

1. INFORMACIÓN GENERAL

Título:

Aplicativo web y móvil para la comercialización de productos en línea de la empresa “Señor de Maca” de la ciudad de Latacunga.

Fecha de inicio:

05 de abril del 2021

Fecha de finalización:

05 de agosto del 2021

Lugar de ejecución:

Provincia de Cotopaxi, ciudad de Latacunga, parroquia San José de Poalo, en la empresa “Señor de Maca”

Facultad que auspicia:

Ciencias de la Ingeniería y Aplicadas.

Carrera que auspicia:

Ingeniería en Informática y Sistemas Computacionales.

Proyecto de investigación vinculado:

Equipo de Trabajo:

Tutor: Ing. Llano Casa Alex Christian.

Postulantes: Galeas Guanochanga Carlos Xavier, Noteno Andi Byron Luis.

Área de Conocimiento:

Información y Comunicación (TIC).

Línea de investigación:

Tecnologías de la Información y Comunicación (TICS) y Diseño Gráfico.

Sub líneas de investigación de la Carrera:

Ciencias informáticas para la modelación de sistemas de información a través del desarrollo de software.

2. DISEÑO INVESTIGATIVO DE LA PROPUESTA TECNOLÓGICA

2.1. TÍTULO DE LA PROPUESTA TECNOLÓGICA

2.2. TIPO DE ALCANCE

Este proyecto tendrá un alto impacto social y económico, tanto para la empresa la cual lo auspicia como para las clientes a los cuales va dirigida, ya que gracias a la automatización de los procesos que se va a realizar con el uso de las TIC, la empresa tendrá un mayor auge en el mercado y al mismo tiempo podrá empezar a entender de una manera más clara cuál es el uso específico de las TIC en el mercado actual, también nos dará un enfoque más práctico de cómo es el uso de las API para validaciones de compras seguras en línea y como estas interactúan con los lenguajes de programación elegidos para el desarrollo del proyecto, y así, de esta manera obtener mejores resultados en la aplicación de los conocimientos que se aplicaran para el avance de la aplicación.

2.3. ÁREA DE CONOCIMIENTO

Área: Información Y Comunicación (TIC)

Sub-Área: Desarrollo y análisis de software y aplicaciones.

2.4. SINÓPSIS DE LA PROPUESTA TECNOLÓGICA

El tema objetivo a realizar en el proyecto es el desarrollo de una aplicación web y móvil para la comercialización de productos en línea y la gestión de productos de la empresa “Señor de Maca”. Para el desarrollo de la aplicación se ha definido un total de 2 usuarios principales los cuales son: el administrador y el cliente, los cuales estarán participando de manera conjunta en uno o más módulos de la aplicación.

2.5. OBJETO DE ESTUDIO Y CAMPO DE ACCIÓN

2.5.1. Objeto De Estudio

Para definir el objeto, el cual es de suma importancia para el desarrollo de la propuesta tecnológica, se debe realizar un análisis del tema en el cual se realizará la investigación y desarrollo, en este caso es: Aplicativo web y móvil para la comercialización de productos en línea de la empresa “Señor de Maca” de la ciudad de Latacunga. Se puede diferenciar dos puntos de suma importancia que se deben identificar, el primero es el problema de investigación, por el cual se lograra obtener el objeto de la investigación. En el tema podemos diferenciar el problema de investigación como: “Un aplicativo web y móvil para

la comercialización de productos en línea”, dando como resultado el siguiente objeto de investigación: “La empresa Señor de Maca de la ciudad de Latacunga”.

2.5.2. Campo De Acción

Y para el campo de acción, el cual está definido por Nomenclatura Internacional de la UNESCO para los campos de Ciencia y Tecnología, tomaremos la siguiente nomenclatura: 3300 Ciencias Tecnológicas / 3304 Tecnología de Los Ordenadores / 3304.17 Sistemas en Tiempo Real. Esta es la nomenclatura que más apegada se encuentra a nuestro problema de investigación.

2.6. SITUACIÓN PROBLEMÁTICA Y PROBLEMA

2.6.1. Situación Problemática

A nivel internacional, podemos ver varias aplicaciones tanto web como móviles que se prestan a dar servicios de manera gratuita o pagada para las diferentes empresas que lo deseen, una de las más reconocidas que se puede encontrar en la actualidad es la aplicación de Play Store, el cual es un servidor de aplicaciones para dispositivos móviles, en la web, pero esta no solo posee aplicaciones móviles, también nos permite realizar pagos con tarjetas de crédito, a través de su propia Api creada por Google, la cual tiene un nivel de seguridad alto para compra y venta de productos en línea. Esto es de mucha ayuda para las empresas, ya que así los usuarios que deseen comprar alguna aplicación o producto a través de su aplicación, podrán hacerlo de una manera muy segura y tener la confianza de que sus datos no serán robados (en su mayor parte) por otras personas.

En cambio, en América Latina el desarrollo de aplicaciones móviles ha ido teniendo un mayor auge en estos últimos años, con lo cual podemos ver que se han creado aplicaciones con las cuales se trata de dar competencia a las mismas que se usan a un nivel mayor internacional. Como ejemplo tenemos la aplicación de Traki, la cual es una aplicación, tanto web como móvil, que se usa para realizar compras de productos varios en línea, la cual sirve en varios países de América Latina, dando servicios, como el pago de productos en un país y que los mismos se puedan retirar en una agencia de otro país.

2.6.2. Problema

A través del paso de los años, el uso de las TIC ha ido tomando gran relevancia para la aplicación y avance de las empresas, la razón de esto es por el mercado que se encuentra siempre cambiante, teniendo como resultado que las empresas que no logren automatizar

de manera eficaz los servicios que ofrecen, queden fuera de este mercado actual. Este mercado también se ve afectado por factores externos al mismo, como es el resultado de la pandemia COVID 19 con la cual vivimos en la actualidad, está ocasionado que las empresas tengan que actualizar sus servicios de manera obligatoria, dando paso a que la implementación de aplicaciones webs y móviles, que puedan generar los servicios que estas empresas proveen de una manera más segura y rápida, sean la solución más adecuada para una mejor atención de sus servicios.

Por otra parte, el desarrollo de aplicaciones, tanto web como móviles, en Ecuador no ha tenido mucho auge en estos últimos años, esto se debe a que la mayor parte de empresas no desean automatizar sus negocios, haciendo que el interés en el desarrollo de las aplicaciones sea muy bajo o se decida por el contrato de una aplicación de otro país pero que esta de cobertura hacia Ecuador. Uno de estos casos es la empresa “Señor de Maca”, la cual al no tener implementada una aplicación web o móvil para la gestión de sus servicios y productos, ha ido perdiendo poco a poco sus clientes, ya que estos prefieren otras empresas que brinden sus servicios de una manera más segura y rápida.

Cómo se puede mejorar la comercialización de productos y ventas en línea de la Empresa “Señor de Maca”, a través del uso de técnicas para el desarrollo de una aplicación web y móvil que permita agilizar los procesos de ventas en la empresa.

2.7. HIPÓTESIS

Con el desarrollo de la aplicación web y móvil, optimizaremos los procesos de comercialización de productos en la empresa Señor de Maca.

2.8. OBJETIVOS

2.8.1. Objetivo General

Desarrollar una aplicación web y móvil con el uso de lenguajes de programación libres, para la comercialización de productos en línea de la empresa “Señor de Maca”.

2.8.2. Objetivos Específicos

- Realizar un estudio bibliográfico sobre el uso y desarrollo de aplicaciones web y móviles para comercialización de productos en línea.
- Recopilar información acerca del uso de la metodología ágil para el desarrollo de aplicaciones de comercialización de productos en línea.
- Aplicar la metodología Scrum para el desarrollo de la aplicación en el entorno web y la metodología Mobile-D para el entorno móvil.

2.9. DESCRIPCIÓN DE LAS ACTIVIDADES Y TAREAS PROPUESTAS CON LOS OBJETIVOS ESTABLECIDOS

Tabla 2.1. Descripción de las actividades y tareas propuestas

Objetivos específicos	Actividades (tareas)	Resultados esperados	Técnicas, Medios e Instrumentos
<p>Realizar un estudio bibliográfico sobre el uso y desarrollo de aplicaciones web y móviles para comercialización de productos en línea.</p>	<ol style="list-style-type: none"> 1. Realizar una investigación bibliográfica acerca del desarrollo de aplicaciones webs y móviles. 2. Investigación de aplicaciones para la comercialización de productos en línea. 	<ol style="list-style-type: none"> 1. Se seleccionó las fuentes bibliográficas más confiables sobre el desarrollo de aplicaciones webs y móviles. 2. Se obtuvo la información necesaria para su aplicación. 	<ul style="list-style-type: none"> • Análisis bibliográfico. • Trabajos e instigaciones relacionadas con el tema.
<p>Recopilar información acerca del uso de la metodología ágil para el desarrollo de aplicaciones de comercialización de productos en línea.</p>	<ol style="list-style-type: none"> 1. Analizar los datos recolectados sobre el desarrollo de aplicaciones de comercialización de productos en línea. 2. Seleccionar una metodología de desarrollo ágil, que cumpla con las necesidades de desarrollo del proyecto. 	<ol style="list-style-type: none"> 1. Se seleccionó el tipo de metodología ágil adecuado para el desarrollo de la aplicación, tanto web como móvil. 	<ul style="list-style-type: none"> • Análisis documental • Análisis de contenido. • Trabajos e instigaciones relacionadas con el tema.

<p>Aplicar la metodología Scrum para el desarrollo de la aplicación en el entorno web y la metodología Mobile-D para el entorno móvil.</p>	<ol style="list-style-type: none"> 1. Modelar y desarrollar el sistema a través del uso de la metodología Scrum para la aplicación web. 2. Modelar y desarrollar el sistema a través del uso de la metodología Mobile-D para la aplicación móvil. 	<ol style="list-style-type: none"> 1. Modelado y desarrollo de cada una de las aplicaciones, usando las metodologías propuestas por aplicación. 	<ul style="list-style-type: none"> • Maquetación del sistema • Pruebas del sistema • Análisis de cumplimiento del sistema.
---	---	--	---

3. JUSTIFICACIÓN

El presente estudio se enfoca en el desarrollo e implementación de una aplicación web y móvil para la comercialización y gestión de los productos en línea de la empresa “Señor de Maca”, ya que por los problemas que ha ocasionado la enfermedad Covid19, tanto a nivel nacional como mundial, el mercado en el cual se desarrolla la empresa ha cambiado drásticamente, haciendo que sea necesario que esta empresa posea una aplicación que funcione en línea las 24 horas del día, 7 días de la semana. De esta manera, el presente proyecto permitirá demostrar la importancia del cambio de modelos de procesos de la empresa “Señor de Maca”, a través del uso de las TIC, y de manera adicional podremos profundizar los conocimientos teóricos sobre el desarrollo de aplicaciones web y móviles en los diferentes lenguajes de programación.

La información necesaria para el desarrollo de las aplicaciones se obtendrá a través de una investigación de campo, realizando una entrevista al propietario de la empresa y encuestas hacia los clientes de la misma, y una investigación documental, por la cual se elegirán las mejores herramientas de desarrollo. Para la ejecución y progreso de los módulos del proyecto, se ha seleccionado una metodología de desarrollo ágil por cada aplicación, para la aplicación web se utilizará la metodología Scrum y para la aplicación móvil la metodología de desarrollo Mobile-D, las cuales permite tener una mejor organización en el proyecto.

El desarrollo de las aplicaciones será de gran importancia para el proceso de ventas que ofrece la empresa, ya que, a través de las mismas se podrá optimizar de una manera más eficaz los servicios que da la empresa y obtendrá un mejor posicionamiento empresarial en el mercado laboral en la cual se encuentra ubicada.

Debido a que la empresa se dedicará a la prestación de un servicio, en este estudio se analizará con detalle cada uno de los componentes del mismo, de manera tal que se logre una prestación eficiente y duradera. También se propondrán otros servicios o productos complementarios al servicio principal que constituye a la compra de productos en línea, tales como: que el cliente pueda agregar productos a una lista de favoritos; para que de esta manera el mismo tenga un acceso más rápido a los productos que este compre con mayor regularidad. El administrador podrá tener estadísticas de compra en su empresa, este tipo de modalidades adicionales son los que servirán para diseñar un servicio plenamente adecuado y que satisfaga convenientemente esas necesidades.

4. BENEFICIARIOS DEL PROYECTO

4.1. DIRECTOS

Propietarios, accionistas y empleados de la empresa “Señor de Maca”, ubicada en la ciudad de Latacunga, provincia de Cotopaxi.

Cantidad de beneficiarios: 10

4.2. INDIRECTOS

Clientes de la empresa “Seño de Maca” y el sector productivo de la construcción.

Cantidad de beneficiarios: 150

5. MARCO TEÓRICO

5.1. ANTECEDENTES

5.1.1. Centros Ferreteros

Alrededor del mundo se sabe que, para poder construir, arreglar cosas dentro de una casa o establecimiento o simplemente poder mejorar otras, es necesario tener materiales especializados para esto, y, por ende, la ferretería es ese lugar. Dentro de este orden, se sabrá que la ferretería se enfoca poder vender y comprar materiales, utensilios o herramientas que se usan comúnmente en diversas áreas, así que su importancia es bastante cuando se trata de la humanidad. [1]

Los centros ferreteros son empresas que se centran en la comercialización de productos que sirven para la construcción de viviendas o edificios en general, estas pueden vender sus productos en grandes o pequeñas cantidades dependiendo de la necesidad del cliente, así mismo, brinda otros servicios adicionales a la venta de productos.

5.1.2. Centros Ferreteros En Ecuador

En Ecuador existe una gran variedad de centros ferreteros a disposición de los clientes en general, a continuación, enunciare algunos de los más importantes:

- Comercial Kywi S.A.
- Importador Ferretero Trujillo.
- Ecuaimco.
- Megabyce Centros Ferreteros S.A.
- Ferremundo.
- CEFEIND.

- Ecuaimco.
- Señor de Maca.

5.2. Aplicaciones Digitales

Una aplicación es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas. Generalmente, son diseñadas para facilitar ciertas tareas complejas y hacer más sencilla la experiencia informática de las personas. [2]

Las aplicaciones son programas totalmente distintos a los sistemas operativos o lenguajes de programación, ya que estas cumplen una función puntual y es pensada para que sea de uso común como por ejemplo manejar finanzas o contabilidad desde hojas de cálculo. [2]

Este tipo de aplicativos tiene tipos específicos en los cuales pueden ejecutarse, al mismo tiempo cada uno de estos tiene funciones específicas que deberá cumplir en su propósito, existen varias plataformas para el desarrollo de las misma, así como diferentes estructuras que puede llevar la misma.

5.2.1. Aplicación Web

Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones. [3]

El protocolo HTTP forma parte de la familia de protocolos de comunicaciones TCP/IP, que son los empleados en Internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores. HTTP se sitúa en el nivel 7 (aplicación) del modelo OSI. [3]

La figura 1 nos muestra como la sección del navegador, la cual es conocida como “Cliente” se comunica a través de algunos protocolos de red, para obtener la información deseada del servidor en el cual se encuentra alojada la aplicación.

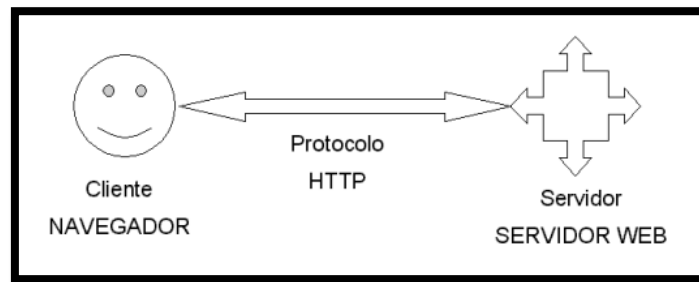


Figura 5.1: Esquema básico de una aplicación web [3]

5.2.2. Tipos De Aplicaciones Web

Hay clasificaciones que dividen sitios Web en multitud de categorías, basándose en los temas que tratan (como salud, noticias o deportes), que a su vez pueden contener otras categorías. Ejemplos de esto son los llamados directorios Web, como son el Open Directory Project (Open Directory Project) y Yahoo! Directory (Yahoo Inc.). Otro caso es la compañía Alexa (Alexa Internet, Inc.), que clasifica los sitios Web por relevancia a través de las estadísticas de visitas y emplea un sistema de categorías similar [4].

Existen muchas maneras en las cuales podemos categorizar los tipos de aplicaciones web. La figura 2 nos muestra una clasificación adicional del tipo de aplicaciones web existentes.

5.2.3. Aplicaciones Móviles

Las aplicaciones móviles o apps son piezas de software diseñadas para ser instaladas y utilizadas en dispositivos móviles, que se adaptan a las limitaciones de estos dispositivos, pero también permiten aprovechar sus posibilidades tecnológicas (por ejemplo, la localización para servicios adaptados al contexto o el acelerómetro en algunos videojuegos) [5].

El objetivo principal de estas aplicaciones es que se puedan acceder desde cualquier dispositivo móvil en cualquier lugar, así mismo estas pueden ejecutarse en cualquier sistema operativo actual sea Android, IOS o en Harmonyos que es el nuevo sistema móvil desarrollado por la empresa Huawei.

5.2.4. Tipos de Aplicaciones Móviles

Debido que los tipos de móviles existentes son muy versátiles, existen muchos tipos de aplicaciones que pueden funcionar en el mismo, debido a esto es necesario tener una clasificación para cada una de estas aplicaciones. Entre las cuales encontramos:

5.2.4.1. Aplicaciones básicas

Las aplicaciones básicas se pueden gestionar simplemente con el envío de mensajes de texto (SMS o MMS). Estas aplicaciones existen desde hace mucho tiempo y, aunque han tenido gran aceptación y uso, actualmente están comenzando a dejar paso a aplicaciones más complejas [6].

5.2.4.2. Webs móviles

Las webs móviles son aquellas webs que ya existen actualmente y que son adaptadas específicamente para ser visualizadas en los dispositivos móviles. Adaptan la estructura de la información a las capacidades del dispositivo, de manera que no saturan a los usuarios y se pueden usar correctamente desde estos dispositivos [6].

5.2.4.3. Aplicaciones web sobre móviles

Las aplicaciones web sobre móviles son aplicaciones que no necesitan ser instaladas en el dispositivo para poder ejecutarse. Están basadas en tecnologías HTML, CSS y Javascript, y que se ejecutan en un navegador. A diferencia de las webs móviles, cuyo objetivo básico es mostrar información, estas aplicaciones tienen como objetivo interactuar con el dispositivo y con el usuario. De esta manera, se le saca un mayor partido a la contextualización [6].

5.2.4.4. Aplicaciones web móviles nativas

Existe un tipo de aplicaciones, llamadas aplicaciones web móviles nativas, que no son aplicaciones web propiamente ni tampoco nativas. Se ejecutan con un navegador o, mejor dicho, con un componente nativo que delega en un navegador, y tienen algunas de las ventajas de las aplicaciones nativas [6].

5.2.4.5. Aplicaciones nativas

Las aplicaciones nativas son las aplicaciones propias de cada plataforma. Deben ser desarrolladas pensando en la plataforma concreta. No existe ningún tipo de estandarización, ni en las capacidades ni en los entornos de desarrollo, por lo que los desarrollos que pretenden soportar plataformas diferentes suelen necesitar un esfuerzo extra [6].

5.3. ASPECTOS TEÓRICOS

Para el desarrollo del proyecto se ha elegido dos tipos de investigación, las cuales nos servirán para la recopilación de información, dos metodologías para el desarrollo de aplicaciones y una serie de herramientas indispensables.

5.3.1. Investigación Científica

Investigar significa llevar a cabo diferentes acciones o estrategias con el fin de descubrir algo. Así, dichos actos se dirigen a obtener y aplicar nuevos conocimientos, explicar una realidad determinada o a obtener maneras de resolver cuestiones y situaciones de interés. La investigación es la base del conocimiento científico, si bien no toda investigación es científica de por sí [7].

Para generar información para el desarrollo de un proyecto es necesario realizar una investigación científica acorde a las necesidades que necesitemos en información, así mismo existe una clasificación para el tipo de investigación que se necesite realizar.

5.3.1.1. Investigación descriptiva

En las investigaciones de tipo descriptiva, llamadas también investigaciones diagnósticas, buena parte de lo que se escribe y estudia sobre lo social no va mucho más allá de este nivel. Consiste, fundamentalmente, en caracterizar un fenómeno o situación concreta indicando sus rasgos más peculiares o diferenciadores [8].

El objetivo de la investigación descriptiva consiste en llegar a conocer las situaciones, costumbres y actitudes predominantes a través de la descripción exacta de las actividades, objetos, procesos y personas. Su meta no se limita a la recolección de datos, sino a la predicción e identificación de las relaciones que existen entre dos o más variables [8].

La investigación descriptiva, según se mencionó, trabaja sobre realidades de hecho y su característica fundamental es la de presentar una interpretación correcta. Esta puede incluir los siguientes tipos de estudios: Encuestas, Casos, Exploratorios, Tipos de Investigación Histórica, descriptiva y experimental Diversas formas de clasificación Básica y aplicada Documental, de campo y mixta Experimental y No experimental Exploratoria, descriptiva y explicativa Transversales y horizontales De campo o laboratorio Según la disciplina que se investiga Causales, De Desarrollo, Predictivos, De Conjuntos, De Correlación [9].

Para utilizar de manera correcta esa metodología se debe, de manera usual, responder a las siguientes preguntas:

- ¿Qué es?
- ¿Cómo es?
- ¿Dónde está?
- ¿De qué está hecho?
- ¿Cómo están sus partes, si las tiene, interrelacionadas?
- ¿Cuánto?

5.3.1.2. Investigación de campo

La investigación de campo recopila los datos directamente de la realidad y permite la obtención de información directa en relación a un problema.

Este tipo de investigación es esencial para realizar otras como la exploratoria, la correlacional o la mixta. De hecho, en el método hipotético-deductivo (el utilizado en economía) suele ser el paso posterior al establecimiento de las hipótesis. Una vez sabemos qué buscamos, debemos recabar datos y para eso se realiza un trabajo de campo [10].

Una investigación de campo o estudio de campo es un tipo de investigación en la cual se adquieren o miden datos sobre un suceso en particular, en el lugar donde suceden. Es decir que, el investigador se traslada hasta el sitio donde ocurre el fenómeno que desea estudiar, con el propósito de recolectar información útil para su investigación [11].

Este tipo de investigación es muy importante, ya que, a través de esta podemos obtener datos o información en tiempo real en el área en donde realizaremos el estudio, haciendo que se puedan construir diseños experimentales con los datos, ahorrando mucho tiempo al momento del análisis de los mismos.

5.3.1.3. Investigación cuantitativa

La investigación cuantitativa es aquella en la que se recogen y analizan datos cuantitativos sobre variables. La investigación cualitativa evita la cuantificación. Los investigadores cualitativos hacen registros narrativos de los fenómenos que son estudiados mediante técnicas como la observación participante y las entrevistas no estructuradas. La diferencia fundamental entre ambas metodologías es que la cuantitativa estudia la asociación o relación entre variables cuantificadas y la cualitativa lo hace en contextos estructurales y situacionales [12].

En ella el contacto del investigador con el sujeto objeto de estudio es prácticamente nulo. Dado que el investigador cuantitativo mantiene este distanciamiento, su postura respecto al sujeto de estudio es la de un intruso, la de un desconocido que aplica un marco de trabajo establecido a priori sobre el objeto de su investigación y que se implica lo menos posible en el contexto social donde se desarrolla el fenómeno a ser estudiado [13].

A través de este tipo de investigación podremos obtener variables y datos más confiables que se relación entre sí. De igual manera nos servirá para tomar la muestra de una población con la cual se desea trabajar en una investigación.

5.3.2. Ingeniería De Software

La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales [14].

La ingeniería de software está compuesta por una serie de modelos que abarcan los métodos, las herramientas y los procedimientos. Estos modelos se denominan frecuentemente paradigmas de la ingeniería del software y la elección de un paradigma se realiza básicamente de acuerdo al tipo del proyecto y de la aplicación, los controles y las entregas a realizar [15].

La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza [16].

5.4. METODOLOGÍAS

La metodología describe, valida, y en cierto modo, prescribe un conjunto de reglas que el investigador debe considerar para construir y validar conocimientos científicos. Es el camino a seguir mediante una serie de operaciones, reglas y procedimientos fijados de antemano, de manera voluntaria y reflexiva, para alcanzar un determinado fin que puede ser material o conceptual [17].

El uso de alguna metodología de investigación permite organizar de una manera más efectiva el desarrollo de un software, a través de una serie de reglas a seguir y delimitaciones a cumplir. Las delimitaciones existentes son únicas por cada metodología y se designan independientemente a cada proyecto. Debido a esto existes varios tipos de

metodologías de desarrollo, que traen sus propias ventajas y desventajas en la aplicación de algún proyecto de desarrollo de software.

5.4.1. Metodologías Ágiles

Los procesos de desarrollo del software rápido se diseñan para producir rápidamente un software útil. El software no se desarrolla como una serie de incrementos, y cada uno de ellos incluye una nueva funcionalidad del sistema [18].

Los procesos de especificación, diseño e implementación están entrelazados. No existe una especificación detallada del sistema, y la documentación del diseño se minimiza o es generada automáticamente por el entorno de programación [18].

Las metodologías ágiles tienen dos diferencias fundamentales con las metodologías tradicionales; la primera es que los métodos ágiles son adaptativos –no predictivos-. La segunda diferencia es que las metodologías ágiles son orientadas a las personas –no orientadas a los procesos.

Las metodologías ágiles son adaptativas. Este hecho es de gran importancia ya que contrasta con la predictibilidad buscada por las metodologías tradicionales. Con el enfoque de las metodologías ágiles los cambios son eventos esperados que generan valor para el cliente [19].

5.4.2. Metodología Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Scrum es una de las metodologías de desarrollo ágil de Software más reconocidas a nivel mundial, su concepción data de los años 80 en análisis realizados por Ikujiro Nonaka e Hirotaka Takeuchi, en el cual resaltaron el trabajo en equipo para el desarrollo de productos y la autonomía que estos deben tener [20].

Scrum al ser una metodología de desarrollo ágil tiene como base la idea de creación de ciclos breves para el desarrollo, que comúnmente se llaman iteraciones y que en Scrum se llamarán “Sprints”.

Al ser parte de las metodologías ágiles, Scrum tiene un ciclo de vida muy parecido a este tipo de metodologías, para entender de manera clara cuál es el funcionamiento de Scrum

en un proyecto debemos tomar en cuenta este ciclo de vida. Según [21] el ciclo de vida de Scrum es:

- **Concepto:** Se define de forma general las características del producto y se asigna el equipo que se encargará de su desarrollo.
- **Especulación:** en esta fase se hacen disposiciones con la información obtenida y se establecen los límites que marcarán el desarrollo del producto, tales como costes y agendas. Se construirá el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno.
- **Exploración:** Se incrementa el producto en el que se añaden las funcionalidades de la fase de especulación.
- **Revisión:** El equipo revisa todo lo que se ha construido y se contrasta con el objetivo deseado.
- **Cierre:** Se entregará en la fecha acordada una versión del producto deseado. Al tratarse de una versión, el cierre no indica que se ha finalizado el proyecto, sino que seguirá habiendo cambios, denominados “mantenimiento”, que hará que el producto final se acerque al producto final deseado.

