

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE: INGENIERÍA EN SISTEMAS DE INFORMACIÓN



Trabajo de Titulación

Tema: Desarrollo de una aplicación web dedicada a la venta de partes y piezas de computadoras para la empresa SATELNET PC.

AUTOR:

Simón Aucatoma

QUITO DM, DICIEMBRE 2023

RESUMEN

Este trabajo de titulación aborda directamente los retos que enfrenta la tienda física "Satelnet PC", especializada en equipos informáticos en Quito, a raíz del cierre temporal debido a la pandemia en 2020. La necesidad de adaptarse al auge del comercio electrónico en el país emerge como una estrategia clave para recuperar y consolidar su posición en el mercado. La iniciativa busca ofrecer a los consumidores una experiencia agradable mediante un catálogo virtual. Este aplicativo web aporta un valor adicional a los procedimientos convencionales que implicaban el registro manual en hojas de cálculo de Excel para gestionar las operaciones diarias de la tienda.

En el desarrollo de la aplicación web, la elección de la metodología XP junto con las herramientas React y Firebase se destaca como una estrategia eficiente. La meticulosa delimitación de los requisitos desempeñó un papel fundamental en la planificación y ejecución eficientes del desarrollo del sistema. Al concluir el desarrollo de la aplicación se confirmó el cumplimiento integral de los objetivos propuestos, validando la efectividad de las decisiones tomadas en términos de metodología y herramientas tecnológicas.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS.....	iii
ÍNDICE DE FIGURAS.....	iv
CAPÍTULO 1: INTRODUCCIÓN	1
1 INTRODUCCIÓN.....	1
1.1 Justificación.....	1
1.2 Planteamiento del Problema.....	2
1.3 Objetivos.....	3
1.3.1 General:.....	3
1.3.2 Específicos:.....	3
1.4 Antecedentes.....	3
1.5 Alcance.....	5
CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA.....	6
1 MARCO TEÓRICO.....	6
1.1 Aplicación Web.....	6
1.2 Framework de Desarrollo Web.....	7
1.2.1 Angular.....	7
1.2.2 React.....	8
1.2.3 Vue.js.....	8
1.3 Lenguajes de Programación.....	9
1.3.1 JavaScript.....	9
1.3.2 Python.....	10
1.3.3 Comparativa Javascript vs Python.....	10
1.4 Base de Datos.....	11
1.4.1 Bases de Datos Relacionales.....	11
1.4.2 Bases de Datos NoSQL.....	12
1.4.3 Principales Diferencias de las Bases de Datos NoSQL y Relacionales.....	12
1.5 Plataforma como Servicio.....	13
1.5.1 Amazon AWS.....	13
1.5.2 Firebase.....	14
1.5.3 Comparación entre AWS y Firebase.....	14
1.6 Metodologías de Desarrollo.....	15
1.6.1 Metodología Ágil.....	15
1.6.2 Metodología Tradicional.....	16
1.6.3 Metodologías Tradicionales vs Ágiles.....	16
1.7 Lenguaje Unificado de Modelado (UML).....	17
1.7.1 Casos de Uso.....	17
1.7.2 Diagrama de Actividades.....	18
CAPÍTULO 3: METODOLOGÍA.....	19
1 METODOLOGÍA DE INVESTIGACIÓN.....	19
1.1 Investigación Aplicada.....	19
1.2 Enfoque Cualitativo.....	19
1.3 Metodología de Desarrollo.....	19
1.3.1 Scrum.....	20
1.3.2 XP.....	21
1.3.3 Comparativa Scrum vs XP.....	23
CAPÍTULO 4: DESARROLLO	24
1 DESARROLLO DE LA APLICACIÓN.....	24
1.1 Fase Inicial: Análisis de los Requerimientos.....	24

1.1.1	Historias de Usuarios	24
1.1.2	Priorización de Historias de Usuarios	25
1.1.3	Iteraciones	26
1.2	<i>Diseño del Sistema</i>	27
1.2.1	Diseño de la Aplicación Web	27
1.2.2	Almacenamiento en Cloud Firestore	28
1.2.3	Funcionamiento de la Base de Datos Cloud Firestore	28
1.3	<i>Iteración 1</i>	31
1.3.1	Planeación	31
1.3.1.1	Historias de Usuarios Prioridad Alta	31
1.3.1.2	Objetivos de la Iteración	33
1.3.1.3	Criterios de Aceptación	33
1.3.2	Diseño	33
1.3.2.1	Diagramas UML para la Visualización de Productos	33
1.3.3	Codificación	35
1.3.4	Pruebas	36
1.3.4.1	Pruebas unitarias	36
1.3.4.2	Pruebas de Aceptación	40
1.4	<i>Iteración 2</i>	43
1.4.1	Planeación	43
1.4.1.1	Historias de Usuarios	43
1.4.1.2	Objetivos de la Iteración	44
1.4.1.3	Criterios de Aceptación	44
1.4.2	Diseño	44
1.4.2.1	Diagramas UML para Registro e Inicio de Sesión	44
1.4.3	Codificación	48
1.4.4	Pruebas	49
1.4.4.1	Pruebas Unitarias	49
1.4.4.2	Pruebas de Aceptación	51
1.5	<i>Iteración 3</i>	52
1.5.1	Planeación	52
1.5.1.1	Historia de Usuarios	53
1.5.1.2	Objetivos de la Iteración	53
1.5.1.3	Criterios de Aceptación	53
1.5.2	Diseño	54
1.5.3	Codificación	56
1.5.4	Pruebas	56
1.5.4.1	Pruebas Unitarias de la Tercera Iteración	56
1.5.4.2	Pruebas de Aceptación	58
1.6	<i>Cierre</i>	59
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES		60
1	CONCLUSIONES Y RECOMENDACIONES	60
1.1	<i>Conclusiones</i>	60
1.2	<i>Recomendaciones</i>	61
Bibliografía		62
Glosario		64
ANEXOS		65

ÍNDICE DE TABLAS

Tabla 1 <i>Comparativa entre lenguajes de programación</i>	10
Tabla 2 <i>Diferencias entre una base de datos relacional y no relacional</i>	13
Tabla 3 <i>Comparación entre aws y firebase</i>	14
Tabla 4 <i>Comparación entre metodologías ágiles y tradicionales</i>	16
Tabla 5 <i>Comparación entre la metodología Scrum y XP</i>	23
Tabla 6 <i>Asignación de historia de usuarios a iteraciones</i>	27
Tabla 7 <i>Historia de usuario número 2</i>	31
Tabla 8 <i>Historia de usuario número 3</i>	31
Tabla 9 <i>Historia de usuario número 4</i>	32
Tabla 10 <i>Historia de usuario número 5</i>	32
Tabla 11 <i>Historia de usuario número 7</i>	32
Tabla 12 <i>Prueba de aceptación de la historia de usuario número 2</i>	40
Tabla 13 <i>Prueba de aceptación de la historia de usuario número 3</i>	40
Tabla 14 <i>Prueba de aceptación de la historia de usuario número 4</i>	41
Tabla 15 <i>Prueba de aceptación de la historia de usuario número 5</i>	41
Tabla 16 <i>Prueba de aceptación de la historia de usuario número 7</i>	42
Tabla 17 <i>Historia de usuario número 1</i>	43
Tabla 18 <i>Historia de usuario número 9</i>	43
Tabla 19 <i>Historia de usuario número 10</i>	44
Tabla 20 <i>Prueba de Aceptación de la historia de usuario número 1</i>	51
Tabla 21 <i>Prueba de Aceptación de la historia de usuario número 9</i>	51
Tabla 22 <i>Prueba de Aceptación de la historia de usuario número 10</i>	52
Tabla 23 <i>Historia de usuario número 6</i>	53
Tabla 24 <i>Historia de usuario número 8</i>	53
Tabla 25 <i>Prueba de Aceptación de la historia de usuario número 6</i>	58
Tabla 26 <i>Prueba de Aceptación de la historia de usuario número 8</i>	59

ÍNDICE DE FIGURAS

Figura 1 <i>Proceso de la metodología XP</i>	22
Figura 2 <i>Árbol de navegación de la aplicación web</i>	27
Figura 3 <i>Colecciones y documentos en Cloud Firestore</i>	30
Figura 4 <i>Caso de uso del catálogo virtual</i>	33
Figura 5 <i>Diagrama de secuencia para buscar producto</i>	34
Figura 6 <i>Diagrama de colaboración para buscar producto</i>	34
Figura 7 <i>Pantalla de la ejecución de Cypress</i>	37
Figura 8 <i>Error de llamadas en el home</i>	37
Figura 9 <i>Prueba unitaria en las llamadas del home exitosa</i>	38
Figura 10 <i>Prueba unitaria para el filtro de precios y categorías</i>	39
Figura 11 <i>Diagrama de secuencia del registro de usuario</i>	45
Figura 12 <i>Diagrama de secuencia del inicio de sesión</i>	45
Figura 13 <i>Diagrama de colaboración de actualización de perfil</i>	46
Figura 14 <i>Diagrama de colaboración del administrador</i>	47
Figura 15 <i>Estándar de codificación adoptado</i>	48
Figura 16 <i>Errores de Firebase durante el registro</i>	49
Figura 17 <i>Prueba de registro de usuario exitosa</i>	50
Figura 18 <i>Caso de uso del carrito de compras</i>	54
Figura 19 <i>Diagrama de secuencia de la compra</i>	55
Figura 20 <i>Diagrama de colaboración para generar prefactura</i>	55
Figura 21 <i>Prueba de agregar productos al carrito</i>	57
Figura 22 <i>Firma de responsabilidad del cliente para documentos del trabajo</i>	65
Figura 23 <i>Carta de aceptación de finalización del software</i>	66

CAPÍTULO 1: INTRODUCCIÓN

1 Introducción

1.1 Justificación

La tienda física denominada "Satelnet PC," especializada en la comercialización de equipos informáticos en Ecuador, experimentó un cierre temporal que abarcó un período de cuatro meses durante el año 2020, en respuesta a la pandemia global. Esta medida, necesaria para salvaguardar la salud pública, resultó en un impacto significativo en los ingresos de la mencionada entidad comercial. Tras su reapertura, la tienda ha enfrentado desafíos persistentes en la recuperación de su volumen de ventas, situación que persiste hasta la fecha correspondiente al año 2023.

En este contexto, es importante destacar que muchas empresas han encontrado en la adopción de nuevas tecnologías un medio efectivo para mejorar su competitividad y rentabilidad. Según la fuente citada de la revista "Ekos" el comercio electrónico en Ecuador ha experimentado un notable crecimiento, alcanzando un volumen de negocio de USD 2.3 mil millones. Este dato representa un incremento de USD 700 millones, equivalente al 43.75% en comparación con los valores registrados en el año 2019. Además, se observa un aumento significativo del 400% en las visitas a sitios web (Cámara Ecuatoriana de Comercio Electrónico, 2021).

Este fenómeno del crecimiento del comercio electrónico resalta la importancia de la adaptación a las tendencias tecnológicas en el contexto empresarial actual. Para las empresas como Satelnet PC, la exploración de estrategias y soluciones basadas en el comercio electrónico podría ser un factor crucial en su intento de recuperar y mejorar su posición en el mercado, dado el cambio en las preferencias y comportamientos de los consumidores, exacerbados por las circunstancias resultantes de la pandemia.

El propietario de este negocio identificó a principios del presente año una oportunidad para promocionar productos a través de las redes sociales, lo que resultó en una respuesta favorable por parte de un considerable número de clientes. Este éxito inicial ha llevado a la necesidad de desarrollar una aplicación web dedicada, cuyo propósito principal es exhibir los productos disponibles junto con sus precios y características detalladas. El propósito detrás de esta iniciativa es proporcionar a los clientes un acceso más conveniente y completo para explorar y evaluar los productos que se encuentran disponibles en la tienda en línea.

1.2 *Planteamiento del Problema*

En el año de ejecución de este proyecto (2023), se constata la carencia de procesos tecnificados en la comercialización de productos en línea. Los consumidores se enfrentan a dificultades relacionadas con la disponibilidad de una variedad suficiente de componentes informáticos. Asimismo, el proceso de cotización de productos se torna engorroso para el propietario de la tienda, y la verificación del inventario de manera manual resulta en una pérdida de tiempo significativa.

Por lo tanto, surge la necesidad de implementar un proyecto de comercio electrónico específicamente diseñado para la tienda "Satelnet PC," que ofrezca una diversidad de opciones, información detallada, precios competitivos y una experiencia de compra conveniente. La finalidad de esta iniciativa es permitir a los consumidores acceder a un catálogo virtual confiable y eficaz para adquirir los componentes que requieren, satisfaciendo así sus necesidades tecnológicas y mejorando significativamente su experiencia de compra.

1.3 *Objetivos*

1.3.1 **General:**

Desarrollar una aplicación web para la venta de partes y piezas de computadoras.

1.3.2 **Específicos:**

Analizar el proceso actual de compras en la empresa SATELNET PC.

Determinar la metodología para el desarrollo de software.

Analizar los requerimientos que son importantes para este proyecto.

Desarrollar el aplicativo web.

1.4 *Antecedentes*

El señor Sergio Aucatoma, originario de la ciudad de Nueva Loja, emigró a la ciudad de Quito en el año 2005, donde se adentró en el campo de la tecnología como técnico de computadoras. Fue durante este período que identificó una oportunidad de emprendimiento al constatar la carencia de suministros y componentes informáticos en su lugar de trabajo, y cuando estaban disponibles, resultaban onerosos para los clientes finales. Movidio por esta observación, optó por iniciar su propia empresa, desempeñando el rol de intermediario entre los principales distribuidores tecnológicos de la ciudad de Quito y sus clientes. A esta pequeña pero prometedora empresa le otorgó el nombre de "SATELNET PC".

La entidad denominada "SATELNET PC" se especializa en la distribución y mantenimiento de equipos informáticos, que abarcan desde computadoras de escritorio, laptops, impresoras, hasta copiadoras, entre otros dispositivos afines. La empresa ha acumulado una robusta experiencia a lo largo de un periodo de una década en el mercado local de la ciudad de Quito, ubicada en el territorio ecuatoriano. Su fundación tuvo lugar en la ciudad de Quito en el año 2011, bajo la dirección y propiedad del señor Sergio Aucatoma. Hasta el presente año 2023,

la empresa tiene su sede matriz una tienda física ubicada en el sector de San Bartolo, al sur de la ciudad de Quito, donde se llevan a cabo tanto las ventas como los servicios de mantenimiento relacionados con equipos de cómputo.

En el contexto de la gestión de esta tienda, el propietario ha implementado diversas estrategias con el objetivo de fortalecer su posición competitiva en el mercado nacional. Una de las estrategias clave adoptadas ha sido la decisión de convertirse en un importador directo para la mayoría de sus productos, dando especial énfasis a las adquisiciones procedentes de China. Esta iniciativa busca consolidar su presencia en el mercado y mejorar su competitividad. En paralelo, en lo que respecta al registro y control de las transacciones de compras y ventas, el propietario ha establecido un sistema que comprende el uso de hojas de cálculo en Excel, respaldado por registros manuales detallados en cuadernos de contabilidad. Asimismo, la gestión del inventario se lleva a cabo de manera manual, implicando la revisión física de los productos almacenados en estanterías o cartones cada vez que un cliente muestra interés por un artículo específico.

A partir del año 2020, el propietario ha adoptado una decisión estratégica relevante, que implica la promoción de los productos de la tienda a través de las redes sociales. Este enfoque ha recibido una respuesta positiva por parte de una gran proporción de la clientela, al punto que, desde el año 2022, la mayoría de las ventas de la empresa se efectúan mediante plataformas de redes sociales. En este contexto, ha surgido una necesidad inminente de desarrollar y mantener un catálogo digital. Esta iniciativa responde a la diversidad geográfica de los clientes, quienes provienen de diversas provincias del Ecuador. El catálogo digital se concibe con el propósito de brindar a los consumidores una visualización detallada y de fácil acceso de los productos disponibles en el inventario de la empresa.

1.5 Alcance

El proyecto de titulación actual alcanzará su conclusión una vez que se hayan cumplido en su totalidad todos los objetivos establecidos y se haya recibido la carta de conformidad por parte del cliente, confirmando que los requisitos y expectativas han sido adecuadamente satisfechos.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA

1 Marco Teórico

En el análisis efectuado por Cardona Arenas y su equipo de investigadores (2022) en relación con el impacto del comercio electrónico en el rendimiento financiero de las pequeñas y medianas empresas en Manizales, Colombia, se enfatiza la relevancia del comercio electrónico como una herramienta fundamental para fomentar la competitividad y la excelencia en el servicio. Conforme a los resultados obtenidos en esta investigación, aquellas compañías que han incorporado estrategias digitales en su proceso de desarrollo empresarial experimentan mayores posibilidades en los actuales mercados, lo que se traduce en niveles superiores de rentabilidad y competitividad. Este fenómeno se atribuye a la capacidad de estas organizaciones para responder con eficacia a las cambiantes dinámicas del mercado y para establecer relaciones de comunicación, conexión y compromiso más sólidas con su base de consumidores, aspectos fundamentales en el contexto empresarial contemporáneo (Cardona Arenas, Quintero Renaud, Mora Quintero, & Castro Cardona, 2022).

1.1 Aplicación Web

Las aplicaciones web son la puerta de entrada a un mundo de posibilidades, donde la conectividad y la accesibilidad se unen para brindar experiencias interactivas y funcionales. A través de tecnologías web como HTML, CSS y JavaScript, estas aplicaciones permiten a los usuarios acceder y utilizar software desde cualquier dispositivo con conexión a internet, sin barreras de instalación ni limitaciones de plataforma¹. Son la respuesta a la demanda de flexibilidad y conveniencia en la era digital (Espinosa-Hurtado, 2021).

