the sub-header 'Complete todos los campos requeridos.' It includes input fields for 'Nombres completos', 'Apellidos completos', 'Correo electrónico', and 'Contraseña', and a 'Registrarme' button. At the bottom, there is a dark footer with four sections: 'MENU' (Home, Tienda, Contacto, Ayuda), 'TIENDA' (Café estándar, Café especial, Café tradicional, Café premium), 'CONTACTO' (Teléfono: +57 312 763 5772, Email: asoprocafengs2014@gmail.com, social media icons), and 'MI PERFIL' (Iniciar sesión). A copyright notice '© All rights reserved. Made with By Nissa Solutions Ltd.' is at the very bottom.

Fuente: Elaboración propia

- **Listado de los productos**

Este módulo contiene una lista de todos los productos disponibles a la venta, adicionalmente se muestra un banner de oferta disponible, posteriormente vemos para ordenar y listar cuantos productos queremos ver, a la izquierda encontraremos un filtro: Buscar por nombre, categorías y rango de precios. Los usuarios finales pueden ver una lista de productos amigable y ordenada, cada producto debe tener un nombre, precio y un botón de agregar al carrito, clic sobre el producto para ver más detalles del producto.

Ilustración 17. Inicio de sesión y registro

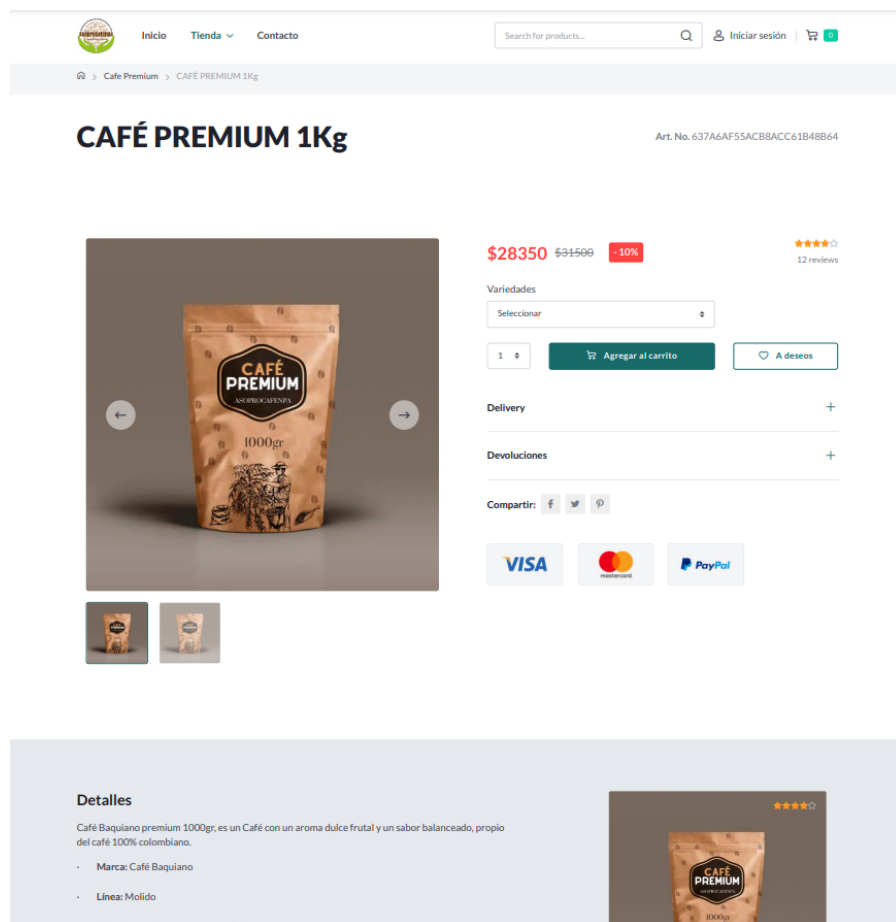


Fuente: Elaboración propia

- **Detalle del producto**

Este módulo muestra la descripción detallada del producto seleccionado, mostrando información relevante para que sea más claro saber lo que se va a comprar, la vista debe tener una gallería de imágenes del producto, nombre, descripción, id del producto, descuento si existe, variedades, cantidad, botón de agregar al carrito. Agregar a deseos, ver calificación y reseñas general del producto en la parte superior derecha, no está disponible la funcionalidad en la presente versión v1.0.

Ilustración 18. Detalle del producto



Fuente: Elaboración propia

- **Carrito de compras**

El carrito de compras contiene detalles más específicos sobre los productos agregados. En este módulo se muestren los productos listados con sus respectivas imágenes añadida por defecto, precio, cantidad, descuento si existe, botón eliminar producto del carrito, dirección de envío, subtotal, total, métodos de pago y caja para enviar mensaje adicional acerca de los productos. También se agregó una entrada para aplicar el cupón, el sistema debe validar el cupón como válido si existe y hacer el descuento correspondiente y el cálculo total.

Ilustración 19. Carrito de compras


[Inicio](#)
[Tienda](#)
[Contacto](#)

Jefferson
🛒

[🏠](#)
[🛒](#)
[📧](#)

[🏠](#)
[🛒](#)
[📧](#)

Informacion General

[Regresar a la tienda](#)

1. Tus productos

	CAFE PREMIUM 1kg	1	\$28350 descuento	Eliminar	Mover a 📌
	CAFE ESPECIAL 1kg	1	\$22950 descuento	Eliminar	Mover a 📌
	CAFE ESTÁNDAR 750gr	1	\$14400 descuento	Eliminar	Mover a 📌

2. Direccion de envío

Destinatario
Jefferson Walteros Sotaban

Numero de documento
1115692064

Numero contacto
3134630373

Dirección de entrega
Calle 3 #8-135 - Barrio el Humilladero - Pasaje Divino Niño

Ubicacion
Colombia - Norte de Santander - Pamplona

Código postal
2415

3. Metodos de envio

☐ Envío Express 1 a 3 días	\$15000
☐ Envío estandar 3 a 6 días	\$11000
☒ Envío Gratis 1 semana	\$0

4. Metodos de pago

☒ Credit card  

Numero de tarjeta

Expiración CVC

☐ PayPal 

☐ Cash on delivery

5. Información adicional (opcional)

Mensaje

Notas sobre su pedido, Notas especiales para la entrega.