En Scrum existen iteraciones que dividen las funcionalidades del proyecto, por cada iteración se debe realizar reuniones, tanto al finalizar la iteración, como reuniones diarias para visualizar sus avances, a través de estas reuniones se tiene el control sobre todas las iteraciones propuestas por el grupo.

Para todo el proceso descrito Scrum tiene componentes los cuales detallan el proceso de desarrollo de la metodología y explican la manera por la cual se debe seguir dicho procedimiento. A continuación, describiremos dichas fases.

5.4.2.1. Las reuniones [21].

5.4.2.1.1. Planificación del Backlog

Se definirá un documento en el que se reflejarán los requisitos del sistema por prioridades. En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración. Se obtendrá además en esta reunión un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint.

5.4.2.1.2. Seguimiento del Sprint

En esta fase se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán:

¿Qué trabajo se realizó desde la reunión anterior?

¿Qué trabajo se hará hasta una nueva reunión?

Inconvenientes que han surgido y qué hay que solucionar para poder

5.4.2.1.3. Revisión del Sprint

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el feedback con el cliente.

5.4.2.2. Los roles [21].

5.4.2.2.1. Product Owner

Es la persona que toma las decisiones, y es la que realmente conoce el negocio del cliente y su visión del producto. Se encarga de escribir las ideas del cliente, las ordena por prioridad y las coloca en el Product Backlog.

5.4.2.2.2. ScrumMaster

Es el encargado de comprobar que el modelo y la metodología funcionan. Eliminará todos los inconvenientes que hagan que el proceso no fluya e interactuará con el cliente y con los gestores.

5.4.2.2.3. Equipo De Desarrollo

Suele ser un equipo pequeño de unas 5-9 personas y tienen autoridad para organizar y tomar decisiones para conseguir su objetivo. Está involucrado en la estimación del esfuerzo de las tareas del Backlog.

5.4.2.2.4. Usuarios

Es el destinatario final del producto.

5.4.2.2.5. Stakeholders

Las personas a las que el proyecto les producirá un beneficio. Participan durante las revisiones del Sprint.

Managers: Toma las decisiones finales participando en la selección de los objetivos y de los requisitos.

5.4.2.3. Elementos de Scrum [21]

- Product Backlog: lista de necesidades del cliente.
- Sprint Backlog: lista de tareas que se realizan en un Sprint.
- Incremento: parte añadida o desarrollada en un Sprint, es un parte terminado y totalmente operativo.

La figura 3 nos muestra cómo se desarrolla el ciclo de vida de Scrum, desde la toma de datos hasta las reuniones diarias del proyecto.

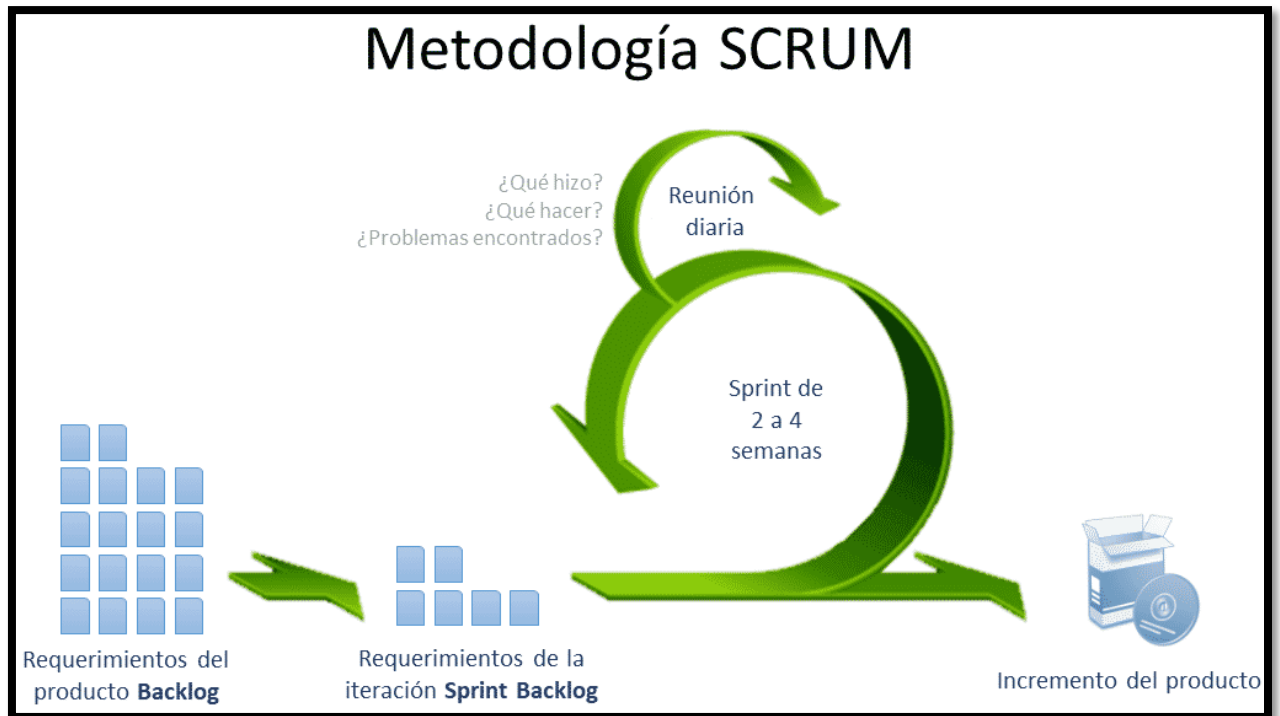


Figura 5.2: Ciclo de desarrollo de Scrum [21]

5.4.3. Metodología Mobile-D

Mobile-D es una metodología específicamente diseñada para el desarrollo de proyectos que traten sobre aplicaciones móviles, la cual es una de las más importantes entre las metodologías existentes para este tipo de aplicaciones. Está basada en las metodologías ágiles de desarrollo, por esta razón comparte gran cantidad de similitudes con las mismas. En esta metodología se recomienda que el equipo de desarrollo no pase de 10 desarrolladores y el plazo máximo para la entrega del proyecto es de 10 semanas.

El método se basa en prácticas ágiles como Extreme Programming y crystal; las prácticas asociadas a Mobile-D incluyen desarrollo basado en pruebas, la programación en parejas, integración continua y refactorización, así como las tareas de mejora de procesos de software [22].

Mobile-D consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas. Las especificaciones completas del método están disponibles en [62]. En la primera fase, Explorar, el equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento de actores, definición del alcance y el establecimiento de proyectos. Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos [22].

La figura 4 nos muestra de una manera más entendible cual es el ciclo de vida de la metodología MOBILE-D, indicando de manera rápido cuales son los puntos a tratar por cada una de las etapas de la misma.

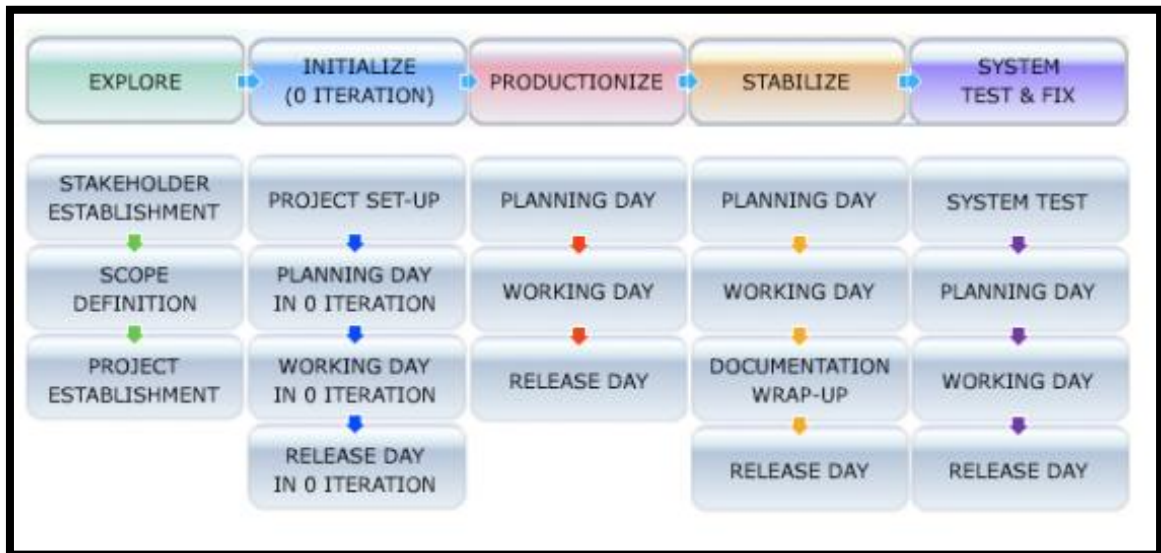


Figura 5.3: Ciclo de desarrollo MOBILE-D [23].

5.4.3.1. Exploración

En esta primera etapa es donde se definirán principalmente los interesados que tendrá el proyecto. Para esto realizaremos una entrevista al dueño de la empresa para conocer a quien va dirigida principalmente la aplicación, de la misma manera con esto se

identificará los posibles interesados que tendrá la misma y por último podremos diferenciar cuantos tipos de actores tendrá la aplicación.

Después de haber identificado los actores de la aplicación podremos darle su respectivo alcance. Donde limitaremos el objetivo principal de la aplicación e iniciaremos con la creación del plan de inicio del proyecto. En donde se desarrolla todos los procesos que iremos realizando en la primera etapa.

Y por último definiremos la arquitectura principal que tendrá la aplicación, delimitando tanto los recursos técnicos que utilizaremos como los recursos humanos que serán necesarios para la culminación del proyecto.

5.4.3.2. Inicialización

En esta fase es en donde prepararemos el inicio del desarrollo del proyecto.

Aquí elegiremos los recursos que necesitaremos para el desarrollo de la aplicación, se elegirán las herramientas necesarias considerando que el proyecto será desarrollado por el momento para Android, de la misma manera comprar los recursos técnicos por el número de programadores que participaran en el proyecto.

También se definirá quien se comunicará con el cliente en todo el desarrollo del proyecto, para la presentación de avances o cambios de necesidades de último momento.

Y por último realizaremos la planificación inicial, en donde detallaremos la arquitectura seleccionada y propondremos lo planes que realizaremos para el desarrollo de la aplicación en todo el proyecto.

5.4.3.3. Fase de producto

En esta fase realizaremos el desarrollo de la aplicación por las iteraciones que tenemos propuestas en la planificación inicial, primero verificaremos los requisitos que son propuestos por el cliente y los dividiremos para el desarrollo de las iteraciones, en esta primera parte se enfocara solo en el desarrollo de las iteraciones correspondientes al tiempo establecido.

También se tratará de acoplar cada interacción al tiempo y necesidad que tenga el desarrollador de dicha iteración, para mejorar el plan por el cual se desarrollada cada una.

Después de haber acabado con las iteraciones del proyecto entraremos al día de trabajo, en el cual, como primera instancia se dará prioridad a la iteración principal de la

aplicación y esta será la primera que se ponga en marcha antes que las demás. El equipo de desarrollo entrara a una etapa de desarrollo en pareja, el cual permitirá que estas puedan ejecutar las iteraciones de manera más efectiva para su lanzamiento.

Y por último entraremos al día de lanzamiento, en el cual pondremos a producción toda la aplicación para las pruebas exclusivas con los empleados de la empresa, aquí verificaremos que todo esté funcionando correctamente al momento de poner en marcha la aplicación en el servidor.

5.4.3.4. Fase de estabilización

En esta sección es en donde realizaremos la integración de todas las partes de la aplicación y verificaremos que esta funcione correctamente, si el equipo de desarrollo trabajo en sus subsistemas aparte, esta será la etapa más importante, ya que aquí verificaremos que las partes de los subsistemas desarrolladas por los demás desarrolladores, se puedan acoplar de manera correcta con la aplicación principal, y que esta no tenga fallas al momento de la integración completa para el lanzamiento.

5.4.3.5. Fase de pruebas

En esta fase es en donde realizaremos el plan de pruebas a proponer para la aplicación, se corregirán errores que la misma tenga y se realizara una serie de documentación para llevar los resultados obtenidos de las pruebas realizadas a cada módulo, si es que algún modulo está funcionando de manera incorrecta o este no da el resultado esperado, se lo corregirá lo antes posible.

Después de la aplicación haya pasado todas las pruebas de manera correcta se la podrá entregar al cliente (usuario final), para la puesta definitiva en marcha y salga de manera oficial al mercado al cual esta designada.

5.5. LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS PARA EL DESARROLLO DEL PROYECTO

5.5.1. Lenguajes De Programación

Un lenguaje de programación es una notación para comunicarle a una computadora lo que deseamos que haga. Desde un punto de vista formal, podemos definirlo como un sistema notacional para describir computaciones en una forma legible tanto como para la maquina como para el ser humano [24].

Un modelo de programación provee (y determina) la visión y métodos de un programador en la construcción de un programa o subprograma. Los diferentes paradigmas son el resultado de los distintos estilos de programación y las diferentes formas de pensar en la solución de problemas (con la solución de múltiples “problemas” se construye una aplicación) [25].

Los lenguajes de programación que se van a utilizar son lenguajes específicos para desarrollo web y móvil, por ende, estos serán creados en un editor de código en particular y su interprete será un navegador web o un sistema operativo móvil.

5.5.2. Framework

Con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta [26].

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones [26].

En desarrollo web podemos definir un Framework como un modelo vista controlador, el cual se encarga de agilizar los métodos por los cuales la parte grafica de la aplicación se comunica con su parte lógica o con su BD, esta también se encarga de dar una mejor seguridad a la página, haciéndola más robusta al momento de ser usada y protegiendo los datos que son enviados en tiempo real y ofreciendo una mayor usabilidad para el programador.

5.5.3. Single Page Application (SPA)

SPA significa Single Page Application, o aplicación de página única en español. El principio detrás de este término es proporcionar una aplicación web basada en los códigos ergonómicos de las aplicaciones pesadas o de tiendas online (Apple Store, Google store, Windows Store, etc.) [27].

Una SPA está compuesta de una única página, el usuario navega a través de varias vistas, pero permaneciendo siempre en esta misma página. El rol del navegador es mucho más importante que en un sitio web, pues toda la lógica del navegador es trasladada. Esto

quiere decir que el navegador debe gestionar los mecanismos de navegación entre las vistas, la recuperación de los datos y la fusión de los datos posterior para generar la página que se debe mostrar, etc. [27].

Técnicamente, una SPA es un sitio donde existe un único punto de entrada, generalmente el archivo `index.html`. En la aplicación no hay ningún otro archivo HTML al que se pueda acceder de manera separada y que nos muestre un contenido o parte de la aplicación, toda la acción se produce dentro del mismo `index.html`.

Aunque solo tengamos una página, lo que sí tenemos en la aplicación son varias vistas, entendiendo por vista algo como lo que sería una pantalla en una aplicación de escritorio. En la misma página, por tanto, se irán intercambiando vistas distintas, produciendo el efecto de que tienes varias páginas, cuando realmente todo es la misma, intercambiando vistas.

La figura 5 nos explica cuál es el ciclo de vida de una SPA comparado con una página web tradicional, como podemos ver la principal diferencia es cuando se pide la petición para ver otra vista de la página, la cual en una TWP se hace directo una nueva petición completa al server y esta devuelve una página diferente recargando toda la infraestructura de la aplicación, en cambio una SPA hace una petición sobre el componente y devuelve un JSON con los datos solicitados, haciendo que la carga de la página sea mucho más liviana y rápida.

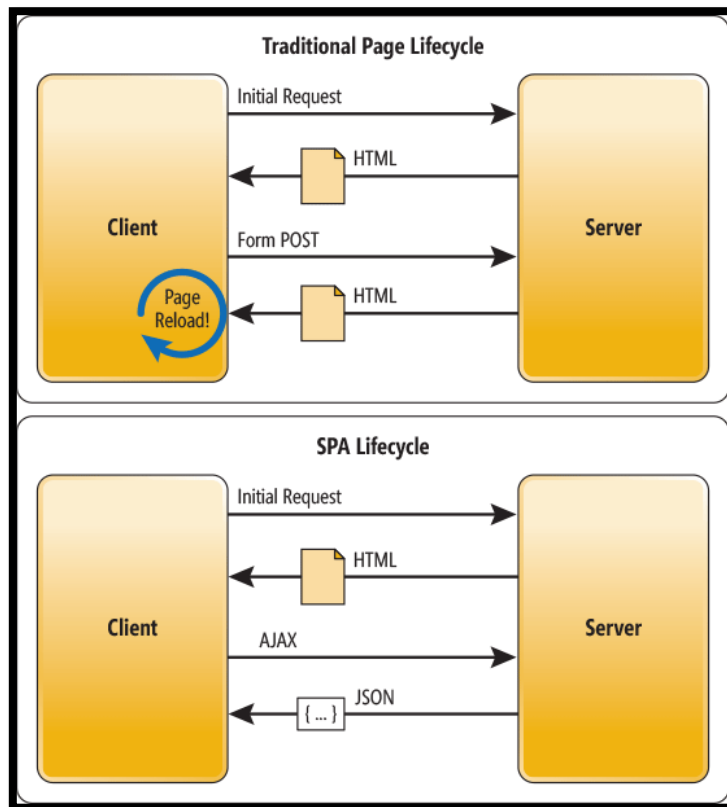


Figura 5.4: Comparación del ciclo de vida de una página web tradicional y una SPA [28]

5.5.4. Patrón De Desarrollo MVC

Fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos. Sus características principales están dadas por el hecho de que, el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas. Este modelo de arquitectura se puede emplear en sistemas de representación gráfica de datos, donde se presentan partes del diseño con diferente escala de aumento, en ventanas separadas [29].

El patrón MVC puede implementarse sin la necesidad de utilizar un framework; no obstante, y a diferencia de aplicarlo de forma manual, el framework obliga al desarrollador a utilizarlo, creando de esta forma un código mucho más robusto. De esta manera se evita el “código spaghetti”, el cual se caracteriza por agregar funcionalidades en capas que no corresponden, lo que incide negativamente sobre el código fuente y el cumplimiento de las características de calidad del producto en relación con la

mantenibilidad, especialmente sobre algunas de sus sub-características como: analizabilidad, modificabilidad y testeabilidad [30].

La figura 6 nos muestra cómo interactúan los módulos existentes en el patrón MVC, donde la vista necesita de la interacción tanto del modelo como del controlador para su funcionamiento y el controlador manda las actualizaciones al modelo para poder realizar las notificaciones en la vista y el modelo adopta la nueva modificación hecha.

5.5.5. JavaScript

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios [32].

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems, como se puede ver en <http://www.sun.com/suntrademarks/> [32].

JavaScript fue desarrollado por Netscape, a partir del lenguaje Java, el cual sigue una filosofía similar, aunque va más allá. Java es un lenguaje de programación por sí mismo, como lo puedan ser C, Pascal o VisualBasic. Esto quiere decir, que se puede ejecutar un programa Java fuera de un navegador. Se hablará más sobre Java en posteriores temas. Pero, repetimos, que la diferencia fundamental es que Java es un lenguaje completo, que puede ser utilizado para crear aplicaciones de todo tipo, mientras que JavaScript sólo “funciona” dentro de una página HTML. Por otro lado, también se puede incluir Java en páginas HTML, tal es el caso de los applets, que se podría traducir como “aplicacioncitas” [33].

JavaScript es uno de los lenguajes de programación más populares que existen actualmente, esto se debe a que ahora las personas pasan mucho tiempo visitando sitios web, y casi todos los sitios web actuales están desarrollados en parte con JavaScript. Antes este lenguaje solo se usaba para modelar páginas web, pero ahora tiene utilidad para muchos procesos que realiza la página, también se puede construir aplicaciones móviles a través de este y se lo utiliza como método para comunicarse a una base de datos, hasta para desarrollo de juegos de escritorio, y para muchos usos más.

5.5.6. MERN Stack

En la actualidad uno de los stack o conjunto de herramientas más utilizadas en el mundillo es MERN. Prácticamente lo tiene todo, es extremadamente versátil, cuenta con una amplia documentación y ejemplos, y una activa comunidad que actúa como un sólido pilar que actualiza, depura y añade funcionalidades en cada nueva versión [34].

Básicamente, la pila MERN es un conjunto de marcos/tecnologías utilizados para el desarrollo web de aplicaciones que consta de MongoDB, React JS, Express JS y Node JS como sus componentes [34].

El conjunto de técnicas MERN tiene su base en JavaScript, y son una de las herramientas más populares en la actualidad para el desarrollo de aplicaciones móviles. Este tipo de herramientas van enfocadas tanto al desarrollo Front – End como al Back – End de una aplicación web. Y todas estas herramientas son basadas en software libre, lo que hará que su implementación sea más sencilla.

5.5.7. React JS

React, también conocido como React.js o ReactJS, es una librería JavaScript OpenSource diseñada para crear interfaces de usuario. Ha sido concebida para facilitar el desarrollo de SPA, Single Page Applications, obteniendo un gran rendimiento y ofreciendo una forma de desarrollo más cercana a la creación de videojuegos que a la de aplicaciones. Esta librería está mantenida por Facebook, Instagram y una gran comunidad de desarrolladores independientes y corporaciones [35].

React intenta ayudar a los desarrolladores a crear aplicaciones web complejas que utilizan un gran intercambio de datos. Su principio es sencillo declarative and composable. React solo se preocupará de la interfaz de usuario de tu app; si piensas en el paradigma de software Modelo-Vista-Controlador (MVC) React es únicamente la V. Esto hace que puedas utilizarlo combinándolo con otras librerías JavaScript o con frameworks como AngularJS, Ember o Backbone. Además, React, gracias a su enorme y creciente comunidad, está generando una gran biblioteca de extensiones, las cuales en muchas ocasiones no se ocupan solo de la interfaz de usuario, sino que te ayudarán a complementar las necesidades de tu App [35].

React te ayuda a crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado en tu aplicación, y React se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien.

5.5.7.1. JSX

Se llama JSX, y es una extensión de la sintaxis de JavaScript. Recomendamos usarlo con React para describir cómo debería ser la interfaz de usuario. JSX puede recordarte a un lenguaje de plantillas, pero viene con todo el poder de JavaScript [36].

La figura 7 nos muestra cómo se usa la sintaxis JSX al momento de realizar un código en React, esta es la mezcla del lenguaje JavaScript con HTML (XML) interactuando entre sí en un solo documento.

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;

ReactDOM.render(
  element,
  document.getElementById('root')
);
```

Figura 5.5: Código JSX en React JS [36].

5.5.7.2. Componentes

Los componentes permiten separar la interfaz de usuario en piezas independientes, reutilizables y pensar en cada pieza de forma aislada [36].

Los componentes pueden referirse a otros componentes en su salida. Esto nos permite utilizar la misma abstracción de componente para cualquier nivel de detalle. Un botón, un cuadro de diálogo, un formulario, una pantalla: en aplicaciones de React, todos son expresados comúnmente como componentes [36].

En la figura 8 nos muestra la estructura de un componente. Los componentes son estructuras que son parte de la SPA. Cada sección que sea única y se pueda reutilizar en la página puede ser dividida en un componente.

```

function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}

function App() {
  return (
    <div>
      <Welcome name="Sara" />
      <Welcome name="Cahal" />
      <Welcome name="Edite" />
    </div>
  );
}

ReactDOM.render(
  <App />,
  document.getElementById('root')
);

```

Figura 5.6: Componente funcional en React Js [36].

5.5.8. Express JS

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla. Express proporciona una delgada capa de características de aplicación web básicas, que no ocultan las características de Node.js que tanto ama y conoce [37].

Express es el framework web más popular de Node, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para: Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas) [38].

Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.

Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta [38].

Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición [38].

Express JS se creó en base a la necesidad de que Node JS pueda interactuar en el desarrollo de aplicaciones web, ya que este tipo de lenguaje iba encaminado a otros propósitos, pero al ser de gran utilidad y poseer mejor compatibilidad con las

herramientas Front de JS se vieron en la necesidad de acoplar una nueva herramienta que permita ser el puente entre Node y el Front de la página web, de esta manera creándose Express JS.

5.5.9. Node JS

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables [39].

En la figura 9 podemos ver un ejemplo de las conexiones simultaneas que puede tener en el servidor, donde se activa el callback si es que hay algo llamando al servidor.

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hola Mundo');
});

server.listen(port, hostname, () => {
  console.log(`El servidor se está ejecutando en http://${hostname}:${port}/`);
});
```

Figura 5.7: Creación de los servicios del servidor[39].

Esto contrasta con el modelo de concurrencia más común de hoy en día, en el que se emplean hilos del Sistema Operativo. Las redes basadas en hilos son relativamente ineficientes y muy difíciles de usar. Además, los usuarios de Node.js están libres de preocuparse por el bloqueo del proceso, ya que no existe. Casi ninguna función en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea. Por ello, es muy propicio desarrollar sistemas escalables en Node.js [39].

El objetivo principal de Node JS es el de permitir ejecutar código JavaScript en el lado del servidor y no solamente en un navegador. Este tiene módulos que funcionan para dar una mejor interactividad al usuario con el servidor, permitiéndole reutilizar funcionalidades de código preinstaladas en el servidor para alguna acción en la sección del Front-End de la aplicación web.

5.5.10. Mongo DB

MongoDB es un software que permite a los usuarios crear y manipular bases de datos presentados en forma de así llamados documentos, los cuales son almacenados en un formato BSON (Binary JSON). La creación de la base de datos no requiere de previa definición de su esquema contrario a bases relacionales. Esta característica permite a los desarrolladores crear un sistema más flexible a futuras modificaciones de la estructura de documentos en comparación con bases de datos relacionales [40].

MongoDB es uno de los sistemas de bases de datos NoSQL más conocidos y debido a esto en muchas ocasiones es contrastado con sistemas relacionales y no relacionales. Sin embargo, MongoDB por si solo ofrece diferentes opciones de implementación, como, por ejemplo, la posibilidad de definir documentos anidados o referenciados, el uso del operador lookup para contrarrestar la necesidad de realizar la operación join entre documentos, la aplicación de índice para diferentes tipos de consultas, para mencionar solo algunas alternativas [40].

NoSQL (Not Only SQL) realmente es una categoría muy amplia para un grupo de soluciones de persistencia que no siguen el modelo de datos relacional, y que no utilizan SQL como lenguaje de consulta; pero, en resumen, las bases de datos NoSQL pueden clasificarse en función de su modelo de datos en las siguientes cuatro categorías [41]:

- Orientadas a clave-valor (Key-Value stores).
- Orientadas a columnas (Wide Column stores).
- Orientadas a documentos (Document stores).
- Orientadas a grafos (Graph databases).

Mongo DB es un base de datos NoSQL y este permite trabajar de manera más flexible con los datos que se almacenan sin la necesidad de construir un esquema para un base de datos relacional SQL, de esta manera la aplicación se vuelve más rápida al momento de llamar los datos del servidor a la página principal.

En la figura 10 podemos observar como son los tipos de datos de una base de datos NoSQL comparados con una base de datos SQL, la diferencia principal es que las bases de datos NoSQL guardan sus tipos de datos en objetos JSON que son enviados desde el backend, esto hace que los datos sean mucho más flexibles ya que no se necesita de una estructura obligatoria como en las bases de datos SQL.