¹ Variedad de Sistemas Operativos y entornos de hardware que permite funcionar una aplicación web.

1.2 Framework de Desarrollo Web

El desarrollo de aplicaciones web se refiere a la creación de software que se ejecuta en un navegador web y se comunica con un servidor a través de Internet. Para facilitar este proceso, se utilizan frameworks que proporcionan una estructura predefinida para el desarrollo de aplicaciones en los cuales pueden incluir herramientas para la gestión de bases de datos, la creación de interfaces de usuario entre otros (Márquez Egea, 2020). Estos marcos proporcionan una base sólida para agilizar el proceso de desarrollo y mejorar la eficiencia en la creación de aplicaciones web.

Un framework de desarrollo web se refiere a un conjunto de elementos que incluyen herramientas, librerías, estándares y prácticas establecidas que permiten a los desarrolladores crear aplicaciones web de manera más efectiva y en menos tiempo. Los desarrolladores hacen uso de los frameworks web para simplificar el proceso de construcción de aplicaciones web, ya que ofrecen una base sólida y predefinida que cubre muchas de las necesidades comunes de una aplicación web (Márquez Egea, 2020). Algunos ejemplos de frameworks web son Angular, React, Vue.js, Django, Laravel y Spring.

1.2.1 Angular

Es un framework de desarrollo de aplicaciones web de código abierto, ha ganado una amplia popularidad en la comunidad de desarrollo. Este framework, concebido y lanzado por Google en 2016, representa una reescritura completa de su predecesor, AngularJS, y está especialmente diseñado para la creación de aplicaciones web SPA² (Schwarz Müller, 2020).

Angular está basado en componentes mediante Modelo Vista Controlador para lo cual utiliza HTML, CSS y TypeScript. Generalmente, toda la aplicación se ejecuta en una sola página HTML proporcionando mayor rapidez con el usuario.

² Se trata de una aplicación de página única en la que todo el contenido se encuentra en una sola página.

1.2.2 React

React es una biblioteca de código abierto escrita en JavaScript, que se utiliza ampliamente para crear interfaces de usuario interactivas y reutilizables en aplicaciones web. Facebook lanzó React en 2013, y ha ganado popularidad debido a su enfoque basado en componentes, lo que significa que se pueden crear componentes de interfaz de usuario reutilizables y ajustables para representar diferentes partes de una aplicación. Otra característica importante de React es su enfoque de programación declarativa, que facilita la actualización y renderizado de los componentes y permite crear interfaces de usuario dinámicas y escalables (Facebook, 2021).

Esta herramienta poderosa de programación en JavaScript que permite una mayor modularidad y reutilización del código. Además, su modelo de programación se basa en estado y props, lo que facilita el manejo de la información y la actualización de los componentes. Todo esto, combinado con su alto rendimiento y eficiencia, ha llevado a React a convertirse en una herramienta muy popular entre los desarrolladores de aplicaciones web en todo el mundo.

1.2.3 Vue.js

Este framework de desarrollo, diseñado por Evan You, vio la luz por primera vez en febrero de 2014 como una versión en fase de prueba. La versión estable inicial, conocida como Vue.js 1.0, fue lanzada en febrero de 2016, y posteriormente se presentó la versión 2.0 en septiembre de 2016. A lo largo del tiempo, se han implementado diversas actualizaciones y mejoras en este framework (Schwarz Müller, 2020).

Según Schwarz Müller (2020) Vue.js es un framework progresivo de JavaScript utilizado para construir interfaces de usuario en aplicaciones web. Vue.js se enfoca en la capa de presentación de una aplicación web, permitiendo a los desarrolladores crear componentes reutilizables que se pueden combinar para construir interfaces de usuario complejas. Además,

Vue.js proporciona un conjunto de herramientas para el enlace de datos bidireccional, la manipulación del DOM³ y la creación de animaciones, lo que facilita el desarrollo de interfaces de usuario dinámicas y responsivas.

Este framework, aunque actualmente carece de recursos y foros en línea, no figura como la primera opción para los desarrolladores. Sin embargo, su potencial de escalabilidad y flexibilidad podría posicionarlo en una posición destacada en un futuro próximo.

1.3 Lenguajes de Programación

Según (Martín Villalba, Rubio, & Urquia Moraleda, 2021), un lenguaje de programación es un conjunto de símbolos, palabras y reglas sintácticas y semánticas que permiten a los programadores escribir programas que pueden ser interpretados o compilados en código ejecutable por una computadora. Los lenguajes de programación se utilizan para crear software, aplicaciones, sistemas operativos, juegos y otros tipos de programas informáticos, y pueden clasificarse en diferentes categorías según su nivel de abstracción, paradigma de programación y aplicación específica. Algunos ejemplos de lenguajes de programación comunes incluyen C++, Java, Python, JavaScript, Ruby, PHP y Swift.

1.3.1 JavaScript

JavaScript es un lenguaje de programación muy popular y versátil, utilizado para desarrollar aplicaciones web y móviles, así como aplicaciones empresariales y de escritorio. Es conocido por su capacidad de interactuar con HTML y CSS, lo que lo hace especialmente adecuado para la programación web, pero también se puede utilizar para crear programas independientes. Además, la continua evolución de JavaScript, junto con la comunidad de desarrolladores de código abierto, ha dado lugar a una amplia variedad de herramientas y

³ Modelo de Objetos del Documento es una representación jerárquica de la estructura de un documento HTML o XML

marcos que lo hacen aún más útil y poderoso (Martín Villalba, Rubio, & Urquia Moraleda, 2021).

1.3.2 Python

Python es un lenguaje de programación de nivel superior ampliamente empleado en el desarrollo de diversas aplicaciones. A diferencia de lenguajes como Java o .NET, Python se considera un lenguaje interpretado, lo que significa que no requiere de un proceso de compilación para ejecutar las aplicaciones escritas en este lenguaje. En su lugar, las aplicaciones se ejecutan directamente mediante un intérprete, sin necesidad de ser traducidas al lenguaje de máquina (Martín Villalba, Rubio, & Urquia Moraleda, 2021).

1.3.3 Comparativa Javascript vs Python

En esta situación, asigno una escala del 1 al 10 para evaluar el desempeño, donde un puntaje más bajo indica debilidades y un puntaje más alto destaca las fortalezas.

Tabla 1

Comparativa entre lenguajes de programación

Características \ Lenguajes de programación	JavaScript	Python
Back-end	9	6
Front-end	9	8
Librerías para la web	10	6
Adaptabilidad web	8	7
Total	36	26

Nota: se puntúa de acuerdo con el desarrollo web.

Se observa que javascript tiene mayor ventaja sobre Python. Por lo que es el lenguaje utilizado en este proyecto de desarrollo web.

1.4 Base de Datos

La definición de base de datos según varios autores es: "Una base de datos es una colección de datos interrelacionados, que se utilizan para diversos fines. Los datos están organizados en tablas que representan objetos o conceptos del mundo real, y las relaciones entre esas tablas se establecen mediante el uso de claves y otros mecanismos" (Silberschatz, Korth, & Sudarshan, 2014, pág. 16).

Las bases de datos son una parte fundamental de cualquier aplicación web, ya que permiten el almacenamiento, la gestión y la recuperación de datos. Hay muchos tipos de bases de datos disponibles para el desarrollo web.

1.4.1 Bases de Datos Relacionales

Las bases de datos relacionales son un tipo de sistema de gestión de bases de datos (DBMS) que almacenan y organizan datos en tablas relacionales. Cada tabla representa una entidad en el mundo real, como clientes, productos o pedidos, y las relaciones entre ellas se definen mediante claves primarias y foráneas. Las bases de datos relacionales permiten realizar consultas complejas mediante el uso de lenguajes de consulta estructurados (SQL) y garantizan la integridad de los datos mediante reglas de integridad referencial. Además, ofrecen características como la escalabilidad, la seguridad y la capacidad de recuperación. Las bases de datos relacionales son ampliamente utilizadas en aplicaciones empresariales y de software en todo el mundo (Elmasri & Navathe, 2016).

Las bases de datos relacionales son las más utilizadas en el desarrollo web. Utilizan tablas para organizar la información en filas y columnas, y se basan en el lenguaje SQL para interactuar con los datos. Ejemplos de bases de datos relacionales incluyen MySQL, PostgreSQL, Oracle y SQL Server.

1.4.2 Bases de Datos NoSQL

Una base de datos NoSQL, también conocida como "no relacional", es un tipo de sistema de gestión de bases de datos que no utiliza tablas relacionales como en las bases de datos relacionales tradicionales. En su lugar, utilizan otros tipos de modelos de datos para almacenar información, como documentos, gráficos o claves/valores. La principal ventaja de las bases de datos NoSQL es que son altamente escalables y pueden manejar grandes volúmenes de datos no estructurados. Sin embargo, pueden requerir más trabajo por parte del desarrollador para garantizar la integridad y coherencia de los datos (Elmasri & Navathe, 2016).

Las bases de datos NoSQL (Not Only SQL) son una alternativa a las bases de datos relacionales. A menudo se utilizan en aplicaciones web modernas que manejan grandes cantidades de datos no estructurados o semi-estructurados. Algunos ejemplos de bases de datos NoSQL son MongoDB, Cassandra, Redis y Firebase Realtime Database.

1.4.3 Principales Diferencias de las Bases de Datos NoSQL y Relacionales.

Según lo mencionado por (Elmasri & Navathe, 2016) las bases de datos con su estructura tabular y esquemas predefinidos, se destaca por su eficacia en la manipulación de datos estructurados y operaciones complejas. En contraste, NoSQL abarca diversos modelos (documentos, clave-valor, columnares, grafos), ofreciendo flexibilidad y adaptabilidad para datos no estructurados y escalabilidad horizontal. La elección entre ambos paradigmas debe basarse en las necesidades específicas del proyecto, considerando la complejidad y naturaleza de los datos a gestionar.

A continuación, presentamos una tabla para ilustrar las diferencias entre las bases de datos no relacionales y las bases de datos relacionales de acuerdo con lo extraído de (Elmasri & Navathe, 2016):

Tabla 2*Diferencias entre una base de datos relacional y no relacional*

Características	BD relacional	BD no relacional
Estructura de datos	Organizado en tablas	Diversos formatos
Esquema	Rígido	Dinámico
Escalabilidad horizontal	Desafiante	Sencillo
Consultas	Es bueno para consultas complejas	Mejor rendimiento en consultas sencillas
Costo y recursos	Mayor costo	Menor costo
Ejemplos de base de datos	MySQL, Oracle	MongoDB, Firestore

Nota: Características relevantes para el proyecto. Fuente: (Elmasri & Navathe, 2016)

1.5 Plataforma como Servicio

La Plataforma como Servicio (PaaS) es un entorno completo de desarrollo y despliegue en la nube que ofrece recursos para la creación de aplicaciones simples o complejas. En lugar de adquirir y administrar sus propios servidores y recursos, se compra el acceso a estos servicios a través de un proveedor de la nube. Esto le permite desarrollar y ejecutar aplicaciones a través de una conexión segura a Internet, pagando solo por los recursos que utiliza, lo que brinda flexibilidad y escalabilidad en costos. (Microsoft Corporation, 2023).

1.5.1 Amazon AWS

AWS (Amazon Web Services) es una plataforma de servicios en la nube que ofrece una amplia gama de servicios de computación, almacenamiento, bases de datos, redes y más. Es una de las principales plataformas de servicios en la nube utilizadas por empresas y desarrolladores de todo el mundo. AWS permite a los usuarios acceder y utilizar recursos informáticos bajo demanda, sin la necesidad de adquirir y mantener su propia infraestructura física. Los servicios de AWS se ofrecen a través de centros de datos distribuidos en diferentes regiones del mundo y se pueden escalar según las necesidades del usuario (Gimenez, 2020).

1.5.2 Firebase

Según la documentación oficial de Firebase, "Firebase es una plataforma de desarrollo de aplicaciones móviles y web que proporciona una base de datos en tiempo real, autenticación de usuarios y alojamiento de archivos" (Google, 2023).

Firebase puede ser una alternativa atractiva para aquellos interesados en una plataforma de desarrollo de aplicaciones web o móviles que ofrezca múltiples servicios integrados. Además, Firebase cuenta con una documentación clara y completa, una comunidad activa de desarrolladores y un aprendizaje gradual, lo que facilita su uso para los principiantes en el desarrollo de aplicaciones. Otra ventaja de Firebase es que ofrece un plan gratuito que permite a los desarrolladores experimentar y probar la plataforma antes de invertir en un plan de pago.

1.5.3 Comparación entre AWS y Firebase

En esta comparativa, considero los factores que tienen un impacto en el desarrollo de una aplicación web de tamaño reducido.

Tabla 3

Comparación entre aws y firebase

Características \ Paas	AWS	Firebase
Nube pública	X	X
Sencilles en el registro de cuenta		X
Registro sin tarjeta de crédito		X
Autenticación de usuarios	X	X
Total	2	4

En esta comparación, el puntaje más alto se asigna a aquellos elementos que tienen un mayor número de marcas de "x". El servicio de Firebase se presenta como la elección idónea para llevar a cabo el desarrollo de la aplicación web propuesta en este caso.

1.6 Metodologías de Desarrollo

Según (Kendall & Kendall, 2018), la metodología de desarrollo se refiere a un enfoque estructurado y sistemático para planificar, diseñar, implementar y mantener sistemas de información. Las metodologías de desarrollo son un conjunto de técnicas, herramientas y procedimientos que se utilizan para guiar el proceso de desarrollo de sistemas y asegurar la calidad del resultado final.

Existen diferentes metodologías de desarrollo de sistemas, como la metodología de desarrollo tradicionales, la metodología ágil, la metodología en espiral y la metodología orientada a objetos, entre otras. Cada una de estas metodologías tiene sus propias ventajas y desventajas, y la elección de la metodología adecuada depende del tipo de proyecto y de las necesidades específicas de la organización.

1.6.1 Metodología Ágil

De acuerdo con la definición de (HNEIF & HOCK OW, 2009), las metodologías ágiles se caracterizan por ser un método iterativo e incremental en el desarrollo de software, centrándose en la entrega colaborativa, rápida y adaptable de software funcional y de alta calidad, en contraste con los enfoques tradicionales que se basan en una planificación y ejecución más rigurosas.

La metodología ágil se adapta mejor a los entornos de desarrollo de software actuales, que a menudo cambian rápidamente y requieren una respuesta flexible y ágil. Existen algunos de estas metodologías de los cuales podemos mencionar a Scrum, Kanban, XP (Extreme Programming) y Lean Development, entre otras.

1.6.2 Metodología Tradicional

La metodología de desarrollo de software tradicional se basa en un enfoque secuencial y lineal, también conocido como modelo en cascada. En este enfoque, las etapas del desarrollo de software, como el análisis de requisitos, el diseño, la implementación y las pruebas, se realizan en secuencia, una después de la otra. Cada etapa debe completarse antes de pasar a la siguiente y cualquier cambio o ajuste requerido puede ser costoso y complicado de implementar. Este enfoque es adecuado para proyectos con requisitos estables y bien definidos, donde los cambios son mínimos. Sin embargo, puede ser inflexible frente a cambios en los requisitos del cliente o necesidades emergentes durante el proceso de desarrollo (Pressman, 2010). Algunos ejemplos de la metodología tradicional incluyen el modelo en cascada, el modelo en V y el modelo en espiral.

1.6.3 Metodologías Tradicionales vs Ágiles

Tabla 4

Comparación entre metodologías ágiles y tradicionales

Características \ Metodologías	Tradicionales	Ágiles
Flexible a cambios en el proceso		X
Alta comunicación con el usuario		X
Entregables a corto ciclo		X
Aplicable a pequeños proyectos	X	X
Total	1	4

La metodología ágil se caracteriza por su flexibilidad y capacidad de respuesta a los cambios, entregando incrementos de software de forma rápida y priorizando la satisfacción del cliente. Por otro lado, la metodología tradicional se enfoca en una planificación exhaustiva y la entrega final del software completo, sin adaptarse fácilmente a los cambios. En este caso

específico, se optará por aplicar la metodología ágil Extreme Programming (XP), la cual es adecuada para proyectos de menor escala que requieren entregas continuas e incrementales.

1.7 Lenguaje Unificado de Modelado (UML)

Según (Ambler, 2005) define a UML como un lenguaje de modelado visual utilizado en el campo del desarrollo de software para representar, visualizar, especificar y documentar los diferentes aspectos de un sistema. Proporciona una notación estándar y un conjunto de diagramas que permiten a los desarrolladores y equipos de proyecto comunicar y comprender de manera efectiva la arquitectura, estructura, comportamiento y relaciones de un sistema. El uso de UML ayuda a facilitar la comunicación, la colaboración y el análisis de requisitos en el proceso de desarrollo de software.

UML desempeña un papel clave en el desarrollo de software al facilitar la comunicación efectiva entre desarrolladores, analistas y partes interesadas mediante una notación visual estándar. Los diagramas UML permiten transmitir de manera comprensible conceptos y diseños complejos, lo que mejora la comprensión del proyecto. Además, UML proporciona una metodología estructurada para analizar y diseñar sistemas de software, utilizando diferentes tipos de diagramas para representar visualmente la estructura, el comportamiento y las interacciones del sistema. Esto ayuda a los desarrolladores a tener una visión clara y precisa del sistema, facilitando el proceso de desarrollo y asegurando la calidad del software resultante.