Order totals

Subtotal: \$65700

Envío: \$0

Descuentos: -\$0

Total a pagar: \$65700

Completar orden

Aplicar cupón

Introduce el código de promoción

Aplicar

Fuente: Elaboración propia

- **Perfil usuario**

El sistema debe permitir que el usuario final llene y agregue otros campos requeridos al momento de la compra, en primer lugar, completar el formulario de “mi perfil”, posteriormente, agregar una dirección de envío y listar las direcciones creadas. Este módulo permitirá listar compras realizadas por cliente en la opción “Mis pedidos” en el menú lateral izquierdo. En la presente versión v1.0 no está disponible la funcionalidad visto recientemente y mis reseñas.

Ilustración 20. Perfil Usuario

The image displays three screenshots of a web application interface, likely for a coffee shop or similar business, showing the user profile, address management, and order details.

Screenshot 1: Mi Perfil (User Profile)

The user profile page shows the following information:

- Nombre:** Jhefferson
- Apellidos:** Walteros
- Correo electrónico:** wjhefferson@gmail.com
- Contraseña:** Nueva Contraseña
- Teléfono:** 3134630378
- N. de documento:** 1115695024
- Fecha de nacimiento:** dd/mm/aaaa
- Genero:** Masculino
- País:** Colombia

Screenshot 2: Nueva Dirección (New Address)

The new address form includes the following fields:

- Destinatario:** Nombre y apellidos completos
- N. de Documento:** Número de documento
- Código Postal:** Teléfono
- Dirección:** Dirección
- País:** Colombia
- Departamento:** Departamento
- Ciudad:** Ciudad
- Activar como principal:** ☐

Screenshot 3: Mis Direcciones (My Addresses)

The list of addresses shows one address:

- Lissy Walteros Contreras**
- Calle 1423-23 Barrio Plaza**
- Código postal: 2015**
- Colombia - Norte de Santander - Pamplona**

Screenshot 4: Mi pedido #6387F7F3F2A5856B64BC81F1 (My Order)

The order details page shows the following information:

- No Pedido:** #6387F7F3F2A5856B64BC81F1
- Trasacción No:** 2H807080F37820908
- Tipo de envío:** Envío Gratis
- Estado:** Procesando
- Fecha:** Nov 30, 2022
- Total:** \$1300

Order Details:

Producto	Precio	Cantidad	Subtotal
Café Especial 1kg	\$2500	1	\$2500
Café Premium 1kg	\$1500	1	\$1500

Nota:

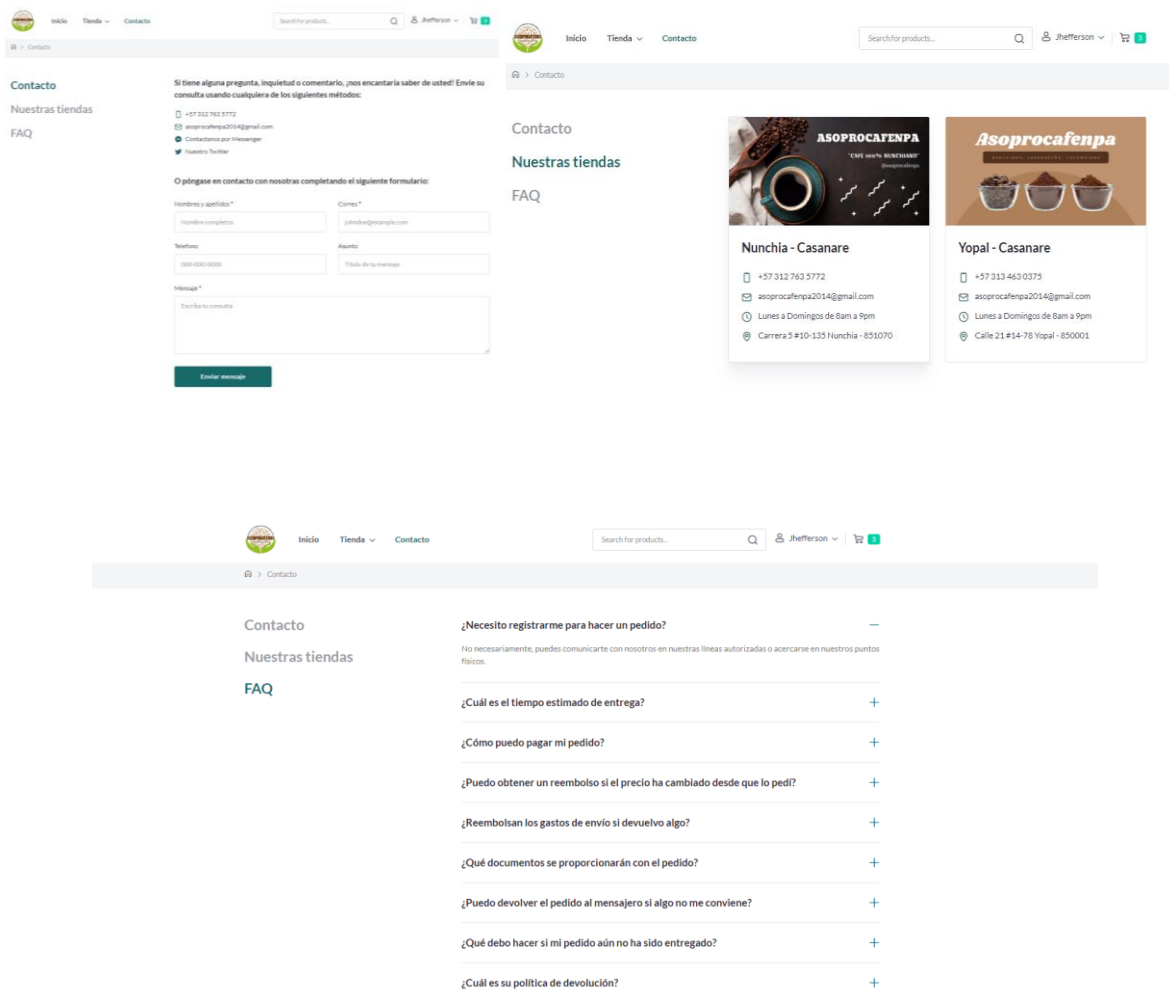
Descuento de producto: 50% Envío: \$0 Total: \$1300

Fuente: Elaboración propia

- **Contacto**

En modulo contacto es importante en el e-commerce por que crean un puente entre clientes finales hacia empresa, es importante que cuente con una buena interfaz de usuario. Aquí se visualiza un menú lateral izquierdo con la funcionalidad de mostrar información relevante de la empresa, junto con un formulario de contacto, dirección de puntos físicos de la tienda en la opción “Nuestras tiendas” y “Faq” lista las preguntas frecuentes de los clientes finales.