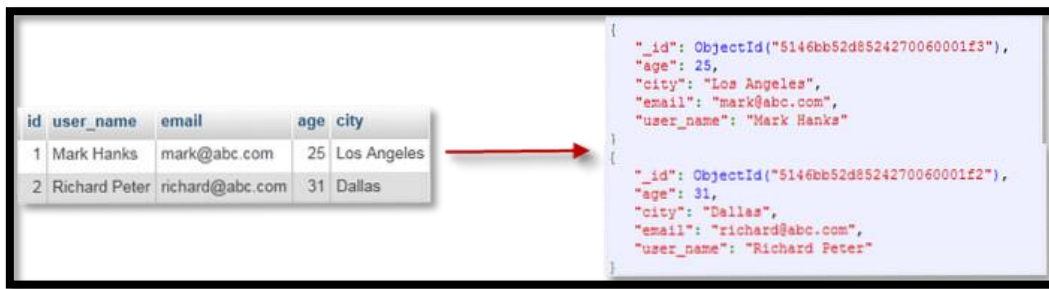


Figura 5.8: Comparación de datos entre SQL y NoSQ [41].

5.5.11. Strapi

Strapi es el CMS sin cabeza de código abierto líder. Strapi ofrece a los desarrolladores la libertad de utilizar sus herramientas y marcos favoritos al tiempo que permite a los editores administrar fácilmente su contenido y distribuirlo en cualquier lugar [42].

La interfaz de Strapi es tan intuitiva no tienes que pensar dos veces para construya sus tipos de contenido. Solo necesitas ingresar un código editor para realizar cambios en el API o su estructura de datos. La estructura del tipo de contenido puede ser flexionado de acuerdo a sus necesidades con decenas de campos disponibles y posibilidades de mezclar y combinar [42].

CMS son las siglas de Content Management System, en español Sistema de Gestión de Contenidos. En síntesis, un CMS te permite crear, organizar, publicar y eliminar contenidos de tu sitio web. Pero, por otro lado, en este artículo entenderás cómo y por qué utilizar un CMS en tu estrategia digital.

Desde el punto de vista del desarrollador un CMS es capaz de ahorrar mucho trabajo de creación de un sitio web, ya que muchas de las tareas repetitivas que se tienen que realizar en cada sitio ya se encuentran implementadas al instalar el CMS. De este modo, el profesional no tiene que preocuparse por hacer cosas como paneles de administración, construir un acceso seguro para que los usuarios puedan entrar con su clave, permitir que los visitantes dejen sus comentarios en el sitio. Y mucho más.

5.5.12. Stripe

Stripe es una compañía de tecnología que crea infraestructuras económicas para Internet. Empresas de todo tamaño desde nuevas startups hasta empresas que cotizan en bolsa aceptan pagos y administran sus negocios online con nuestro software [43].

A pesar de que las empresas en Internet crecen con más rapidez que el resto de la economía, en la actualidad solo un 3 por ciento del comercio global se realiza electrónicamente.

La complejidad regulatoria, un sistema financiero global excesivamente intrincado y la escasez de ingenieros están limitando el impacto de la economía en Internet [43].

La eliminación de las barreras al comercio electrónico ayuda a más empresas nuevas a establecerse, acelera el crecimiento de las empresas existentes y aumenta la producción económica y el comercio en todo el mundo [43].

6. MATERIALES Y MÉTODOS

Para el desarrollo de aplicaciones, tanto aplicaciones webs como móviles, existen varias técnicas y métodos los cuales otorgan, en su mayor parte, una mejor comprensión del tema y facilitan su desarrollo.

En este caso, estas aplicaciones serán desarrolladas para la comercialización de productos de la empresa “Señor de Maca”, para lo cual, se utilizará las técnicas que den mayor facilidad en la investigación y que agilicen el proceso por el cual se desarrollarán estas aplicaciones.

6.1. TIPO DE INVESTIGACIÓN

6.1.1. Investigación Descriptiva

A través de la investigación descriptiva se podrá conocer cuáles son las actividades y procesos que realiza la empresa, de esta manera se logrará determinar cuál es el problema a solventar con el desarrollo de la aplicación, de la misma manera permite construir una hipótesis en base a los datos recolectados.

6.1.2. Investigación De Campo

Con la investigación de campo se podrá recopilar los datos de las ventas de los productos que oferta la empresa, las necesidades de venta que esta necesita y las preferencias de compra de los clientes en base a los productos que ofrece la empresa. Para la recolección de los datos mencionados se utilizará dos herramientas en específico: la entrevista al propietario de la empresa y las encuestas a los clientes de la misma.

6.2. MÉTODOS DE RECOLECCIÓN DE INFORMACIÓN

6.2.1. Entrevista

A través del uso de la entrevista se podrá obtener la información relevante para el desarrollo de las aplicaciones, esta entrevista será realizada al propietario de la empresa para conocer el punto de vista que este tenga sobre la viabilidad de la aplicación, la necesidad que se quiere solventar, como se está manejando el proceso de venta que la empresa tiene, y que es lo que el propietario solicita.

Este es el método más efectivo que existe para la recolección de datos, en donde se aplica un cuestionario, el cual debe ser lo más preciso y claro posible, para de esta manera, obtener los datos más confiables para el desarrollo de la aplicación.

6.2.2. Encuesta

Mediante la encuesta se podrá analizar cuáles son las preferencias de los usuarios al momento de realizar una compra en la empresa, y las opiniones de los empleados estratégicos sobre el uso de la aplicación. De los datos obtenidos de dicha encuesta se logrará tener un desarrollo del proyecto más acorde a las necesidades del propietario y de los usuarios que utilizaran la aplicación.

Esta encuesta será aplicada a una muestra obtenida del total de la población que interactúa con la empresa, de esta manera, se podrá analizar el nivel de aceptación que tiene la propuesta tecnología en la población afectada.

6.3. DETERMINACIÓN DE LA POBLACIÓN Y MUESTRA

6.3.1. Muestreo

A través del muestreo se podrá obtener una fracción de la población a la cual debemos aplicar el estudio de campo. Esto es de gran ayuda ya que permitirá agilizar el proceso por el cual obtendremos la información para el desarrollo del proyecto, obteniendo resultados reales.

6.3.2. Muestra

La muestra será el resultado de las fórmulas obtenidas al realizar el cálculo de la población. Esta muestra será la que se utilizará para realizar el total de encuestas que se necesita en la investigación de campo.

6.3.3. Población

La población es el total de clientes que tiene aproximadamente la empresa. El dato del total de la población se obtendrá de las charlas realizadas con el propietario de la empresa.

Al momento en que se tiene la población se procederá a realizar las formulas necesarias para obtener la muestra deseada de la población especificada.

6.3.3.1. Cálculo de la población

Con el siguiente procedimiento se calculará la muestra que debemos obtener con una población finita.

6.3.3.2. Población y muestra

Población total a analizar: 150.

6.3.3.3. Descripción de las variables

n = Tamaño de muestreo buscado.

N = Tamaño de la población.

Z = Nivel de confianza.

e = Error máximo aceptado.

p = Probabilidad de que ocurra el evento.

q = Probabilidad de que no ocurra el evento.

6.3.3.4. Descripción de los datos

N = 150.

Z = 95% (1.96).

e = 5%.

p = 50%.

q = 50%.

6.3.3.5. Cálculo de la muestra

$$n = (N * Z^2 * p * q) / (e^2 * (N - 1) + Z^2 * p * q)$$

$$n = (150 * 1.962 * 50 * 50) / (0.052 (150 - 1) + 1.962 * 50 * 50)$$

$$n = 1440600 / 9604.3725$$

$$n = 148.$$

La muestra a realizar para el proyecto será de 148 participantes.

Existen algunas estrategias por las cuales permiten tener resultados homogéneos y reducir el tamaño de los datos y la variabilidad de los resultados que se obtienen, por lo cual indica que la muestra en una población no tan numerosa, como la que se tiene actualmente, no debería sobrepasar de 30 a 40 participantes.

La muestra que se ha obtenido es de 148 participantes, de los cuales tomaremos el 25% del total a encuestar.

6.3.3.6. Descripción de los variables

n=muestra población finita

m= margen de error y similitud 25%

n1= población encuestar

6.3.3.7. Descripción de los datos

n=148

m=25%

n1=?

6.3.3.8. Cálculo de la muestra

$n1 = (n * m) / 100$

$n1 = (148 * 25) / 100$

n1= 37

Del total de 148 participantes obtenidos la muestra final se encaminará a 37 personas a las cuales se les realizará las encuestas creadas.

6.4. METODOLOGÍAS DE DESARROLLO DEL PROYECTO

6.4.1. Metodología Scrum

Al no contar con mucho tiempo para el desarrollo y entrega de la propuesta tecnológica, se ha optado por la elección de una metodología ágil, en este caso, la metodología Scrum. A través de esta se podrá obtener una mejor organización de la aplicación, esta permite dividir las funcionalidades del proyecto por iteraciones, de las cuales podemos seleccionar su orden de desarrollo, dependiendo de cuál sea la más importante para el proyecto. De esta manera se logrará minimizar los riesgos y errores que pueda presentar la aplicación al momento de realizar algún cambio o agregar alguna funcionalidad extra.

En Scrum se identificará 3 actores principales que son:

- **Product Owner:** Es el responsable de maximizar el valor del trabajo del equipo de desarrollo. La maximización del valor del trabajo viene de la mano de una buena gestión del Product Backlog.
- **Scrum Master:** Es el responsable de que las técnicas Scrum sean comprendidas y aplicadas en la organización. Es el manager de Scrum, un líder que se encarga de

eliminar impedimentos o inconvenientes que tenga el equipo dentro de un sprint, aplicando las mejores técnicas para fortalecer el equipo de marketing digital.

- Equipo de desarrollo: Son los encargados de realizar las tareas priorizadas por el Product Owner. Es un equipo multifuncional y auto-organizado. Son los únicos que estiman las tareas del product backlog, sin dejarse influenciar por nadie.

6.4.2. Metodología Mobile-D

Mobile-D es una metodología ágil, exclusivamente para el desarrollo de aplicaciones móviles, permite interactuar constantemente entre el equipo de trabajo con el cliente, así como de responder rápidamente a los cambios que se puedan producir durante la etapa de desarrollo del proyecto, permitiendo la reducción de tiempos de producción.

Exploración: En esta primera etapa es donde se definirán principalmente los interesados que tendrá el proyecto. Para esto se realizará una entrevista al dueño de la empresa para conocer a quien va dirigida principalmente la aplicación, de la misma manera con esto se identificará los posibles interesados que tendrá la misma y por último se podrá diferenciar cuantos tipos de actores tendrá la aplicación.

Inicialización: Aquí se elegirá los recursos que se necesita para el desarrollo de la aplicación, se seleccionará las herramientas necesarias considerando que el proyecto será desarrollado por el momento para Android, de la misma manera, comprar los recursos técnicos por el número de programadores que participaran en el proyecto.

Fase de producto: En esta fase se realizará el desarrollo de la aplicación por las iteraciones que están propuestas en la planificación inicial, primero hay que verificar los requisitos que son propuestos por el cliente y dividirlos para el desarrollo de las iteraciones, en esta primera parte se enfocara solo en el desarrollo de las iteraciones correspondientes al tiempo establecido.

Fase de estabilización: En esta sección es en donde se realizará la integración de todas las partes de la aplicación y se verificará que esta funcione de manera correcta, si el equipo de desarrollo trabajo en sus subsistemas aparte, esta será la etapa más importante, ya que aquí se verificará que las partes de los subsistemas desarrolladas por los demás desarrolladores, se puedan acoplar de manera correcta con la aplicación principal, y que esta no tenga fallas al momento de la integración completa para el lanzamiento.

7. RESULTADOS

7.1. RESULTADOS DE LA ENTREVISTA Y ENCUESTAS

7.1.1. Resultados De Las Encuestas

El análisis se realizó en base a las encuestas aplicadas a los clientes y empleados de relevancia de la empresa “Señor de Maca”. El formato de la encuesta aplicada la podrá encontrar en la sección de anexos numeral 2. Y los análisis respectivos a cada una de las preguntas se encuentran en la sección de anexos numeral 14.

7.1.2. Reporte De La Entrevista

El presente análisis está basado en la entrevista realizada al gerente de la empresa, el modelo de la entrevista realiza lo puede encontrar en la sección de anexos numeral 4.

Con la entrevista realizada se logró deducir que la empresa no cuenta con un sistema que facilite sus ventas, debido a que la publicidad es realizada solo en el sector, es decir de manera limitada impidiendo que las personas de lugares más lejanos puedan conocer los productos que se ofertan.

Al ser una empresa nueva no se encuentra actualizada en el campo de la tecnología, es por ello que se vio necesario desarrollar un sistema por el cual se pueda mejorar las ventas, brindar un mejor servicio y poder lograr que el negocio crezca.

Con este análisis se obtuvo el punto clave que fue las características o funcionalidades que desean tener en la aplicación móvil y la página web, para lograr cumplir el objetivo y cubrir cada una de las necesidades del cliente y así tener una aceptación dentro del mercado.

7.2. HERRAMIENTAS DE PROGRAMACIÓN

7.2.1. Herramientas Principales De Desarrollo

7.2.1.1. React Js

React Js ha permitido tener un mejor orden lógico al momento de la creación de los componentes de la aplicación. De esta manera se logró tener una aplicación más compacta y fácil de entender para futuros programadores que deseen modificar objetos del proyecto. De la misma manera, se obtuvo un mejor manejo de las herramientas facilitadas por la comunidad de desarrollo de React, clasificando todas estas herramientas en dependencias que se declaran dentro de los reglamentos de uso de la aplicación.

7.2.1.2. Express Js

Express permitió tener un ambiente de desarrollo más cómodo y fácil. Permitiendo crear tablas y puntos de acceso para la base de datos de manera fácil y sencilla. Al tener un régimen menos estricto que SQL, se puede trabajar de manera más sencilla y obteniendo las mismas funcionalidades con un mejor rendimiento que SQL.

7.2.1.3. Mongo DB

Ya que todo el proyecto está basado en JS, Mongo DB se convierte en la mejor opción de uso para Bases de Datos, esta permitió una mejor estructura con los datos del sistema, ya que todos estos son manejados en formato de objetos, con Mongo DB se puede guardar todos estos datos en formato JSON, haciendo que su presentación sea mucho más ligera y al momento en el que se recuperan los datos, se hacen de una manera mucho más rápida y sencilla, logrando que los datos de nuestra aplicación carguen más rápido y sean más sencillos de utilizar.

7.2.1.4. Amazon Web Services

Al no ser recomendable guardar imágenes en una base de datos que sean más específicamente para información de la empresa y clientes. Se ha tomado la decisión de tener un almacenamiento aparte en la nube que sea exclusivamente para el uso de este tipo de archivos. AWS permite tener este tipo de almacenamiento de una manera rápida y sencilla. El resultado a la implementación de este servicio en la aplicación es una mejora considerable al momento de cargar las imágenes en la página, así también, nuestros objetos en la base de datos, al no tener una carga adicional y pesada, se renderizan más rápido en la aplicación y son muchos más livianos en la base de datos.

7.2.1.5. Stripe

Ya que se trabajará con compra y venta de productos en línea, es muy necesario tener una manera de realizar el pago de los productos de una manera rápida y segura. Con el uso de Stripe, se ha logrado desarrollar el módulo de pago a través de una tarjeta de crédito de manera exitosa. De esta manera se ha logrado implementar un pago asegurado por parte del comprador de la manera más segura posible con las herramientas de pago y seguridad que maneja Stripe.

7.2.2. Interfaces De La Aplicación

En esta sección se mostrará las interfaces más importantes que tienen las aplicaciones, y de esta manera se dará a conocer cómo funciona el sistema planteado.

7.2.2.1. Interfaz de descripción de productos

A través de esta funcionalidad el cliente puede seleccionar un producto que le interese, ver la información del mismo, ver su precio y si este se encuentra en descuento o no. Si el cliente tiene una cuenta creada y ha iniciado sesión con esta en la página, podrá añadir el producto que está observando a su lista de favoritos, de esta manera podrá tener un acceso más rápido al producto en futuras ocasiones. También el cliente podrá añadir el producto a su carrito de compras, para posteriormente poder realizar la compra del mismo. Para visualizar la interfaz diríjase a la sección de Anexos, imagen 10.60.

7.2.2.2. Interfaz de categorías de productos

En esta interfaz el cliente podrá visualizar todas las categorías que la empresa ofrece para la venta de sus productos, al momento que el cliente selecciona una categoría, se despliega la lista de productos existentes en la misma para el este puede seleccionar cualquier producto que desee. También el cliente podrá buscar los productos que el desee a través de la sección de búsqueda que existe en la parte superior de la interfaz. Para visualizar la interfaz diríjase a la sección de Anexos, imagen 10.61.

7.2.2.3. Interfaz de carrito de compra

En esta interfaz se mostrará todos los productos que el cliente haya añadido a su carrito de compras. La interfaz describirá cada producto existente, así como su precio, descuento, podrá modificar la cantidad que desea del producto, y se mostrará tanto el valor total por producto como el valor total de la compra de todo el carrito existente. Después le pedirá al usuario validar ciertas peticiones antes de proceder con el pago, como aceptar los términos y condiciones de envío, o la dirección a la cual el usuario desea mandar los productos. Por ultimo cuando haya completado todos estos pasos se mostrará un pequeño formulario en el cual el cliente podrá ingresar una tarjeta de crédito para validar su compra. Después podrá comprar los productos y si la compra se valida, todos estos se eliminaran del carrito de compras actual. Para visualizar la interfaz diríjase a la sección de Anexos, imagen 10.62.

7.2.2.4. Interfaz de datos del cliente

Esta interfaz le permite al cliente visualizar sus datos de usuario, también podrá modificarlos en caso de ser necesarios, podrá realizar la gestión de sus direcciones de entrega de productos, por otro lado, al navegar por la sección de usuario también podrá ver su lista de productos favoritos, podrá ver sus pedidos realizados y podrá visualizar los datos informativos que la empresa posee. Para visualizar la interfaz diríjase a la sección de Anexos, imagen 10.63.

7.2.2.5. Interfaz de edición de productos del administrador

En esta interfaz el administrador podrá realizar la gestión de los productos que existan en la empresa y que el mismo haya agregado. Podrá visualizar los datos más relevantes del producto en la interfaz principal y al momento de presionar un producto se logrará ver todos los datos del mismo. Este siempre ira agregado a una categoría así que el administrador primero deberá llenar una categoría nueva para poder ingresar los productos deseados. Para visualizar la interfaz diríjase a la sección de Anexos, imagen 10.64.

En esta interfaz el administrador podrá ver el total de pedidos de productos que realizan los clientes. Aquí se podrá visualizar el ID de cada pedido realizado, los productos, la cantidad, por quien fue realizado y cuál es el estado de entrega. Al ingresar a cada pedido se podrá ver los detalles del producto y la dirección por la cual fue pedida y podrá cambiar el estado del pedido cuando esté listo para su despacho. Para visualizar la interfaz diríjase a la sección de Anexos, imagen 10.65.

7.2.3. Capturas De Fragmentos De Código

En esta sección se describirá parte del código que se utilizó para el desarrollo de la aplicación, de la misma manera, solo se tomará en cuenta los datos más importantes para su respectiva descripción.

7.2.3.1. Validación de entrada de datos en formularios

La validación de los datos que ingresa tanto el cliente como el administrador son de gran importancia para que la aplicación funcione de manera correcta. Para realizar estas validaciones se ha utilizado dos librerías muy importantes para validación de datos, las cuales son “YUP”, que permite realizar las validaciones que sean necesarias agregando funciones y condiciones a la necesidad de la aplicación, y “FORMIK”, el cual permite el

manejo de los eventos que suceden en el formulario cuando el usuario utiliza el botón que dispara las acciones. Gracias a estas librerías se puede agilizar el proceso de programación y evitar el exceso de código innecesario en la misma. Para visualizar el código diríjase a la sección de Anexos, imagen 10.66.

7.2.3.2. Navegación entre componentes de la aplicación

Al estar únicamente mostrando una página en la aplicación, se necesitará una manera por la cual llamar los componentes que se visualizaran en la misma. React tiene librerías especiales para el uso de estas rutas, en la aplicación se implementó NEXT JS, el cual es un framework que proviene de React, este ya tiene su propio sistema de rutas establecida. Por el cual solo se debe agregar el componente que necesitamos mostrar en la página principal, y mandarlo a llamar con el uso del “USE ROUTER”, que viene implementado con las librerías del framework mencionado. De esta manera facilita la visualización del componente en la página y se puede realizar de una manera más rápida la validación de rutas públicas y privadas que va a tener la aplicación. Para visualizar el código diríjase a la sección de Anexos, imagen 10.67.

7.2.3.3. Uso de las Api

Al estar utilizando Express y haber creado los “END POINTS” de cada tabla de la base de datos, es posible comunicarse con la misma a través de peticiones asíncronas que la página realiza con el servidor y la base de datos. En estas peticiones se especificará el tipo de acción que se quiere realizar, seguido de los datos que se van a insertar o la consulta que se desea hacer. Al utilizar mongo BD se puede agilizar el ingreso de datos, ya que solo se obtiene el form data que viene con un objeto el cual es compatible con el modelo de datos usado por mongo, agregando todo el objeto mandado y sin tener que desestructurar el mismo, a menos que sea necesario. Para visualizar el código diríjase a la sección de Anexos, imagen 10.68.

7.2.3.4. Api para acciones de cliente registrados

Este tipo de api es especial y diferente de las otras de la aplicación, a través de esta se puede validar que el usuario haya ingresado con su cuenta en la página para realizar alguna acción. Este tipo de validación es de gran ayuda ya que hay cosas que solo puede hacer un cliente activo, como agregar productos al carrito, realizar una compra, modificar sus datos, y otras acciones en general, de esta manera se logrará delimitar lo que puede

hacer un usuario sin cuenta a otro con una cuenta activa. Para visualizar el código diríjase a la sección de Anexos, imagen 10.69.

7.3. APLICACIÓN DE LA METODOLOGÍA SCRUM

7.3.1. Roles Scrum

Tabla 7.1. Roles Scrum, Programador 1

Nombre	Carlos Galeas
Rol	Programador.
Responsabilidad	Planificar las actividades que se llevaran a cabo en el proyecto. Desarrollo de las aplicaciones a través del uso de algún lenguaje de programación.
Información de contacto.	carlos.galeas4113@utc.edu.ec
	0996117253

Tabla 7.2. Roles Scrum, Analista 2

Nombre	Byron Noteno
Rol	Analista, Tester.
Responsabilidad	Levantar los requerimientos de la manera más clara y precisa posible. Realizar las pruebas necesarias a los módulos de la aplicación en busca de posibles errores o bugs de sistema.
Información de contacto.	byron.noteno4@utc.edu.ec
	0968530292

Tabla 7.3. Roles Scrum, Scrum Master

Nombre	Ing. Alex Llano
Rol	Scrum Master.
Responsabilidad	<p>Revisar y aprobar los requerimientos de la aplicación.</p> <p>Revisar y aprobar las interfaces creadas de la aplicación.</p> <p>Revisar que las pruebas se hayan realizado de manera correcta.</p> <p>Revisar que la documentación tenga un orden lógico y lleve lo más esencial del proyecto.</p>
Información de contacto.	alex.llano9864@utc.edu.ec
	0999969302

Tabla 7.4. Roles Scrum, Product Owner

Nombre	Ing. José David Pichucho Coyachamin
Rol	Product Owner
Responsabilidad	<p>Delimitar las actividades que el equipo de desarrollo va a realizar.</p> <p>Seleccionar los puntos más importantes en el desarrollo de la aplicación.</p> <p>Recomendación sobre las herramientas de programación en el desarrollo de la aplicación.</p>
Información de contacto.	joda_david@hotmail.com
	0984644674

7.3.2. Propósito

Al aplicar la entrevista al propietario de la empresa “Señor de Maca” y la encuesta a 37 clientes, empleados de la empresa, se han obtenido los datos, que nos muestran una

viabilidad positiva para la implementación de la aplicación web y móvil. También la propuesta tuvo una gran aceptación por parte de la población mencionada.

7.3.3. Perspectiva Del Producto

La aplicación web está diseñada para ejecutarse en cualquier navegador que se haya actualizado en al menos 5 versiones más recientes y soporte las herramientas de desarrollo planteadas en JSX. La aplicación móvil puede ejecutarse en cualquier sistema operativo móvil actual (Android o IOS). De esta manera la aplicación brindara los sus servicio y productos de la manera más rápida y eficaz posible, a través de su menú de opciones.

7.3.4. Funciones Del Proyecto

Para el desarrollo de la aplicación web y móvil para la comercialización de productos en línea de la empresa “Señor de Maca”, se dividirá el sistema en módulos, los cuales pueden tener sub módulos, de acuerdo a las necesidades planteadas por el propietario de la empresa. Estos módulos se dividirán en dos actores principales, los cuales son:

Usuario (cliente de la empresa):

- Módulo de gestión de su cuenta.
- Módulo de gestión de compras.
- Módulo de mensajería instantánea.

Administrador (gerente de la empresa “Señor de Maca”)

- Módulo de gestión de usuarios.
- Módulo de gestión de administradores.
- Módulo de gestión de categorías.
- Módulo de gestión de marcas.
- Módulo de gestión de productos.
- Módulo de gestión de pedidos.
- Módulo de gestión de información empresarial.

7.3.5. Historias De Usuario

Tabla 7.5. Historias de usuario

Cliente	Administrador.
Como cliente, quiero visualizar la lista de productos existentes en la aplicación.	Como gerente, deseo subir productos que estén a la venta en mi compañía a la aplicación.
Como cliente, quiero visualizar la lista de categorías de productos existentes en la aplicación.	Como gerente, deseo subir categorías de mis productos que tenga mi compañía a la aplicación.
Como cliente, quiero buscar la existencia de algún producto en la aplicación a través de un buscador.	Como gerente, deseo relacionar las categorías que he subido con los productos existentes.
Como cliente, quiero agregar algún producto que me sea de agrado a una lista de favoritos solo mía.	Como gerente, deseo tener una manera por la cual pueda editar los datos informativos de mi empresa.
Como cliente, quiero agregar algún producto que desee comprar a algún carrito de compras.	Como gerente, deseo tener una sección en donde pueda ver que clientes se han creado una cuenta en la aplicación.
Como cliente, quiero modificar la cantidad que deseo del producto a comprar.	Como gerente, deseo tener una sección en la cual pueda ver los pedidos que ha realizado mis clientes para poder dar despacho a los mismos cuando se realiza una compra.
Como cliente, quiero agregar direcciones por las cuales me puedan enviar los productos que quiero comprar.	Como gerente, deseo subir mis marcas de productos que tiene mi empresa.
Como cliente, quiero seleccionar una forma de envío del mi producto comprado.	Como gerente, deseo poder modificar los datos de marcas cada vez que sea necesario.
Como cliente, quiero pagar mis productos comprados de una manera rápida y segura.	Como gerente, deseo poder agregar más cuentas de administrador.
Como cliente, quiero visualizar los datos informativos de la empresa.	Como gerente, deseo tener una interfaz de aplicación fácil de utilizar y amigable al mismo tiempo.