1.7.1 Casos de Uso

En el ámbito del desarrollo de software, el uso de UML desempeña un papel fundamental al posibilitar una comunicación efectiva entre los diversos actores involucrados, tales como desarrolladores, analistas y partes interesadas (Booch, Rumbaugh, & Jacobson, 2005). A través de una notación visual estandarizada, los diagramas UML permiten transmitir conceptos y diseños complejos de forma comprensible, mejorando así la comprensión y

colaboración en el proyecto. Además, UML proporciona una metodología estructurada para el análisis y diseño de sistemas de software, utilizando diferentes tipos de diagramas como los de clases, secuencia y actividad, que visualizan la estructura, el comportamiento y las interacciones del sistema (Booch, Rumbaugh, & Jacobson, 2005). Esta aproximación facilita la comprensión del sistema, mejora la planificación y ejecución del desarrollo de software, y contribuye a la calidad del producto final.

1.7.2 Diagrama de Actividades

Según (Pressman, 2010), en su obra "Ingeniería del software: Un enfoque práctico", se utiliza el diagrama de actividades como una herramienta visual en la ingeniería de software para representar el flujo de actividades o procesos dentro de un sistema. Este tipo de diagrama consiste en nodos que representan las actividades y flechas que indican las transiciones entre ellas. Su objetivo es facilitar la comprensión y análisis del flujo de trabajo, eventos, decisiones y actividades paralelas en un sistema específico.

CAPÍTULO 3: METODOLOGÍA

1 Metodología de Investigación

1.1 Investigación Aplicada

La investigación aplicada es un enfoque de investigación que busca aplicar los conocimientos y descubrimientos científicos de manera práctica para abordar problemas reales. Según (Álvarez-Gayou, 2003), este tipo de investigación se enfoca en utilizar esos conocimientos para resolver problemas específicos, mejorar procesos, desarrollar nuevas tecnologías y tomar decisiones informadas en diversos campos como la medicina, la ingeniería y la agricultura. A diferencia de la investigación pura, la investigación aplicada tiene una orientación práctica y busca generar resultados tangibles y beneficios concretos para la sociedad.

1.2 Enfoque Cualitativo

El enfoque cualitativo en la investigación se basa en la recolección y análisis de datos no numéricos para comprender fenómenos complejos y explorar la realidad desde la perspectiva de los participantes. Según (Creswell, 2014), este enfoque se centra en la comprensión profunda de los contextos, significados y experiencias de los individuos, utilizando métodos como entrevistas, observación participante y análisis de contenido. A través del enfoque cualitativo, se busca capturar la riqueza y la diversidad de los datos cualitativos, permitiendo una interpretación más completa de los fenómenos estudiados.

1.3 Metodología de Desarrollo

La metodología de desarrollo de software es un enfoque sistemático para planificar, diseñar, implementar y mantener aplicaciones de software. Según (Sommerville, 2011), una metodología de desarrollo proporciona un conjunto de principios, prácticas y técnicas que guían el proceso de desarrollo, desde la concepción del proyecto hasta la entrega del producto final.

Estas metodologías, como Scrum, XP y RUP⁴, ofrecen estructuras y herramientas para gestionar los recursos, el tiempo y los requisitos del proyecto de manera eficiente. Al seguir una metodología de desarrollo, las organizaciones pueden mejorar la calidad del software, reducir los costos y los riesgos, y asegurar la satisfacción del cliente.

1.3.1 Scrum

Scrum es una metodología ágil ampliamente utilizada en el desarrollo de software. Según (Schwaber & Sutherland, 2017), Scrum se basa en un enfoque iterativo e incremental, donde el equipo de desarrollo trabaja en ciclos cortos y fijos llamados "sprints". Durante cada sprint, se planifican, desarrollan y prueban las funcionalidades del software, permitiendo una entrega continua de valor al cliente. Además, Scrum fomenta la colaboración, la comunicación y la autoorganización del equipo, lo que contribuye a una mayor flexibilidad y capacidad de respuesta ante los cambios. Al adoptar Scrum, las organizaciones pueden mejorar la productividad, la transparencia y la satisfacción del cliente, al tiempo que reducen los riesgos asociados con el desarrollo de software.

Scrum consta de tres fases principales: planificación, desarrollo y revisión. En la fase de planificación, se define el objetivo del sprint y se crean las tareas y el plan de trabajo. Durante la fase de desarrollo, el equipo trabaja en la implementación de las tareas asignadas, siguiendo los principios de colaboración y autoorganización. Finalmente, en la fase de revisión, se realiza una evaluación del trabajo realizado y se obtiene retroalimentación del cliente. (Schwaber & Sutherland, 2017), estas fases se repiten en ciclos iterativos llamados "sprints", que suelen tener una duración de dos a cuatro semanas, permitiendo la entrega continua de valor al cliente y la adaptación a los cambios en el proyecto.

⁴ Rational Unified Process es un marco de desarrollo de software que se basa en principios iterativos e incrementales.

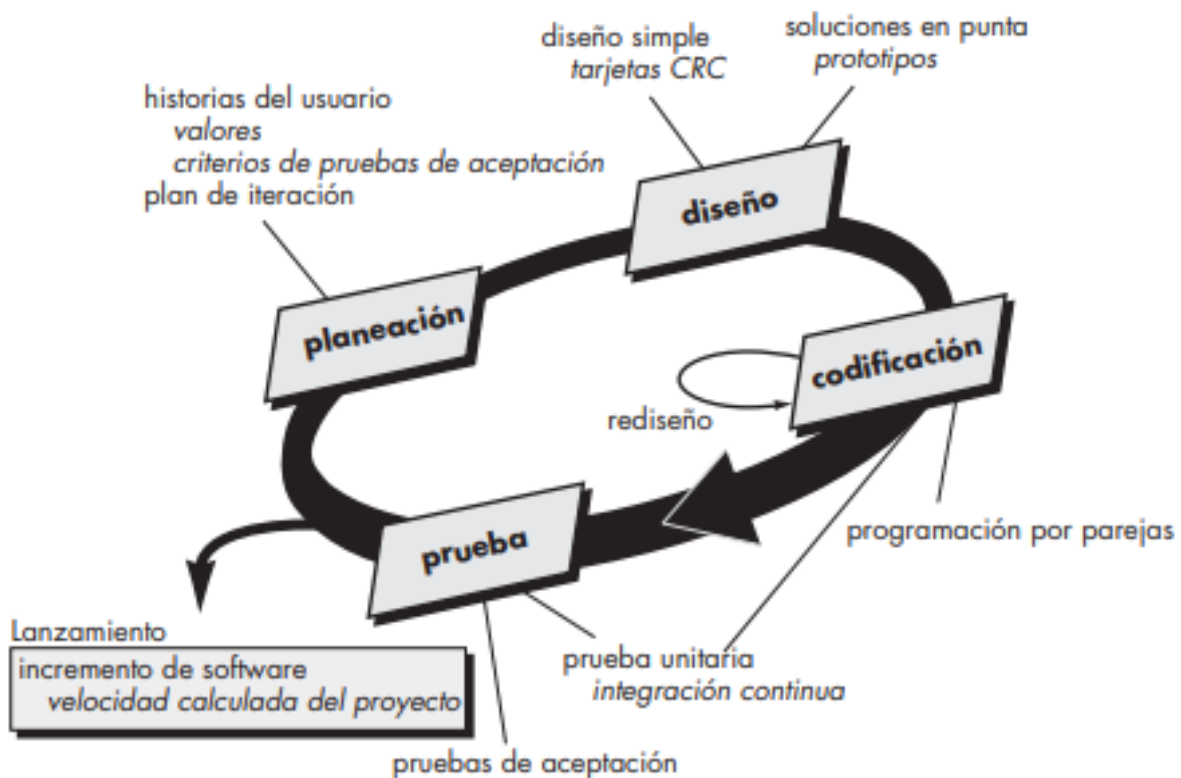
1.3.2 XP

Extreme Programming (XP) es una metodología ágil de desarrollo de software que enfatiza la simplicidad, la comunicación constante y la retroalimentación frecuente. (Pressman, 2010) señala que XP se basa en una serie de prácticas, como la programación en parejas, las pruebas unitarias continuas y la integración frecuente, que permiten a los equipos responder rápidamente a los cambios y entregar software funcional de alta calidad. Además, XP promueve una relación colaborativa entre los desarrolladores y los clientes, fomentando la participación de los clientes en el proceso de desarrollo. Al aplicar XP, las organizaciones pueden lograr una mayor flexibilidad, productividad y satisfacción tanto para el equipo de desarrollo como para los usuarios finales.

El proceso de Programación Extrema (XP) se basa en un enfoque iterativo e incremental en el desarrollo de software. Comprende etapas como la planificación, el diseño, la codificación, las pruebas y la integración. En cada iteración, se priorizan las historias de usuario, se diseñan, implementan y prueban, buscando la retroalimentación constante y la mejora continua. Este enfoque ágil permite adaptarse rápidamente a los cambios y entregar software funcional de alta calidad en tiempos cortos. En la figura 1 se observa el proceso de la metodología XP.

Figura 1

Proceso de la metodología XP



Nota: Proceso cíclico según las iteraciones definidas. Fuente: (Pressman, 2010).

Este proceso se caracteriza por realizar entregas funcionales en cada iteración, lo que facilita la interacción y mejora continua. Comienza con la fase de planeación, donde se definen las funcionalidades y criterios de pruebas de aceptación. Luego, en la fase de diseño, se crea un prototipo o diagrama de UML para visualizar el componente a desarrollar. A continuación, se pasa a la fase de codificación, donde se realizan pruebas unitarias constantes para eliminar errores. La última fase es la de pruebas, donde el cliente valida las funcionalidades solicitadas. Una vez que todas las iteraciones pasan las pruebas de aceptación, se realiza la entrega del producto final y se cierra el proyecto.

1.3.3 Comparativa Scrum vs XP

Tabla 5

Comparación entre la metodología Scrum y XP

Características \ Metodologías	Scrum	XP
Flexible a cambios en el proceso	7	9
Conocimiento previo de la metodología	4	8
Simplicidad del proceso	6	8
Prácticas técnicas	4	9
Roles específicos	9	5
Aplicable a pequeños proyectos	9	9
Total	39	48

En este proyecto, se opta por la metodología XP para el desarrollo de software debido a su énfasis en la flexibilidad y la simplicidad, lo cual se ajusta a las necesidades del proyecto. Dado que la asignación de roles no es un factor relevante debido a las limitaciones de personal, la metodología XP resulta más adecuada para este proyecto.

CAPÍTULO 4: DESARROLLO

1 Desarrollo de la Aplicación

Durante el desarrollo de la aplicación en XP, el equipo realiza integraciones continuas, practica la refactorización⁵ constante y realiza pruebas unitarias constantemente. Se enfoca en un diseño simple y funcional, entregando incrementos de funcionalidad en iteraciones cortas y frecuentes. Esto permite adaptarse a los cambios y obtener retroalimentación temprana de los usuarios.

Herramientas necesarias para el desarrollo de la aplicación:

- ✓ Visual Studio Code
- ✓ JavaScript
- ✓ Node.js
- ✓ HTML
- ✓ CSS
- ✓ Firebase (servicio de autenticación de usuarios, base de datos y hosting)

1.1 Fase Inicial: Análisis de los Requerimientos

Estas actividades iniciales permiten establecer una base sólida para comenzar con las iteraciones en XP, enfocándose en la entrega de valor al cliente de manera incremental y adaptativa.

1.1.1 Historias de Usuarios

Las historias de usuarios son descripciones narrativas breves de las funcionalidades que los usuarios desean en el software. Escritas desde la perspectiva del usuario, expresan qué quieren lograr y por qué es importante para ellos. Estas historias capturan los requisitos desde

⁵ Se refiere al proceso de reestructurar y mejorar el código existente sin cambiar su comportamiento externo.

el punto de vista del usuario y enfocan en proporcionar valor a los usuarios finales. Estas historias tienen similitud con los requerimientos funcionales del sistema. A continuación, listamos los requerimientos funcionales.

1. Crear Registro de usuarios e inicio de sesión
2. Crear categorías de productos
3. Buscar productos por categoría
4. Crear filtro de precios ascendente y descendente
5. Mostrar precios y descuentos
6. Crear carrito de compras.
7. Agregar stock del producto y avisar si no está disponible
8. Generar prefactura
9. Crear un usuario administrador para insertar, actualizar y eliminar productos.
10. Permitir actualizar datos de los clientes en su perfil.

Los requerimientos no funcionales del sistema se mencionan a continuación:

Usabilidad. - Sencillez en el uso de la interfaz de usuario.

Seguridad. - Autenticación, el cifrado de datos y las medidas para proteger el sistema contra amenazas externas.

Escalabilidad. - El sistema tiene que ser adaptativo a cambios en caso de que haya múltiples transacciones y aumento rápido de volúmenes de datos.

Disponibilidad. - Se requiere que el sistema esté disponible las 24 horas.

1.1.2 Priorización de Historias de Usuarios

La priorización de historias de usuario en XP implica considerar el valor de negocio, los riesgos y dependencias, el esfuerzo y tiempo estimado, y la retroalimentación del cliente. Se

seleccionan las historias más importantes para ser implementadas primero, adaptando la priorización según las necesidades cambiantes del proyecto.

1. HU2 Crear categorías de productos
2. HU3 Buscar productos por categoría
3. HU4 Crear filtro de precios ascendente y descendente
4. HU5 Mostrar precios y descuentos si los hubiera
5. HU7 Agregar stock del producto y avisar si no está disponible
6. HU1 Crear Registro de usuarios e inicio de sesión
7. HU9 Crear un usuario administrador para insertar, actualizar y eliminar productos.
8. HI10 Permitir actualizar datos de los clientes en su perfil.
9. HU6 Crear carrito de compras.
10. HU8 Generar prefactura.

1.1.3 Iteraciones

Se define la cantidad de iteraciones compuestas de historias de usuarios o tareas, para ser completados dentro del período de tiempo definido, generalmente entre una a cuatro semanas. Estas iteraciones permiten ajustar y adaptar el trabajo en función de la retroalimentación recibida, garantizando una entrega incremental del software. Para el desarrollo de este proyecto se crean tres iteraciones cada una integrada de varias tareas relacionadas. Cada iteración consta de cuatro fases: planeación, diseño, codificación y pruebas.

Tabla 6

Asignación de historia de usuarios a iteraciones

Iteraciones	Historia de usuarios	Duración
1	2,3,4,5,7	2 semanas
2	1,9,10	1 semana
3	6,8	1 semana

Nota: Cada iteración consta de varias historias de usuario, además se define la duración de cada iteración.

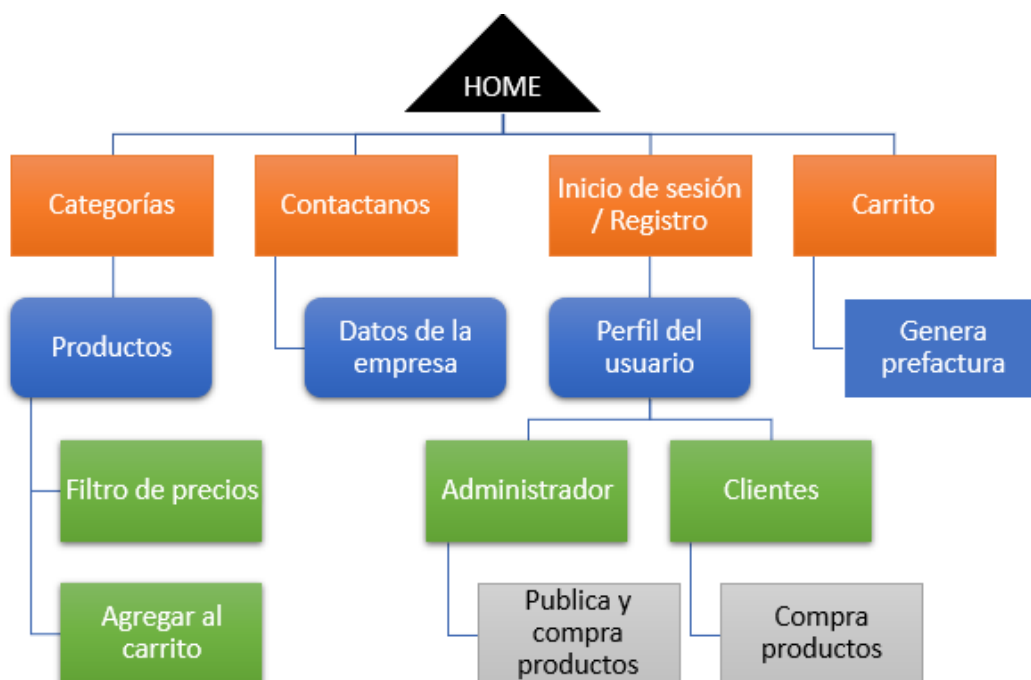
1.2 *Diseño del Sistema*

1.2.1 **Diseño de la Aplicación Web**

Se procede a realizar un diseño de la aplicación web según los requisitos obtenidos.

Figura 2

Árbol de navegación de la aplicación web



Se observa la estructura jerárquica de la aplicación, proporcionando una orientación tanto en la fase de programación como para el usuario final.