Ilustración 21. Modulo contacto



Fuente: Elaboración propia

5.1.3.2.2. Administrador

- **Inicio de sesión**

Este módulo incluye el acceso al sistema de gestión de comercio electrónico (E-Commerce) de tal forma que los usuarios con acceso serán únicamente aquellos previamente registrados manualmente por la base de datos.

Ilustración 22. Modulo Iniciar sesión

Inicio de sesión ASOPROCAFENPA

Inicie sesión en su cuenta utilizando el correo electrónico y la contraseña proporcionados por el administrador.

Correo electrónico

Su dirección de correo electrónico

Contraseña

Tu contraseña

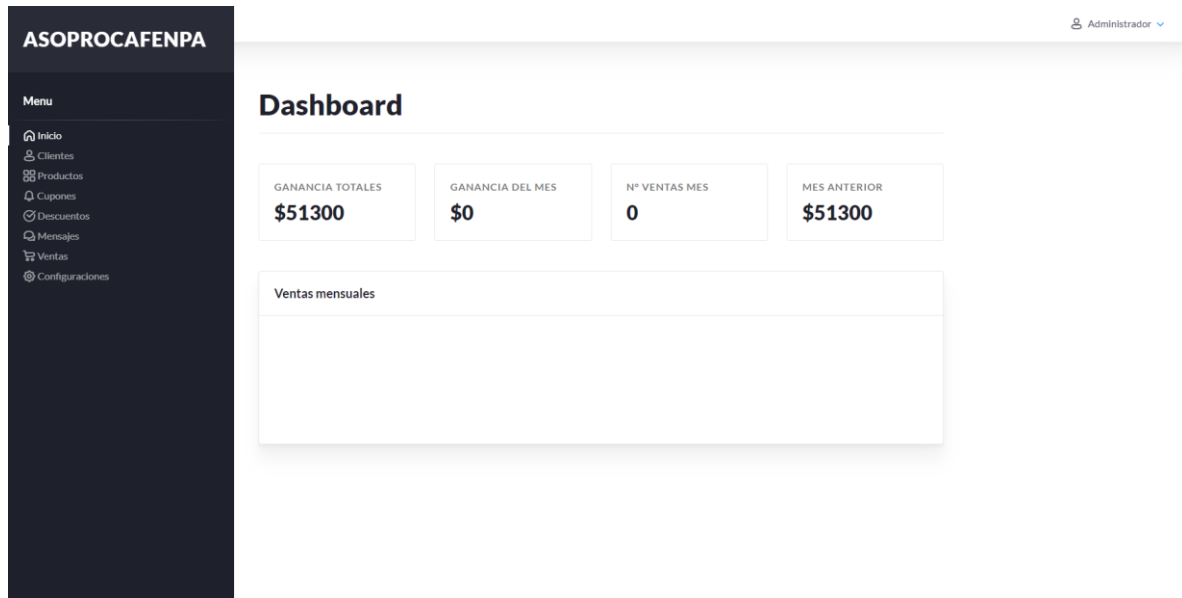
Ingresar

Fuente: Elaboración propia

- **Dashboard**

El presente modulo gestiona y analiza la información de las ventas generadas en el E-Commerce es fundamental para hacer un seguimiento del estado de la empresa, aquí lista las ganancias totales, ganancias del mes, N° de ventas del mes y total del mes anterior.

Ilustración 23. Dashboard Administrativo



Fuente: Elaboración propia

- **Gestor de usuarios**

El gestor de usuarios hace un listado de los usuarios registrados en el E-Commerce, con la finalidad de hacer un control con los clientes finales. El módulo permite filtrar, registrar, editar y eliminar usuarios.

Ilustración 24. Gestor de usuarios

ASOPROCAFENPA

Administrador

Clientes

Este módulo lista todos los clientes de la tienda. [Registrar nuevo >](#)

Filtro de clientes

Apellidos @ Correo

#	Nombres	Apellidos	Correo	Opciones
1	Vincent	Murray	vicentmurray@gmail.com	OPCIONES ▾
2	Jhefferson	Walteros	wjhefferson@gmail.com	OPCIONES ▾
3	Rosy	Cruz	rossy@gmail.com	OPCIONES ▾
4	Natalia	Sepulveda	natalia@gmail.com	OPCIONES ▾

1

Fuente: Elaboración propia

- **Productos**

Esta sección lista los productos que han sido previamente creados y listos para verse en el E-Commerce, teniendo funciones como registrar, editar, eliminar productos, filtrar, agregar inventarios, exportar a Excel el inventario, crear variedades y agregar galería de fotos.

Ilustración 25. Productos

ASOPROCAFENPA

Menu

- Inicio
- Clientes
- Productos**
- Cupones
- Descuentos
- Mensajes
- Ventas
- Configuraciones

Productos

Este módulo lista todos los Productos de la tienda. [Registrar nuevo >](#)

Filtro de Productos

Título del producto [Filtrar](#) [Borrar Filtro](#)

[Exportar Excel](#)

Título	Stock	Precio	Categoría	Ventas	OP
CAFÉ ESTÁNDAR 250Gr	100	5800	Cafe Estándar	0	OPCIONES ▾ Editar producto Eliminar producto Inventario Variedades Galería
CAFÉ ESTÁNDAR 500Gr	1500	11800	Cafe Estándar	0	
CAFÉ ESTÁNDAR 750Gr	1000	16000	Cafe Estándar	0	
CAFÉ ESTÁNDAR 1Kg	1000	20000	Cafe Estándar	0	OPCIONES ▾
CAFÉ TRADICIONAL 250Gr	1000	6000	Cafe Tradicional	0	OPCIONES ▾

Fuente: Elaboración propia

- **Cupones**

En esta sección se listan los cupones de descuentos creados para el E-Commerce, ya sea que estén disponibles o el límite cupones de canje haya terminado. Este módulo permite crear, editar y eliminar cupones.

Ilustración 26. Cupones

ASOPROCAFENPA

Administrador

Cupones

Este módulo lista todos los cupones de la tienda. [Registrar nuevo >](#)

Filtro de cupones

Buscar por código [Filtrar](#)

Código	Tipo	Valor	Límite	Opciones
MICUPO2023	valor fijo	1000	10 restantes	OPCIONES ▾
MICUPO2022	Porcentaje	50	10 restantes	OPCIONES ▾

« 1 »

Fuente: Elaboración propia

- **Descuentos**

En esta sección se listan los descuentos creados para el E-Commerce, ya sea que estén disponibles o la vigencia haya terminado. Este módulo permite registrar, editar, eliminar y el estado de los descuentos.