7.3.6. Requerimientos Funcionales

- El administrador podrá agregar, modificar, eliminar productos en la aplicación.
- El administrador podrá agregar, modificar, eliminar categorías relacionadas a los productos.
- El administrador podrá agregar, modificar, eliminar marcas de productos.
- El administrador podrá agregar, modificar la información general que tiene la página sobre la empresa.
- El administrador podrá modificar el estado de entrega de productos comprados por el usuario.
- El administrador podrá agregar, modificar, eliminar usuarios existentes en la aplicación.
- El administrador podrá agregar, modificar nuevos administradores para la gestión de la aplicación.
- El usuario podrá agregar productos a un carrito de compra.
- El usuario podrá seleccionar cuantos productos quiere comprar dependiendo de la cantidad que exista en stock.
- El usuario podrá crear, modificar, eliminar direcciones para la entrega de productos.
- El usuario podrá crearse una cuenta y modificar los datos de la misma.
- El usuario podrá elegir si quiere ir al local a retirar los productos o entrega a domicilio.
- El usuario podrá pagar los productos seleccionados a través de una tarjeta de crédito.
- El usuario podrá visualizar los productos que ha comprado en una sección de pedidos.
- El usuario podrá contactarse con la empresa a través de un formulario de mensajes.

7.3.7. Requerimientos No Funcionales

- El sistema debe ser capaz de procesar N transacciones por segundo.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
- El sistema debe contar con manuales de usuario estructurados adecuadamente.
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final

- El sistema debe poseer interfaces gráficas bien formadas.

7.3.8. Definición del Product Backlog

Entre el StakeHolder y el Product Owner su pudo definir todas las necesidades encontradas para la aplicación en el siguiente Product Backlog:

Tabla 7.6. Product Backlog

Ítem	Descripción	Estimación (Horas)
1	Análisis y diseño de bases de datos, casos de usos diagramas de clases.	30
2	Selección de herramientas de programación y modelado de la aplicación.	25
3	Desarrollo de la aplicación dirigida al administrador.	120
4	Desarrollo de la aplicación dirigida al cliente.	160
Total:		335

7.3.9. Definición De Los Sprints

Tabla 7.7. Sprint 1

Sprint 1		
Análisis y diseño de bases de datos, casos de usos diagramas de clases.		
Ítem	Descripción	Estimado (Horas)
1	Analizar y diseñar los diagramas de casos de usos	5
2	Analizar y crear modelo entidad relación	5
3	Analizar y crear modelo relación	9
4	Analizar y crear diagrama de clases	11
Total:		30

Tabla 7.8. Sprint 2

Sprint 2 Selección de herramientas de programación y modelado de la aplicación.		
Ítem	Descripción	Estimado (Horas)
1	Investigación de las herramientas de programación.	5
2	Creación de las aplicaciones en base a las herramientas seleccionadas	10
3	Determinación de herramientas y librerías a utilizar para la aplicación	8
4	Investigación y acoplamiento de las librerías necesarias para pagas en la aplicación.	8
Total:		25

Tabla 7.9. Sprint 3

Sprint 3 Desarrollo de la aplicación dirigida al administrador.		
Ítem	Descripción	Estimado (Horas)
1	Crear modelos en el sistema e interfaces del sistema.	15
2	Importar controladores y funciones necesarias para el funcionamiento del sistema.	15
3	Crear las vistas y validaciones necesarias para el usuario.	10
4	Crear los módulos de la aplicación.	60
5	Pruebas sobre los módulos creados	20
Total:		120

Tabla 7.10. Sprint 4

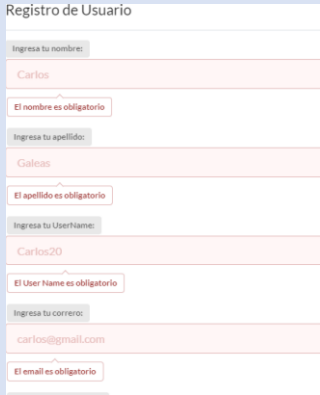
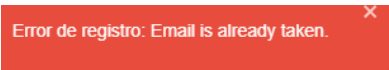
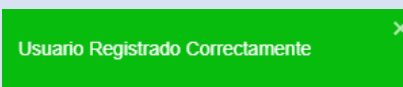
Sprint 4 Desarrollo de la aplicación dirigida al cliente.		
Ítem	Descripción	Estimado (Horas)
1	Crear modelos en el sistema e interfaces del sistema.	15
2	Importar controladores y funciones necesarias para el funcionamiento del sistema.	15
3	Crear las vistas y validaciones necesarias para el usuario.	10
4	Crear los módulos de la aplicación.	60
5	Pruebas sobre los módulos creados	20
6	Realizar las pruebas de acoplamiento con la parte del administrador.	10
7	Crear el manual de usuario del administrador	10
8	Crear el manual de usuario del cliente	10
9	Subir la aplicación a producción.	10
Total:		160

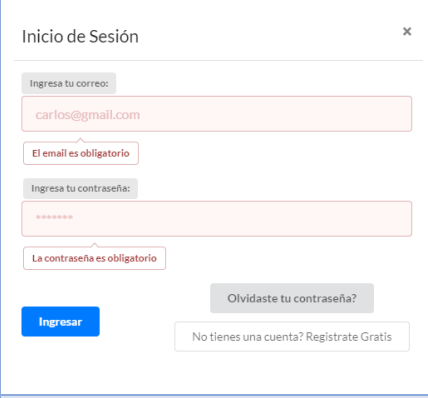
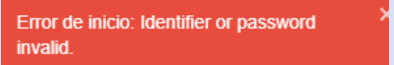
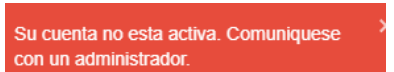
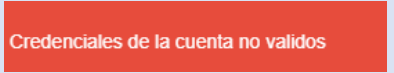

7.3.10. Pruebas De Software En Los Módulos

En esta sección se presentará las pruebas de software realizadas en base a los módulos estipulados en cada uno de los sprints descritos. Al tener una gran cantidad de módulos en la aplicación, solo se describirá las pruebas de los módulos más importantes de las aplicaciones, tanto del lado del usuario como la del administrador.

7.3.10.1. Pruebas de inicio de sesión y registro del cliente

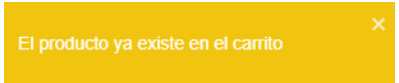

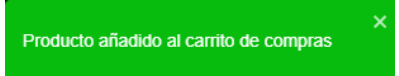
Tabla 7.11. Pruebas de inicio de sesión del cliente

No.	Variables	Se Espera	Se Obtuvo	Imagen
1	Formulario de registro vacío.	El sistema tiene que mostrar en rojo los campos que son obligatorios.	El sistema muestra los campos en rojo que no están completos.	 <p>Registro de Usuario</p> <p>Ingresar tu nombre:</p> <p>Carlos</p> <p>El nombre es obligatorio</p> <p>Ingresar tu apellido:</p> <p>Galeas</p> <p>El apellido es obligatorio</p> <p>Ingresar tu Username:</p> <p>Carlos20</p> <p>El User Name es obligatorio</p> <p>Ingresar tu correo:</p> <p>carlos@gmail.com</p> <p>El email es obligatorio</p>
2	Correo ya registrado en la aplicación	El sistema tiene que mostrar una alerta con el mensaje: ‘Error de registro: Correo ya registrado’.	El sistema muestra la alerta con el mensaje descrito.	 <p>Error de registro: Email is already taken. ✕</p>
3	Usuario registrado.	El sistema tiene que mostrar una alerta con el mensaje: ‘Usuario registrado correctamente’.	El sistema muestra la alerta con el mensaje descrito.	 <p>Usuario Registrado Correctamente ✕</p>

4	Formulario de inicio de sesión vacío.	El sistema tiene que mostrar en rojo los campos que son obligatorios.	El sistema muestra los campos en rojo que no están completos	 <p>Inicio de Sesión</p> <p>Ingresar tu correo: carlos@gmail.com El email es obligatorio</p> <p>Ingresar tu contraseña: ***** La contraseña es obligatorio</p> <p>Ingresar</p> <p>Olvidaste tu contraseña? No tienes una cuenta? Regístrate Gratis</p>
5	Datos de inicio de sesión incorrectos	El sistema tiene que mostrar una alerta con el mensaje: 'Error de inicio: Correo o contraseña incorrectas'.	El sistema muestra la alerta con el mensaje descrito.	 <p>Error de inicio: Identifier or password invalid.</p>
6	Cuenta de usuario inactiva	El sistema tiene que mostrar una alerta con el mensaje: 'Su cuenta no está activa'.	El sistema muestra la alerta con el mensaje descrito.	 <p>Su cuenta no esta activa. Comuniquese con un administrador.</p>
7	Inicio de sesión con una cuenta que no es de usuario,	El sistema tiene que mostrar una alerta con el mensaje: 'Credenciales de cuenta no validos'.	El sistema muestra la alerta con el mensaje descrito.	 <p>Credenciales de la cuenta no validos</p>
8	Inicio de sesión correcto.	El sistema tiene que mostrar una alerta con el mensaje: 'Bienvenido: nombre de usuario'.	El sistema muestra la alerta con el mensaje descrito.	 <p>Bienvenido: Carlos Galeas</p>

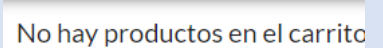

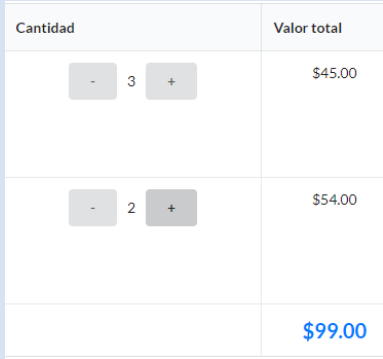
7.3.10.2. Pruebas para agregar productos al carrito del cliente




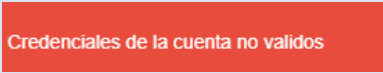

Tabla 7.12. Pruebas de agregar productos al carrito del cliente

No.	Variables	Se Espera	Se Obtuvo	Imagen
1	Usuario sin inicio de sesión.	El sistema tiene que mostrar una alerta con el mensaje: ‘Debe estar logueado para comprar’.	El sistema muestra la alerta con el mensaje descrito	
2	Producto ya existe en el carrito de compra.	El sistema tiene que mostrar una alerta con el mensaje: ‘El producto ya existe en el carrito’.	El sistema muestra la alerta con el mensaje descrito.	
3	Producto fuera de stock.	El sistema tiene que deshabilitar el botón de compra y poner el texto: ‘No hay stock’	El sistema deshabilita le botón y cambia el mensaje dentro del mismo..	
4	Producto añadido al carrito de compra.	El sistema tiene que mostrar una alerta con el mensaje: ‘Producto añadido al carrito de compra’.	El sistema muestra la alerta con el mensaje descrito	

7.3.10.3. Pruebas para realizar compras de productos


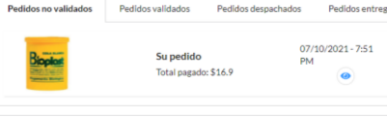

Tabla 7.13. Pruebas de compras de productos

No.	Variables	Se Espera	Se Obtuvo	Imagen
1	Sin productos agregados al carrito.	El sistema tiene que mostrar la interfaz de carrito de compra con el mensaje 'No hay productos en el carrito'.	El sistema muestra la interfaz con la descripción indicada.	
2	Con productos en el carrito.	El sistema tiene que mostrar una tabla con los datos de los productos que fueron agregados al carrito por el usuario..	El sistema muestra la tabla con los productos descritos.	
3	Incremento o decremento de productos pedidos	El sistema tiene que cambiar la cantidad de productos, precio unitario y precio total cuando el usuario presione el botón de incremento o decremento de un producto.	El sistema muestra los nuevos datos descrito en la tabla para cada producto que fuese aumentado.	

4	Al no aceptar los términos ni tener una dirección de envío.	El sistema no tiene que mostrar el botón de compra hasta que las opciones de términos y condiciones estén activas.	El sistema no muestra el formulario si las opciones dichas no están activas.	
5	Al aceptar los términos ni tener una dirección de envío.	El sistema tiene que mostrar el botón de compra existe para los productos..	El sistema muestra el formulario de compra de productos.	
6	Opción de pedido a través de depósito bancario.	El sistema tiene que mostrar el botón con unas indicaciones adicionales de cómo realizar la compra con un depósito bancario.	El sistema muestra el botón con los datos necesario de su descripción.	
7	Pedido a través de depósito bancario.	El sistema tiene que mostrar una alerta con el mensaje: ‘Pedido realizado’, cuando se pide un producto a través de un depósito bancario.	El sistema muestra la alerta con el mensaje descrito.	
8	Opción de compra a través de una tarjeta de crédito.	El sistema tiene que mostrar el formulario para ingresar una tarjeta de crédito para la compra de productos.	El sistema muestra el formulario descrito.	

7.3.10.4. Pruebas para validar pedidos realizados

Tabla 7.14. Pruebas de validaciones de pedidos

No.	Variables	Se Espera	Se Obtuvo	Imagen
1	Sin productos pedidos.	El sistema tiene que mostrar la interfaz de pedidos con el mensaje 'No tienes compras por validar'	El sistema muestra la interfaz con la descripción indicada.	
2	Con productos pedidos.	El sistema tiene que mostrar una tabla con los datos de los productos que fueron pedidos por el usuario	El sistema muestra la tabla con los productos descritos.	
3	Sin haber seleccionado una imagen para subir como voucher.	El sistema tiene que mostrar una alerta con el siguiente mensaje 'Debe seleccionar una imagen'	El sistema muestra la alerta con el mensaje descrito.	

7.4. APLICACIÓN DE LA METODOLOGÍA MOBILE-D

7.4.1. Exploración

En esta fase se definió cuáles son los requisitos iniciales de la aplicación y a los usuarios principales de la misma.

7.4.1.1. Declaración del grupo de interés

Las personas involucradas en el desarrollo de la aplicación son:

- Desarrolladores:

Son los especialistas de software, cuya función en específica es la codificación del aplicativo móvil en base a las funcionalidades presentadas por el cliente.

- Gerente de la empresa Señor de Maca:

Es la persona que administra la empresa a la cual va dirigida el aplicativo, esta se encarga de definir los requisitos iniciales para el funcionamiento de la aplicación.

- Cliente:

Es la persona que compra los productos o servicios que ofrece la empresa Señor de Maca.

7.4.1.2. Requisitos iniciales

Se desarrolló un aplicativo móvil híbrido con el uso del frameworks React Native que permita realizar la compra de productos en línea para los clientes que posee la empresa Señor de Maca.

Para realizar dicha aplicación se ha utilizado la información obtenida de parte del gerente, la cual se transformó en requerimientos. Estos requerimientos están detallados en la metodología Scrum utilizada para el desarrollo del aplicativo web. Para visualizar dichos requerimientos diríjase a la sección 5.3, Metodología Scrum, del documento.

7.4.1.3. Definición de alcance

El alcance del aplicativo móvil supone tener algunas limitaciones y posibles supuestos.

- Limitaciones:

Las posibles limitaciones que tendrá el aplicativo móvil son las siguientes:

- La aplicación móvil requiere de manera obligatoria de una conexión a internet para navegar y comprar productos en la misma.

- Los dispositivos móviles en los cuales se instalará la aplicación deben tener al menos un sistema operativo Android 6 o IOS 10.

En este caso, la aplicación móvil solo trabajara en la plataforma Android.

- Establecimiento de la categoría:

Los posibles supuestos que se han encontrado son los siguientes:

- El servicio de entregas de la aplicación móvil solo podrá ser utilizado para las personas que vivan por las zonas aledañas a la ciudad de Latacunga.
- La aplicación obtendrá sus datos de la Api construida en base a la aplicación web principal.

7.4.2. Inicialización

En esta fase se describirá las fases que se contemplan para el desarrollo de la aplicación y las configuraciones necesarias para su desarrollo.

7.4.2.1. Configuración del ambiente de desarrollo

Para esta actividad el participante principal será el desarrollador, aquí se definirá el ambiente de trabajo idóneo para el desarrollo de la aplicación y se definirán las herramientas necesarias a utilizar.

- Tipo de proyecto: Aplicación hibrida.
- Framework para la aplicación móvil: React Native, React.
- Framework para la aplicación web: Express, Node JS.

7.4.2.2. Preparación del ambiente de desarrollo

Se realizará una instalación de todas las herramientas necesarias para el desarrollo de la aplicación. Entre las cuales tenemos:

- Visual Studio Code.
- Node Js.
- Mongo DB Compas.
- Express Js.
- React Native.
- React.
- Postman.

7.4.2.3. Capacitaciones

Por el uso de nuevas tecnologías en desarrollo de aplicaciones híbridas, fue necesario que los desarrolladores sigan un curso orientado al desarrollo de este tipo de aplicaciones. Y de esta manera logren tener un mayor conocimiento sobre el tema y solventen las dudas existentes sobre el tema.

7.4.2.4. Planificación de las fases

Tabla 7.15. Planificación de fases para el desarrollo de la aplicación

Fase	Iteración	Descripción
Exploración	Iteración 1	Establecimiento de requisitos, alcance, limitaciones, supuestos y grupo de interés de la aplicación.
Inicialización	Iteración 1	Análisis de requisitos iniciales, instalación de herramientas necesarias, capacitaciones de los desarrolladores.
Producción	Iteración 2	Desarrollo e implementación de la funcionalidad de registro e inicio de sesión del cliente.
		Desarrollo e implementación de la funcionalidad de navegación entre categorías, productos de la empresa y acción de añadir productos al carrito de compras.
		Desarrollo e implementación de la funcionalidad de carrito de compras del cliente
	Iteración 3	Desarrollo e implementación de la funcionalidad de selección de método de envío y selección de método de compra del cliente.

		Desarrollo e implementación de la funcionalidad de visualización de órdenes de pedido y validaciones de pedidos del cliente.
		Desarrollo e implementación de la funcionalidad de edición de datos personales del cliente
Estabilización	Iteración 4	Refactorización de las interfaces de la aplicación que sean necesarias.
		Establecimiento de las interfaces definitivas de la aplicación.
Pruebas al sistema	Iteración 4	Se realizara una evaluación a las pruebas del sistema
		Análisis de los resultados obtenidos.

7.4.2.5. Diagrama de bases de datos

Para revisar las tablas existentes de la base de datos diríjase a la sección 8.6 Modelo de base de datos, en la sección de anexos.

7.4.2.6. Esquema de navegabilidad

En esta sección se describe el diagrama de navegabilidad de la aplicación para tener una imagen más clara de la forma de navegación de la aplicación. Para visualizar el esquema de navegabilidad diríjase a la Sección de anexos, Imagen 10.71.

7.4.2.7. Diagramas de casos de uso

Para revisar los diagramas de casos de uso existentes diríjase a la sección 8.8 Casos de uso del sistema, en la sección de anexos.

7.4.2.8. Producción

En esta fase se desarrollará las interfaces descritas en la fase de planificación.

A continuación, se describirá las interfaces más importantes para el desarrollo de la aplicación móvil.

7.4.2.9. Inicio de sesión del cliente

El cliente podrá iniciar sesión en la aplicación para acceder a sus servicios, o en caso de no tener una cuenta, podrá crearse una cuenta de usuario nueva. Para visualizar la interfaz diríjase a la Sección de anexos, Imagen 10.72.

7.4.3. Visualización Del Producto Deseado

El cliente podrá seleccionar un producto de la lista de categorías existente o buscar un producto en la barra de búsqueda. Al momento de seleccionar un producto el cliente podrá visualizar los datos del mismo y tiene la posibilidad de añadir al carrito de compra o añadir a su lista de favoritos dicho producto. Para visualizar la interfaz diríjase a la Sección de anexos, Imagen 10.73.

7.4.4. Carrito De Compra

Los productos seleccionados para su compra aparecerán en esta interfaz, los productos podrán ser aumentados hasta la cantidad deseada por el cliente, se podrá seleccionar un método de entrega y método de compra para proceder con dicha compra. Las órdenes

realizadas por el cliente se guardarán en la sección de Mis pedidos. Para visualizar la interfaz diríjase a la Sección de anexos, Imagen 10.74.

7.4.5. Datos Personales

Esta interfaz mostrara la información del cliente que se encuentre logueado. El cliente puede modificar sus datos personales las veces que sea necesario. Desde esta interfaz el cliente podrá ingresar a la sección de Mis pedidos. Para visualizar la interfaz diríjase a la Sección de anexos, Imagen 10.75.

7.4.6. Mis Pedidos

En esta interfaz se mostrarán los pedidos o compras realizadas por el cliente. Estas estarán clasificadas por compras realizadas con tarjeta de crédito o con depósito bancario. En caso de con depósito bancario, el cliente podrá subir un voucher para validar el pedido realizado. Para visualizar la interfaz diríjase a la Sección de anexos, Imagen 10.76.

7.4.7. Estabilización

En esta fase describiremos los estándares de codificación aplicados y la estructura del directorio resultante del desarrollo de la aplicación.

7.4.7.1. Estándares de codificación

- Componentes: los nombres de los componentes están escritos la primera letra en mayúscula, de esta manera podremos diferenciarlos de las vistas y serán más fáciles de utilizar al momento de utilizarlo en algún servicio o vista.
- Vistas: los nombres de las vistas estarán escritos en minúsculas.
- Servicios: los servicios adicionales creados en la aplicación serán nombrados en letras minúsculas más la palabra ‘use’ al principio de cada uno, ya que esta es la nomenclatura que maneja React para el uso de sus hooks.

7.4.7.2. Estructura de los directorios

Se especifica la estructura del directorio de la aplicación móvil. Para visualizar la estructura de los directorios diríjase a la Sección de anexos, Imagen 10.77.

7.4.8. Pruebas

7.4.8.1. Pruebas funcionales

Se comprueba que las vistas de las interfaces mencionadas en el desarrollo de la aplicación cumplan con las funciones establecidas.

Tabla 7.16. Pruebas funcionales.

Id	Vista	Cumplimiento
1	Inicio de Sesión o registro de cliente.	Hecho
2	Navegación entre productos, categorías y búsqueda.	Hecho
3	Agregar un producto al carrito de compras.	Hecho
4	Realizar un pedido de productos a través del carrito de compras	Hecho
5	Visualización y modificación de los datos del cliente.	Hecho
6	Visualización de órdenes de pedido y validación de pedidos realizados.	Hecho
7	Cerrar sesión en la aplicación	Hecho

7.5. CONFIGURACIÓN DEL SERVIDOR DE DESPLIEGUE

Para el alojamiento de las aplicaciones web, tanto del administrador como usuario, se ha seleccionado los servidores y dominios de Amazon Web Services (AWS), ya que cuentan con una infraestructura compatible con las herramientas usadas en el desarrollo de las aplicaciones, y cuenta con servicios adicionales que se utilizarán para las múltiples funciones que tienen las aplicaciones.

Para acceder a todos los servicios que ofrece AWS se debe poseer una cuenta activa en la página, de preferencia con una suscripción anual para tener acceso a todos los servicios, de manera adicional, debe tener creado un usuario de desarrollo en la consola de AWS.

7.5.1. Uso De AWS Amplify Como Servidor De Las Aplicaciones

Se ha implementado las aplicaciones en AWS Amplify por los múltiples beneficios que brinda esta plataforma para aplicaciones basadas en Node JS, esta permite seleccionar que tipo de aplicación es, permite configurar por consola los lineamientos para realizar el despliegue de la aplicación, y tiene un ambiente de desarrollo e implementación más cómodo porque permite realizar cambios en la aplicación sin la necesidad de realizar un cambio total de todos los componentes de la misma, solamente montando los cambios realizados en la plataforma Github.

7.5.1.1. Carga de los archivos en la plataforma Github

AWS Amplify cuenta con una conexión de archivos de la plataforma Github, esto lo hace para optimizar los procesos de carga de la aplicación, extrayendo los datos encontrados en Github y solo generando el despliegue de la aplicación, esto hace que el servidor tenga menos peso de carga de archivos y que estos sean más susceptibles al cambio, ya que las posibles modificaciones se las hacen en tiempo real desde Github.

Para poder subir proyectos a Github se debe tener creado una cuenta en la misma aplicación, de manera adicional, desde el equipo de cómputo por el cual se va a subir la aplicación, debe tener instalado la aplicación de consola Git de manera global. Y tenerla configurada con el usuario creado en Github.

Ahora para subir la aplicación, se debe dirigir a la página de Github, iniciar sesión, y crear un nuevo proyecto. Github pedirá que se agregue un nombre y descripción al proyecto. Después de llenar estos datos, como lo muestra la figura 10.78 en anexos, nos darán los códigos que se tienen que agregar en la consola de la carpeta de la aplicación para subir el proyecto a Github.

Al agregar los códigos descritos en la consola de la carpeta de la aplicación, Git subirá automáticamente todo el repositorio con los archivos existentes a Github. La figura 10.79, en anexos, muestra la estructura de archivos del proyecto en Github.

7.5.1.2. Creación y despliegue de la aplicación en AWS Amplify

Una vez se tenga agregado el proyecto que se va a subir al servidor en Github, se procederá a cargar este a los servicios Amplify de AWS.

Para la contratación del servidor nos ubicaremos en el panel de búsqueda de AWS, buscar “AWS Amplify” e ingresar a la primera opción que nos muestre la búsqueda.

Los servicios de Amplify nos mostrarán que clase de aplicación queremos subir, en este caso seleccionaremos la opción de aplicaciones basadas en Node Js y React Js. Amplify tiene varias formas de cargar archivos para realizar su desarrollo, como lo muestra la figura 10.80, en anexos, se seleccionará la carga a través de Github.

Después Amplify pedirá que se dé acceso a una cuenta de Github desde la que se realizará el desarrollo de la aplicación existente, al dar acceso con el usuario deseado, como se puede visualizar en la figura 10.81, en anexo, se mostrará los proyectos que se tiene creados en la cuenta de Github, se debe seleccionar el que se desea subir al host.

Una vez seleccionado el proyecto, se debe configurar la codificación que llevara nuestra aplicación, a través de esta codificación Amplify realizará el desarrollo respectivo y agregará la aplicación al host en configuración, en función de las herramientas a usar en la aplicación o la versión que queramos cambiar, modificaremos el código pre establecido. La figura 10.82, en anexos, muestra el código utilizado por Amplify para realizar el desarrollo de la aplicación.

Después de configurar el código, Amplify realizará todas las configuraciones necesarias para subir la aplicación, como nos muestra la Figura 10.83, en anexos, después de realizar todas las verificaciones necesarias, Amplify pone en línea la aplicación para su uso.

Al abrir el enlace que nos proporciona Amplify se puede verificar que la aplicación se encuentra funcionando de manera correcta en el host adquirido.

7.5.1.3. Configuración del dominio de la aplicación

Una vez que la aplicación se encuentre funcionando correctamente en el host se deberá configurar el dominio por el cual se va a ingresar a la aplicación, aquí se puede utilizar cualquier dominio deseado de cualquier página, para ese caso, se utilizará dominios adquiridos en AWS Route 53, los cuales se explicará su adquisición en la siguiente sección.

Con el dominio adquirido, se trasladará a la sección de Manejo de Dominios de Amplify, existente en el host, como nos muestra la figura 10.84, en anexos, el dominio predeterminado será el que nos ofrece Amplify por defecto. Como no es el dominio que se necesita en la aplicación será cambiado por uno nuevo.