1.2.2 Almacenamiento en Cloud Firestore

Se optó por emplear una base de datos NoSQL para el presente proyecto, fundamentando esta elección en varias consideraciones estratégicas. Dada la naturaleza pequeña del proyecto y la posibilidad de cambios estructurales con el crecimiento del negocio, la flexibilidad inherente a las bases de datos NoSQL se presenta como una ventaja significativa. Este tipo de bases de datos permite una adaptación más ágil a modificaciones en el esquema de datos sin imponer restricciones rígidas.

Adicionalmente, se destacan las ventajas proporcionadas por la plataforma Firebase, la cual integra servicios de hosting, base de datos y autenticación de usuarios. Esta integración simplifica la gestión global de servicios al consolidarlos en una única plataforma. Cabe resaltar que la gratuidad de Firebase y su alojamiento en la nube aseguran tanto la economía del proyecto como la seguridad de los datos almacenados.

Finalmente, se consideró que tanto los recursos hardware como humanos son más accesibles económicamente en Firebase en comparación con un entorno local. Esta ventaja se vuelve especialmente relevante en situaciones de escalabilidad rápida, donde Firebase ofrece una solución costo-efectiva para satisfacer las demandas del proyecto en crecimiento.

1.2.3 Funcionamiento de la Base de Datos Cloud Firestore

Para entender el funcionamiento de una base de datos NoSQL es necesario hacer unas similitudes con las bases de datos relacionales.

1. Colección - Tabla:
 - Relacional (SQL): Una tabla.
 - Firestore (NoSQL): Una colección.

2. Documento - Registro:

- Relacional (SQL): Una fila o registro en una tabla.
- Firestore (NoSQL): Un documento.

3. Campos - Columnas:

- Relacional (SQL): Campos predefinidos en una tabla.
- Firestore (NoSQL): Campos dinámicos dentro de un documento.

Los documentos se estructuran con pares de clave valor a continuación un ejemplo de la estructura en Firestore:

```
colección: "usuarios"
```

```
|
```

```
|-- documento: "usuario1" (ID de documento automático)
```

```
| |
```

```
| |-- campo: "nombre" - "Diana Xin"
```

```
| |-- campo: "correoElectrónico" - "diana@yahoo.com"
```

```
|
```

```
|-- documento: "usuario2" (ejemplo de un ID =xJ1zERmnqUhqo5HOuwwW)
```

```
|
```

```
|-- campo: "nombre" - "Ana Calderón"
```

```
|-- campo: "correoElectrónico" - "ana@yahoo.com"
```

La clave en el documento “usuario1” son “nombre” y “correoElectrónico” estos no se pueden repetir dentro del mismo documento, mientras que los valores son “Diana Xin” y “diana@yahoo.com” respectivamente.

En este proyecto, creamos cuatro colecciones llamadas "usuarios", "compras", "productos" y "categorías". Para conectar estas colecciones entre sí, usamos referencias clave en la estructura de los documentos en Firestore.

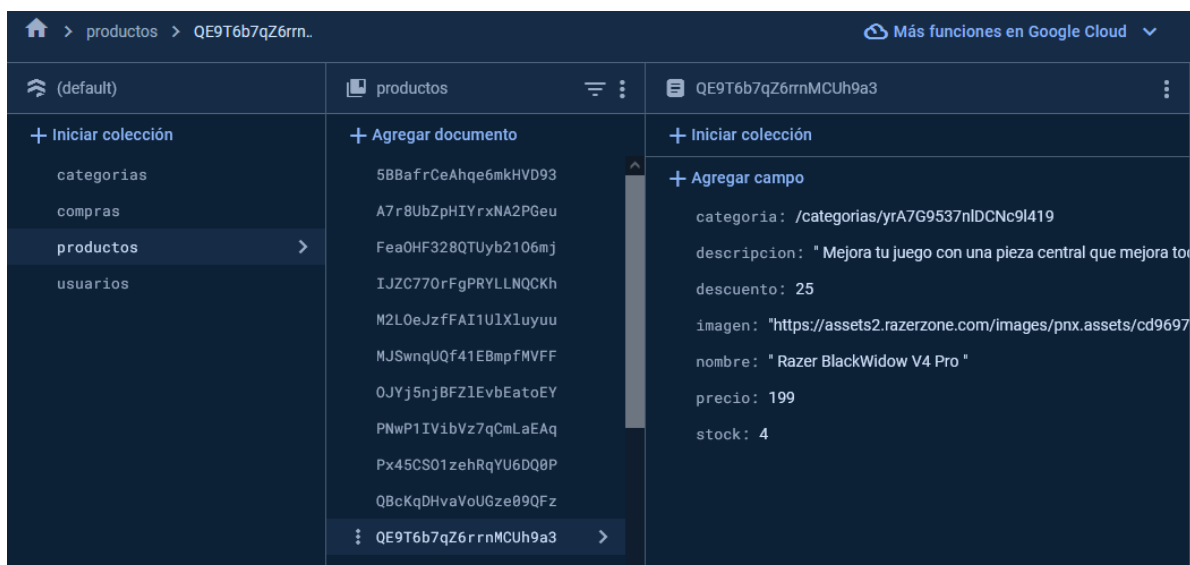
En específico, la relación entre productos y categorías se configura mediante una referencia en cada documento de "productos", donde se incluye la clave "categoría". Esta estructura facilita la organización y clasificación de los productos dentro de categorías específicas.

Por otro lado, las compras se asocian con los usuarios mediante la utilización de la clave "usuarioId". Esta información se extrae directamente desde la colección "usuarios" y se incorpora en cada documento de la colección "compras". Este proceso se ejecuta cuando un usuario inicia sesión y procede a realizar una compra, estableciendo así una vinculación directa entre la transacción y el perfil del usuario.

A continuación, se presenta un fragmento representativo de las colecciones y documentos implementados en Firestore.

Figura 3

Colecciones y documentos en Cloud Firestore



Nota: Se muestra las colecciones y documentos creados para el proyecto.

1.3 Iteración 1

1.3.1 Planeación

En esta fase se colocan las historias de usuarios prioritarias y relacionadas. Además, se plantean los objetivos y criterios de aceptación.

1.3.1.1 Historias de Usuarios Prioridad Alta

Tabla 7

Historia de usuario número 2

Historias de usuario	
Número: 2	Nombre de historia: crear categorías de productos.
Programador: Simón Aucatoma	Iteración: 1
Prioridad en negocio: alta	Prioridad en desarrollo: media
Descripción: el usuario podrá visualizar los productos con descripción y fotos en seis categorías.	
Observaciones: Ninguna	

Tabla 8

Historia de usuario número 3

Historias de usuario	
Número: 3	Nombre de historia: Buscar productos por categoría.
Programador: Simón Aucatoma	Iteración: 1
Prioridad en negocio: alta	Prioridad en desarrollo: media
Descripción: el usuario podrá buscar los productos mediante la opción de categorías.	
Observaciones: Ninguna	

Tabla 9*Historia de usuario número 4*

Historias de usuario	
Número: 4	Nombre de historia: Crear filtro de precios ascendente y descendente
Programador: Simón Aucatoma	Iteración: 1
Prioridad en negocio: alta	Prioridad en desarrollo: media
Descripción: el usuario podrá visualizar los productos en cada categoría según el orden de precios ascendente o descendente.	
Observaciones: Ninguna	

Tabla 10*Historia de usuario número 5*

Historias de usuario	
Número: 5	Nombre de historia: Mostrar precios y descuentos.
Programador: Simón Aucatoma	Iteración: 1
Prioridad en negocio: alta	Prioridad en desarrollo: media
Descripción: el usuario podrá visualizar los productos con sus precios y en caso de haber descuento se muestra junto al precio normal.	
Observaciones: Ninguna	

Tabla 11*Historia de usuario número 7*

Historias de usuario	
Número: 7	Nombre de historia: Agregar stock del producto y avisar si no está disponible.
Programador: Simón Aucatoma	Iteración: 1
Prioridad en negocio: Alta	Prioridad en desarrollo: media
Descripción: los productos muestran la cantidad del stock y en caso de haberse agotado tendrá el aviso de no disponible.	
Observaciones: Ninguna	

1.3.1.2 Objetivos de la Iteración

- Buscar productos por categoría
- Visualizar productos con fotos y descripción
- Listar productos por precio más barato primero

1.3.1.3 Criterios de Aceptación

- Los productos pueden ser encontrados según la categoría seleccionada.
- Los productos deben incluir mínimo una foto de muestra
- Los productos se ordenan por filtro de precios.

1.3.2 Diseño

1.3.2.1 Diagramas UML para la Visualización de Productos

Se diseña el caso de uso para la primera iteración con las principales funcionalidades.

Figura 4

Caso de uso del catálogo virtual

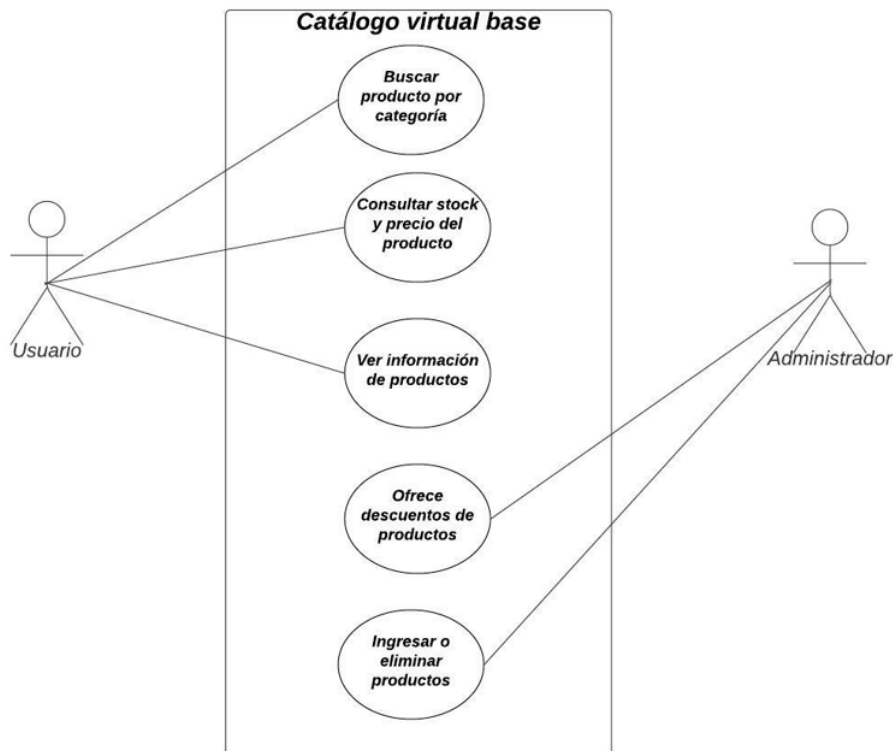
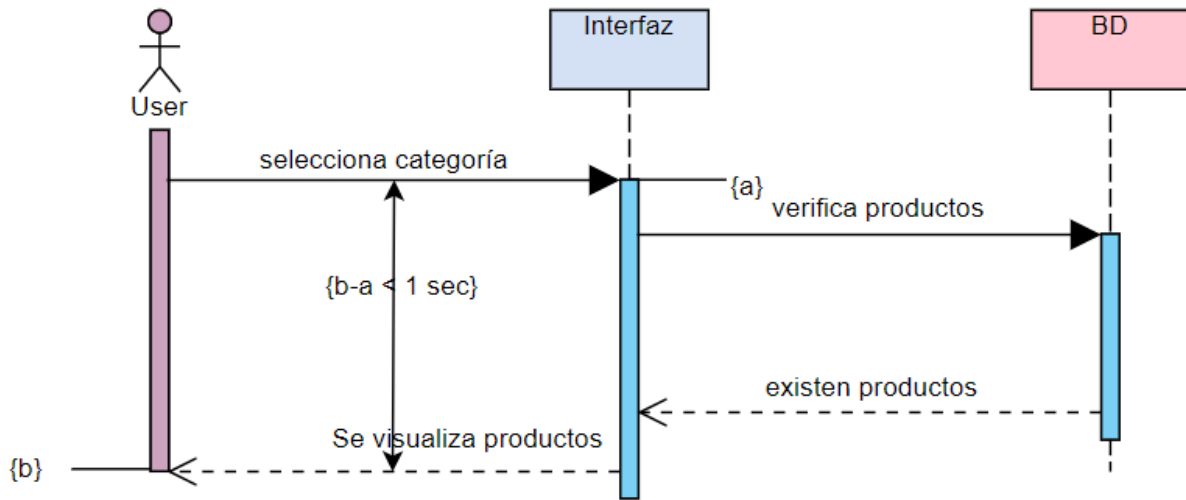


Figura 5

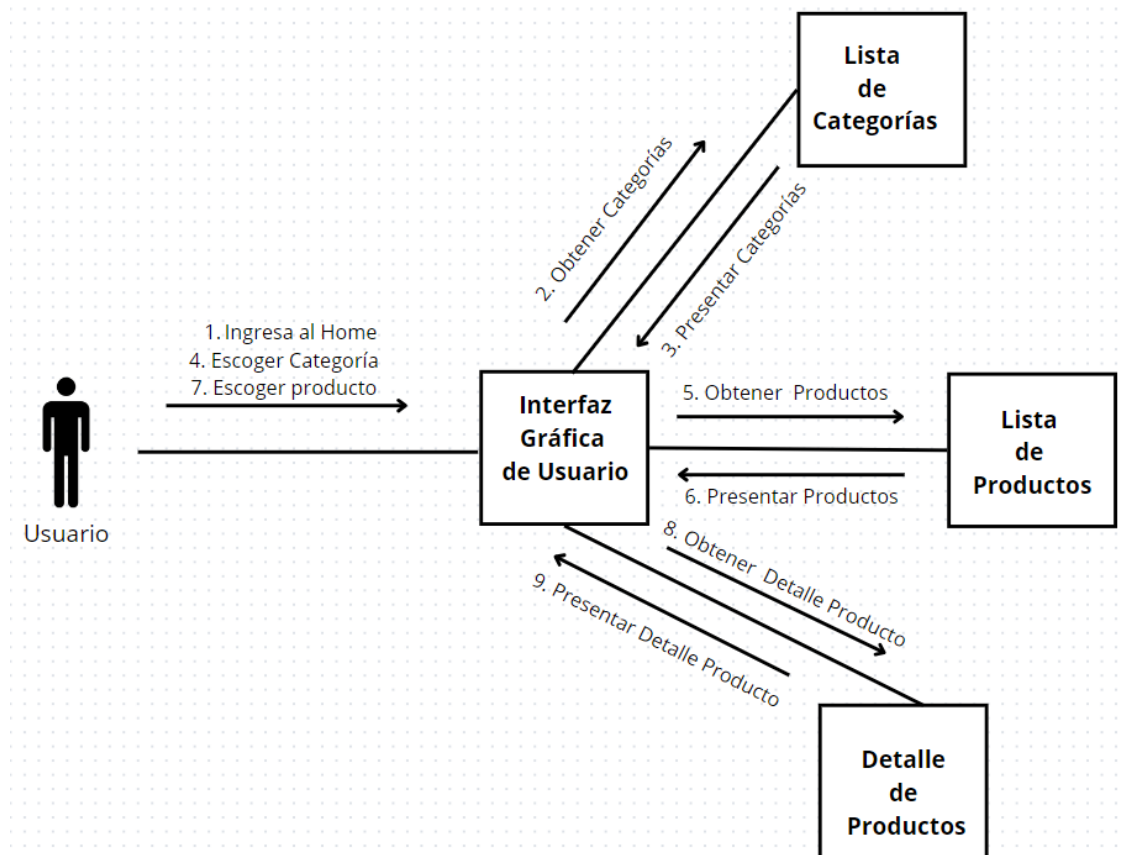
Diagrama de secuencia para buscar producto



Nota: Se observa la secuencia de los objetos para buscar productos.

Figura 6

Diagrama de colaboración para buscar producto



Este diagrama proporciona a los desarrolladores una herramienta útil para iniciar la creación de su primer conjunto de pautas en el desarrollo de los componentes de este proceso.

1.3.3 Codificación

Se implementa las funcionalidades establecidas en esta iteración mediante la programación. También, se realiza las pruebas unitarias constantemente para corregir errores de código. Con el propósito de facilitar la lectura del código y asegurar un mantenimiento eficiente a largo plazo, se adopta como convención la siguiente nomenclatura:

Nombre de variables y funciones

- Utilizar nombres significativos.
- Seguir la convención de capitalización camelCase.
- Evitar nombres excesivamente largos.

Comentarios

- Incluir comentarios claros y concisos.
- Evitar comentarios obvios o innecesarios.

Estructura del código

- Organizar el código de manera lógica y coherente.
- Separar visualmente bloques de código con líneas en blanco.

Manejo de errores

- Utilizar excepciones y mensajes de error descriptivos.

Versionado de código:

- Utilizar GitHub para el control de versiones.
- Mantener mensajes de commit descriptivos según la versión.

1.3.4 Pruebas

1.3.4.1 Pruebas unitarias

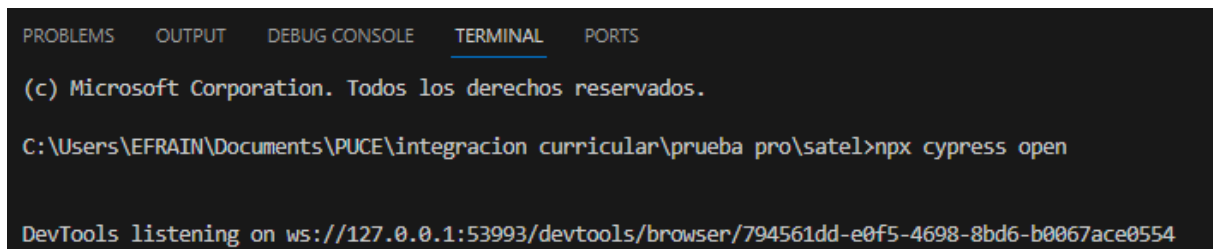
Se realiza las pruebas unitarias para verificar la correcta implementación de las funcionalidades. Se adopta esta medida como previsión para corregir los errores antes de que los usuarios los experimenten. De esta manera se anticipa a corregir los errores antes que los usuarios.