Ilustración 27. Descuentos

ASOPROCAFENPA

Menu

- Inicio
- Clientes
- Productos
- Cupones
- Descuentos
- Mensajes
- Ventas
- Configuraciones

Descuento

Este módulo lista todos los descuentos de la tienda. [Registrar nuevo >](#)

Filtro de descuento

Título del Descuento

Título	Descuento	Vigencia	Estado	OP
Oferta Fin de año	10%	(2022-10-15) - (2022-12-30)	Expirado	OPCIONES

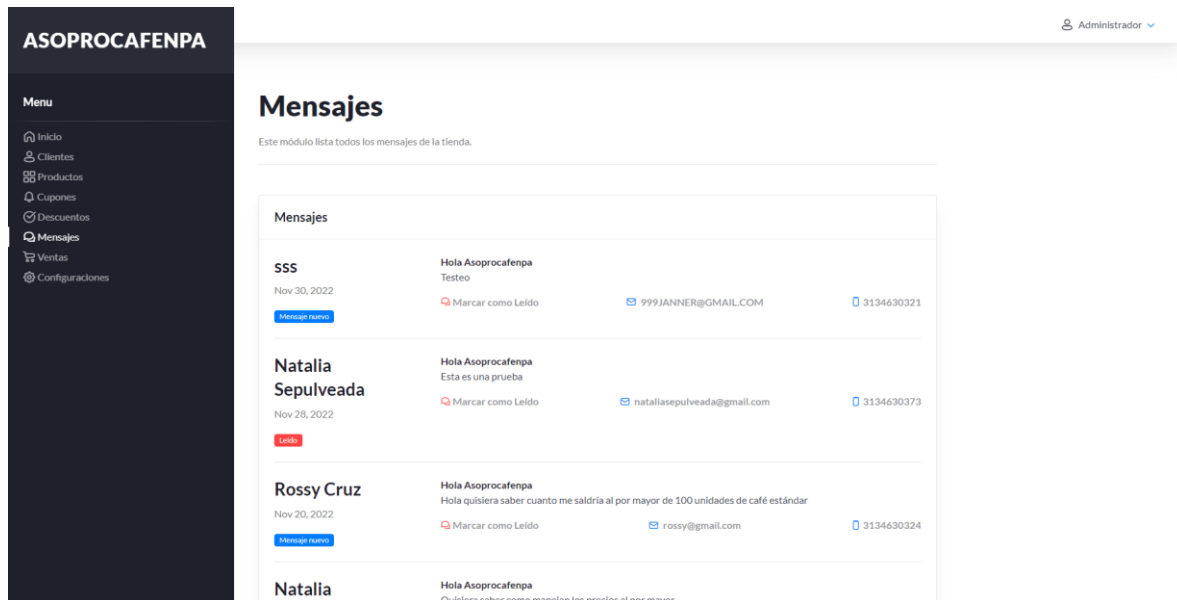
1

Fuente: Elaboración propia

- **Mensajes**

En esta sección se listan los mensajes enviados desde el formulario de contacto del E-Commerce.

Ilustración 28. Modulo Mensajes



Fuente: Elaboración propia

- **Ventas**

En esta sección se listan todas las ventas del E-Commerce ordenadas por fecha, Este módulo permite ver el cliente quien realizo la compra y ver los detalles.

Ilustración 29. Modulo ventas

ASOPROCAFENPA Administrador

Menu

- Inicio
- Clientes
- Productos
- Cupones
- Descuentos
- Mensajes
- Ventas**
- Configuraciones

Ventas

Este módulo lista todas las ventas de la tienda.

Filtro de ventas

Desde: dd/mm/aaaa Hasta: dd/mm/aaaa

Filtrar

Jhefferson Walteros **Transaccion**

#025-000005 Nov 30, 2022 Envío Gratis

\$51300

1

ASOPROCAFENPA

Menu

- Inicio
- Clientes
- Productos
- Cupones
- Descuentos
- Mensajes
- Ventas**
- Configuraciones

Ventas

Este módulo lista los detalles de las ventas de la tienda.

Detalle de la venta

Cliente:	Jhefferson	Precio envío:	\$0
No Pedido:	#6387F7F3F2A5856B64BC81F1	Transaccion No:	2HB07080F57820908
Tipo de envío:	Envío Gratis	Estado:	Procesando
Fecha:	Nov 30, 2022	Total	51300

Direccion de envío

Jhefferson Walteros
Sotaban
Calle 3 #8-135 - Barrio el Humilladero - Pasaje Divino Niño
Colombia - Norte de Santander - Pamplona
Codigo postal: 2415 [Direccion de envío](#)

Nota

	CAFE ESPECIAL 1Kg	Precio:	Cantidad:	Subtotal:
		\$25500	1	\$25500
	CAFE PREMIUM 1Kg	Precio:	Cantidad:	Subtotal:
		\$31500	1	\$31500
Descuento de productos: 10%		Envío:	\$0	Total:
				\$51300

[Regresar](#)

Fuente: Elaboración propia

- **Configuraciones**

En esta sección se hacen las configuraciones del panel administrativo y se crean y eliminan las categorías que tiene la empresa.

Ilustración 30. Configuraciones

ASOPROCAFENPA

Menu

- Inicio
- Clientes
- Productos
- Cupones
- Descuentos
- Mensajes
- Ventas
- Configuraciones**

Configuraciones

Este módulo se establece las configuraciones de la tienda.

Configuraciones

Nombre de la tienda: ASOPROCAFENPA

Facturación: 001 000001

Categorías: Título Icono

[Agregar](#)

Categoría	Icono	Eliminar
Cafe Estándar	col-bag	
Cafe Especial	col-layouts	
Cafe Tradicional	col-home	
Cafe Premium	col-star	

Logo: Arrastre una imagen

[Subir imagen](#)

[Guardar cambios](#)