Se agregará un nuevo dominio presionando el botón de agregar dominio, en esta sección nos pedirá elegir un dominio el cual debe estar activo en el lugar que fue adquirido, al seleccionar el dominio podremos realizar las configuraciones de re direccionamiento que va a tener nuestro dominio al momento que el usuario ingrese en él, la figura 10.85, en anexos, nos muestra las configuraciones necesarias para lograr aplicar el dominio en el host.

Al utilizar un dominio propio de AWS te da la facilidad de la configuración automática del certificado SSL al agregar dicho dominio a un host de Amplify.

La figura 10.86, en anexos, nos muestra la configuración del certificado SSL, re direccionamiento y activación del dominio que realiza Amplify de manera automática al momento de agregar un servidor propio de AWS al host de la aplicación.

Una vez realizada la configuración automática de los dominios, Amplify seleccionara el nuevo dominio como el predeterminado para el uso de la aplicación.

La figura 10.87, en anexos, muestra el resultado final de la configuración del host con el nuevo dominio adquirido.

7.5.1.4. Costos de implementación del host con el dominio establecido

Primero se calculará el valor total de un host con su servidor y después se calculará el total de ambas aplicaciones.

AWS Amplify tiene un método para cobrar por sus hosts dependiendo de la cantidad de visitas y el número de cambios que se realiza en la aplicación de manera mensual.

Se hará el cálculo en base a un promedio de 10000 usuario activos diarios y con dos actualizaciones mensuales de la aplicación.

- Cargos mensuales de implementación

Cargos mensuales de implementación = número de actualizaciones al mes * tiempo promedio de creación.

Cargos mensuales de implementación = $(2 * 3) * 0.01$.

Cargos mensuales de implementación = 0.06 USD.

- Cargos de alojamiento mensuales

Datos: tamaño de la aplicación web = 100 MB, tamaño promedio de la página solicitada = 1,5 MB

GB de uso por mes = usuarios activos por día * tamaño promedio de la página * días =
 $10\,000 * (1,5/1024) * 30 = 439,45$ GB

GB de almacenamiento por mes = tamaño de la aplicación web * número de creaciones mensuales = $(100/1024) * 2 = 0,19$ GB

Cargos mensuales de alojamiento = $439,45 * 0,15$ USD + $0,19 * 0,023$ USD = 65,92 USD

- Cargos mensuales totales.

Cargos totales = Cargos de alojamiento mensuales + Cargos mensuales de implementación

Cargos totales = 65.92 + 0.06

Cargos totales = 65.98

Al tener dos aplicaciones los costos se duplicarían, por ende, los costos totales a pagar por mes son de 131.96 USD para mantener las dos aplicaciones en línea.

De manera adicional, tendría un costo anual adicional de 24 USD para mantener los dominios en línea.

7.5.2. Registro De Dominios Con AWS Route 53

A continuación, se va a describir como fue el procedimiento para la contratación y configuración de los dominios que van a utilizar las aplicaciones con la contratación del servicio de dominios Route 53 que ofrece AWS.

Para la contratación del dominio nos ubicaremos en el panel de búsqueda de AWS, buscar "Route 53" e ingresar a la primera opción que nos muestre la búsqueda.

7.5.2.1. Contratación de los dominios

En la pantalla de servicios de Route 53 se encuentra la sección de dominios, en donde se podrá gestionar los dominios contratados, se dará click en "Agregar nuevo dominio". Se desplegará la interfaz para comprar nuevos dominios.

Lo primero que debe hacer es escribir el nombre de dominio, elegir el dominio de nivel superior y comprobar que el dominio se encuentre disponible, como lo muestra la figura 10.88, en anexos, el dominio que se va a contratar se encuentra libre para la compra.

Al tener añadido el dominio al carrito, se procederá a continuar con la compra, en la siguiente sección solamente pedirá los datos personales o empresariales de la persona que este contratando los servicios.

Al tener los datos de contacto se procederá a realizar la compra, AWS mostrará el total a pagar y solamente pedirá que aceptes los términos y condiciones para realizar la compra de los dominios.

Cuando se presione el botón de realizar el pedido, el monto total se cobrará automáticamente de la tarjeta de crédito asociada a la cuenta principal del usuario. Ahora como muestra la figura 10.90, en anexos, el dominio entrara en fase de registro. El dominio tardara en activarse de 1 a 3 días.

Y para finalizar, como lo muestra la figura 10.91, en anexos, los dominios pasan a la sección de dominios registrados y quedan listos para el uso de algún host web.

7.6. ESTIMACIÓN DE COSTOS

La estimación de costos es de suma importancia para el desarrollo de las aplicaciones. De esta manera se puede determinar cuál será el capital total que necesitará la aplicación para su desarrollo e implementación.

El presupuesto para el desarrollo del presente proyecto se trabajará por la duración de 4 meses y 8 días, y 8 horas productivas se estima un costo de 3.916.4 dólares americanos.

Para revisar el total del análisis de costos estipulados diríjase a la sección 10.5. Anexo C: Estimación de costos.

7.7. TABLA DE COMPROBACIÓN DE LA HIPÓTESIS

Tabla 7.17. Tabla comparativa – Resultados de la aplicación en la empresa

Método de venta de la empresa antes de las aplicaciones	Método de venta de la empresa con las aplicaciones
<ul style="list-style-type: none"> ➤ Los pedidos se receptan en el local y personalmente. 	<ul style="list-style-type: none"> ➤ Los pedidos se realizan desde la comodidad de su hogar o lugar donde se encuentre el cliente.
<ul style="list-style-type: none"> ➤ Los pedidos se guardaban manualmente en un cuaderno de registros y en algunos casos dichos registros se extraviaban dificultando el proceso del mismo. 	<ul style="list-style-type: none"> ➤ Los pedidos se almacenan en la base de datos de la aplicación brindando seguridad y evitando que se pierdan los datos registrados.
<ul style="list-style-type: none"> ➤ La empresa no llegaba a un gran número de clientes, solamente se limitaba en el sector que se encuentra ubicada. 	<ul style="list-style-type: none"> ➤ La empresa mejoro paulatinamente con la aplicación ya que se dio a conocer y llego a más posibles clientes.
<ul style="list-style-type: none"> ➤ Las ventas de la empresa eran bajas por el motivo de que tenían poca cantidad de clientes además de que la pandemia empeoró la situación económica de la misma. 	<ul style="list-style-type: none"> ➤ La aplicación ayuda a mejorar las ventas de la empresa con la facilidad, eficiencia, comodidad y seguridad para el cliente al realizar un pedido.

8. CONCLUSIONES Y RECOMENDACIONES

8.1. CONCLUSIONES

- A través del estudio metodológico aplicado se ha evidenciado que el uso de nuevas tecnologías para la optimización de procesos para las empresas es de gran importancia, ya que estas herramientas permiten a las empresas tener un mejor control de sus actividades y agilizar los servicios que otorgan a sus clientes. De la misma manera, el uso de nuevas herramientas nos ha permitido tener un mejor ambiente desarrollo y tener mejores beneficios para la empresa, como es el caso del uso de Mongo DB, que nos permitió tener una respuesta más rápida y sencilla de utilizar al momento de traer información de esta, así mismo, con React aumentamos la velocidad de respuesta que tiene la aplicación y la carga que esta realiza al servidor, haciendo que nuestra aplicación sea más ligera y menos tediosa de usar para el usuario.
- A través de la metodología Scrum se ha desarrollado todos los módulos necesarios para la finalización de las aplicaciones para la empresa “Señor de Maca”, de esta manera, se ha presentado una aplicación acorde a las necesidades planteadas y que tiene una gran aceptación en la población a la cual va dirigida, de esta manera, obteniendo los resultados deseados por los clientes.
- Gracias al uso e implementación de nuevas tecnologías para el desarrollo de aplicaciones, hemos logrado crear interfaces de usuario que sean claras, intuitivas, llamativas y fáciles de usar, cumpliendo con todos los requerimientos funcionales y de seguridad que deben tener cada una de estas y agilizando todos los procesos descritos.

8.2. RECOMENDACIONES

- A la vez que pasa el tiempo la empresa se expande y es necesario implementar nuevas tecnologías y nuevas funcionalidades en las aplicaciones para con ello ofrecer una mejor experiencia de usuario a todos los clientes de la empresa.
- Es de suma importancia medir el nivel de aceptación y acogida que tienen las aplicaciones en los clientes, para así mantenerlas actualizadas dependiendo de las necesidades que tanto los clientes como el administrador den a conocer.
- Para que el rendimiento de las aplicaciones sea óptimo y no pierda velocidad, se debe seguir utilizando el servidor contratado por los programadores, o en caso de cambio, contratar uno que tenga una eficiencia equiparable con el servidor actual de la aplicación.
- Para el uso correcto de la aplicación, tanto del cliente como del administrador, o si se tiene alguna duda acerca del funcionamiento de alguna parte de la misma, se debe revisar el manual de usuario, en la cual se describen de manera precisa el funcionamiento de las aplicaciones.

9. BIBLIOGRAFÍA

- [1] H. Global, «¿Qué es un ferretero?,» Notas de prensa Online, 1 Junio 2019. [En línea]. Available: <https://shern.net/que-es-un-ferretero/>. [Último acceso: 19 Mayo 2021].
- [2] GCFGlobal, «¿Qué son las aplicaciones?,» GCFGGlobal, 15 Febrero 2018. [En línea]. Available: <https://edu.gcfglobal.org/es/cultura-tecnologica/que-son-las-aplicaciones-o-programas/1/>. [Último acceso: 19 Mayo 2021].
- [3] S. L. Mora, Programación de aplicaciones web: historia, principios básicos y clientes web, Alicante: Club Universitario, 2012.
- [4] J. P. Pérez, *Estudio y clasificación*, Valencia, 2013.
- [5] J. M. Aguado, «Tendencias evolutivas del contenido digital en aplicaciones móviles,» *El profesional de la información*, vol. 24, nº 6, pp. 787-795, 2015.
- [6] R. R. Vique, Métodos para el desarrollo de aplicaciones móviles, Catalunya: Creative Commons, 2012.
- [7] O. C. Mimenza, «Los 15 tipos de investigación (y características),» *Psicología y Mente*, 18 Diciembre 2015. [En línea]. Available: <https://psicologiaymente.com/miscelanea/tipos-de-investigacion>. [Último acceso: 19 Mayo 2021].
- [8] F. Morales, Conozca 3 tipos de investigación: Descriptiva, Exploratoria y Explicativa, 2012.
- [9] S. TRES, METODOLOGÍA DE LA INVESTIGACIÓN I, 2008.
- [10] E. R. Arias, «Investigación de campo,» Conomipedia, [En línea]. Available: <https://economipedia.com/definiciones/investigacion-de-campo.html>. [Último acceso: 19 Mayo 2021].
- [11] A. Cajal, «Investigación de campo: características, diseño, técnicas, ejemplos,» lifeder, [En línea]. Available: <https://www.lifeder.com/investigacion-de-campo/>. [Último acceso: 19 Mayo 2021].
- [12] S. PITA FERNÁNDEZ, «Investigación cuantitativa y cualitativa,» *Cad Aten Primaria*, vol. 9, pp. 76-78, 2002.

- [13] N. U. BINDA, «Investigación cuantitativa e investigación cualitativa: buscando las ventajas de las diferentes metodologías de investigación,» *Revista de Ciencias económicas*, vol. 31, nº 2, pp. 179-187, 2013.
- [14] R. S. PRESSMAN, *Ingeniería del software*, Educacion, 1988.
- [15] Z. CATALDI, «Ingeniería de software educativo,» *Proceedings del V Congreso Internacional de Ingeniería Informática*, pp. 185-199, 1999.
- [16] I. Sommerville, *Ingeniería del software*, Pearson educación., 2005.
- [17] A. Vélez, *Metodología de la investigación*, Medellín: EAFIT, 2011.
- [18] R. S. Pressman, *Ingeniería de Software I*, 2011.
- [19] A. N. CADAVID, «Revisión de metodologías ágiles para el desarrollo de software,» *Prospectiva*, vol. 11, nº 2, pp. 30-39, Prospectiva.
- [20] C. RODRÍGUEZ, «¿ Por qué implementar Scrum?,» *Revista Ontare*, vol. 3, nº 1, pp. 125-144, 2015.
- [21] T. GALLEGO, *Metodologia scrum*, 2012.
- [22] Y. D. A. BALAGUERA, «Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles,» *Revista de Tecnología*, vol. 12, nº 2, pp. 111-123, 2013.
- [23] P. BLANCO, «Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone,» *Dr. en Ing. Sist. Telemáticos*, pp. 1-30, 2009.
- [24] F. G. BELLAS, R. M. UNANUE y V. D. F. FERNÁNDEZ, *Lenguajes de programación y procesadores*, Editorial Centro de Estudios Ramon Areces SA, 2016.
- [25] L. DE PROGRAMACIÓN, *Lenguajes de programación*, ESCUELA SUPERIOR POLITECNICA DEL LITORAL.: Tesis Doctoral, 2009.
- [26] J. J. GUTIÉRREZ, «¿ Qué es un framework web,» Mayo 2014. [En línea]. Available: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.. [Último acceso: 19 Mayo 2021].
- [27] S. OLLIVIER y G. U. R. Y. PIERRE-ALEXANDRE, *AngularJS: Desarrolle hoy las aplicaciones web de mañana*, Ediciones ENI, 2016.

- [28] GitBooks, «Desarrollo de aplicaciones web.,» GitBooks, [En línea]. Available: https://juanda.gitbooks.io/webapps/content/spa/arquitectura_de_un_spa.html?q=. [Último acceso: 19 Mayo 2021].
- [29] Y. D. GONZÁLEZ y Y. F. ROMERO, «Patrón Modelo-Vista-Controlador,» *Telemática*, vol. 11, n° 1, pp. 47-57, 2012.
- [30] L. PANTOJA y C. PARDO, «Evaluando la Facilidad de Aprendizaje de Frameworks mvc en el Desarrollo de Aplicaciones Web,» *Publicaciones e investigación*, vol. 10, pp. 129-142, 2016.
- [31] E. BASCÓN PANTOJA, «Ernesto. El patrón de diseño Modelo-Vista-Controlador (MVC) y su implementación en Java Swing,» *Acta Nova*, vol. 2, n° 4, pp. 493-507., 2004.
- [32] J. E. PÉREZ, *introduccion a JavaScript*, 2019.
- [33] T. NAVARRETE, *El lenguaje JavaScript*, Argentina, 2006.
- [34] M. Parada, «MERN Stack: Qué es y qué ventajas ofrece,» OpenWebinars, 21 Octubre 2020. [En línea]. Available: <https://openwebinars.net/blog/mern-stack-que-es-y-que-ventajas-ofrece/>. [Último acceso: 19 Mayo 2020].
- [35] J. JIMENEZ, *Descubre React.*, Leanpub, 2016.
- [36] F. Inc, «React,» React, [En línea]. Available: <https://es.reactjs.org/docs/introducing-jsx.html>. [Último acceso: 19 Mayo 2021].
- [37] I. StrongLoop, «Express,» Express, [En línea]. Available: <https://expressjs.com/es/>. [Último acceso: 19 Mayo 2021].
- [38] Mozilla, «Introducción a Express/Node,» MDN Web Docs, [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction. [Último acceso: 19 Mayo 2021].
- [39] O. Foundation, «Acerca de Node.js®,» nodejs, [En línea]. Available: <https://nodejs.org/es/about/>. [Último acceso: 19 Mayo 2021].
- [40] K. e. a. CALVO, «MongoDB: alternativas de implementar y consultar documentos.,» de *IX Congreso Internacional de Computación y Telecomunicaciones.*, Lima, 2017.
- [41] H. G. DEL BUSTO y O. Y. ENRÍQUEZ, «Bases de datos NoSQL,» *Telemática*, vol. 11, n° 3, pp. 21-33, 2013.

- [42] Strapi, «Strapi,» Strapi, [En línea]. Available: <https://strapi.io/>. [Último acceso: 19 Mayo 2021].
- [43] Stripe, «Stripe,» Stripe, [En línea]. Available: <https://stripe.com/es-us/about>. [Último acceso: 19 Mayo 2021].

10.ANEXOS

ANEXO A: Hojas de vida de los investigadores

10.1. HOJA DE VIDA DEL TUTOR



Datos Informativos personal Docente

Datos Personales

APELLIDOS: Llano Casa

NOMBRES: Alex Christian

ESTADO CIVIL: Casado(a)

CÉDULA DE CIUDADANÍA: 0502589864

NÚMERO DE CARGAS FAMILIARES: 3

LUGAR Y FECHA DE NACIMIENTO: Latacunga, 09-11-1986

DIRECCIÓN DOMICILIARIA: Parroquia Tamicuchi, Barrio Cajón Veracruz, Calle Batalla de Tarqui

CELULAR: 0999969302

TELÉFONO:

EMAIL INSTITUCIONAL: alex.llano9864@utc.edu.ec



ESTUDIOS REALIZADOS Y TÍTULOS OBTENIDOS:

NIVEL	TÍTULO OBTENIDO	FECHA DE REGISTRO	CÓDIGO DEL REGISTRO CONESUP O SENESCYT
TERCER	INGENIERO EN INFORMATICA Y SISTEMAS COMPUTACIONALES	2010-07-06	1020-10-1002344
CUARTO	MASTER UNIVERSITARIO EN INGENIERIA DE SOFTWARE Y SISTEMAS INFORMATICOS	2018-08-13	7241128506

HISTORIAL PROFESIONAL:

FACULTAD EN LA QUE LABORA: Ciencias de la Ingeniería y Aplicadas

ÁREA DEL CONOCIMIENTO EN LA CUAL SE DESEMPEÑA: Ciencias, Subárea: Informática

FECHA DE INGRESO A LA UTC: 29/11/2018

FIRMA

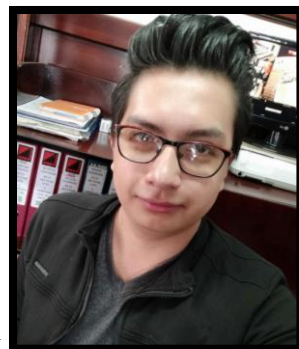
10.2. HOJA DE VIDA DE LOS INVESTIGADORES

10.2.1. Hoja de vida investigador 1

HOJA DE VIDA

DATOS PERSONALES

NOMBRES:	CARLOS XAVIER
APELLIDOS:	GALEAS GUANOCHANGA
CÉDULA DE CIUDADANÍA:	172623411-3
LUGAR DE NACIMIENTO:	PICHINCHA-MACHACHI
FECHA DE NACIMIENTO:	20/MAYO/1997
EDAD:	24 AÑOS
NACIONALIDAD:	ECUATORIANO
ESTADO CIVIL:	SOLTERO
TELÉFONO:	0996117253
E-MAIL:	carlos.galeas4113@utc.edu.ec.com
DOMICILIO:	PICHINCHA-MACHACHI

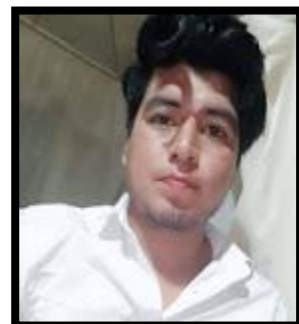


ESTUDIOS REALIZADOS

PRIMARIA	ESCUELA FISCAL “JOSÉ MEJÍA LEQUERICA” (PICHINCHA-MACHACHI)
SECUNDARIA	UNIDAD EDUCATIVA “MACHACHI” (PICHINCHA-MACHACHI)

10.2.2. Hoja de vida investigador 2

HOJA DE VIDA



DATOS PERSONALES

NOMBRES: BYRON LUIS
APELLIDOS: NOTENO ANDI
CÉDULA DE CIUDADANÍA: 220008256-4
LUGAR DE NACIMIENTO: ORELLANA-COCA
FECHA DE NACIMIENTO: 30/MAYO/1992
EDAD: 29 AÑOS
NACIONALIDAD: ECUATORIANO
ESTADO CIVIL: SOLTERO
TELÉFONO: 0968530292
E-MAIL: byron.noteno4@gmail.com
DOMICILIO: ORELLANA-COCA

ESTUDIOS REALIZADOS

PRIMARIA ESCUELA FISCAL MIXTA
"PRESIDENTE TAMAYO" (COCA-
ORELLANA)

SECUNDARIA UNIDAD EDUCATIVA "SEGUNDO
TORRES" (LATACUNGA-COTOPAXI)

10.3. FORMULARIO DE ENCUESTA

ANEXO B: Formularios de Encuesta y Entrevista

UNIVERSIDAD TÉCNICA DE COTOPAXI
DIRECCIÓN DE VINCULACIÓN CON LA SOCIEDAD
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA INFORMÁTICA Y SISTEMAS COMPUTACIONALES

OBJETIVO: Recaudar información confiable para conocer la opinión de los ciudadanos sobre la empresa “Señor De Maca” y la importancia que pueda tener el desarrollo de la aplicación web y móvil para la comercialización de los productos en línea.

INDICACIONES: Lea detenidamente cada uno de los ítems y responda con completa sinceridad, puede elegir las opciones que usted crea conveniente. Recuerde que la encuesta es anónima y confidencial.

ENCUESTA

1. **¿Conoce Ud. la empresa “Señor De Maca”?**

- SI
- NO

2. **¿Cómo adquiere sus productos?**

- Personalmente.
- Encomienda.
- En línea

3. **¿Le gustaría facilitar sus compras?**

- SI
- NO

4. **¿Con que frecuencia Ud. utiliza páginas web o aplicaciones móviles en su vida diaria?**

- Siempre
- Casi siempre

- A veces
- Casi nunca
- Nunca

5. ¿Usted cree que la tecnología facilita las actividades de las personas?

- SI
- NO
- TAL VEZ

6. ¿Conoce algún sistema de compras?

- SI
- NO

7. ¿Le gustaría realizar sus comprar en línea?

- SI
- NO
- TAL VEZ

8. ¿Qué es lo que busca al momento de realizar compras en línea?

- Buen servicio
- Facilidad
- Eficacia
- Confiabilidad
- Seguridad
- Comodidad
- Ahorro de tiempo

9. ¿Cree usted que un sistema de compras online ayuda a los ciudadanos?

- SI
- NO

10. ¿Utilizaría la aplicación web y móvil de la empresa “Señor De Maca” para adquirir sus productos?

- SI
- NO
- TAL VEZ
-

10.4. FORMULARIO DE ENTREVISTA

**UNIVERSIDAD TÉCNICA DE COTOPAXI
DIRECCIÓN DE VINCULACIÓN CON LA SOCIEDAD
FACULTAD DE CIENCIAS DE LA INGENIERÍA Y APLICADAS
CARRERA INFORMÁTICA Y SISTEMAS COMPUTACIONALES**

19 de Mayo del 2021

ENTREVISTA

1) ¿A qué se dedica su empresa?

A la comercialización de productos.

2) ¿La empresa tiene página web o aplicación móvil?

No.

3) ¿Tiene alguna estrategia para poder llegar a los clientes?

No, por el momento solo contamos con la publicidad que se ha colocado en el mismo lugar de la empresa.

4) ¿Usted cree que la tecnología es muy importante para su empresa?

Creo que si es importante porque en la actualidad a la tecnología la encontramos en muchos campos y está siendo de gran ayuda para todas las personas, mejorando en muchos de los aspectos la calidad de vida. Además, que hay que tener en cuenta que la tecnología en los últimos años ha tenido un gran avance y cada día sigue en progreso.

5) ¿Usted cree que las redes sociales pueden ayudar a dar publicidad a su empresa?

Sí, porque hoy en día las redes sociales forman parte de las personas en su día a día, ya es muy extraño que una persona no entre a sus redes sociales así sea una vez al día, además de que es la forma más rápida de llegar a los clientes.

6) ¿Qué necesidades tecnológicas tiene para su empresa?

Al ser una empresa sin mucha antigüedad no ha sido posible aun estar involucrado en la tecnología así que la empresa por el momento lo que necesita es empezar a actualizarse y adentrarse en el mundo de la tecnología. Empezando por un sistema que permita que los productos que se ofertan lleguen a la mayor cantidad de clientes y así que la empresa sea muy conocida.

7) ¿Usted cree que el Sistema beneficiaría al desarrollo de su empresa?

Creo que es de gran ayuda para que los clientes puedan escoger adquirir sus productos en esta empresa por las facilidades que el sistema va a brindar.

8) ¿Con qué sistemas operativos trabaja la empresa?

Por el momento la empresa cuenta con un computador que trabaja con Windows el cual es utilizado para diferentes actividades.

9) ¿Qué necesita que haga el sistema a desarrollarse?

Por el momento la prioridad es poder comercializar los productos y también brindar a los clientes la información necesaria tanto de la empresa como de los productos que están disponibles para la venta.

10) ¿Qué funcionalidades le gustaría que estén en el sistema?

Como ya le mencionaba lo más importante en este momento es la comercialización de nuestros productos y por debería existir un catálogo donde el cliente tenga toda la información de nuestros productos, pueda solicitarlos y comprarlos. Además, también sería de gran utilidad tener un control tanto de las ventas como de los clientes.

10.5. Estimación de costos

ANEXO C: Estimación de costos de las aplicaciones

10.5.1. Gastos Directos

Tabla 10.5.1: Gastos Directos.

DETALLE	CANTIDAD	VALOR U.	VALOR TOTAL
Impresora	1	300	300
Software Web	1	100	100
Software Móvil	1	50	50
TOTAL			450

10.5.2. Gastos Indirectos

Tabla 10.5.2: Gastos Indirectos

DETALLE	CANTIDAD	VALOR U.	VALOR TOTAL
Luz	2	30	60.00
Internet	1	50	50.00
Transporte	2	0.50	1.00
Comida	2	2	4.00
TOTAL			115

10.5.3. Gastos Total

Tabla 10.5.3: Gastos Total

DETALLE	CANTIDAD	VALOR U.	VALOR TOTAL
Gastos Directos	1	450	450
Gastos Indirectos	1	115	115

TOTAL	565
--------------	------------

10.5.4. Gastos Imprevistos

Donde se sacará el 20% de los gastos totales correspondiente a los gastos directos e indirectos para gastos imprevistos de algún tipo para el proyecto.

$$g_i = \frac{20(565)}{100}$$

$$g_i = 113$$

Tabla 10.5.4: Gastos Imprevistos

DETALLE	CANTIDAD	TOTAL
Gastos Imprevistos	20%	113
TOTAL		113

10.5.5. Gastos Totales

Tabla 10.5.5: Gastos Totales

DETALLE	VALOR TOTAL
Gastos Directos	450
Gastos Indirectos	115
Gastos Imprevistos	113
TOTAL	678

10.6. MÉTRICAS DE ESTIMACIÓN DE COSTOS

10.6.1. Puntos de función

Para la estimación de costos del proyecto se realizará mediante la norma IFPUB.

A continuación, trataremos temas claves como: Puntos de función sin ajustar, Puntos de función ajustados, Factores de ajuste, con el fin de poder establecer la complejidad del proyecto, un tiempo estimado y el costo final del mismo.

En la tabla se define cada una de las funciones según su tipo y complejidad obtenida en el IFPUG, el cual permitirá tener un valor adecuado al número de funcionalidades de los aplicativos.