En este proyecto para las pruebas unitarias hemos optado por utilizar la herramienta Cypress, que se encarga de verificar de manera individual las funciones y fragmentos de código presentes en cada archivo de código fuente desarrollada en JavaScript. El proceso se inicia instalando las dependencias necesarias en el entorno de desarrollo JavaScript de Visual Studio Code. Posteriormente, procedemos a ejecutar la herramienta mediante el comando "npx cypress open". En este punto, se nos brinda la opción de seleccionar el navegador de preferencia para obtener un informe detallado de las pruebas realizadas.

Durante la ejecución, los fragmentos de código que presentan errores se resaltan en color rojo, acompañados de un mensaje de error que facilita la identificación de posibles problemas. En caso de encontrar errores, se lleva a cabo la corrección pertinente en el código subyacente. En contraste, si todas las verificaciones son exitosas y el código es correcto, las funciones correspondientes se presentan en un estado aprobatorio resaltado en color verde, indicando que han pasado las pruebas con éxito. Este proceso sistemático y detallado de verificación y corrección contribuye significativamente a la calidad y confiabilidad del código desarrollado.

Figura 7

Pantalla de la ejecución de Cypress



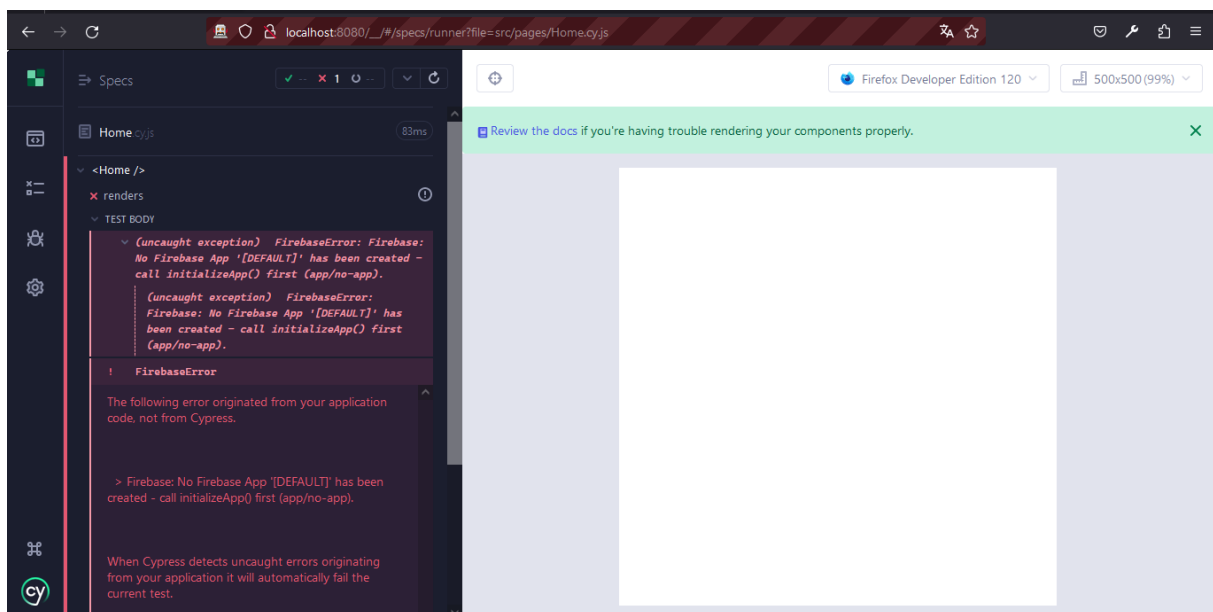
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
(c) Microsoft Corporation. Todos los derechos reservados.  
C:\Users\EFRAIN\Documents\PUCE\integracion curricular\prueba pro\satel>npx cypress open  
DevTools listening on ws://127.0.0.1:53993/devtools/browser/794561dd-e0f5-4698-8bd6-b0067ace0554
```

Nota: se ejecuta Cypress desde la consola terminal de Visual Studio Code.

Se procede a realizar la ejecución de pruebas unitarias en el archivo fuente "home.js", el cual se asocia con la página de inicio de la aplicación. Durante este proceso, se detectaron errores relacionados con la secuencia incorrecta de llamadas, originados por intentar operaciones en Firebase sin haber inicializado previamente la conexión. Se detallan estos inconvenientes en el informe generado por Cypress, que se muestra en la figura subsiguiente.

Figura 8

Error de llamadas en el home



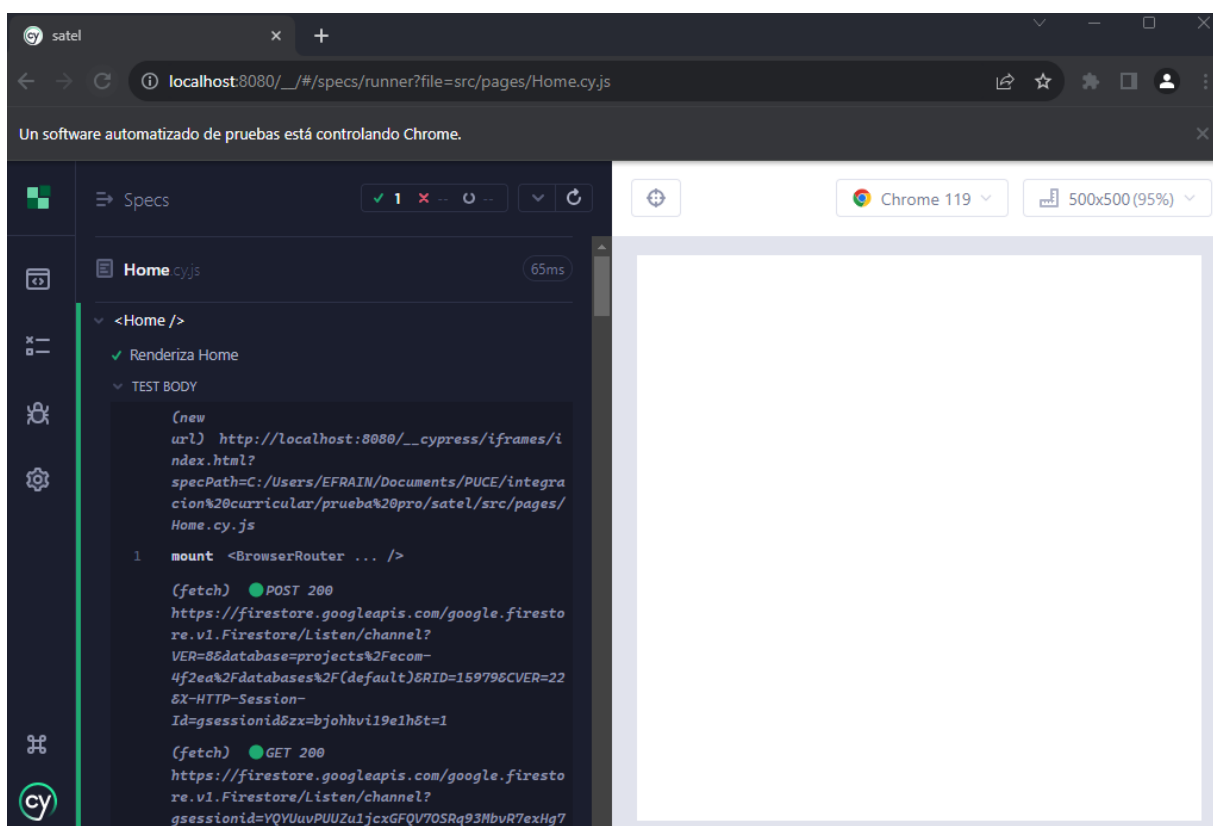
Nota: Firebase aún no está inicializado cuando se intenta usar sus servicios.

Luego de revisar detenidamente el informe generado por Cypress, se lleva a cabo un proceso de corrección, enfocándose en la organización de las llamadas para garantizar que los servicios de Firebase se inicialicen primero. Además, se realiza una verificación exhaustiva de las credenciales de configuración proporcionadas por Firebase para asegurar su exactitud y alineación con los requisitos de nuestro proyecto.

Esta fase de corrección es esencial para asegurar que la inicialización de los servicios fundamentales, como Firebase, ocurra de manera adecuada y antes de cualquier operación que dependa de estos servicios. Asimismo, la verificación de las credenciales es crucial para garantizar una conexión segura y válida con los recursos de Firebase.

Figura 9

Prueba unitaria en las llamadas del home exitosa

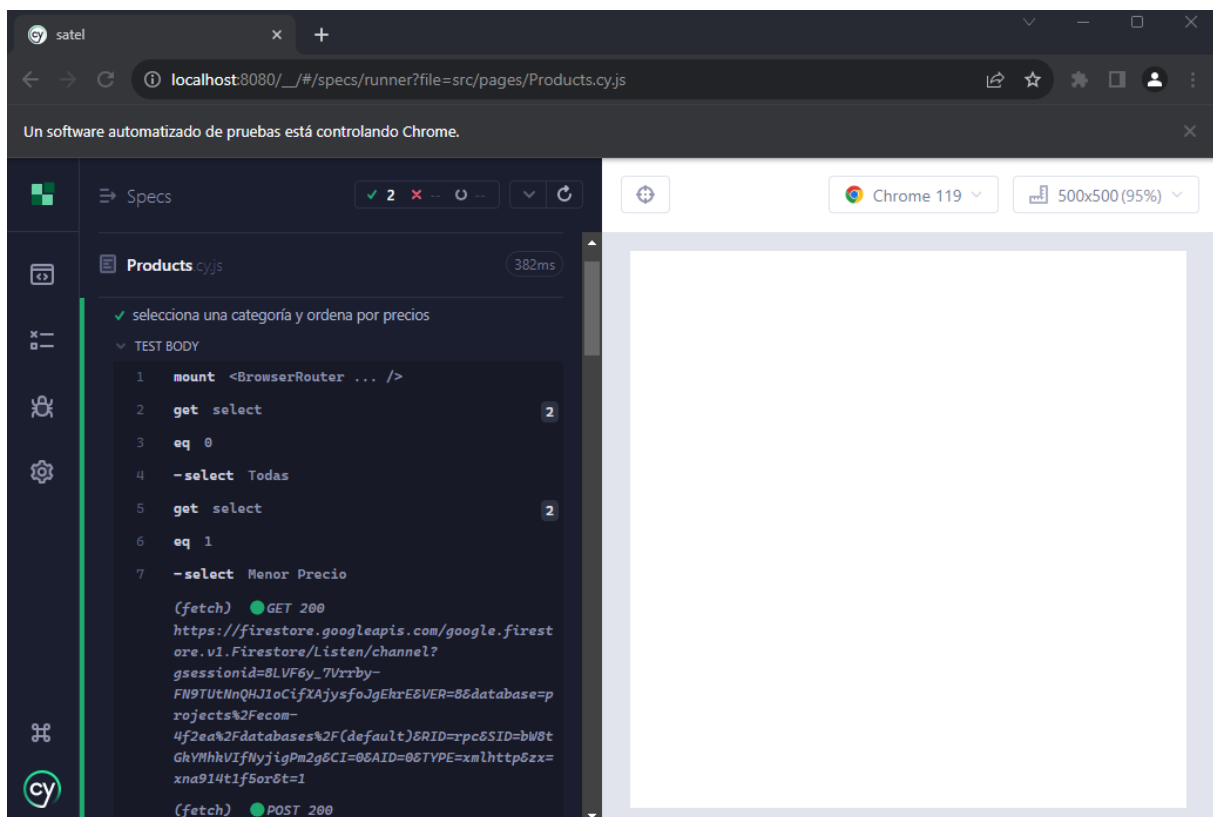


La prueba unitaria se ha vuelto a ejecutar utilizando la herramienta Cypress, y se evidencia que en esta ocasión la prueba ha sido superado exitosamente. Este resultado positivo indica que las correcciones implementadas, particularmente la organización de las llamadas y la verificación de las credenciales de configuración de Firebase, han surtido efecto favorable en la funcionalidad deseada. Este proceso de revisión y ajuste resalta el compromiso con la calidad y la integridad del código, contribuyendo a un desarrollo más robusto y confiable de la aplicación.

La siguiente prueba ejecutada es para verificar los productos se visualizan por categoría y orden de precios.

Figura 10

Prueba unitaria para el filtro de precios y categorías



Nota: las funcionalidades de categorías y filtro de precios están correctas.

1.3.4.2 Pruebas de Aceptación

Las pruebas de aceptación para cada historia de usuario planteadas en esta iteración son:

Tabla 12

Prueba de aceptación de la historia de usuario número 2

Caso de Prueba de Aceptación	
Código: PA_HU2	Historia de usuario: HU02
Nombre: Creación de categorías de productos.	
Descripción: se probará el sistema con la visualización en la interfaz de seis categorías de productos.	
Condiciones de ejecución: se verifica que haya productos en las categorías.	
Pasos de ejecución: <ol style="list-style-type: none">1. Ingresar al Home2. Verificar Categorías de productos.	
Resultado esperado: existen productos en categorías.	
Evaluación de la prueba: Aceptado	

Tabla 13

Prueba de aceptación de la historia de usuario número 3

Caso de Prueba de Aceptación	
Código: PA_HU3	Historia de usuario: HU03
Nombre: Buscar productos por categoría.	
Descripción: se probará el sistema con la implementación de un filtro por categorías de productos, al ser seleccionado una categoría se desplegará la lista de productos relacionados a la misma clase.	
Condiciones de ejecución: se verifica el filtro de las categorías y presentación de los productos segmentados.	
Pasos de ejecución: <ol style="list-style-type: none">1. Ingresar al Home2. Verificar filtro de Categorías.3. Escoger un ítem del filtro4. Verificar despliegue de productos segmentados.	
Resultado esperado: los productos pueden ser buscado por categorías.	
Evaluación de la prueba: Aceptado	

Tabla 14*Prueba de aceptación de la historia de usuario número 4*

Caso de Prueba de Aceptación	
Código: PA_HU4	Historia de usuario: HU04
Nombre: Creación de filtro de precios.	
Descripción: Se prueba sistema con la lista de productos mostrados en orden ascendente y descendente según el filtro seleccionado.	
Condiciones de ejecución: se verifica que los productos estén ordenados según los precios.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Ingresar al Home. 2. Verificar Categorías de productos. 3. Escoger filtro de menor a mayor precio. 4. Presentación de productos con precios más baratos primero. 	
Resultado esperado: los productos se ordenan según el precio ascendente o descendente.	
Evaluación de la prueba: Aceptado	

Tabla 15*Prueba de aceptación de la historia de usuario número 5*

Caso de Prueba de Aceptación	