Fuente: Elaboración propia

5.1.3.3. Software Backend

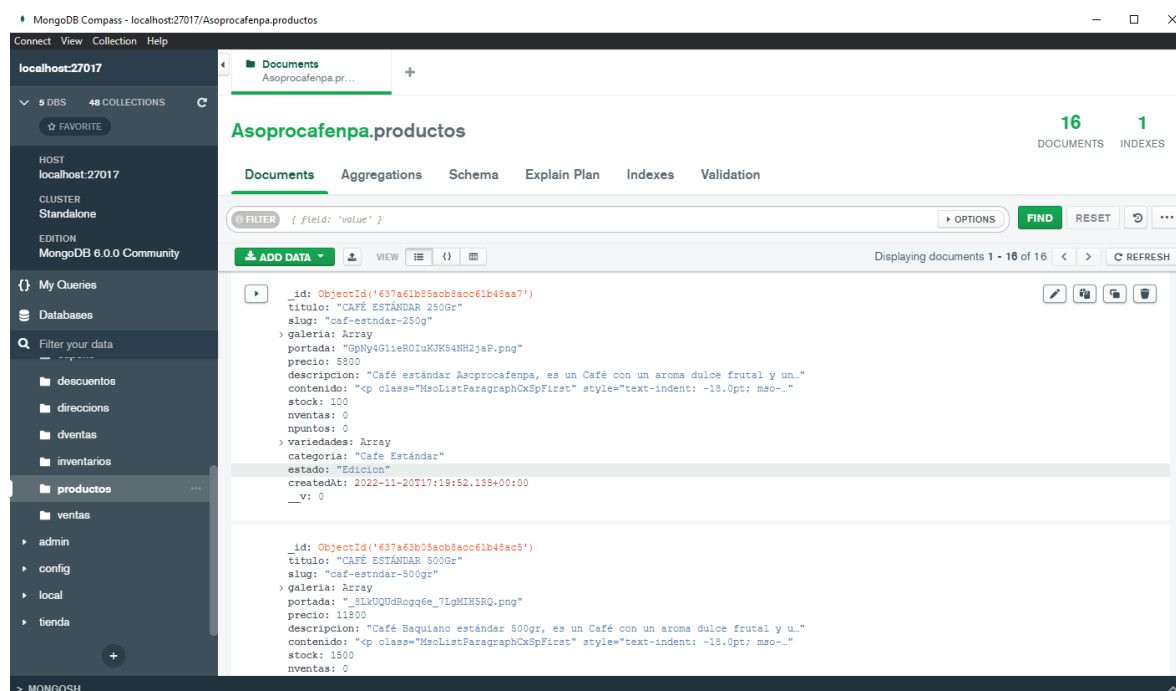
Para el desarrollo del backend fue construido en NodeJS y como gestor de base de datos se usó MongoDB una base de datos no relación que se almacenan usando el formato JSON, perfecta para almacenar grandes volúmenes de datos.

Por seguridad de la empresa ASOPROCAFENPA se ilustrará una base de datos teniendo previa autorización, ya que puede contener datos sensibles de usuarios.

- **Productos**

En la ilustración 31, se muestra una base de datos en formato JSON listando los productos actualmente creados en el E-Commerce.

Ilustración 31. Productos



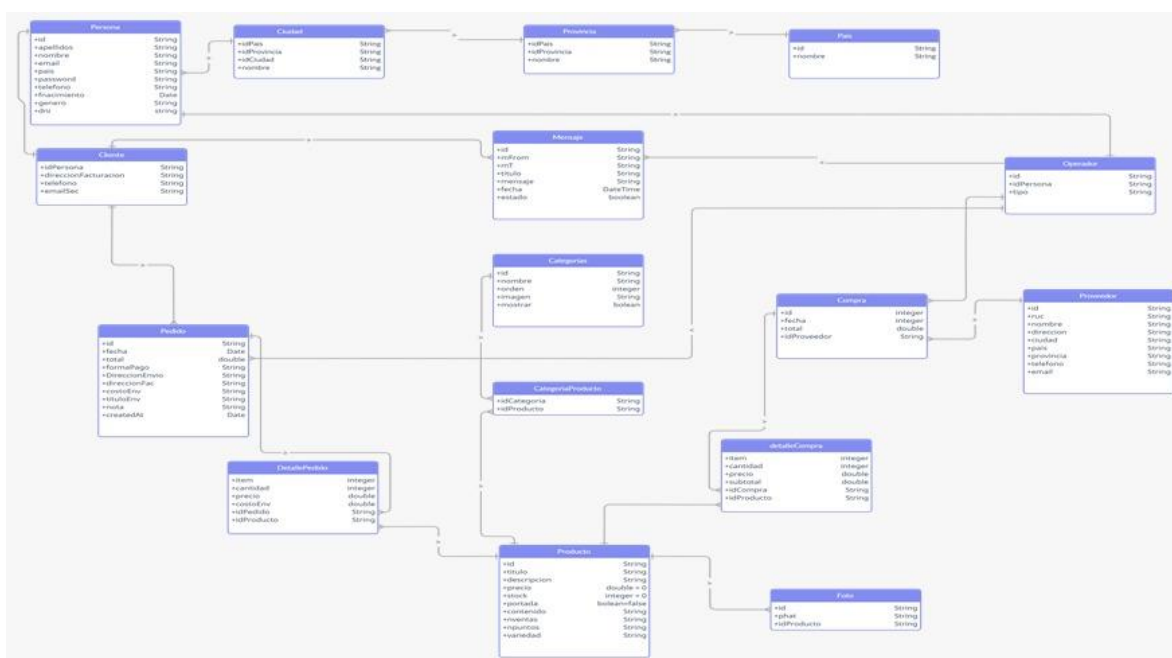
Fuente: Elaboración propia

5.1.3.3.1. Modelo de datos

El siguiente modelo de base de datos se utilizó para llevar a cabo el desarrollo del comercio electrónico basado en los requerimientos propuestos para el proyecto. Este es un modelo de bases de datos orientado a documentos, seleccionado para almacenar grandes cantidades de datos y a futuro no tener limitaciones para escalar el E-Commerce, así mismo, se utilizó para almacenar, recuperar y administrar los datos semiestructurados.

Esta base de datos documental almacena cada uno del registro y los datos relacionados en un mismo documento y están listo para ser consultados desde el gestor de bases de datos MongoDB.