Tabla 10.6.1: Puntos de función

Tipo/Complejidad	Bajo	Media	Alta
Entrada Externa (EI)	3pf	4pf	6pf
Salida Externa (EO)	4pf	5pf	7pf
Consulta Externa(EQ)	3pf	4pf	5pf
Archivo Lógico Interno (ILF)	7pf	10pf	15pf
Archivo de Interfaz Externo (EIF)	5pf	7pf	10pf

Interacción con el Usuario

- EI: Actualizar, Insertar, Eliminar.
- EO: Listar, Informe de un Reporte, Mensaje.
- EQ: Buscar.
- ILF: Número de tablas de una base de datos.
- EIF: Consumir Datos-Sistema Externo.

10.6.2. Puntos de función sin ajustar (PFSA)

Se presenta cada una de las funciones de los aplicativos, las mismas que son evaluadas con el nivel de complejidad media.

Requisitos Específicos

Tabla 10.6.2: Puntos de función sin ajustar (Requisitos Específicos)

#	Descripción	Complejidad	Valor
RN01	El usuario podrá agregar productos a un carrito de compra.	EI	4pf
RN02	El usuario podrá seleccionar cuantos productos quiere	EI	4pf

	comprar dependiendo de la cantidad que exista en stock.		
RN03	El usuario podrá crear, modificar, eliminar direcciones para la entrega de productos.	EI	4pf
RN04	El usuario podrá crearse una cuenta y modificar los datos de la misma.	EI	4pf
RN05	El usuario podrá elegir si quiere ir al local a retirar los productos o entrega a domicilio.	EO	5pf
RN06	El usuario podrá pagar los productos seleccionados a través de una tarjeta de crédito.	EO	5pf
RN07	El usuario podrá visualizar los productos que ha comprado en una sección de pedidos.	EO	5pf
RN08	El usuario podrá contactarse con la empresa a través de un formulario de mensajes.	EO	5pf
RN09	El administrador puede agregar, modificar, eliminar productos en la aplicación.	EI	4pf
RN10	El administrador puede agregar, modificar, eliminar	EI	4pf

	categorías relacionadas a los productos.		
RN11	El administrador puede agregar, modificar, eliminar marcas de productos.	EI	4pf
RN12	El administrador puede agregar, modificar la información general que tiene la página sobre la empresa.	EI	4pf
RN13	El administrador puede modificar el estado de entrega de productos comprados por el usuario.	EI	4pf
RN14	El administrador podrá agregar, modificar, eliminar usuarios existentes en la aplicación.	EI	4pf
RN15	El administrador podrá agregar, modificar nuevos administradores para la gestión de la aplicación.	EI	4pf
RN16	El sistema cuenta con una base de datos de 5 tablas.	ILF	35pf
TOTAL			99pf

Se presenta el número de funcionalidades por cada tipo, en donde se calculó el total de puntos de función sin ajustar (PFSA) dado en resultado a continuación.

Tabla 10.6.3: Total de puntos de función sin ajustar

Tipo / Complejidad	Nº Funcionalidades	Media / Alta	Total
Entrada Externa (EI)	11	4pf	44
Salida Externa (EO)	4	5pf	20
Consulta Externa(EQ)	0	4pf	0
Archivo Lógico Interno (ILF)	5	7pf	35
TOTAL PFSA			99

10.6.3. Factores de ajuste

En el presente apartado procedemos a describir el cálculo de los factores de ajuste mediante las métricas de la norma IFPUG, con la finalidad de emplear el resultado en la determinación de los puntos de función ajustados.

Tabla 10.6.4: Factores de ajuste

#	Factores de Ajuste	Puntuación
1	Comunicación de Datos	2
2	Funciones Distribuidos	2
3	Prestaciones	3
4	Facilidad de Configuración	3
5	Tasa de transición	2
6	Entrada de Datos en Línea	4
7	Diseño para la eficacia del usuario final	3
8	Actualización de datos en línea	3
9	Complejidad del proceso lógico interno de la aplicación	3

10	Reusabilidad del código para otras aplicaciones	3
11	Facilidad de Instalación	4
12	Facilidad de operación	3
13	Localización múltiple	2
14	Facilidad de Cambios	3
TOTAL		40

10.6.4. Puntos de función ajustados

En el presente apartado trata acerca del cálculo de los Puntos de función ajustados (PFA), para dicho cálculo se utilizará los PFSA y los FA datos que se determinaron en los apartados anteriores, la formula aplicada será la siguiente:

$$\text{PFA} = \text{PFSA} * [0,65 + (0,01 * \text{FA})]$$

DATOS

PFSA: 99

FA: 40

Remplazamos los datos en la formula

$$\text{PFA} = 99 * [0,65 + (0,01 * 40)]$$

$$\text{PFA} = 103.95$$

El valor de los puntos de función ajustados es: 103.95

10.6.5. Estimación de esfuerzo requerido (tiempo)

En el siguiente apartado trata acerca del procedimiento a realizar para el cálculo del tiempo estimado para el desarrollo del proyecto.

Tabla 10.6.5: Estimación de esfuerzo requerido

Lenguaje	HORAS PF PROMEDIO	LINEA DE CODIGO POR PF
Lenguaje de 3ra generación	15	100
Lenguaje de 4ta generación (PHP)	8	20

Cálculo de Hora/Hombre

$$H/h = PFA * \text{hora promedio}$$

$$h/h = 103.95 * 8$$

$$\mathbf{h/h = 831.6}$$

Cálculo días /hombre

$$\text{Horas productivas} = 5$$

$$1 \text{ mes} = 20 \text{ días}$$

$$D/h = (h/h) / \text{horas productivas}$$

$$d/h = 415.8 / 5$$

$$\mathbf{d/h = 83.16}$$

Cálculo mes /hombre

$$m/h = (d/h) / \text{mes}$$

$$m/h = 207.9 / 20$$

$$\mathbf{m/h = 4.158}$$

Cálculo para dos programadores

$$h/p = 831.6 / 2 \text{ (programadores)}$$

$$\mathbf{h/p = 415.8}$$

Interpretación de resultados

$$m/h = 4.158$$

$$\text{Meses} = 4 \text{ MESES}$$

$$\text{días} = 20 * 0.4$$

$$\mathbf{\text{días} = 8}$$

Por el presente proyecto tendría un lapso de 4 meses con 8 días y 8 horas con dos programadores

$$\text{Costo} = (\# \text{programadores} * \text{meses duración} * \text{suelo}) + \text{otros valores}$$

$$\text{Costo} = (2 * 4.158 * 400) + 590$$

$$\mathbf{\text{Costo} = 3.916,4}$$

Interpretación

El presupuesto para el desarrollo del presente proyecto se trabajará por la duración de 4 meses y 8 días, y 8 horas productivas se estima un costo de 3.916.4 dólares americanos.

10.7. MODELO DE BASE DE DATOS

ANEXO D: Diagramas de las aplicaciones

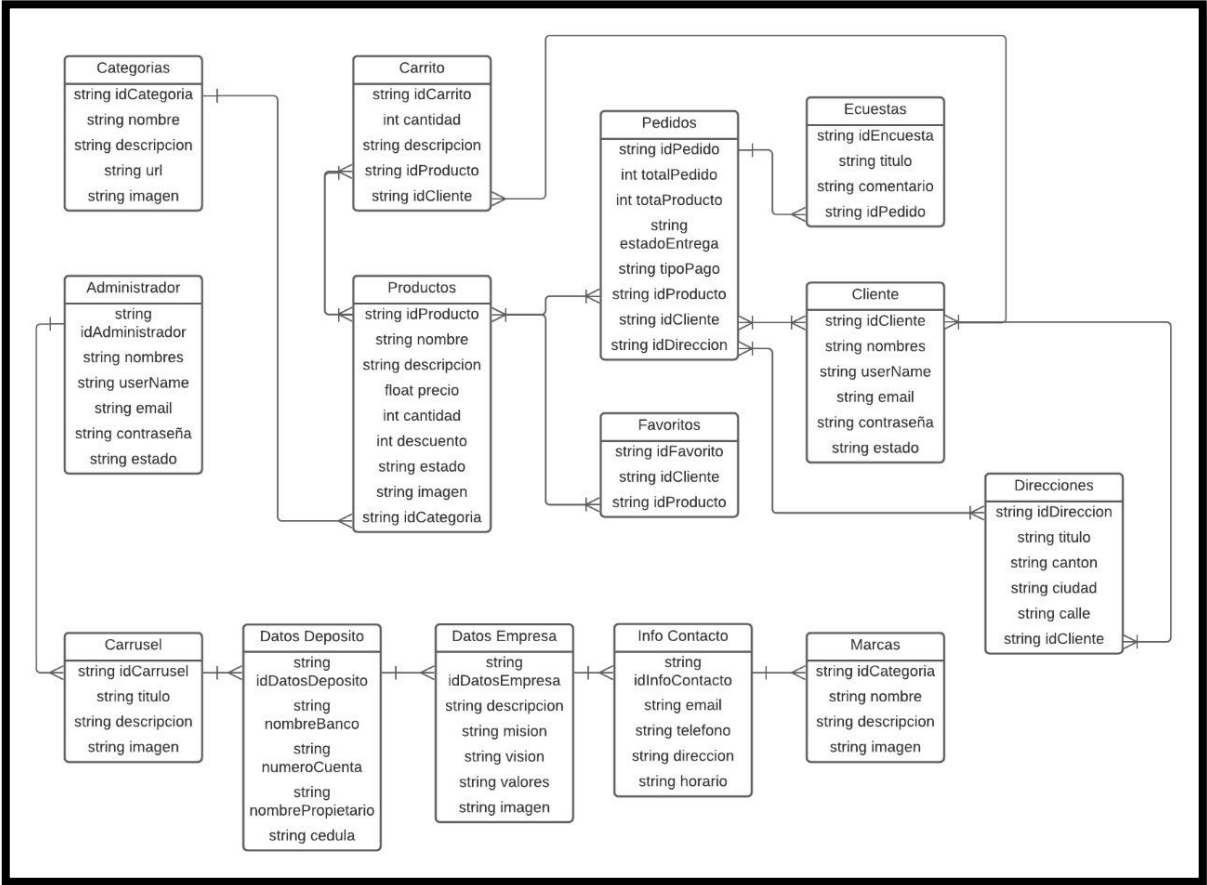


Figura 10.1: Diagrama de Bases de Datos

10.8. DIAGRAMA DE CLASES DEL SISTEMA

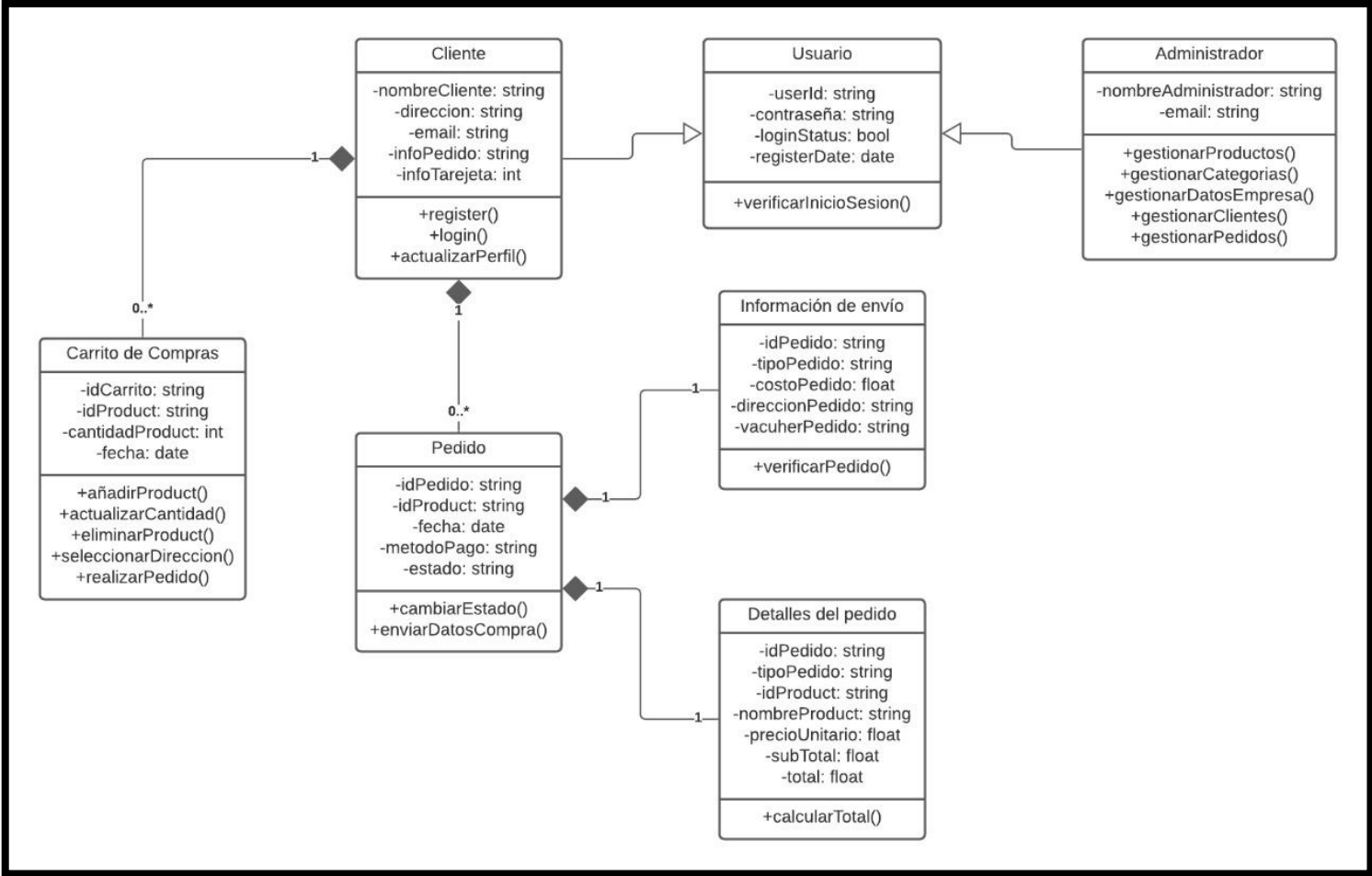


Figura 10.2: Diagrama de Clases

10.9. DIAGRAMA DE COMPONENTES

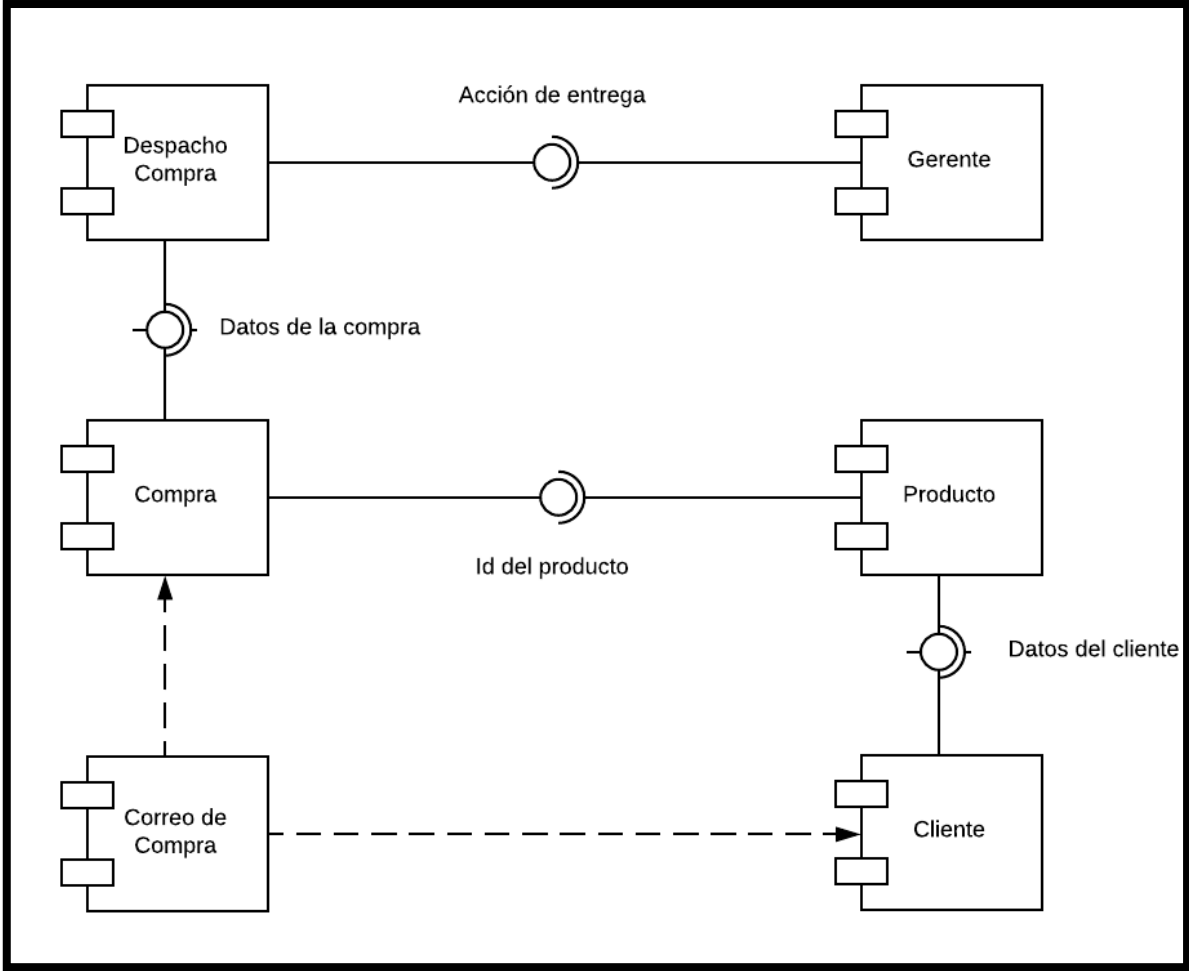


Figura 10.3: Diagrama de componentes de la aplicación

10.10. CASOS DE USO

ANEXO E: Casos de uso de la aplicación

- Diagrama general de casos de uso administrador.

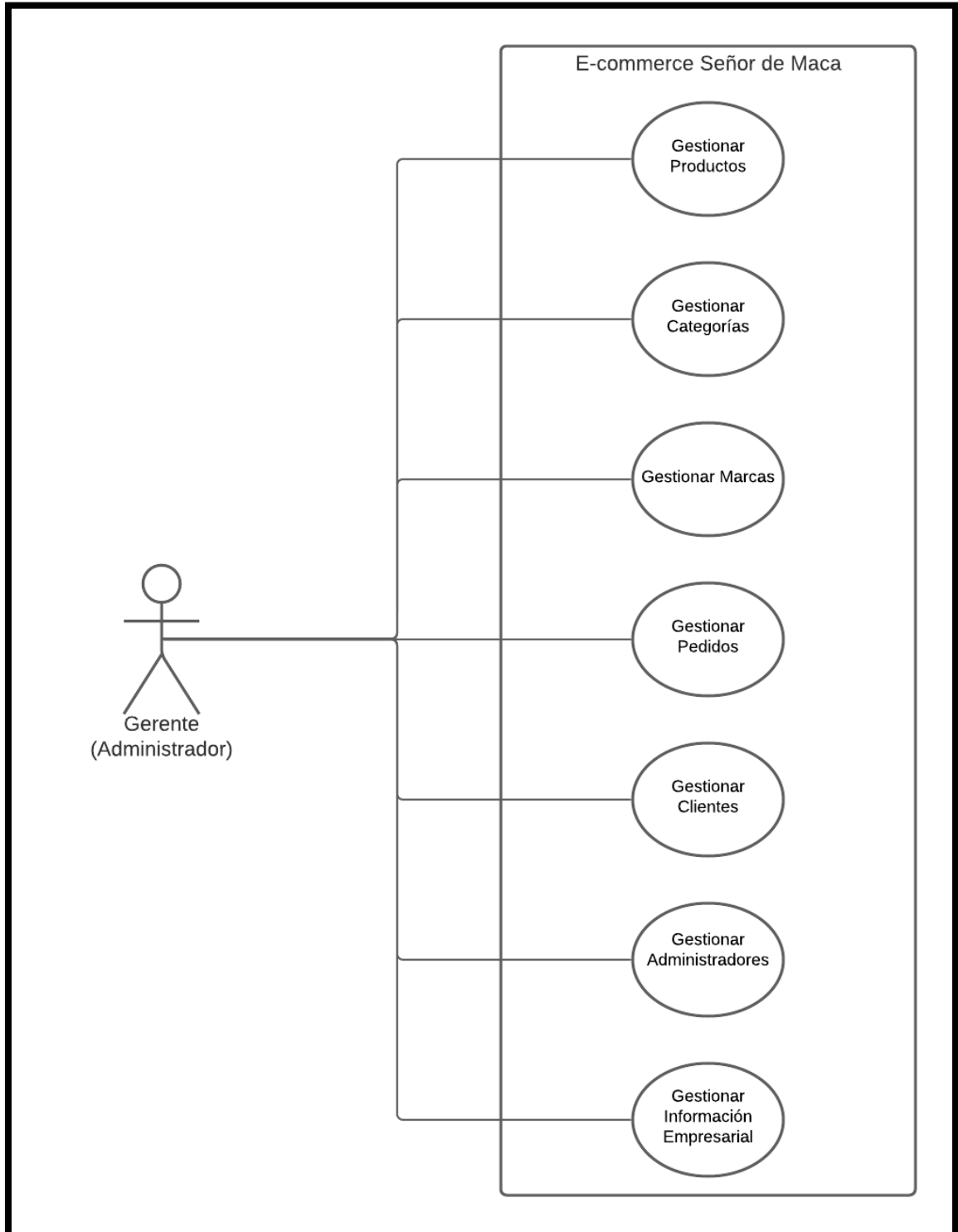


Figura 10.4: Diagrama general de casos de uso administrador

- Diagrama general de casos de uso Cliente.

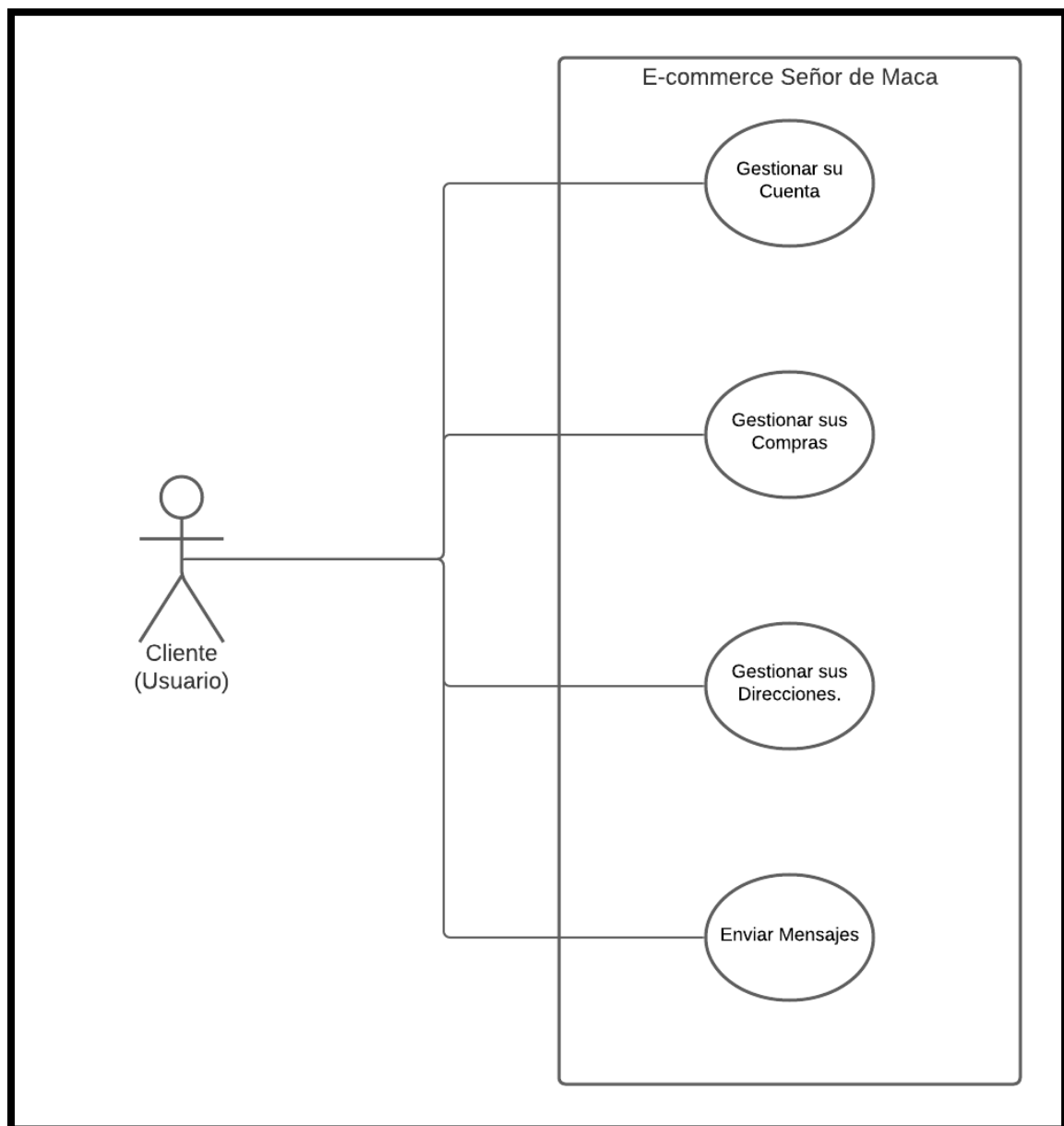


Figura 10.5: Diagrama general de casos de uso Cliente

10.11. Casos de uso A detalle Administrador

10.11.1. Gestionar productos

Extends= Agregar productos, modificar productos, eliminar productos.

Tabla 10.10.1: Caso de uso a detalle (Gestionar Productos)

N ^a CU:	CU 001
Nombre:	Gestionar productos.
Descripción	El gerente podrá realizar una gestión de los productos existentes en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.
Flujo principal:	<ol style="list-style-type: none">1. El gerente visualiza la interfaz de inicio.2. El gerente visualiza el menú de opciones.3. El gerente selecciona la opción de “Productos.”4. El sistema despliega la interfaz de productos.5. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar los productos existentes.

10.11.2. Agregar productos

Tabla 10.10.2: Caso de uso a detalle (Agregar Productos)

N ^a CU:	CU 001.1
Nombre:	Agregar Productos.
Descripción	El gerente podrá agregar un nuevo producto a la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de productos.
Flujo principal:	<ol style="list-style-type: none">1. El gerente visualiza la interfaz de productos.2. El gerente presiona el botón “Agregar Productos” de la interfaz.3. El sistema despliega la interfaz de para agregar productos.

	<ol style="list-style-type: none"> 4. El sistema muestra el formulario para agregar un nuevo producto. 5. El gerente visualiza el formulario. 6. El gerente llena los campos del formulario presentado. 7. El gerente presiona el botón “Guardar” del formulario. 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema da como resultado que los datos son correctos. 10. El sistema guarda los datos del nuevo producto en la base de datos. 11. El sistema manda el mensaje “Producto agregado”. 12. El sistema muestra la interfaz de datos del producto.
Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 10. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 11. El sistema no agrega los datos a la Base de Datos. 12. El sistema mostrar el formulario con los datos que agrego el gerente. 13. El sistema regresa al paso 5 del flujo principal.
Post condición:	El nuevo producto será agregado a la base de datos.