Código: PA_HU5	Historia de usuario: HU05
Nombre: Mostrar precios y descuentos.	
Descripción: se muestra precios a todos los productos, adicionalmente se visualiza si tiene descuento algún producto.	
Condiciones de ejecución: se verifica precios y descuento de productos.	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar una categoría de productos. 2. Observar los precios y descuentos en la lista de productos. 	
Resultado esperado: los productos se visualizan con sus precios.	
Evaluación de la prueba: Aceptado	

Tabla 16*Prueba de aceptación de la historia de usuario número 7*

Caso de Prueba de Aceptación	
Código: PA_HU7	Historia de usuario: HU07
Nombre: Controlar stock de productos.	
Descripción: se verifica cantidad de productos en stock, además notificación de productos agotados.	
Condiciones de ejecución: productos disponibles en la tienda.	
Pasos de ejecución: <ol style="list-style-type: none">1. Seleccionar productos de la categoría.2. Agregar a carrito.3. Generar comprar y validad stock	
Resultado esperado: los productos se visualizan con stock actualizado.	
Evaluación de la prueba: Aceptado	

En respuesta a la retroalimentación del usuario, se solicita la modificación del color de la aplicación y la incorporación del logo de la empresa. Además, se requiere que, en la parte inferior, se visualice la misión de la empresa. Posteriormente, una vez realizadas las modificaciones solicitadas, se presenta la nueva versión al usuario para su revisión y aprobación.

Todas las pruebas antes descritas son aprobadas por el usuario, de esta manera se valida que las funcionalidades están satisfechas y se procede a pasar a la siguiente iteración.

1.4 Iteración 2

1.4.1 Planeación

En esta fase se colocan las historias de usuarios de prioridad alta, media y relacionadas. Además, se plantean los objetivos a cumplirse en esta fase y los criterios de la prueba de aceptación.

1.4.1.1 Historias de Usuarios

Tabla 17

Historia de usuario número 1

Historias de usuario	
Número: 1	Nombre de historia: Registro de usuarios e inicio de sesión
Programador: Simón Aucatoma	Iteración: 2
Prioridad en negocio: alta	Prioridad en desarrollo: media
Descripción: el usuario podrá registrarse mediante correo electrónico. Si el usuario quiere iniciar sesión con contraseña errónea notificarle.	
Observaciones:	

Tabla 18

Historia de usuario número 9

Historias de usuario	
Número: 9	Nombre de historia: Crear un usuario administrador para insertar y eliminar productos.
Programador: Simón Aucatoma	Iteración: 2
Prioridad en negocio: media	Prioridad en desarrollo: alta
Descripción: el usuario administrador puede agregar, modificar o eliminar un producto de la tienda desde su cuenta.	
Observaciones:	

Tabla 19

Historia de usuario número 10

Historias de usuario	
Número: 10	Nombre de historia: Permitir actualizar datos de los clientes en su perfil.
Programador: Simón Aucatoma	Iteración: 2
Prioridad en negocio: media	Prioridad en desarrollo: alta
Descripción: los usuarios pueden cambiar datos como contraseña y correo en su perfil.	
Observaciones:	

1.4.1.2 Objetivos de la Iteración

- Registrar usuarios y validar autenticación de usuarios
- Crear un usuario administrador para hacer CRUD⁶ a la colección de productos.

1.4.1.3 Criterios de Aceptación

- Permite el registro de usuarios y validación de usuarios registrados.
- El usuario administrador puede eliminar o insertar productos al catálogo.

1.4.2 Diseño

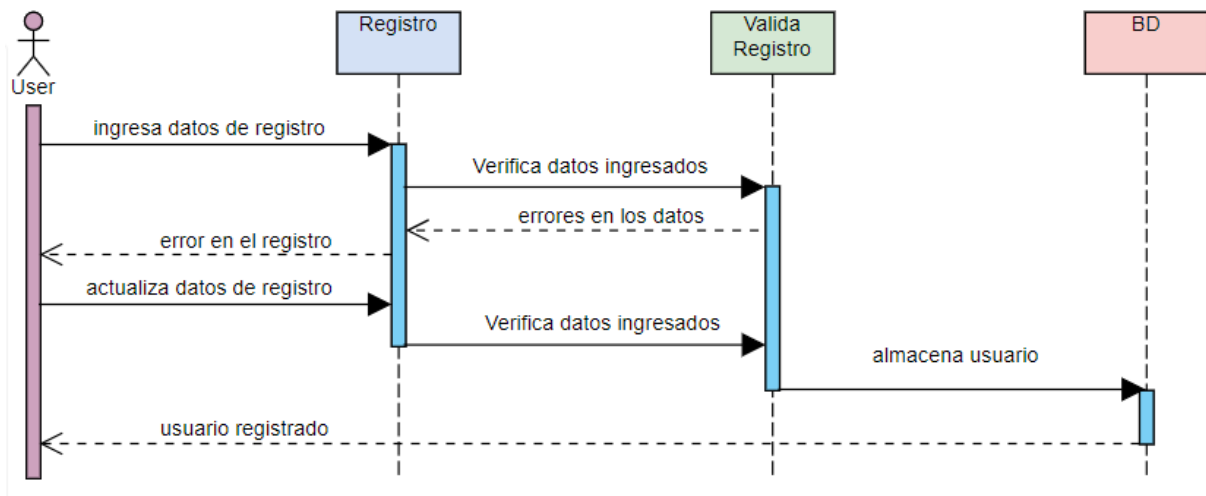
1.4.2.1 Diagramas UML para Registro e Inicio de Sesión.

Se elaboraron los diagramas de secuencia y colaboración correspondientes a la segunda iteración con el propósito de proporcionar una guía de las acciones a seguir. Estos diagramas están específicamente enfocados en las operaciones relacionadas con el registro e inicio de sesión.

⁶ Es un acrónimo que se refiere a las operaciones básicas en sistemas que gestionan datos almacenados (crear, leer, actualizar y eliminar).

Figura 11

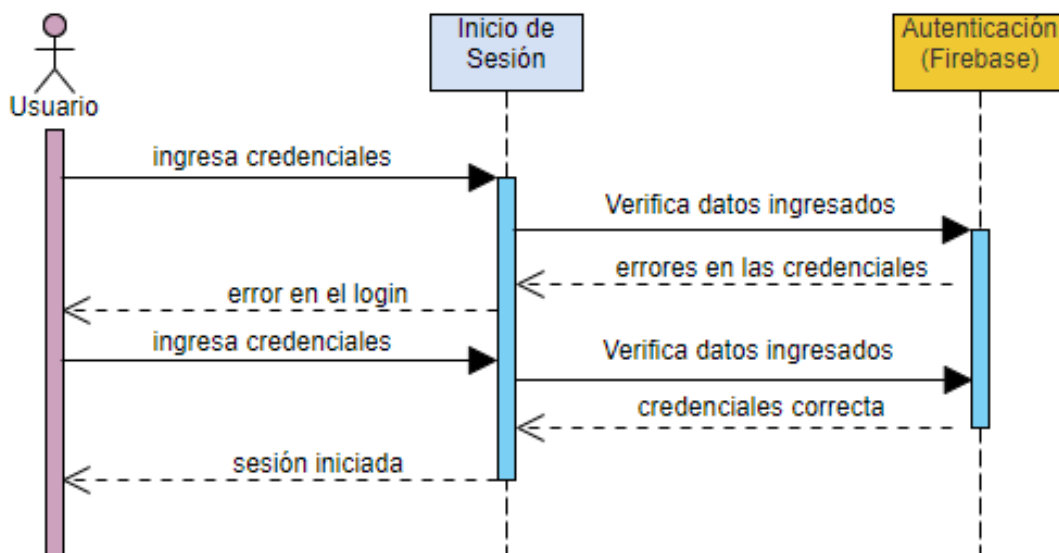
Diagrama de secuencia del registro de usuario



Nota: la base de datos y el servicio de autenticación están alojadas en firebase.

Figura 12

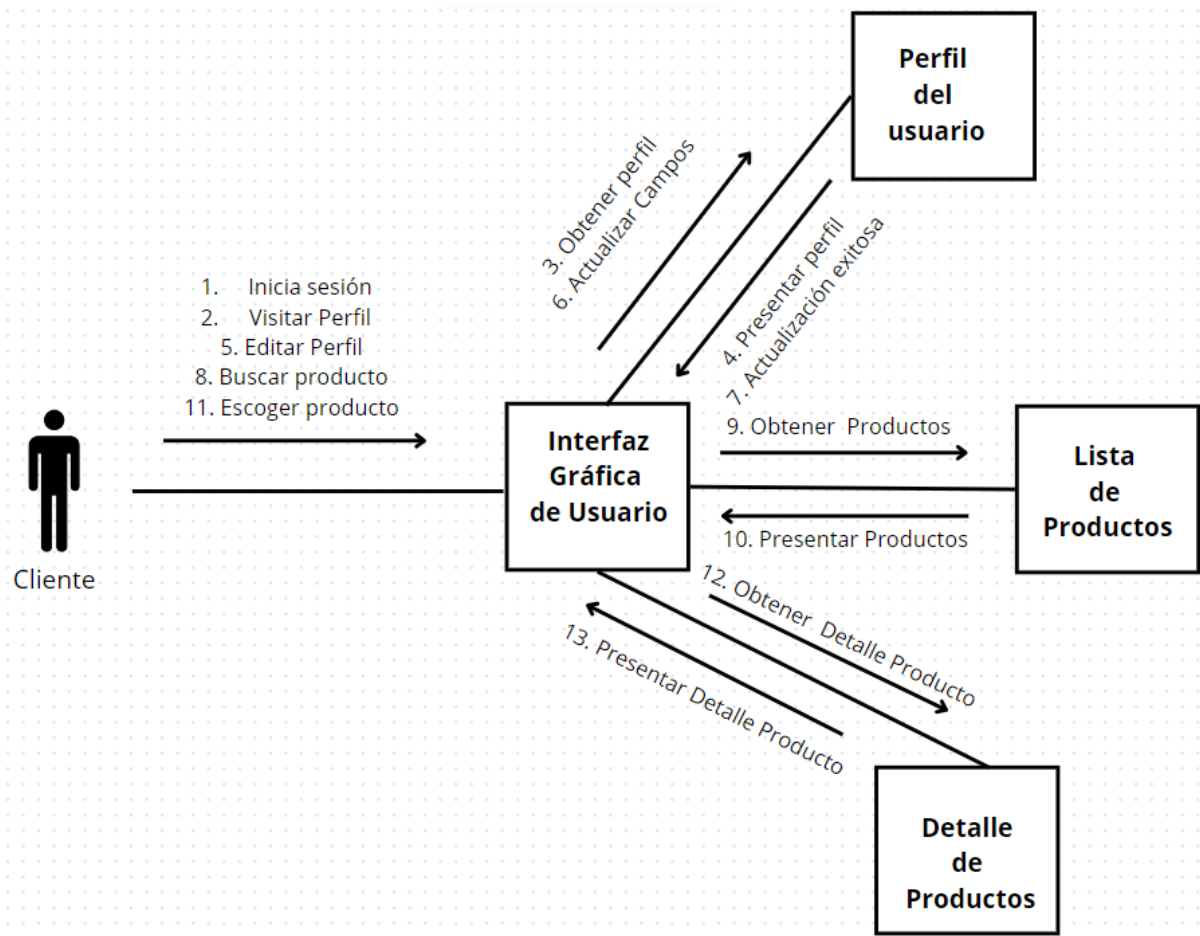
Diagrama de secuencia del inicio de sesión



Nota: La autenticación se realiza en tiempo real con los datos almacenados en Firebase.

Figura 13

Diagrama de colaboración de actualización de perfil

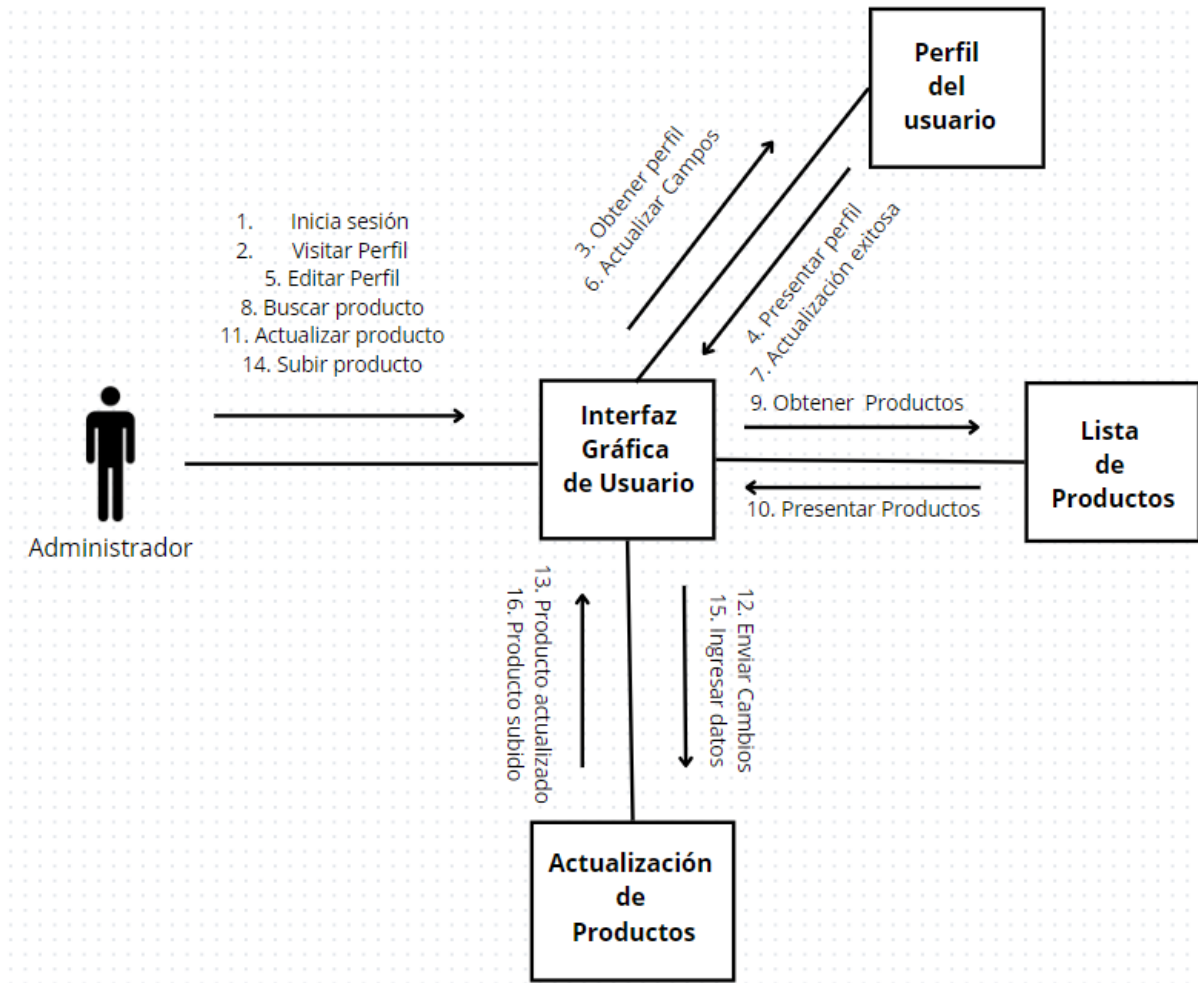


Nota: Cliente actualiza perfil y visualiza productos de la tienda

Los usuarios registrados, una vez autenticados con sus credenciales, tienen la capacidad de acceder a su cuenta, donde pueden actualizar su perfil, visualizar productos y agregarlos al carrito para su compra posterior. Por otro lado, el administrador posee las mismas funciones que un usuario registrado, con la adición de realizar operaciones CRUD en relación con los productos. Esto implica la capacidad de agregar nuevos productos al catálogo, actualizar información sobre productos ya publicados y eliminar aquellos productos que ya no se desean mostrar en la tienda en línea.

Figura 14

Diagrama de colaboración del administrador



Nota: se visualiza los permisos del administrador para hacer crud a los productos.

Estos diagramas se han concebido con el propósito de brindar una visualización clara y detallada de las funciones esenciales relacionadas con el registro de usuarios y la gestión del CRUD de productos. En el caso del usuario administrador se agrega manualmente desde la consola de firebase un campo adicional “EsAdmin” del tipo boolean luego de que el documento se haya creado mediante el registro del usuario.

1.4.3 Codificación

Se procede a programar las funcionalidades especificadas en las historias de usuarios y los diagramas UML el cual permite entender con mayor facilidad el flujo de los componentes a desarrollar. Adicionalmente se realizan las pruebas unitarias constantemente para conseguir un código de alta valor. Con el propósito de facilitar la lectura del código y asegurar un mantenimiento eficiente a largo plazo, se adopta como las convecciones, nomenclatura y repositorio mencionados en la primera iteración (variables y funciones camelCase, repositorio GitHub, comentarios, entre otros). A continuación, se muestra una figura con los estándares de codificación antes descrito:

Figura 15

Estándar de codificación adoptado