Ilustración 32. Modelo base de datos



Fuente: Elaboración propia

5.1.3.4. Verificación de las buenas practicas

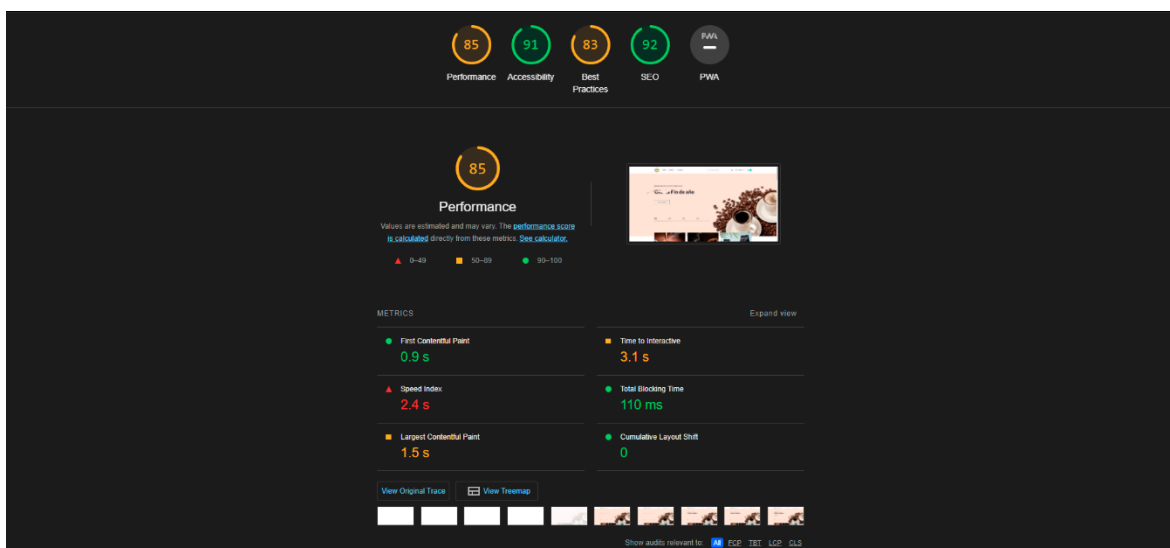
Las buenas prácticas de programación requieren el uso de varios principios y metodologías para crear, mantener e implementar software seguro.

En este proceso se hizo uso de la herramienta automatizada Google Lighthouse integrada en el navegador de Google Chrome para verificar las buenas prácticas, SEO, rendimiento y accesibilidad del E-Commerce.

- **Rendimiento**

En la ilustración 36, podemos ver que el rendimiento de la pagina tiene un buen desempeño a la hora de cargar el contenido, se puede mejorar optimizando imágenes, pero esto afectaría la calidad de ellas.

Ilustración 33. Prueba de rendimiento e-commerce

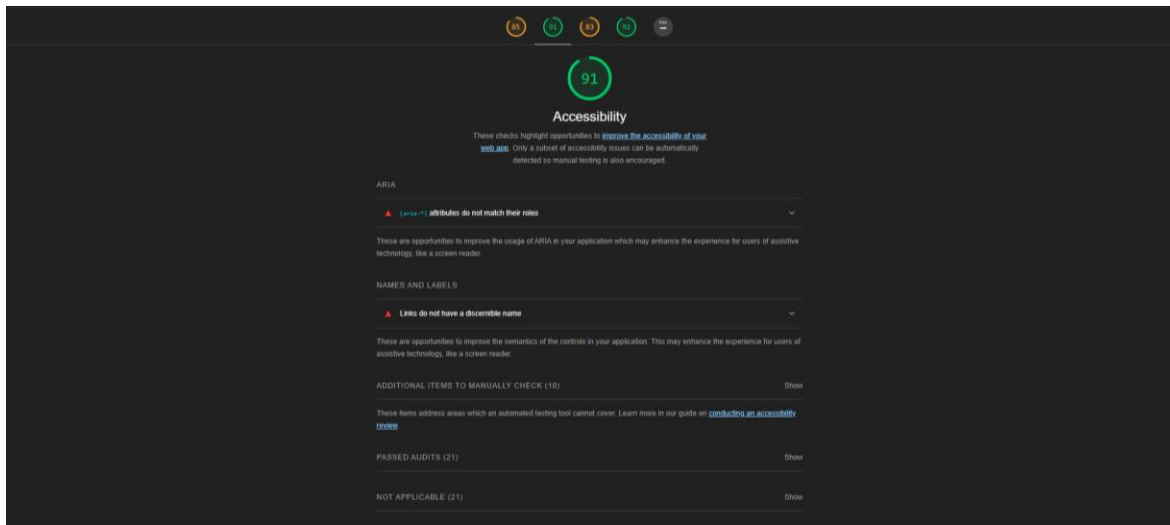


Fuente: Elaboración propia

- **Accesibilidad**

En accesibilidad la ilustración 37, registra que el E-Commerce es de fácil de uso para personas con limitaciones físicas, además que se adapta para gran mayoría de dispositivos móviles.

Ilustración 34. Prueba de accesibilidad e-commerce

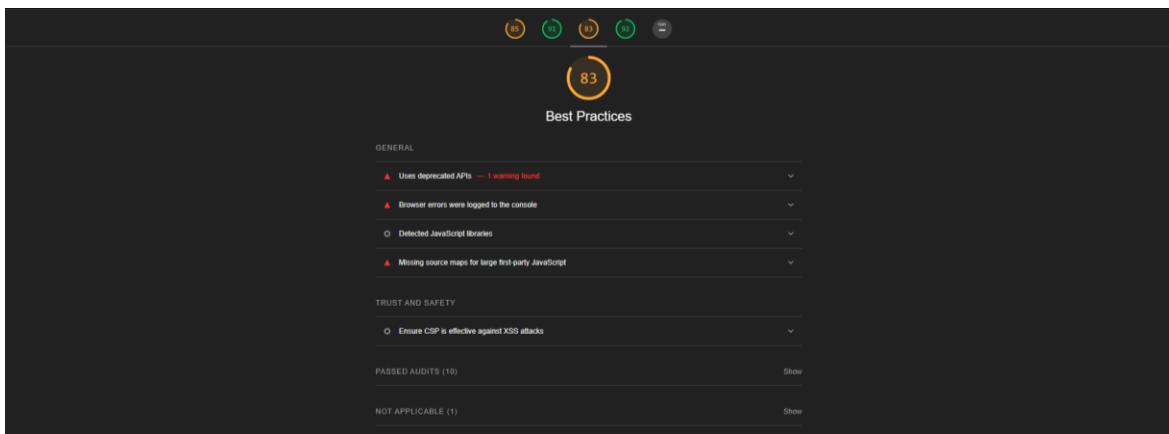


Fuente: Elaboración propia

- **Mejores Practicas**

En best practices Google Lighthouse analiza los aspectos de seguridad de la página web y si ha utilizado alguna codificación TLS, si el contenido viene de fuentes seguras o si los framework de JavaScript las puede ver confiables.