10.11.3. Modificar productos.

Tabla 10.10.3: Caso de uso a detalle (Modificar Productos)

Nª CU:	CU 001.2
Nombre:	Modificar Productos.
Descripción	El gerente podrá modificar un producto existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de productos.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de productos. 2. El sistema busca si existen datos de productos agregados en la base de datos. 3. El sistema encuentra los productos en la base de datos. 4. El sistema muestra en una tabla los productos existentes de la base de datos.

	<ol style="list-style-type: none"> 5. El gerente visualiza la lista de productos existentes de la tabla. 6. El gerente selecciona el producto que desea modificar. 7. El sistema toma los datos del producto seleccionado. 8. El sistema despliega la interfaz para modificar un producto. 9. El sistema muestra el formulario de modificación de productos con los datos que obtuvo de la selección del gerente. 10. El gerente visualiza el formulario. 11. El gerente modifica los datos que desee del producto seleccionado. 13. El gerente presiona el botón “Guardar” del formulario. 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema da como resultado que los datos son correctos. 16. El sistema modifica los nuevos datos en el producto seleccionado de la base de datos. 17. El sistema manda el mensaje “Producto modificado”. 18. El sistema muestra la interfaz de datos del producto.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de productos agregados en la base de datos. 3. El sistema no encuentra productos existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay productos.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 15. El sistema realiza la validación de los datos que manda el gerente. 16. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 17. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 18. El sistema no agrega los datos a la Base de Datos. 19. El sistema mostrar el formulario con los datos que agregó el gerente. 20. El sistema regresa al paso 5 del flujo principal.
Post condición:	Los nuevos datos ingresados serán modificados en el producto de la base de datos.

10.11.4. Eliminar productos.

Tabla 10.10.4: Caso de uso a detalle (Eliminar Productos)

Nª CU:	CU 001.3
Nombre:	Eliminar Productos.

Descripción	El gerente podrá eliminar un producto existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de productos.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de productos. 2. El sistema busca si existen datos de productos agregados en la base de datos. 3. El sistema encuentra los productos en la base de datos. 4. El sistema muestra en una tabla los productos existentes de la base de datos. 5. El gerente visualiza la lista de productos existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada producto en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID del producto seleccionado. 11. El sistema busca el producto en la base de datos por su ID. 12. El sistema elimina el producto encontrado. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con los productos existentes.
Flujo alternativo 1:	<ol style="list-style-type: none"> 7. El sistema busca si existen datos de productos agregados en la base de datos. 8. El sistema no encuentra productos existentes en la base de datos. 9. El sistema muestra el mensaje en la tabla “No hay productos.” 10. El gerente visualiza el mensaje. 11. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 9. El gerente presiona el botón “No, cancelar” de la ventana. 10. El sistema cancela la captura de ID. 11. El sistema cierra la ventana modal. 12. El sistema regresa al paso 1 del flujo principal.
Post condición:	El producto seleccionado será eliminado de la base de datos.

10.11.5. Gestionar categorías.

Extends= Agregar categorías, modificar categorías, eliminar categorías.

Tabla 10.10.5: Caso de uso a detalle (Gestionar Categorías)

N ^a CU: CU 002	
Nombre:	Gestionar categorías.
Descripción	El gerente podrá realizar una gestión de las categorías existentes en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de inicio. 2. El gerente visualiza el menú de opciones. 3. El gerente selecciona la opción de “Categorías.” 4. El sistema despliega la interfaz de categorías. 5. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar las categorías existentes.

10.11.6. Agregar categorías.

Tabla 10.10.6: Caso de uso a detalle (Agregar Categorías)

N ^a CU: CU 002.1	
Nombre:	Agregar Categorías.
Descripción	El gerente podrá agregar una nueva categoría a la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de categorías.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Categorías. 2. El gerente presiona el botón “Agregar Categorías” de la interfaz. 3. El sistema despliega la interfaz de para agregar categorías. 4. El sistema muestra el formulario para agregar una nueva categoría. 5. El gerente visualiza el formulario. 6. El gerente llena los campos del formulario presentado. 7. El gerente presiona el botón “Guardar” del formulario. 8. El sistema realiza la validación de los datos que manda el gerente.

	<ol style="list-style-type: none"> 9. El sistema da como resultado que los datos son correctos. 10. El sistema guarda los datos de la nueva categoría en la base de datos. 11. El sistema manda el mensaje “Categoría agregada”. 12. El sistema muestra la interfaz de datos de la categoría.
Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 10. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 11. El sistema no agrega los datos a la Base de Datos. 12. El sistema mostrar el formulario con los datos que agrego el gerente. 13. El sistema regresa al paso 5 del flujo principal.
Post condición:	La nueva categoría será agregada a la base de datos.

10.11.7. Modificar categorías.

Tabla 10.10.7: Caso de uso a detalle (Modificar Categorías)

Nª CU:	CU 002.2
Nombre:	Modificar Categorías.
Descripción	El gerente podrá modificar una categoría existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de categorías.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de categorías. 2. El sistema busca si existen datos de categorías agregados en la base de datos. 3. El sistema encuentra las categorías en la base de datos. 4. El sistema muestra en una tabla las categorías existentes de la base de datos. 5. El gerente visualiza la lista de categorías existentes de la tabla. 6. El gerente selecciona la categoría que desea modificar. 7. El sistema toma los datos de la categoría seleccionada. 8. El sistema despliega la interfaz para modificar una categoría.

	<ol style="list-style-type: none"> 9. El sistema muestra el formulario de modificación de categorías con los datos que obtuvo de la selección del gerente. 10. El gerente visualiza el formulario. 11. El gerente modifica los datos que desee de la categoría seleccionada. 13. El gerente presiona el botón “Guardar” del formulario. 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema da como resultado que los datos son correctos. 16. El sistema modifica los nuevos datos en la categoría seleccionada de la base de datos. 17. El sistema manda el mensaje “Categoría modificado”. 18. El sistema muestra la interfaz de datos de la categoría.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de categorías agregados en la base de datos. 3. El sistema no encuentra categorías existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay categorías.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 16. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 17. El sistema no agrega los datos a la Base de Datos. 18. El sistema mostrar el formulario con los datos que agrego el gerente. 19. El sistema regresa al paso 5 del flujo principal.
Post condición:	Los nuevos datos ingresados serán modificados en la categoría de la base de datos.

10.11.8. Eliminar categorías.

Tabla 10.10.8: Caso de uso a detalle (Eliminar Categorías)

Nº CU:	CU 002.3
Nombre:	Eliminar Categorías.
Descripción	El gerente podrá eliminar una categoría existente en la aplicación.
Usuario:	Administrador.

Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de categorías.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de categorías. 2. El sistema busca si existen datos de categorías agregados en la base de datos. 3. El sistema encuentra las categorías en la base de datos. 4. El sistema muestra en una tabla las categorías existentes de la base de datos. 5. El gerente visualiza la lista de categorías existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada categoría en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID de la categoría seleccionada. 11. El sistema busca la categoría en la base de datos por su ID. 12. El sistema elimina la categoría encontrada. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con las categorías existentes.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de categorías agregados en la base de datos. 3. El sistema no encuentra categorías existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay categorías.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 9. El gerente presiona el botón “No, cancelar” de la ventana. 10. El sistema cancela la captura de ID. 11. El sistema cierra la ventana modal. 12. El sistema regresa al paso 1 del flujo principal.
Post condición:	La categoría seleccionada será eliminada de la base de datos.

10.11.9. Gestionar marcas

Extends= Agregar marcas, modificar marcas, eliminar marcas.

Tabla 10.10.9: Caso de uso a detalle (Gestionar Marcas)

N^{ra} CU:

CU 003

Nombre:	Gestionar marcas.
Descripción	El gerente podrá realizar una gestión de las marcas existentes en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.
Flujo principal:	<ol style="list-style-type: none"> 6. El gerente visualiza la interfaz de inicio. 7. El gerente visualiza el menú de opciones. 8. El gerente selecciona la opción de “Marcas.” 9. El sistema despliega la interfaz de marcas. 10. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar las marcas existentes.

10.11.10. Agregar Marcas.

Tabla 10.10.10: Caso de uso a detalle (Agregar Marcas)

N^{ra} CU:	CU 003.1
Nombre:	Agregar Marcas.
Descripción	El gerente podrá agregar una nueva marca a la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de marcas.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de marcas. 2. El gerente presiona el botón “Agregar Marca” de la interfaz. 3. El sistema despliega la interfaz de para agregar marcas. 4. El sistema muestra el formulario para agregar una nueva marca. 5. El gerente visualiza el formulario. 6. El gerente llena los campos del formulario presentado. 7. El gerente presiona el botón “Guardar” del formulario. 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema da como resultado que los datos son correctos. 10. El sistema guarda los datos de la nueva marca en la base de datos. 11. El sistema manda el mensaje “Marca agregada”. 12. El sistema muestra la interfaz de datos de la marca.

Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 10. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 11. El sistema no agrega los datos a la Base de Datos. 12. El sistema mostrar el formulario con los datos que agrego el gerente. 13. El sistema regresa al paso 5 del flujo principal.
Post condición:	La nueva marca será agregada a la base de datos.

10.11.11. Modificar Marcas.

Tabla 10.10.11: Caso de uso a detalle (Modificar Marcas)

Nª CU:	CU 003.2
Nombre:	Modificar Marcas.
Descripción	El gerente podrá modificar una marca existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de marcas.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de marcas. 2. El sistema busca si existen datos de marcas agregados en la base de datos. 3. El sistema encuentra las marcas en la base de datos. 4. El sistema muestra en una tabla las marcas existentes de la base de datos. 5. El gerente visualiza la lista de marcas existentes de la tabla. 6. El gerente selecciona la marca que desea modificar. 7. El sistema toma los datos de la marca seleccionada. 8. El sistema despliega la interfaz para modificar una marca. 9. El sistema muestra el formulario de modificación de marcas con los datos que obtuvo de la selección del gerente. 10. El gerente visualiza el formulario. 11. El gerente modifica los datos que desee de la marca seleccionada. 12. El gerente presiona el botón “Guardar” del formulario. 13. El sistema realiza la validación de los datos que manda el gerente.

	<p>14. El sistema da como resultado que los datos son correctos.</p> <p>15. El sistema modifica los nuevos datos en la marca seleccionada de la base de datos.</p> <p>16. El sistema manda el mensaje “Marca modificado”.</p> <p>17. El sistema muestra la interfaz de datos de la marca.</p>
Flujo alternativo 1:	<p>2. El sistema busca si existen datos de marcas agregados en la base de datos.</p> <p>3. El sistema no encuentra marcas existentes en la base de datos.</p> <p>4. El sistema muestra el mensaje en la tabla “No hay marcas.”</p> <p>5. El gerente visualiza el mensaje.</p> <p>6. El gerente regresa el paso 1 del flujo principal.</p>
Flujo alternativo 2:	<p>13. El sistema realiza la validación de los datos que manda el gerente.</p> <p>14. El sistema encuentra que falta de ingresar datos en algún campo del formulario.</p> <p>15. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”.</p> <p>16. El sistema no agrega los datos a la Base de Datos.</p> <p>17. El sistema mostrar el formulario con los datos que agrego el gerente.</p> <p>18. El sistema regresa al paso 5 del flujo principal.</p>
Post condición:	Los nuevos datos ingresados serán modificados en la marca de la base de datos.

10.11.12. Eliminar Marcas.

Tabla 10.10.12: Caso de uso a detalle (Eliminar Marcas)

Nª CU:	CU 003.3
Nombre:	Eliminar Marcas.
Descripción	El gerente podrá eliminar una marca existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de marcas.
Flujo principal:	<p>1. El gerente visualiza la interfaz de marcas.</p> <p>2. El sistema busca si existen datos de marcas agregados en la base de datos.</p> <p>3. El sistema encuentra las marcas en la base de datos.</p> <p>4. El sistema muestra en una tabla las marcas existentes de la base de datos.</p>

	<ol style="list-style-type: none"> 5. El gerente visualiza la lista de marcas existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada marca en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID de la marca seleccionada. 11. El sistema busca la marca en la base de datos por su ID. 12. El sistema elimina la marca encontrada. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con las marcas existentes.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de marcas agregados en la base de datos. 3. El sistema no encuentra marcas existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay marcas.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 9. El gerente presiona el botón “No, cancelar” de la ventana. 10. El sistema cancela la captura de ID. 11. El sistema cierra la ventana modal. 12. El sistema regresa al paso 1 del flujo principal.
Post condición:	La marca seleccionada será eliminada de la base de datos.

10.11.13. Gestionar pedidos.

Extends= Despachar pedidos, eliminar pedidos.

Tabla 10.10.13: Caso de uso a detalle (Gestionar Pedidos)

N^{ra} CU:	CU 004
Nombre:	Gestionar pedidos.
Descripción	El gerente podrá realizar una gestión de los pedidos existentes en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.

Flujo principal:	<ol style="list-style-type: none"> 6. El gerente visualiza la interfaz de inicio. 7. El gerente visualiza el menú de opciones. 8. El gerente selecciona la opción de “Pedidos.” 9. El sistema despliega la interfaz de pedidos. 10. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar los pedidos existentes.

10.11.14. Despachar pedidos

Tabla 10.10.14: Caso de uso a detalle (Despachar Pedidos)

Nª CU:	CU 004.1
Nombre:	Despachar Pedidos.
Descripción	El gerente podrá despachar un pedido existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de pedidos.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de pedidos. 2. El sistema busca si existen datos de pedidos agregados en la base de datos. 3. El sistema encuentra los pedidos en la base de datos. 4. El sistema muestra en una tabla los pedidos existentes de la base de datos. 5. El gerente visualiza la lista de pedidos existentes de la tabla. 6. El gerente selecciona el pedido que desea despachar. 7. El sistema toma los datos del pedido seleccionado. 8. El sistema despliega la interfaz para despachar un pedido. 9. El sistema muestra la opción para cambiar de estado al pedido que selecciono el gerente. 10. El gerente visualiza la opción. 11. El gerente cambio el estado del pedido de a entregado. 15. El gerente presiona el botón “Guardar” de la interfaz. 16. El sistema realiza la validación de los datos que manda el gerente. 17. El sistema da como resultado que los datos son correctos. 18. El sistema modifica el estado del producto a entregado. 19. El sistema manda el mensaje “Producto Entregado”. 20. El sistema muestra la interfaz de datos del producto.

Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de pedidos agregados en la base de datos. 3. El sistema no encuentra pedidos existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay pedidos.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Post condición:	El pedido seleccionado cambiara de estado ha Entregado en la base de datos.

10.11.15. Eliminar Pedidos

Tabla 10.10.15: Caso de uso a detalle (Eliminar Pedidos)

N^{ra} CU:	CU 004.2
Nombre:	Eliminar Pedidos.
Descripción	El gerente podrá eliminar un pedido existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de productos.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de pedidos. 2. El sistema busca si existen datos de pedidos agregados en la base de datos. 3. El sistema encuentra los pedidos en la base de datos. 4. El sistema muestra en una tabla los pedidos existentes de la base de datos. 5. El gerente visualiza la lista de pedidos existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada pedido en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID del pedido seleccionado. 11. El sistema busca el pedido en la base de datos por su ID. 12. El sistema elimina el pedido encontrado. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con los pedidos existentes.

Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de pedidos agregados en la base de datos. 3. El sistema no encuentra pedidos existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay pedidos.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 9. El gerente presiona el botón “No, cancelar” de la ventana. 10. El sistema cancela la captura de ID. 11. El sistema cierra la ventana modal. 12. El sistema regresa al paso 1 del flujo principal.
Post condición:	El pedido seleccionado será eliminado de la base de datos.

10.11.16. Gestionar clientes

Extends= Agregar clientes, modificar clientes, eliminar clientes.

Tabla 10.10.16: Caso de uso a detalle (Gestionar Clientes)

Nª CU:	CU 005
Nombre:	Gestionar Clientes.
Descripción	El gerente podrá realizar una gestión de los clientes existentes en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de inicio. 2. El gerente visualiza el menú de opciones. 3. El gerente selecciona la opción de “Clientes.” 4. El sistema despliega la interfaz de clientes. 5. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar los clientes existentes.

10.11.17. Agregar clientes.

Tabla 10.10.17: Caso de uso a detalle (Agregar Clientes)

Nª CU:	CU 005.1
---------------	-----------------

Nombre:	Agregar Clientes.
Descripción	El gerente podrá agregar un nuevo cliente a la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de clientes.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de clientes. 2. El gerente presiona el botón “Agregar Clientes” de la interfaz. 3. El sistema despliega la interfaz de para agregar clientes. 4. El sistema muestra el formulario para agregar un nuevo cliente. 5. El gerente visualiza el formulario. 6. El gerente llena los campos del formulario presentado. 7. El gerente presiona el botón “Guardar” del formulario. 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema da como resultado que los datos son correctos. 10. El sistema guarda los datos del nuevo cliente en la base de datos. 11. El sistema manda el mensaje “Cliente agregado”. 12. El sistema muestra la interfaz de datos del cliente.
Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 10. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 11. El sistema no agrega los datos a la Base de Datos. 12. El sistema mostrar el formulario con los datos que agrego el gerente. 13. El sistema regresa al paso 5 del flujo principal.
Post condición:	El nuevo cliente será agregado a la base de datos.

10.11.18. Modificar clientes.

Tabla 10.10.18: Caso de uso a detalle (Modificar Clientes)

Nª CU:	CU 005.2
Nombre:	Modificar Clientes.

Descripción	El gerente podrá modificar un cliente existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de clientes.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de clientes. 2. El sistema busca si existen datos de clientes agregados en la base de datos. 3. El sistema encuentra los clientes en la base de datos. 4. El sistema muestra en una tabla los clientes existentes de la base de datos. 5. El gerente visualiza la lista de clientes existentes de la tabla. 6. El gerente selecciona el cliente que desea modificar. 7. El sistema toma los datos del cliente seleccionado. 8. El sistema despliega la interfaz para modificar un cliente. 9. El sistema muestra el formulario de modificación de clientes con los datos que obtuvo de la selección del gerente. 10. El gerente visualiza el formulario. 11. El gerente modifica los datos que desee del cliente seleccionado. 13. El gerente presiona el botón “Guardar” del formulario. 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema da como resultado que los datos son correctos. 16. El sistema modifica los nuevos datos en el cliente seleccionado de la base de datos. 17. El sistema manda el mensaje “Cliente modificado”. 18. El sistema muestra la interfaz de datos del cliente.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de clientes agregados en la base de datos. 3. El sistema no encuentra clientes existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay clientes.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 16. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 17. El sistema no agrega los datos a la Base de Datos. 18. El sistema mostrar el formulario con los datos que agregó el gerente. 19. El sistema regresa al paso 5 del flujo principal.

Post condición:	Los nuevos datos ingresados serán modificados en el cliente de la base de datos.
------------------------	--

10.11.19. Eliminar clientes.

Tabla 10.10.19: Caso de uso a detalle (Eliminar Clientes)

Nª CU:	CU 005.3
Nombre:	Eliminar Clientes.
Descripción	El gerente podrá eliminar un cliente existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de clientes.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de clientes. 2. El sistema busca si existen datos de clientes agregados en la base de datos. 3. El sistema encuentra los clientes en la base de datos. 4. El sistema muestra en una tabla los clientes existentes de la base de datos. 5. El gerente visualiza la lista de clientes existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada cliente en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID del cliente seleccionado. 11. El sistema busca el cliente en la base de datos por su ID. 12. El sistema elimina el cliente encontrado. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con los clientes existentes.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de clientes agregados en la base de datos. 3. El sistema no encuentra clientes existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay clientes.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.

Flujo alternativo 2:	13. El gerente presiona el botón “No, cancelar” de la ventana. 14. El sistema cancela la captura de ID. 15. El sistema cierra la ventana modal. 16. El sistema regresa al paso 1 del flujo principal.
Post condición:	El cliente seleccionado será eliminado de la base de datos.

10.11.20. Gestionar administradores.

Extends= Agregar administradores, modificar administradores, eliminar administradores.

Tabla 10.10.20: Caso de uso a detalle (Gestionar Administradores)

Nª CU:	CU 006
Nombre:	Gestionar Administradores.
Descripción	El gerente podrá realizar una gestión de los Administradores que pueden realizar acciones en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de inicio. 2. El gerente visualiza el menú de opciones. 3. El gerente selecciona la opción de “Configuraciones.” 4. El sistema despliega la interfaz de Configuraciones. 5. El gerente visualiza la interfaz de Configuraciones. 6. El gerente selecciona la opción “Usuarios” de la interfaz. 7. El sistema despliega la interfaz de Administradores. 8. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar los administradores existentes.

10.11.21. Agregar administradores.

Tabla 10.10.21: Caso de uso a detalle (Agregar Administradores)

Nª CU:	CU 006.1
Nombre:	Agregar Administradores.
Descripción	El gerente podrá agregar un nuevo administrador a la aplicación.
Usuario:	Administrador.

Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de administradores.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de administradores. 2. El gerente presiona el botón “Agregar Nuevo Administrador” de la interfaz. 3. El sistema despliega la interfaz de para agregar administradores. 4. El sistema muestra el formulario para agregar un nuevo administrador. 5. El gerente visualiza el formulario. 6. El gerente llena los campos del formulario presentado. 7. El gerente presiona el botón “Guardar” del formulario. 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema da como resultado que los datos son correctos. 10. El sistema guarda los datos del nuevo administrador en la base de datos. 11. El sistema manda el mensaje “Administrador agregado”. 12. El sistema muestra la interfaz de datos del administrador.
Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 10. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 11. El sistema no agrega los datos a la Base de Datos. 12. El sistema mostrar el formulario con los datos que agrego el gerente. 13. El sistema regresa al paso 5 del flujo principal.
Post condición:	El nuevo administrador será agregado a la base de datos.

10.11.22. Modificar administradores.

Tabla 10.10.22: Caso de uso a detalle (Modificar Administradores)

N^{ra} CU:	CU 006.2
Nombre:	Modificar Administradores.
Descripción	El gerente podrá modificar un administrador existente en la aplicación.
Usuario:	Administrador.

Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de administradores.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de administradores. 2. El sistema busca si existen datos de administradores agregados en la base de datos. 3. El sistema encuentra los administradores en la base de datos. 4. El sistema muestra en una tabla los administradores existentes de la base de datos. 5. El gerente visualiza la lista de administradores existentes de la tabla. 6. El gerente selecciona el administrador que desea modificar. 7. El sistema toma los datos del administrador seleccionado. 8. El sistema despliega la interfaz para modificar un administrador. 9. El sistema muestra el formulario de modificación de administradores con los datos que obtuvo de la selección del gerente. 10. El gerente visualiza el formulario. 11. El gerente modifica los datos que desee del administrador seleccionado. 13. El gerente presiona el botón “Guardar” del formulario. 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema da como resultado que los datos son correctos. 16. El sistema modifica los nuevos datos en el cliente seleccionado de la base de datos. 17. El sistema manda el mensaje “Administrador modificado”. 18. El sistema muestra la interfaz de datos del administrador.
Flujo alternativo 1:	<ol style="list-style-type: none"> 14. El sistema realiza la validación de los datos que manda el gerente. 15. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 16. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 17. El sistema no agrega los datos a la Base de Datos. 18. El sistema mostrar el formulario con los datos que agrego el gerente. 19. El sistema regresa al paso 5 del flujo principal.
Post condición:	Los nuevos datos ingresados serán modificados en el administrador de la base de datos.

10.11.23. Eliminar administradores.

Tabla 10.10.23: Caso de uso a detalle (Eliminar Administradores)

Nª CU:	CU 006.3
Nombre:	Eliminar Administradores.
Descripción	El gerente podrá eliminar un administrador existente en la aplicación.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de administradores.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de administradores. 2. El sistema busca si existen datos de administradores agregados en la base de datos. 3. El sistema encuentra los administradores en la base de datos. 4. El sistema muestra en una tabla los administradores existentes de la base de datos. 5. El gerente visualiza la lista de administradores existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada administrador en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID del administrador seleccionado. 11. El sistema busca el administrador en la base de datos por su ID. 12. El sistema elimina el administrador encontrado. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con los administradores existentes.
Flujo alternativo 2:	<ol style="list-style-type: none"> 9. El gerente presiona el botón “No, cancelar” de la ventana. 10. El sistema cancela la captura de ID. 11. El sistema cierra la ventana modal. 12. El sistema regresa al paso 1 del flujo principal.
Post condición:	El administrador seleccionado será eliminado de la base de datos.

10.11.24. Gestionar información empresarial.

Extends= Agregar marcas, modificar marcas, eliminar marcas.

Tabla 10.10.24: Caso de uso a detalle (Gestionar Información Empresarial)

N ^a CU: CU 007	
Nombre:	Gestionar Información Empresarial.
Descripción	El gerente podrá realizar una gestión de la información empresarial existente que se ve desde la página del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de inicio.
Flujo principal:	<ol style="list-style-type: none"> 11. El gerente visualiza la interfaz de inicio. 12. El gerente visualiza el menú de opciones. 13. El gerente selecciona la opción de “Agregar Datos de empresa, Info Contacto o Carrusel”. 14. El sistema despliega la interfaz de Datos de Empresa, Info Contacto o Carrusel. 15. El gerente realizara alguna gestión a su necesidad.
Post condición:	El gerente podrá gestionar la Información Empresarial existente.

10.11.25. Agregar Datos de empresa.

Tabla 10.10.25: Caso de uso a detalle (Agregar Datos de Empresa)

N ^a CU: CU 007.1	
Nombre:	Agregar Datos de Empresa
Descripción	El gerente podrá agregar los datos de la empresa que se visualizan en la página del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Datos de empresa.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Datos de empresa. 2. El sistema muestra el formulario para agregar los datos empresariales. 3. El gerente visualiza el formulario. 4. El gerente llena los campos del formulario presentado. 5. El gerente presiona el botón “Guardar” del formulario.

	<ol style="list-style-type: none"> 6. El sistema realiza la validación de los datos que manda el gerente. 7. El sistema da como resultado que los datos son correctos. 8. El sistema guarda los datos de empresa en la base de datos. 9. El sistema manda el mensaje “Guardado”. 10. El sistema muestra la interfaz de datos de empresa.
Flujo alternativo 1:	<ol style="list-style-type: none"> 6. El sistema realiza la validación de los datos que manda el gerente. 7. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 8. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 9. El sistema no agrega los datos a la Base de Datos. 10. El sistema mostrar el formulario con los datos que agregó el gerente. 11. El sistema regresa al paso 5 del flujo principal.
Post condición:	Los datos empresariales son guardados en la base de datos.

10.11.26. Modificar Datos de empresa.

Tabla 10.10.16: Caso de uso a detalle (Modificar Datos de Empresa)

Nª CU:	CU 007.2
Nombre:	Modificar Datos de Empresa
Descripción	El gerente podrá modificar los datos de la empresa que se visualizan en la página del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Datos de empresa.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Datos de empresa. 2. El sistema muestra el formulario para agregar los datos empresariales. 3. El sistema muestra el formulario con los datos existentes en la base de datos 4. El gerente visualiza el formulario. 5. El gerente llena los campos del formulario presentado. 6. El gerente presiona el botón “Guardar” del formulario.