```
77
78     // Mapea todos los productos
79     let products = querySnapshot.docs.map(async (doc) => {
80     // Obtiene la categoría del producto
81     const snapCategoria = await getDoc(doc.data()["categoria"])
82     // Construye un objeto de datos del producto
83     let docData = doc.data()
84     docData["categoria"] = snapCategoria.data()["nombre"]
85     // Devuelve el objeto del producto
86     return { id: doc.id, data: docData }
87     });
88
89     // Mapea productos con descuento
90     let discountProducts = querySnapshotD.docs.map(async (doc) => {
91     const snapCategoria = await getDoc(doc.data()["categoria"])
92     let docData = doc.data()
93     docData["categoria"] = snapCategoria.data()["nombre"]
94     return { id: doc.id, data: docData }
95     });
96
```

Nota: La figura es un extracto del código fuente home.js el cual incluye la codificación de la página principal de la aplicación web.

1.4.4 Pruebas

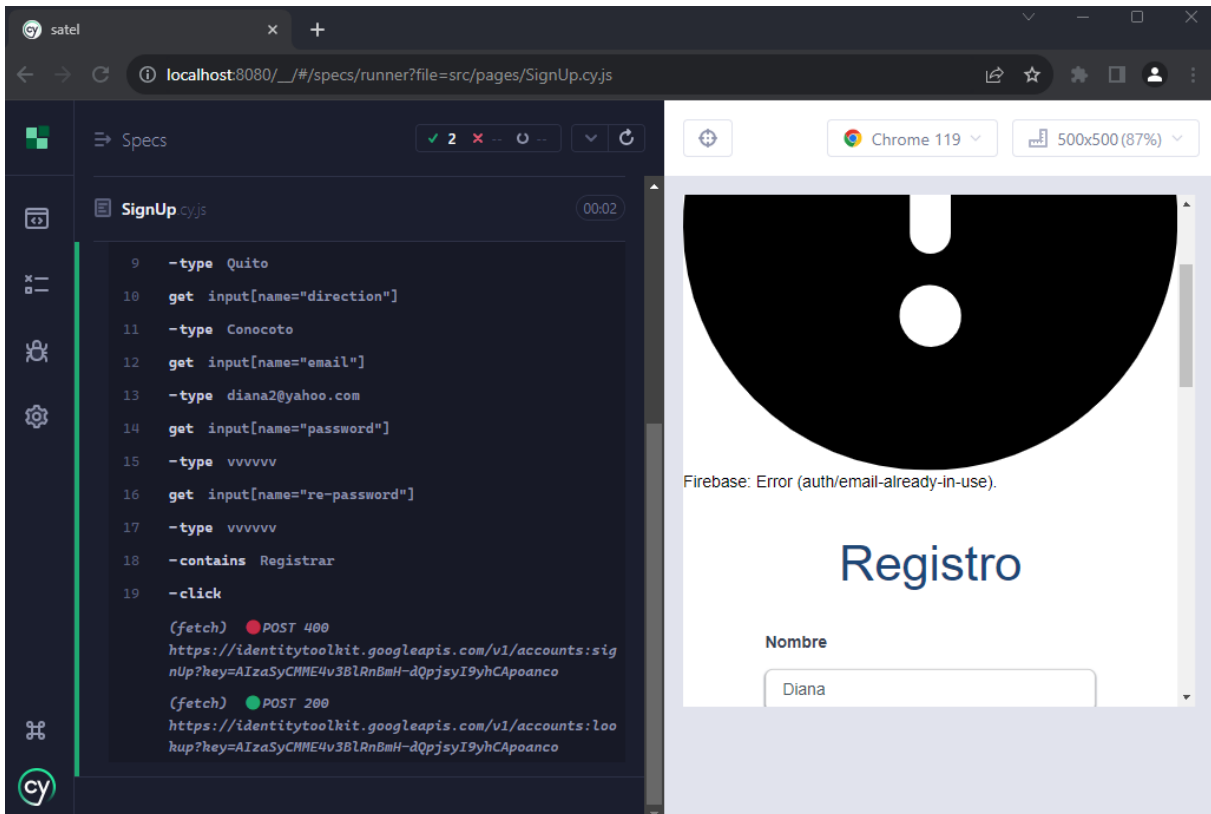
1.4.4.1 Pruebas Unitarias

Se realizan varias pruebas unitarias al implementar las nuevas funcionalidades a fin de verificar la correcta integración de los diferentes componentes.

Se realiza la prueba unitaria para la funcionalidad de creación de usuario, evidenciando que, si un usuario ya está registrado con el mismo correo, la creación no se permite. Además, se observa que se muestra el mensaje de error predeterminado de Firebase.

Figura 16

Errores de Firebase durante el registro

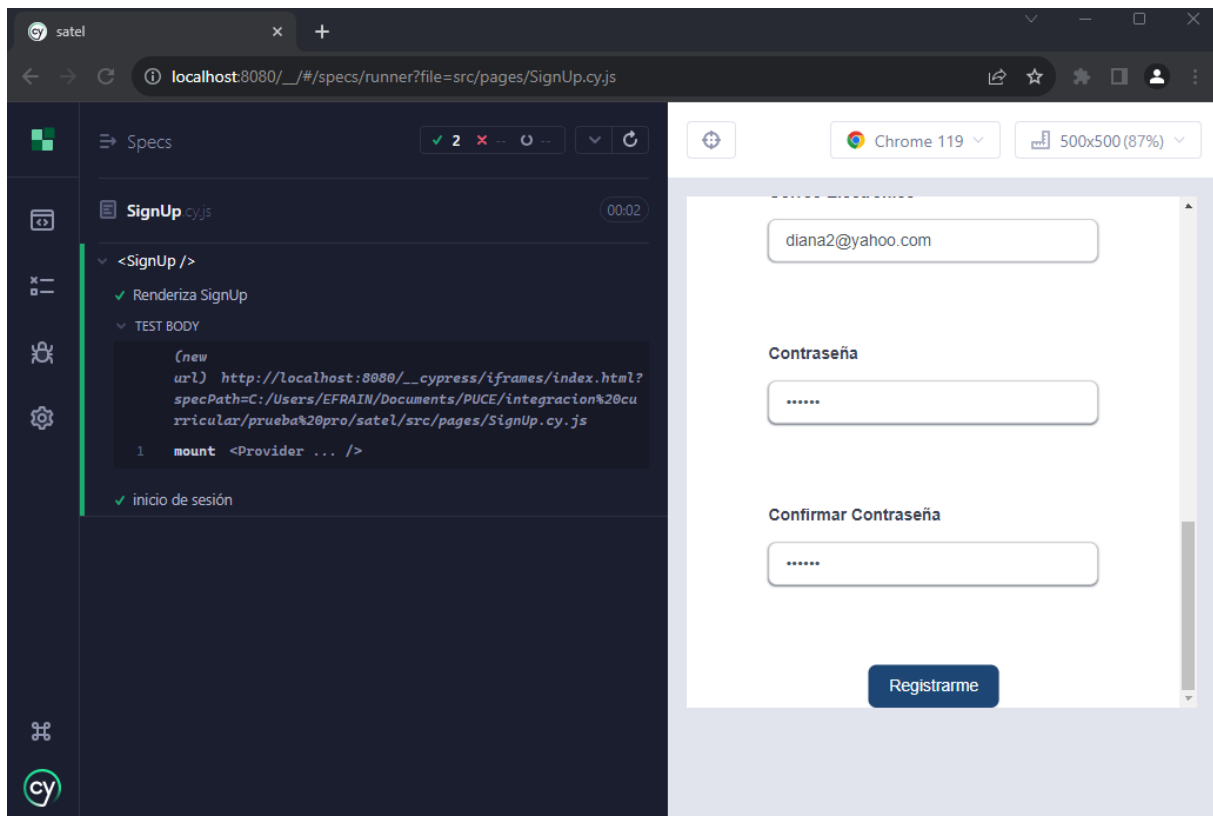


Nota: se visualiza el error en inglés presentado por el servicio de Firebase

Para abordar este problema, se implementa el método "catch" junto con una condición, permitiendo la emisión de un mensaje específico para el caso de registro de usuario “el correo ya está en uso, por favor ingrese otro”. En caso contrario, se imprime el error del sistema.

Figura 17

Prueba de registro de usuario exitosa



Nota: figura obtenida a través de la herramienta Cypress en el navegador web.

Las pruebas unitarias fueron corregidas exitosamente durante esta segunda iteración, lo que contribuye a reducir la probabilidad de errores de codificación en las futuras integraciones. Asimismo, estas correcciones aseguran que las pruebas de aceptación no presenten inconvenientes para los usuarios al ser evaluadas en la siguiente etapa del desarrollo.

1.4.4.2 Pruebas de Aceptación

Estas pruebas se realizan según los requerimientos definidos en las historias de usuarios para validar con los criterios de aceptación establecidos en la fase de planeación.

Tabla 20

Prueba de Aceptación de la historia de usuario número 1

Caso de Prueba de Aceptación	
Código: PA_HU1	Historia de usuario: HU01
Nombre: Inicio de sesión y registro de usuarios	
Descripción: se probará el sistema con la creación de nuevos usuarios y el inicio de sesión.	
Condiciones de ejecución: se verifica iniciar sesión con usuario incorrecto luego con datos verdaderos. Se hace registro de usuario y se valida si permite el acceso con los datos de registro.	
Pasos de ejecución: <ol style="list-style-type: none">1. Registrar usuario2. Verificar datos de registro en perfil.3. Cerrar sesión e iniciar sesión nuevamente.	
Resultado esperado: Almacena de manera correcta los usuarios y valida inicio de sesión.	
Evaluación de la prueba: Aceptado	

Tabla 21

Prueba de Aceptación de la historia de usuario número 9

Caso de Prueba de Aceptación	
Código: PA_HU9	Historia de usuario: HU09
Nombre: Crear usuario administrador	
Descripción: el usuario administrador tiene permisos de insertar, modifica y elimina los productos en la base de datos.	
Condiciones de ejecución: usuario administrador hace CRUD a productos en la base de datos.	
Pasos de ejecución: <ol style="list-style-type: none">1. Iniciar sesión usuario administrador.2. Agregar nuevo producto3. Actualiza stock de productos4. Eliminar un producto agotado.	
Resultado esperado: El administrador puede insertar y eliminar productos.	
Evaluación de la prueba: Aceptado	

Tabla 22

Prueba de Aceptación de la historia de usuario número 10

Caso de Prueba de Aceptación	
Código: PA_HU10	Historia de usuario: HU10
Nombre: Actualizar perfil de usuario	
Descripción: el usuario modifica contraseña, cambia el correo electrónico de su cuenta.	
Condiciones de ejecución: usuario registrado	
Pasos de ejecución: <ol style="list-style-type: none">1. Iniciar sesión usuario2. En perfil de la cuenta cambiar contraseña3. Cerrar sesión4. Iniciar sesión con la nueva contraseña.	
Resultado esperado: El usuario actualiza su contraseña.	
Evaluación de la prueba: Aceptado	

Las tres pruebas de aceptación ejecutados por el usuario son aceptadas por lo que se cumple con los requerimientos especificados y se procede a la última iteración.

1.5 Iteración 3

1.5.1 Planeación

En esta iteración se toma las historias de usuarios restantes, luego se procede a desarrollar los objetivos de la iteración a cumplir. Por último, se crean los criterios de aceptación.

1.5.1.1 Historia de Usuarios

Tabla 23

Historia de usuario número 6

Historias de usuario	
Número: 6	Nombre de historia: Carrito de compras.
Programador: Simón Aucatoma	Iteración: 3
Prioridad en negocio: Alta	Prioridad en desarrollo: media
Descripción: el usuario podrá visualizar el botón de carrito en la parte superior, además en cada producto necesita tener la opción de agregar al carrito. Dentro de carrito se puede actualizar cantidad de producto o eliminar productos.	
Observaciones:	

Tabla 24

Historia de usuario número 8

Historias de usuario	
Número: 8	Nombre de historia: Generar prefactura.
Programador: Simón Aucatoma	Iteración: 3
Prioridad en negocio: media	Prioridad en desarrollo: baja
Descripción: cuando el usuario tengo productos en el carrito y seleccione comprar se genera un archivo formato pdf con los detalles de producto, precios e información tanto de cliente como de la tienda.	
Observaciones:	

1.5.1.2 Objetivos de la Iteración

- Agregar productos al carrito
- Generar la prefactura cuando compre los artículos del carrito.

1.5.1.3 Criterios de Aceptación

- Productos disponibles se agregan al carrito
- Se visualiza la factura con los datos de tienda y cliente.

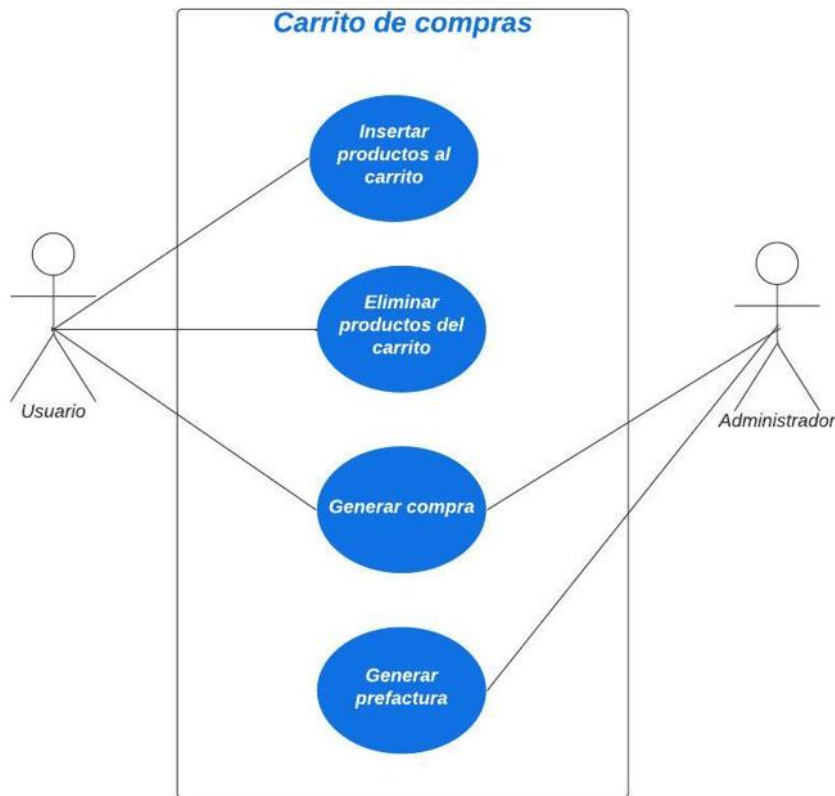
1.5.2 Diseño

4.4.2.1 Diagramas UML del Carrito de Compras.

Se diseñan los diagramas UML para visualizar los componentes a desarrollar en la última fase. En este paso se hace un rediseño debido a que será la etapa final del desarrollo.

Figura 18

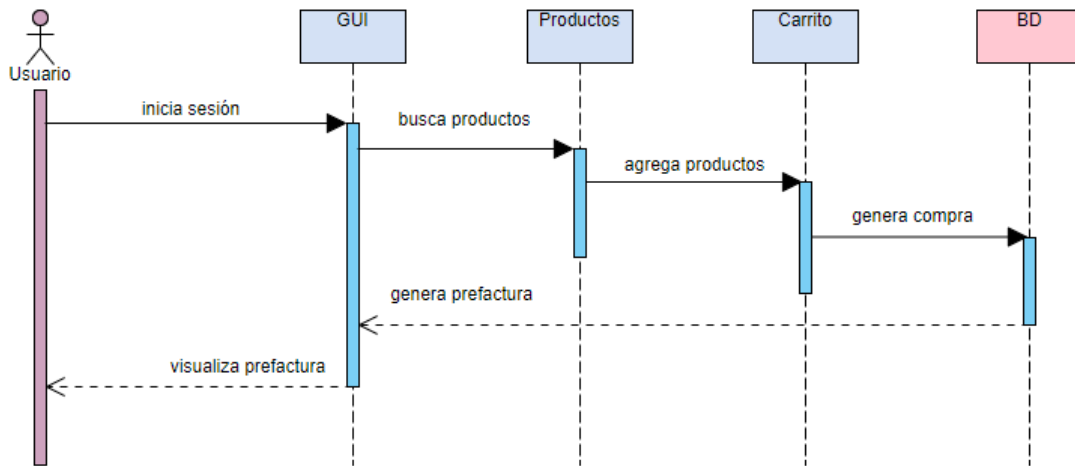
Caso de uso del carrito de compras



En el marco de esta aplicación, se ha implementado una funcionalidad que permite al cliente gestionar el contenido de su carrito de compras de manera efectiva. El cliente tiene la capacidad de agregar o eliminar productos del carrito, lo que proporciona una experiencia de compra personalizada y adaptable a sus necesidades específicas. La interfaz de usuario presenta de manera clara y detallada la cantidad y el precio de los productos añadidos al carrito, así como el monto total a pagar por los productos seleccionados.

Figura 19

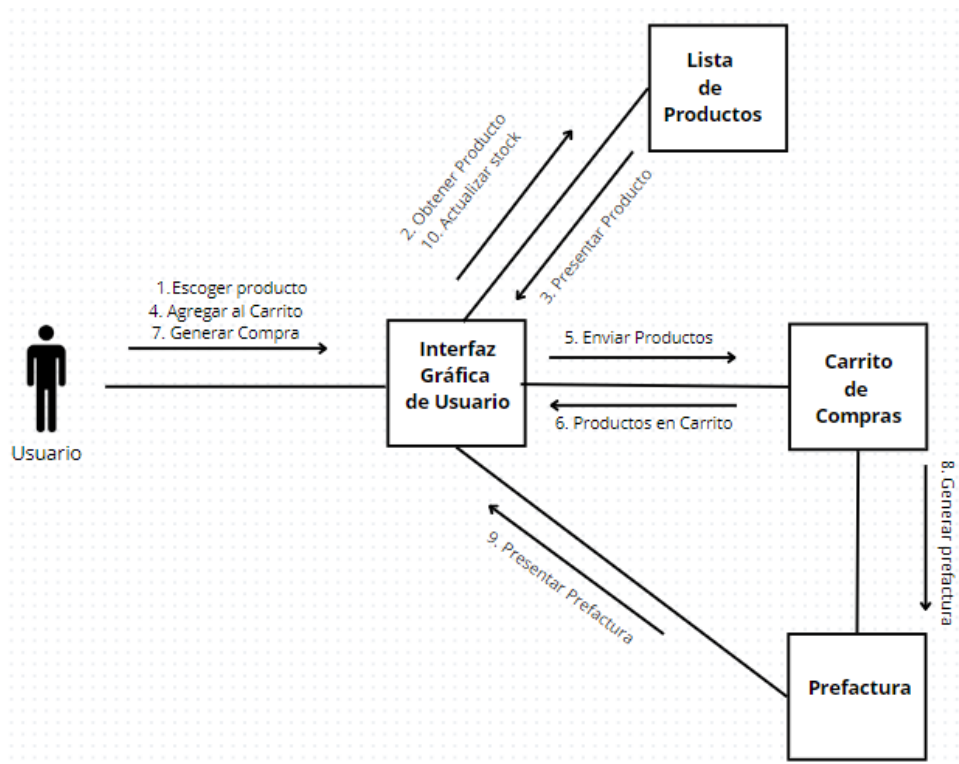
Diagrama de secuencia de la compra



Nota: las compras se almacenan en Firestore

Figura 20

Diagrama de colaboración para generar prefactura



Nota: el digrama visualiza la interacción entre los componentes para generar la prefactura.

1.5.3 Codificación

Se codifica los componentes diseñados en los diagramas de UML antes presentado, también se toma de guía mediante los requerimientos de las historias de usuarios. Los criterios de aceptación favorecen para empezar a programar de acuerdo con los parámetros de evaluación. Las pruebas unitarias permitirán corregir errores de código anticipadamente.

En cuanto a la nomenclatura se adopta las mencionadas de la primera iteración para facilitar la lectura del código y asegurar un mantenimiento eficiente a largo plazo. Entre las principales convenciones se destacan: nombre de variables y funciones con camelCase, manejo de excepciones, control de versiones con el repositorio de GitHub.

1.5.4 Pruebas

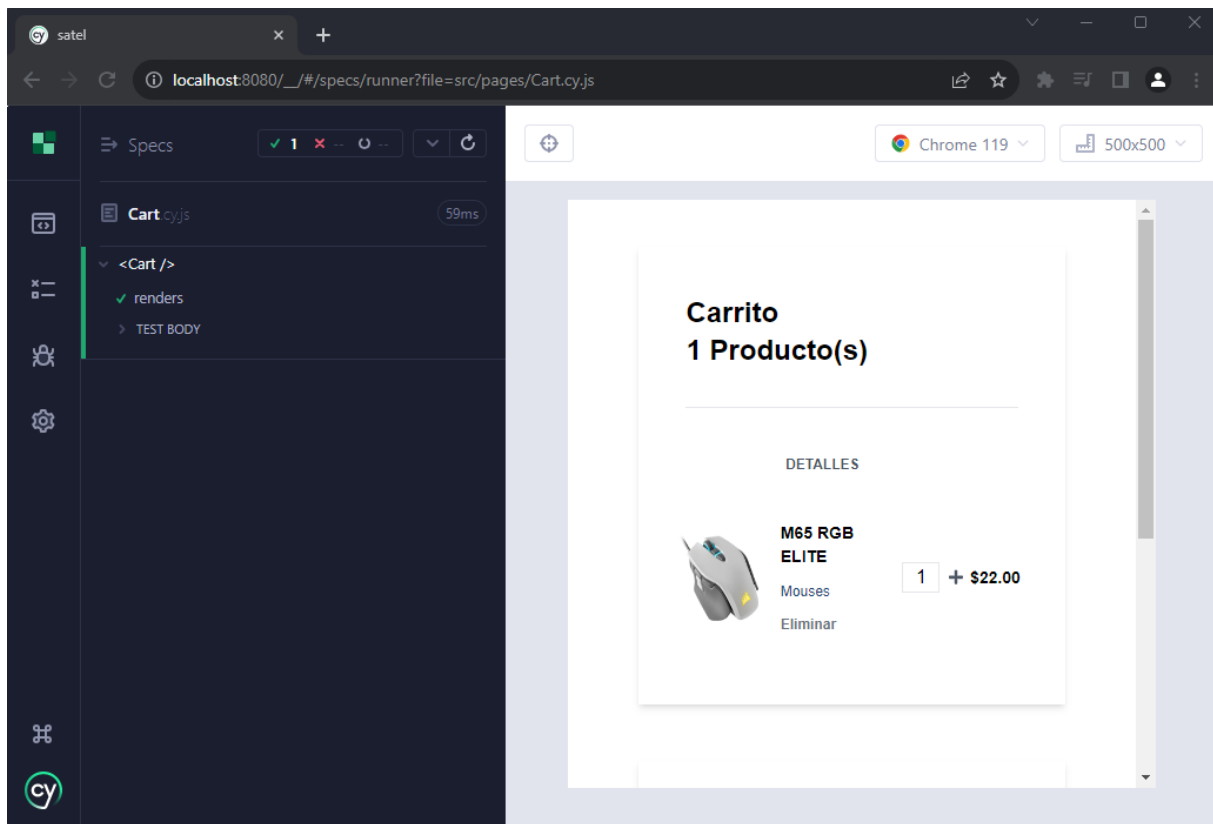
1.5.4.1 Pruebas Unitarias de la Tercera Iteración

La última fase es crucial hacer varias pruebas ya que se implementa las ultimas funcionalidades y puede generar errores de integración los cuales pueden resultar costosos resolver más adelante.

A continuación, se lleva a cabo la ejecución de la prueba, la cual se centra en evaluar la capacidad de renderización de la página. El propósito principal es verificar la precisión y actualización de los elementos presentados en la interfaz gráfica. Este proceso de prueba se realiza con el objetivo de garantizar que la página se represente de manera correcta y acorde con las últimas actualizaciones implementadas en el código. Este enfoque contribuye a mantener la integridad visual de la aplicación y asegura que los usuarios experimenten una interfaz coherente y actualizada.

Figura 21

Prueba de agregar productos al carrito



Nota: se observa que la renderización se completó con éxito.

Este sistema exhibe de manera transparente el precio asociado a los productos añadidos, así como un monto total acumulado correspondiente a la selección actual de productos. Al completar la compra, se genera una prefactura en formato PDF, proporcionando al cliente una representación documentada de la transacción. Además, para mantener un registro integral, la información de esta transacción se almacena en la base de datos, asegurando así la trazabilidad y la disponibilidad de historiales de compras. Este enfoque no solo facilita la experiencia del cliente, sino que también respalda la gestión eficiente de datos y la contabilidad del negocio.

La práctica de reutilizar código ha sido fundamental en la creación de funciones exentas de errores de codificación, permitiendo una eficiente gestión del desarrollo al aprovechar bloques de código previamente probados y validados en diversas instancias. Esta estrategia no solo ha favorecido la coherencia en la implementación de funciones, sino que también ha contribuido a la reducción de posibles errores, al fomentar la consistencia y la aplicación uniforme de lógica de programación probada. Este enfoque se ha revelado como efectivo al optimizar el tiempo necesario para ejecutar pruebas unitarias adicionales, facilitando así la transición hacia pruebas de aceptación con el usuario según lo planificado en cada iteración del desarrollo. En la última iteración, no se encontró observaciones por parte del cliente. Este ciclo iterativo de desarrollo, evaluación continua y ajustes ha contribuido al éxito del proceso, asegurando la calidad del producto y la alineación con las expectativas del cliente.

1.5.4.2 Pruebas de Aceptación

Tabla 25

Prueba de Aceptación de la historia de usuario número 6

Caso de Prueba de Aceptación	
Código: PA_HU6	Historia de usuario: HU06
Nombre: Carrito de compras.	
Descripción: se verifica que los productos se agregan al carrito y se actualiza la cantidad a comprar.	
Condiciones de ejecución: productos disponibles en cada categoría.	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar productos de la categoría y agregar al carrito. 2. Observar el carrito con los productos agregados correctamente. 	