Ilustración 35. Buenas practicas

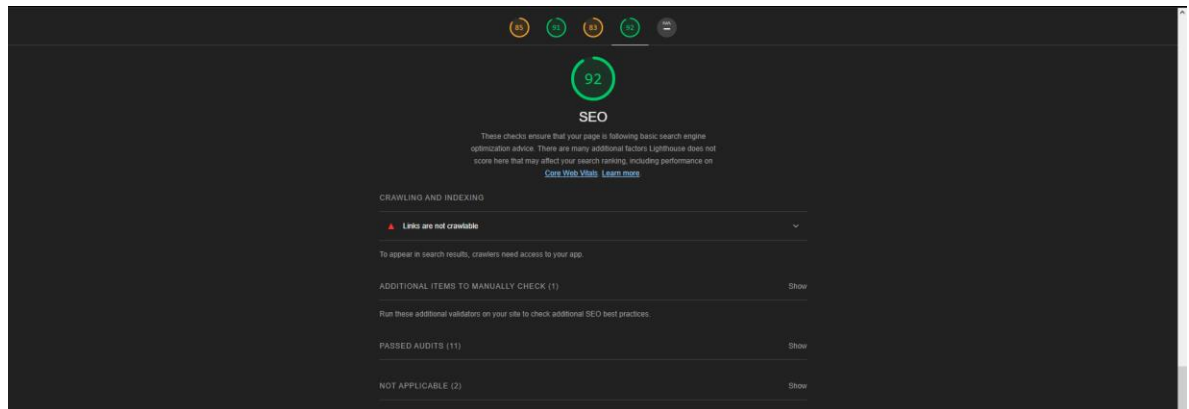


Fuente: Elaboración propia

- **SEO**

Google Lighthouse analiza mediante varias pruebas el nivel de visibilidad en los buscadores, tener un buen porcentaje nos ayuda a parecer en los primeros resultados.

Ilustración 36. Prueba SEO



Fuente: Elaboración propia

Al finalizar la práctica empresarial con la implementación del sitio web E-Commerce se logró crear con éxito el sistema de gestión de ventas online para la empresa ASOPROCAFENPA, con el fin de ser reconocida en el mercado y aumentar su número de clientes, crecer en la comercialización de sus productos, así mismo la facturación electrónica.

Con el desarrollo de esta tienda virtual aumentara las ventas, ya que automatiza los procesos de atención al cliente, siendo una gestión rápida de adquirir información, comprar los productos y ser despachados ágilmente de manera segura. Teniendo en cuenta que los comercios electrónicos tienen la ventaja de estar disponibles 24/7 para los clientes finales, logrando conseguir un incremento de las utilidades en el margen de ganancias

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

La finalización de este proyecto de grado se profundizó en temas de E-Commerce y usabilidad, se aprendió en detalle los pros y contras del comercio electrónico, además, se utilizó un enfoque scrum, realizando entregas parciales donde se comprobó el avance del proyecto y no como tal una sola entrega, lo que permite realizar ajustes sobre la marcha si es necesario.

En el proceso del desarrollo del E-Commerce mi inmersión profunda en el desarrollo web y siendo más específicos en el Framework Angular, puedo afirmar que estas tecnologías han crecido enormemente en la última década, esto se debe a las amplias herramientas entregadas a los desarrolladores y por su puesto la comunidad que lo rodea.

Como resultado del desarrollo del sitio web E-Commerce potenciará el reconocimiento de la empresa, logrando generar un mayor valor a la experiencia del usuario en cuanto a la venta de sus productos, así como expandir la visibilidad de los mismos y con ayuda de estas herramientas ser cada día más competitivos en el mercado.

Gracias a la implementación del comercio electrónico se beneficiará la empresa ASOPROCAFENPA y por supuesto sus clientes, quienes ahora verán más amigable, cómoda y rápida la manera de comprar sus productos.

Finalmente, la ejecución e implementación del comercio electrónico, es una valiosa experiencia para mi crecimiento personal y profesional, permitiéndome adquirir una serie de habilidades importantes para mejorar como Ingeniero de sistemas.

6.2. Recomendaciones

En Colombia el comercio electrónico avanza diariamente y comercializar catálogos en tiendas virtuales tiene sus limitaciones, por ello, el presente proyecto se pensó de tal manera que se pueda escalar a futuro y alcanzar aquellos modelos de negocio que han sido un gran éxito.

Se recomienda para futuros proyectos utilizar un controlador de versiones, como GitLab, que permite hacer una copia de seguridad del proyecto en la nube, evitando pérdidas y cambios de información. Si sucede una de las situaciones mencionadas anteriormente se puede restaurar a una versión anterior.

Si se quiere llegar a clientes finales en un menor tiempo esperado, se recomienda establecer una estrategia de marketing digital, ya que es un propósito fundamental del comercio electrónico.

Los ataques cibernéticos ocurren a diario y ningún sistema por más seguro que sea siempre está vulnerable al robo de información personal y financiera, por esto, se recomienda mantener el E-Commerce actualizado y construir módulos de alerta de actividades sospechosas que comprometan la identidad de los clientes.

BIBLIOGRAFIAS

- Cabrera, D. (2012). Análisis y Aplicación de la Estrategia de Comercio Electrónico en el Portal de Estrategia de Comercio Electrónico. <https://dspace.ups.edu.ec/bitstream/123456789/2065/15/UPS-CT002377.pdf>
- Vista de Teoría y conceptos de la innovación tecnológica. (s. f.-b). <https://journal.universidadean.edu.co/index.php/Revista/article/view/938/905>
- Teoria del empresario innovador de schumpeter - Descuadrando. (s. f.). https://descuadrando.com/Teoria_del_empresario_innovador_de_schumpeter
- Aguiar, F., Lotero, A. y Vega, N. (2015). Procesador y comercializador de café artesanal en la ciudad de Rodantillo. [https://www.intep.edu.co/Es/Usuarios/Institucional/Emprendimiento/2018_1/Publicaciones Tesis cafe de la montana.pdf](https://www.intep.edu.co/Es/Usuarios/Institucional/Emprendimiento/2018_1/Publicaciones/Tesis_cafe_de_la_montana.pdf)
- Arias, J., Zúñiga, P., Viteri, A. y Vilela, M. (2020). Innovaciones en portales adaptativos para mejorar la gestión del comercio electrónico en tiempos de pandemia. Revista de la Universidad Autónoma de los Andes, Universidad y Sociedad Ecuatoriana Vol. 12, No. S1, 2020, pp. 145-150
- Aspís , A., Pertusi , IC, & Nieva, H. (2016). comercio electrónico comercio electrónico. Argentina: Falso .

Barrero, J., Villacreses, C., Choez, J., Figueroa, V. y Maldonado, K. (2021). El impacto de la plataforma Shopify en el comercio electrónico. Mayo , 14 (5), 8-98.
<https://publicaciones.uci.cu/index.php/serie/article/view/866/743>

Ballesteros, M. (2019). Plaza del Café .
[https://repository.urosario.edu.co/bitstream/handle/10336/20276/BallesterosMercad](https://repository.urosario.edu.co/bitstream/handle/10336/20276/BallesterosMercado-)
o-

Bermillón , A. y Cerruti, O. (1993). Implementar y administrar la gestión integral de calidad editorial 2000. Versión 2

Beltrán , O. y Camargo, L. (2016). PTVISAS Diagnóstico de Implementación de la Estrategia de Comercio Electrónico. Universidad Hall . bogotá Distrito de Columbia.[https://ciencia.lasalle.edu.co/ Administración de Empresas/1343](https://ciencia.lasalle.edu.co/Administración%20de%20Empresas/1343)

Cafeteros, F. de. (2020). precio del cafe <https://federaciondefcafeteros.org/wp/publicaciones/>

Carrión, J. (2020). El impacto del comercio electrónico en las PYMES de Providencia de El Oro. Universidad Metropolitana. Ecuador. marzo-abril , 12 (2), 474-480.
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202020000200473

Castaño, J. y Jurado, S. (2006). comercio electrónico España: Editex .

Cámara de Comercio de Colombia. (2020). comercio electrónico _ <https://www.ccce.org.co>

Castaño V. y Echaverry J. (2016). Programa de Mejoramiento de Comercializadores de Café Orgánico Producido en Finca Los Ángeles en Balboa, Departamento Seccional Pereira, Universidad Libre de Risaralda

- A. Duque (2018). Comercio electrónico: Innovando con las redes sociales Delaware. *Gestión de Tecnología e Innovación* , 13 (1), 56-65. como -extendido:
[//dagcmkpagjlhakfdhnbomgmjdpkdklff/enhanced-reader.html?pdf=https%3A%2F%2Fbrxt.mendeley.com%2Fdocument%2Fcontent%2F8fd677-4d57-3438-a1d8-ad7aed1d5761](https://dagcmkpagjlhakfdhnbomgmjdpkdklff/enhanced-reader.html?pdf=https%3A%2F%2Fbrxt.mendeley.com%2Fdocument%2Fcontent%2F8fd677-4d57-3438-a1d8-ad7aed1d5761)
- Schiaffarino, A. (2019, 7 agosto). *Modelo cliente servidor*. Infranetworking.
<https://blog.infranetworking.com/modelo-cliente-servidor/>
- Herrera, C. (2022, 10 octubre). *Base De Datos Relacionales. Que Son Y Para Qué Sirve*. Carlos Herrera. <https://carlos-herrera.com/creacion-de-base-de-datos-relacional/>
- A., D. (2022b, julio 25). *¿Qué es Angular y cuáles son sus ventajas?* Tutoriales Hostinger.
<https://www.hostinger.es/tutoriales/que-es-angular>
- Angular. (s. f.). <https://angular.io/guide/what-is-angular>
- Clark, J. (2021g, septiembre 13). *Los 10 mejores marcos de trabajo de backend*. Back4App Blog. <https://blog.back4app.com/es/los-10-mejores-marcos-de-trabajo-de-backend/>
- DARIAS PÉREZ. (2021, 9 septiembre). *Gestor de Base de datos: Qué es, Funcionalidades y Ejemplos*. Intelequia. <https://intelequia.com/blog/post/2949/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>
- Maldonado, J. (2022b, marzo 8). *5 gestores de bases de datos para diversas aplicaciones*. DocPath - A Step Ahead in Document Software. <https://www.docpath.com/5-gestores-de-bases-de-datos-para-diversas-aplicaciones/?lang=es>
- Kothari, A. (2022, 6 septiembre). *Los 12 mejores marcos de JavaScript (JS) que debe conocer*. Geekflare. <https://geekflare.com/es/best-javascript-frameworks/>

OpenJS Foundation. (s.f.). Acerca de Node.js. Obtenido de nodejs.org:
<https://nodejs.org/es/about/>

Urquiaga, J. C. (2018). <https://devcode.la/blog/que-es-nodejs/>. Obtenido de devcode.

Henry, R. (2021, 1 noviembre). ¿Qué es Node.js y para qué se utiliza? | Henry Blog. Henry.
<https://blog.soyhenry.com/que-es-node-js-y-para-que-se-utiliza/>

Gacelaweb, G. (2020, 23 octubre). *Qué es Node.js y para qué sirve*. Gacelaweb.
<https://www.gacelaweb.com/que-es-nodejs-y-para-que-sirve/>

Rubenfa, R. (2014, 3 febrero). *MongoDB: qué es, cómo funciona y cuándo podemos usarlo (o no)*. Genbeta. <https://www.genbeta.com/desarrollo/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>

ramirez, A. (2019, 25 marzo). *7 razones para usar MongoDB en tus proyectos*. Platzi.
<https://platzi.com/blog/7-razones-mongodb/>

Cervantes Maceda, H., Castro Careaga & Velasco-Elizondo. (2015, enero). *Arquitectura de software. Conceptos y ciclo dedesarrollo*. Arquitectura de software. Conceptos y ciclo dedesarrollo. https://www.researchgate.net/profile/Perla-Velasco-Elizondo/publication/281137715_Arquitectura_de_Software_Conceptos_y_Ciclo_de_Desarrollo/links/57144e1408aeebe07c0641ab/Arquitectura-de-Software-Conceptos-y-Ciclo-de-Desarrollo.pdf

Sergio, S. (2017, 4 enero). *Introducción a JSON Web Tokens (JWT)*. Platzi.
<https://platzi.com/blog/introduccion-json-web-tokens/>

Vallecillo, L. Fuentes y A. (2004). Una introducción a los perfiles UML. Univ. Malaga

AngularZone. (2015). *ANGULARJS HISTORY*. AngularZone.
<https://angularzone.wordpress.com/angularjs-history/>

AngularJS. (2016). *AngularJS by Google*. AngularJS by Google. <https://angularjs.org/>

Java desde 0, A. (2020, 17 diciembre). *Introducción al routing de Angular*. Java desde 0.

<https://javadesde0.com/introduccion-al-routing-de-angular/>

Createx Studio. (s. f.). *Createx | E-Commerce - Home*.

<https://createx.createx.studio/ecommerce/index.html>