Resultado esperado: los productos se visualizan en el carrito.	
Evaluación de la prueba: Aceptado	

Tabla 26*Prueba de Aceptación de la historia de usuario número 8*

Caso de Prueba de Aceptación	
Código: PA_HU8	Historia de usuario: HU08
Nombre: Generar prefactura.	
Descripción: se realiza la compra, se verifica la prefactura con datos del cliente y de la tienda.	
Condiciones de ejecución: productos disponibles, lector de archivos con extensión pdf.	
Pasos de ejecución: <ol style="list-style-type: none">1. Agregar producto al carrito.2. Generar comprar3. Visualizar documento de prefactura	
Resultado esperado: la prefactura se genera correctamente.	
Evaluación de la prueba: Aceptado	

Las pruebas son evaluadas y aceptadas por el usuario por lo que la integración final del incremento de software ha sido exitosa. Se da por finalizado las fases de pruebas a la vez que se corrobora que cumple con todos los requerimientos establecidos al inicio del proyecto.

1.6 Cierre

Una vez finalizadas y aprobadas todas las iteraciones por parte del usuario, se lleva a cabo la entrega del producto final y se procede al cierre del desarrollo de software. En este punto, se ha completado exitosamente el proceso y se pone fin al ciclo de trabajo.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

1 Conclusiones y Recomendaciones

1.1 Conclusiones

- 1.1.1 Se logró una comprensión exhaustiva del proceso actual de compras en SATELNET PC, identificando desafíos en la gestión del inventario y la promoción de productos. Esta situación ha generado la oportunidad estratégica de trasladarse de una tienda física a una plataforma en línea, lo cual se considera una solución para mejorar la eficiencia y la visibilidad en el mercado.
- 1.1.2 Tras una meticulosa evaluación de las metodologías de desarrollo de software rápido, Scrum y XP, se optó por la metodología XP debido a su óptima adaptabilidad a las necesidades específicas del proyecto. Esta elección proporcionó un marco estructurado y efectivo para el desarrollo del aplicativo web.
- 1.1.3 Se realizó un análisis minucioso de los requerimientos funcionales y no funcionales para el proyecto, empleando historias de usuarios. Este enfoque se implementó con el propósito de asegurar que el aplicativo web cumpla de manera efectiva con las expectativas y necesidades del usuario final.
- 1.1.4 La fase de desarrollo del aplicativo web se llevó a cabo de manera exitosa, respaldada por la cuidadosa elección de la metodología de software y la implementación del backend como servicio (BaaS). En este caso, Firebase optimizó la gestión de recursos en el backend.

1.2 Recomendaciones

- 1.2.1 Se sugiere realizar un análisis detallado del giro del negocio para diseñar una solución integral que se alinee con las expectativas y necesidades del usuario. El enfoque debe centrarse en proporcionar una experiencia excepcional de usuario con el producto desarrollado.
- 1.2.2 Se recomienda seleccionar cuidadosamente los requisitos del proyecto, considerando el plazo acordado. Esto evitará incumplimientos y proporcionará una comprensión clara del tiempo necesario para el desarrollo del aplicativo. Delimitar de manera precisa el alcance del desarrollo contribuirá a evitar malentendidos entre el usuario y el desarrollador.
- 1.2.3 En proyectos pequeños, se aconseja la selección de una metodología ágil, ya que estas permiten la interacción continua y directa con el cliente durante el desarrollo. La flexibilidad inherente a estas metodologías facilita la adaptación eficiente a cambios que puedan surgir a lo largo del proyecto, garantizando una mayor satisfacción y alineación con las necesidades del cliente.

BIBLIOGRAFÍA

- Álvarez-Gayou, J. (2003). *Cómo hacer investigación cualitativa. Fundamentos y metodología* (Primera ed.). México: Ediciones Paidós. Obtenido de <http://www.derechoshumanos.unlp.edu.ar/assets/files/documentos/como-hacer-investigacion-cualitativa.pdf>
- Ambler, S. (2005). *The Elements of UML 2.0 Style* (Primera ed.). New York: Cambridge University Press.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (Segunda ed.). Massachusetts: Addison-Wesley Professional.
- Cáceres, L. (2020). *Cómo triunfar en el mundo digital*. Penguin Random House Grupo Editorial.
- Cámara Ecuatoriana de Comercio Electrónico. (2021). En 2021, el comercio electrónico mantendrá un crecimiento sostenido en Ecuador. *Ekos*. Recuperado el 2 de Septiembre de 2023, de <https://ekosnegocios.com/articulo/en-2021-el-comercio-electronico-mantendra-un-crecimiento-sostenido-en-ecuador>
- Cardona Arenas, C., Quintero Renaud, S., Mora Quintero, M., & Castro Cardona, J. (2022). Influencia del comercio electrónico en el desempeño financiero de las pymes en Manizales, Colombia. *INNOVAR*, 32(84), 75-96. Recuperado el 12 de Septiembre de 2023, de <https://revistas.unal.edu.co/index.php/innovar/article/view/100594/83123>
- Creswell, J. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (Cuarta ed.). Los Ángeles: SAGE Publications.
- Elmasri, R., & Navathe, S. (2016). *Fundamentals of database systems* (Séptima ed.). Boston: Pearson.
- Espinosa-Hurtado, R. (2021). Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web. *CEDAMAZ*, 11(2), 133-141. doi:<https://doi.org/10.54753/cedamaz.v11i2.1182>
- Facebook. (20 de Octubre de 2021). *Reac*. Obtenido de <https://reactjs.org/>
- Jiménez, M. (20 de Julio de 2020). *Amazon Web Services (AWS): ¿qué es y qué ofrece?* Recuperado el 28 de Septiembre de 2023, de Hiberus: <https://www.hiberus.com/crecemos-contigo/amazon-web-services-aws-que-es-y-que-ofrece/>
- Google. (2023). *Firebase*. Recuperado el 12 de Octubre de 2023, de <https://firebase.google.com/>
- HNEIF, M., & HOCK OW, S. (2009). REVIEW OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT. *International Journal of Research and Reviews in Applied Sciences*, 1(1). Recuperado el 15 de Octubre de 2023, de https://www.arpapress.com/Volumes/Vol1/IJRRAS_1_01.pdf
- Jiménez Builes, J., Ramírez Bedoya, D., & Branch Bedoya, J. (2019). Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP. *Politécnica*(30), 55-69. doi:<https://doi.org/10.33571/rpolitec.v15n30a6>
- Kendall, K., & Kendall, J. (2018). *Systems Analysis and Design* (Décima ed.). México: Pearson.

- Márquez Egea, N. (2020). Frameworks de desarrollo web back-end. *Dialnet*(36), 9-12. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=7220420>
- Martín Villalba, C., Rubio, M., & Urquia Moraleda, A. (2021). *Lenguajes de programación*. UNED.
- Microsoft Corporation. (2023). *¿Qué es PaaS?* Recuperado el 4 de Octubre de 2023, de Azure: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-paas>
- Pressman, R. (2010). *Ingeniería del software: un enfoque práctico* (Séptima ed.). México: McGraw-Hill.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Obtenido de Scrum: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- Schwarz Müller, M. (18 de Marzo de 2020). *Angular vs React vs Vue - Mis pensamientos*. Obtenido de <https://academind.com/tutorials/angular-vs-react-vs-vue-my-thoughts>
- Silberschatz, A., Korth, H., & Sudarshan, S. (2014). *Fundamentos de base de datos* (Sexta ed.). Madrid: McGraw Hill.
- Sommerville, I. (2011). *Ingeniería de software* (Novena ed.). México: Pearson.

GLOSARIO

CRUD

(Crear, leer, actualizar y eliminar) es un acrónimo que se refiere a las operaciones básicas en sistemas que gestionan datos almacenados.

DOM

(Modelo de Objetos del Documento) es una representación jerárquica de la estructura de un documento HTML o XML

Incremental

Se refiere a las entregas parciales del producto en cada fase del desarrollo y la última entrega es un incremento de la anterior.

Plataforma

Sistemas Operativos y entornos de hardware que permite funcionar una aplicación web.

Prefactura

Es el documento que antecede a la factura en una transacción.

Refactorización:

Se refiere al proceso de reestructurar y mejorar el código existente sin cambiar su comportamiento externo.

RUP

(Rational Unified Process) es un marco de desarrollo de software que se basa en principios iterativos e incrementales.

SPA

Se trata de una aplicación de página única en la que todo el contenido se encuentra en una sola página.

ANEXOS

Figura 22

Firma de responsabilidad del cliente para documentos del trabajo

Quito, 14 de noviembre del 2023

Carta de Colaboración

A quien corresponda,

El presente documento tiene como propósito comunicar la participación entre el Sr. Sergio Aucatoma con cédula de ciudadanía 1724393531 propietario de SATELNET PC y el estudiante Simón Efraín Aucatoma Rea, identificado con la cédula de ciudadanía 2100673355, en el trabajo de titulación "Desarrollo de una aplicación web dedicada a la venta de partes y piezas de computadoras para la empresa SATELNET PC". Durante las reuniones sostenidas hasta la fecha actual, se han alcanzado los siguientes puntos:

- ✓ Se han identificado las características fundamentales de la aplicación a través de la definición de diez historias de usuario.
- ✓ Se han establecido criterios de aceptación para cada entrega del producto, garantizando la alineación del desarrollo con las expectativas.
- ✓ Se han diseñado diez casos de pruebas de aceptación con códigos PA_HUx, proporcionando una estructura clara para la validación de las funcionalidades. Estas pruebas han sido superadas exitosamente a gusto del cliente.
- ✓ Se han definido tres entregables claves para distintas etapas del desarrollo del proyecto.
- ✓ El cliente ha participado activamente en reuniones de revisión, contribuyendo con una retroalimentación oportuna y significativa para el mejoramiento continuo del proyecto.

Este acuerdo de colaboración se considerará válido hasta la finalización del proyecto, prevista para el 17 de noviembre del 2023.

Este acuerdo refleja el compromiso mutuo de ambas partes y busca garantizar una colaboración efectiva para el éxito del proyecto.

Firma de los colaboradores:


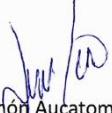
 Sergio Aucatoma C.C. 1724393531 Propietario de SATELNET PC	 Simón Aucatoma Rea C.C. 2100673355 Estudiante de Ingeniería de Sistemas de Información
---	--

Figura 23

Carta de aceptación de finalización del software



SATELNET PC
TEGNOLOGIA A SU ALCANSE

RUC:
1724393531001

VENTAS AL POR MAYOR Y MENOR DE EQUIPOS PARTES Y PIEZAS DE COMPUTACION, LAPTOPS, TABLES, ACCESORIOS, MANTENIMIENTO Y REPARACION DE COMPUTADORAS, IMPRESORAS, ACTUALIZACIONES SOFTWARE Y HARDWARE

Quito, 17 de noviembre del 2023

CERTIFICADO

Yo, Sr. Sergio Aucatoma con cédula de ciudadanía 1724393531 propietario de SATELNET PC.

Certifico,

Que el estudiante Simón Efrain Aucatoma Rea, portador de la cédula de ciudadanía 2100673355, perteneciente a la carrera de Ingeniería en Sistemas de Información de la Pontificia Universidad Católica del Ecuador, ha completado con éxito el desarrollo del aplicativo web titulado "Desarrollo de una aplicación web dedicada a la venta de partes y piezas de computadoras para la empresa SATELNET PC" como parte de su trabajo de titulación. El producto entregado cumple con los requisitos establecidos al inicio del proyecto, y como cliente, me encuentro satisfecho con el resultado obtenido.

Le extiendo los mejores deseos en todos sus proyectos futuros.

Atentamente,



Sergio Danilo Aucatoma R.
RUC: 1724393531001
Gerente Propietario
Telf. 0995502497