	<ol style="list-style-type: none"> 7. El sistema realiza la validación de los datos que manda el gerente. 8. El sistema da como resultado que los datos son correctos. 9. El sistema guarda los datos de empresa en la base de datos. 10. El sistema manda el mensaje “Guardado”. 11. El sistema muestra la interfaz de datos de empresa.
Flujo alternativo 1:	<ol style="list-style-type: none"> 7. El sistema realiza la validación de los datos que manda el gerente. 8. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 9. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 10. El sistema no agrega los datos a la Base de Datos. 11. El sistema mostrar el formulario con los datos que agregó el gerente. 12. El sistema regresa al paso 5 del flujo principal.
Post condición:	Los datos empresariales son modificados en la base de datos.

10.11.27. Agregar Información de Contacto.

Tabla 10.10.27: Caso de uso a detalle (Agregar Información de Contacto)

N^{ra} CU:	CU 007.3
Nombre:	Agregar Información de Contacto
Descripción	El gerente podrá agregar la información de contacto que se visualiza en la página del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Información de Contacto.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Información de Contacto. 2. El sistema muestra el formulario para agregar la Información de Contacto 3. El gerente visualiza el formulario. 4. El gerente llena los campos del formulario presentado. 5. El gerente presiona el botón “Guardar” del formulario. 6. El sistema realiza la validación de los datos que manda el gerente. 7. El sistema da como resultado que los datos son correctos. 8. El sistema guarda la Información de Contacto en la base de datos.

	<ol style="list-style-type: none"> 9. El sistema manda el mensaje “Guardado”. 10. El sistema muestra la interfaz de Información de Contacto.
Flujo alternativo 1:	<ol style="list-style-type: none"> 6. El sistema realiza la validación de los datos que manda el gerente. 7. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 8. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 9. El sistema no agrega los datos a la Base de Datos. 10. El sistema mostrar el formulario con los datos que agrego el gerente. 11. El sistema regresa al paso 5 del flujo principal.
Post condición:	La Información de Contacto es guardada en la base de datos.

10.11.28. Modificar Información de Contacto.

Tabla 10.10.28: Caso de uso a detalle (Modificar Información de Contacto)

Nª CU:	CU 007.4
Nombre:	Modificar Información de Contacto
Descripción	El gerente podrá modificar la información de contacto que se visualiza en la página del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Información de Contacto.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Información de Contacto. 2. El sistema muestra el formulario para agregar la Información de Contacto. 3. El sistema muestra el formulario con los datos existentes en la base de datos 4. El gerente visualiza el formulario. 5. El gerente llena los campos del formulario presentado. 6. El gerente presiona el botón “Guardar” del formulario. 7. El sistema realiza la validación de los datos que manda el gerente. 8. El sistema da como resultado que los datos son correctos. 9. El sistema guarda la Información de Contacto en la base de datos. 10. El sistema manda el mensaje “Guardado”.

	11. El sistema muestra la interfaz de Información de Contacto.
Flujo alternativo 1:	<ol style="list-style-type: none"> 7. El sistema realiza la validación de los datos que manda el gerente. 8. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 9. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 10. El sistema no agrega los datos a la Base de Datos. 11. El sistema mostrar el formulario con los datos que agrego el gerente. 12. El sistema regresa al paso 5 del flujo principal.
Post condición:	La Información de Contacto es modificada en la base de datos.

10.11.29. Agregar Información Carrusel.

Tabla 10.10.29: Caso de uso a detalle (Agregar Información de Carrusel)

Nª CU:	CU 007.5
Nombre:	Agregar Información Carrusel.
Descripción	El gerente podrá agregar una nueva imagen con descripción para mostrar en el carrusel que se visualiza en la interfaz del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Carrusel.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Carrusel. 2. El gerente presiona el botón “Agregar Carrusel” de la interfaz. 3. El sistema despliega la interfaz de para agregar Imágenes. 4. El sistema muestra el formulario para agregar una nueva imagen. 5. El gerente visualiza el formulario. 6. El gerente llena los campos del formulario presentado. 7. El gerente presiona el botón “Guardar” del formulario. 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema da como resultado que los datos son correctos. 10. El sistema guarda los datos de la nueva Imagen en la base de datos. 11. El sistema manda el mensaje “Imagen agregada”. 12. El sistema muestra la interfaz de datos del Carrusel.

Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema realiza la validación de los datos que manda el gerente. 9. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 10. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 11. El sistema no agrega los datos a la Base de Datos. 12. El sistema mostrar el formulario con los datos que agrego el gerente. 13. El sistema regresa al paso 5 del flujo principal.
Post condición:	La Imagen para Carrusel será agregada a la base de datos.

10.11.30. Modificar Carrusel.

Tabla 10.10.30: Caso de uso a detalle (Modificar Información de Carrusel)

Nª CU:	CU 007.6
Nombre:	Modificar Carrusel.
Descripción	El gerente podrá modificar una imagen con descripción existente para mostrar en el carrusel que se visualiza en la interfaz del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Carrusel.
Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Carrusel. 2. El sistema busca si existen datos de Imágenes agregados en la base de datos. 3. El sistema encuentra las Imágenes en la base de datos. 4. El sistema muestra en una tabla las Imágenes existentes de la base de datos. 5. El gerente visualiza la lista de Imágenes existentes de la tabla. 6. El gerente selecciona la Imagen que desea modificar. 7. El sistema toma los datos de la Imagen seleccionada. 8. El sistema despliega la interfaz para modificar una Imagen. 9. El sistema muestra el formulario de modificación de Imágenes con los datos que obtuvo de la selección del gerente. 10. El gerente visualiza el formulario. 11. El gerente modifica los datos que desee de la Imagen seleccionada.

	<p>12. El gerente presiona el botón “Guardar” del formulario.</p> <p>13. El sistema realiza la validación de los datos que manda el gerente.</p> <p>14. El sistema da como resultado que los datos son correctos.</p> <p>15. El sistema modifica los nuevos datos en la Imagen seleccionada de la base de datos.</p> <p>16. El sistema manda el mensaje “Imagen modificado”.</p> <p>17. El sistema muestra la interfaz de datos de la Imagen.</p>
Flujo alternativo 1:	<p>2. El sistema busca si existen datos de Imágenes agregados en la base de datos.</p> <p>3. El sistema no encuentra marcas existentes en la base de datos.</p> <p>4. El sistema muestra el mensaje en la tabla “No hay Imágenes.”</p> <p>5. El gerente visualiza el mensaje.</p> <p>6. El gerente regresa el paso 1 del flujo principal.</p>
Flujo alternativo 2:	<p>13. El sistema realiza la validación de los datos que manda el gerente.</p> <p>14. El sistema encuentra que falta de ingresar datos en algún campo del formulario.</p> <p>15. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”.</p> <p>16. El sistema no agrega los datos a la Base de Datos.</p> <p>17. El sistema mostrar el formulario con los datos que agregó el gerente.</p> <p>18. El sistema regresa al paso 5 del flujo principal.</p>
Post condición:	Los nuevos datos ingresados serán modificados en la Imagen de la base de datos.

10.11.31. Eliminar Carrusel.

Tabla 10.10.31: Caso de uso a detalle (Eliminar Información de Carrusel)

Nª CU:	CU 007.7
Nombre:	Eliminar Carrusel.
Descripción	El gerente podrá eliminar una imagen con descripción existente para mostrar en el carrusel que se visualiza en la interfaz del cliente.
Usuario:	Administrador.
Precondición:	El administrador debe haber iniciado sesión en la página y debe encontrarse en la sección de Carrusel.

Flujo principal:	<ol style="list-style-type: none"> 1. El gerente visualiza la interfaz de Carrusel. 2. El sistema busca si existen datos de Imágenes agregados en la base de datos. 3. El sistema encuentra las Imágenes en la base de datos. 4. El sistema muestra en una tabla las Imágenes existentes de la base de datos. 5. El gerente visualiza la lista de Imágenes existentes de la tabla. 6. El gerente da click en el icono de basurero que existe en cada Imagen en la tabla. 7. El sistema muestra una ventana modal con el mensaje “Esta seguro que quiere eliminar este registro” y los botones “Si, confirmar”, “No, cancelar”. 8. El gerente visualiza las opciones de la ventana modal. 9. El gerente presiona el botón “Si, confirmar” de la ventana. 10. El sistema obtiene el ID de la Imagen seleccionada. 11. El sistema busca la Imagen en la base de datos por su ID. 12. El sistema elimina la Imagen encontrada. 13. El sistema manda el mensaje “Registro Eliminado”. 14. El sistema muestra nuevamente la tabla con las Imágenes existentes.
Flujo alternativo 1:	<ol style="list-style-type: none"> 2. El sistema busca si existen datos de Imágenes agregados en la base de datos. 3. El sistema no encuentra marcas existentes en la base de datos. 4. El sistema muestra el mensaje en la tabla “No hay Imágenes.” 5. El gerente visualiza el mensaje. 6. El gerente regresa el paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 9. El gerente presiona el botón “No, cancelar” de la ventana. 10. El sistema cancela la captura de ID. 11. El sistema cierra la ventana modal. 12. El sistema regresa al paso 1 del flujo principal.
Post condición:	<p>La Imagen seleccionada será eliminada de la base de datos.</p>

10.12. Casos de uso A detalle Cliente

10.12.1. Gestionar su cuenta

Extends= Registrarse, modificar datos de su cuenta.

Tabla 10.11.1: Caso de uso a detalle (Gestionar Cuenta)

N ^a CU: CU 008	
Nombre:	Gestionar su cuenta.
Descripción	El cliente podrá gestionar una cuenta que tenga en la página.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de inicio de la aplicación.
Flujo principal:	<ol style="list-style-type: none">1. El cliente visualiza la interfaz de inicio.2. El cliente presiona la opción “Iniciar Sesión” del menú de opciones de la página.3. El sistema despliega los formularios para iniciar sesión.4. El cliente gestiona su cuenta dependiendo del tipo de acción.
Post condición:	El cliente podrá observar la manera para gestionar su cuenta.

10.12.2. Registrarse.

Tabla 10.11.2: Caso de uso a detalle (Registrarse)

N ^a CU: CU 008.1	
Nombre:	Registrarse.
Descripción	El cliente podrá registrarse en la página para acceder a los beneficios de la misma.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de iniciar sesión de la aplicación.
Flujo principal:	<ol style="list-style-type: none">1. El cliente visualiza la interfaz de Inicio de Sesión.2. El cliente presiona el botón “Crear una cuenta” de la interfaz.3. El sistema muestra el formulario para crearse una cuenta.4. El cliente visualiza el formulario para crearse una cuenta5. El cliente llena los campos del formulario presentado.6. El cliente presiona el botón “Registrarse” del formulario.

	<ol style="list-style-type: none"> 7. El sistema realiza la validación de los datos que manda el cliente. 8. El sistema da como resultado que los datos son correctos. 9. El sistema guarda los datos del nuevo cliente en la base de datos. 10. El sistema manda el mensaje “Registro Correcto”. 11. El sistema muestra la interfaz de Inicio de Sesión.
Flujo alternativo 1:	<ol style="list-style-type: none"> 7. El sistema realiza la validación de los datos que manda el cliente. 8. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 9. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 10. El sistema no agrega los datos a la Base de Datos. 11. El sistema muestra el formulario con los datos que agregó el gerente. 12. El sistema regresa al paso 5 del flujo principal.
Post condición:	El nuevo cliente es registrado en la base de datos.

10.12.3. Modificar datos de su cuenta.

Tabla 10.11.3: Caso de uso a detalle (Modificar Datos de su cuenta)

Nª CU:	CU 008.2
Nombre:	Modificar datos de su cuenta.
Descripción	El cliente podrá cambiar los datos de su cuenta de usuario
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de inicio de la aplicación y debe haber iniciado sesión en la misma.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la lista de opciones que tiene la interfaz de inicio. 2. El cliente selecciona la opción que marca si nombre se usuario. 3. El sistema despliega un menú flotante con las opciones que tiene el cliente. 4. El cliente visualiza las opciones. 5. El cliente elige la opción “Mis Datos”. 6. El sistema busca los datos del cliente en la base de datos. 7. El sistema encuentra los datos buscados.

	<ol style="list-style-type: none"> 8. El sistema muestra la interfaz para cambiar los datos del cliente con sus datos respectivos. 9. El cliente mira la interfaz con sus datos. 10. El cliente modifica los datos que desee cambiar. 12. El cliente presiona el botón “Modificar Datos” del formulario. 13. El sistema realiza la validación de los datos que manda el cliente. 14. El sistema da como resultado que los datos son correctos. 15. El sistema modifica los datos del cliente enviados en la base de datos 16. El sistema manda el mensaje “Datos Modificados”. 17. El sistema muestra la interfaz Mi cuenta.
Flujo alternativo 1:	<ol style="list-style-type: none"> 13. El sistema realiza la validación de los datos que manda el cliente. 14. El sistema encuentra que falta de ingresar datos en algún campo del formulario. 15. El sistema marca en la parte inferior del campo faltante del formulario el mensaje “El campo es obligatorio”. 16. El sistema no agrega los datos a la Base de Datos. 17. El sistema mostrar el formulario con los datos que agrego el gerente. 18. El sistema regresa al paso 8 del flujo principal.
Post condición:	Los nuevos datos ingresados serán modificados en el cliente de la base de datos.

10.12.4. Gestionar sus compras.

Extends= Agregar un producto al carrito, Modificar la cantidad de pedido., Eliminar producto del carrito, Realizar pago, Ver pedidos.

Tabla 10.11.4: Caso de uso a detalle (Gestionar sus comprar)

Nª CU:	CU 009
Nombre:	Gestionar sus compras.
Descripción	El cliente podrá gestionar una compra que vaya a realizar.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de inicio de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de inicio. 2. El cliente presiona la opción “Carrito” del menú de opciones de la página. 3. El sistema la interfaz de carrito.

	4. El cliente gestiona su pedido como crea conveniente.
Post condición:	El cliente podrá observar la manera de gestionar su carrito de compra.

10.12.5. Agregar un producto al carrito.

Tabla 10.11.5.: Caso de uso a detalle (Agregar un Producto al carrito)

Nª CU: CU 009.1	
Nombre:	Agregar un producto al carrito
Descripción	El cliente podrá agregar un producto que desee comprar a su carrito de compra.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de productos de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de productos de la aplicación. 2. El cliente selecciona un producto que desee comprar. 3. El sistema muestra la interfaz de datos del producto. 4. El cliente visualiza la interfaz del producto a comprar. 5. El cliente selecciona la opción comprar de la interfaz de producto. 6. El sistema valida que haya un usuario logueado y activo en la página y base de datos. 7. El sistema encuentra el usuario. 8. El sistema valida que no exista el producto actualmente en el carrito del usuario activo y en la base de datos. 9. El sistema no encuentra el producto en el carrito del usuario. 10. El sistema agrega el producto a la tabla carrito en la base de datos con el usuario activo. 11. El sistema muestra el mensaje “Producto agregado al carrito”. 12. El cliente mira el mensaje del sistema.
Flujo alternativo 1:	<ol style="list-style-type: none"> 6. El sistema valida que haya un usuario logueado y activo en la página y base de datos. 7. El sistema no encuentra un usuario logueado y activo. 8. El sistema manda el mensaje “Para comprar necesita Iniciar Sesión”. 9. El cliente mira la notificación del sistema. 10. El cliente regresa al paso 1 del flujo principal.

Flujo alternativo 2:	<ol style="list-style-type: none"> 8. El sistema valida que no exista el producto actualmente en el carrito del usuario activo y en la base de datos. 9. El sistema encuentra el producto en la sección de carrito del usuario en la base de datos. 10. El sistema manda el mensaje “El producto ya está agregado al carrito”. 11. El cliente visualiza el mensaje. 12. El cliente regresa al paso 1 del flujo principal
Post condición:	El producto seleccionado por el cliente es agregado a la sección de carrito de la base de datos

10.12.6. Modificar la cantidad de productos pedidos.

Tabla 10.11.6: Caso de uso a detalle (Modificar la Cantidad de Productos Pedidos)

Nª CU:	CU 009.2
Nombre:	Modificar la cantidad de productos pedidos.
Descripción	El cliente podrá cambiar la cantidad de productos que va a comprar de los que él ha registrado en el carrito de compras.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de productos de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz principal de la aplicación. 2. El cliente selecciona el icono de “Carrito de compras” del menú de opciones. 3. El sistema valida que haya un usuario logueado y activo en la página y base de datos para ingresar a la sección de Carrito. 4. El sistema encuentra el usuario. 5. El sistema busca los productos que se han agregado al carrito por el cliente. 6. El sistema encuentra los productos. 7. El sistema despliega la interfaz de productos con la información encontrada de productos colocada en tablas. 8. El cliente visualiza la interfaz con los botones para agregar más cantidad de productos. 9. El cliente presiona el botón “+” del producto que quiera aumentar la cantidad. 10. El sistema valida que la cantidad actual no sobrepase la cantidad de productos existentes en la aplicación. 11. El sistema encuentra correcto el aumento de cantidad. 12. El sistema hace el cambio de cantidad en la base de datos en la tabla del producto.

	<p>13. El sistema muestra el cambio en la cantidad de producto y precio total del pedido.</p> <p>14. El cliente visualiza el cambio en la tabla.</p>
Flujo alternativo 1:	<p>3. El sistema valida que haya un usuario logueado y activo en la página y base de datos.</p> <p>4. El sistema no encuentra un usuario logueado y activo.</p> <p>5. El sistema manda el mensaje “Necesita iniciar sesión para ingresar a carrito”.</p> <p>6. El cliente mira la notificación del sistema.</p> <p>7. El cliente regresa al paso 1 del flujo principal.</p>
Flujo alternativo 2:	<p>5. El sistema busca los productos que se han agregado al carrito por el cliente.</p> <p>6. El sistema no encuentra algún producto agregado por el cliente en la base de datos.</p> <p>7. El sistema manda el mensaje “No hay productos en el carrito”.</p> <p>8. El cliente mira el mensaje del sistema.</p> <p>9. El cliente regresa al paso 1 del flujo principal.</p>
Flujo alternativo 3:	<p>10. El sistema valida que la cantidad actual no sobrepase la cantidad de productos existentes en la aplicación.</p> <p>11. El sistema encuentra que la cantidad pedida sobrepasa a la cantidad de productos existentes.</p> <p>12. El sistema manda el mensaje “Cantidad de productos máxima”.</p> <p>13. El sistema deshabilita el botón “+” de la interfaz de Carrito.</p> <p>14. El cliente observa los cambios.</p> <p>15. El cliente regresa al paso 1 del flujo principal.</p>
Post condición:	La cantidad del producto en el carrito cambia dependiendo de la decisión del cliente en la base de datos.

10.12.7. Eliminar productos del carrito.

Tabla 10.11.7: Caso de uso a detalle (Eliminar Productos del carrito)

Nª CU:	CU 009.3
Nombre:	Eliminar productos del carrito.
Descripción	El cliente podrá eliminar productos que haya agregado con anterioridad a su carrito de compras

Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de productos de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz principal de la aplicación. 2. El cliente selecciona el icono de “Carrito de compras” del menú de opciones. 3. El sistema valida que haya un usuario logueado y activo en la página y base de datos para ingresar a la sección de Carrito. 4. El sistema encuentra el usuario. 5. El sistema busca los productos que se han agregado al carrito por el cliente. 6. El sistema encuentra los productos. 7. El sistema despliega la interfaz de productos con la información encontrada de productos colocada en tablas. 8. El cliente visualiza la interfaz con los botones para agregar más cantidad de productos. 9. El cliente presiona el botón “X” del producto que quiera eliminar. 10. El sistema recoge el ID del producto seleccionado. 11. El sistema busca el producto en la base de datos con el ID. 12. El sistema encuentra el producto seleccionado 13. El sistema elimina el producto de la base de datos. 14. El sistema manda el mensaje “Producto eliminado”. 15. El sistema muestra la nueva lista de productos en el carrito. 16. El cliente observa los cambios generados por el sistema.
Flujo alternativo 1:	<ol style="list-style-type: none"> 3. El sistema valida que haya un usuario logueado y activo en la página y base de datos. 4. El sistema no encuentra un usuario logueado y activo. 5. El sistema manda el mensaje “Necesita iniciar sesión para ingresar a carrito”. 6. El cliente mira la notificación del sistema. 7. El cliente regresa al paso 1 del flujo principal.
Flujo alternativo 2:	<ol style="list-style-type: none"> 5. El sistema busca los productos que se han agregado al carrito por el cliente. 6. El sistema no encuentra algún producto agregado por el cliente en la base de datos. 7. El sistema manda el mensaje “No hay productos en el carrito”. 8. El cliente mira el mensaje del sistema. 9. El cliente regresa al paso 1 del flujo principal.

Post condición:	La cantidad del producto en el carrito cambia dependiendo de la decisión del cliente en la base de datos.
------------------------	---

10.12.8. Realizar el pago.

Tabla 10.11.8: Caso de uso a detalle (Realizar Pago)

Nª CU:	CU 009.4
Nombre:	Realizar pago
Descripción	El cliente podrá pagar por los productos pedidos.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de productos de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz principal de la aplicación. 2. El cliente selecciona el icono de “Carrito de compras” del menú de opciones. 3. El sistema valida que haya un usuario logueado y activo en la página y base de datos para ingresar a la sección de Carrito. 4. El sistema encuentra el usuario. 5. El sistema busca los productos que se han agregado al carrito por el cliente. 6. El sistema encuentra los productos. 7. El sistema despliega la interfaz de productos con la información encontrada de productos colocada en tablas. 8. El cliente sigue los pasos para realizar la compra. 9. El cliente acepta los términos y condiciones para la compra de productos. 10. El cliente selecciona el método de entrega de sus productos. 11. El cliente selecciona la dirección de entrega que desea. 12. El sistema valida que se haya seleccionado la dirección de entrega y se haya aceptado los términos y condiciones. 13. El sistema encuentra los datos validados. 14. El sistema muestra el formulario para ingresar una tarjeta de crédito para proseguir con el pago. 15. El cliente visualiza el formulario. 16. El cliente llena los campos necesitados del formulario. 17. El cliente selecciona la opción pagar del formulario. 18. El sistema valida los datos que son mandados. 19. El sistema valida correctamente que los datos de la tarjeta de crédito sean correctos. 20. El sistema manda los datos encriptados a la aplicación de pagos Stripe. 21. Stripe valida que la tarjeta de crédito sea real.

	<ol style="list-style-type: none"> 22. Stripe manda el token de compra para realizar el dispatch de los pedidos. 23. El sistema recibe el token con los datos mandados. 24. El sistema valida que los datos del token recibido sean correctos. 25. El sistema verifica correctamente los datos del token y empieza con la selección de los productos agregados al carrito. 26. El sistema manda los productos existentes en el carrito hacia la tabla de pedidos. 27. El sistema elimina los productos existentes en Carrito del usuario que hace el pedido. 28. El sistema busca los productos que son mandados en la tabla de productos. 29. El sistema reduce la cantidad de productos dependiendo de la cantidad mandada por el pedido. 30. El sistema manda el mensaje “Compra realizada con éxito”. 31. El sistema muestra la interfaz de pedidos del usuario. 32. El cliente visualiza la interfaz de pedidos.
<p>Flujo alternativo 1:</p>	<ol style="list-style-type: none"> 3. El sistema valida que haya un usuario logueado y activo en la página y base de datos. 4. El sistema no encuentra un usuario logueado y activo. 5. El sistema manda el mensaje “Necesita iniciar sesión para ingresar a carrito”. 6. El cliente mira la notificación del sistema. 7. El cliente regresa al paso 1 del flujo principal.
<p>Flujo alternativo 2:</p>	<ol style="list-style-type: none"> 5. El sistema busca los productos que se han agregado al carrito por el cliente. 6. El sistema no encuentra algún producto agregado por el cliente en la base de datos. 7. El sistema manda el mensaje “No hay productos en el carrito”. 8. El cliente mira el mensaje del sistema. 9. El cliente regresa al paso 1 del flujo principal.
<p>Flujo alternativo 3:</p>	<ol style="list-style-type: none"> 12. El sistema valida que se haya seleccionado la dirección de entrega y se haya aceptado los términos y condiciones. 13. El sistema encuentra que una de las dos validaciones no está realizada. 14. El sistema manda el mensaje “Para comprar debe seleccionar una dirección y aceptar los términos y condiciones”. 15. El cliente visualiza el mensaje que manda el sistema. 16. El cliente regresa al paso 7 del flujo principal.

Flujo alternativo 4:	<p>18. El sistema valida los datos que son mandados.</p> <p>19. El sistema encuentra que los datos de la tarjeta de crédito son incorrectos.</p> <p>20. El sistema manda el mensaje “Datos de pago incorrectos”.</p> <p>21. El cliente visualiza el mensaje.</p> <p>22. El cliente regresa al paso del flujo 15 principal.</p>
Post condición:	La compra de los productos agregados al carrito se hace de manera exitosa en Stripe y se genera los datos respectivos en la tabla de pedidos.

10.12.9. Ver pedidos.

Tabla 10.11.7: Caso de uso a detalle (Ver Pedidos)

N ^a CU: CU 009.5	
Nombre:	Ver pedidos
Descripción	El cliente podrá visualizar los pedidos de los productos que fueron comprados.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de productos de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de productos de la aplicación. 2. El cliente selecciona la opción que tiene su nombre de usuario del menú de opciones. 3. El sistema muestra un menú despegable. 4. El cliente selecciona la opción “Pedidos” del menú despegable. 5. El sistema busca los pedidos existentes del usuario. 6. El sistema encuentra los productos existentes. 7. El sistema muestra los pedidos realizado en una tabla. 8. El cliente visualiza los datos de pedidos. 9. El cliente selecciona un pedido de la tabla. 10. El sistema muestra una ventana emergente con los datos del pedido realiza.
Flujo alternativo 1:	<ol style="list-style-type: none"> 1. El sistema busca los pedidos existentes del usuario. 2. El sistema no encuentra los pedidos en la base de datos. 3. El sistema muestra el mensaje “No existen pedidos”. 4. El cliente visualiza el mensaje. 5. El cliente regresa al paso 1 del flujo principal

Post condición:	El cliente podrá visualizar todos los pedidos realizados.

10.12.10. Gestionar direcciones.

Extends: Agregar, modificar, eliminar sus direcciones

Tabla 10.11.10: Caso de uso a detalle (Gestionar sus direcciones)

N ^a CU:	CU 010
Nombre:	Gestionar sus direcciones.
Descripción	El cliente podrá gestionar sus direcciones de compra.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de inicio de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de inicio. 2. El cliente presiona la opción que tiene su nombre de usuario del menú de opciones. 3. El sistema muestra un menú despegable con las opciones de usuario. 4. El cliente presiona la opción “Direcciones” del menú. 5. El sistema valida que exista un usuario activo en el sistema y base de datos. 6. El sistema encuentra al usuario. 7. El sistema muestra la interfaz de direcciones con los datos del usuario.
Flujo alternativo 1:	<ol style="list-style-type: none"> 5. El sistema valida que exista un usuario activo en el sistema y base de datos. 6. El sistema no encuentra un usuario activo en la aplicación. 7. El sistema manda el mensaje “Debe estar logueado para acceder a Mis direcciones”. 8. El sistema muestra la interfaz de inicio. 9. El cliente vuelve al paso 1 del flujo principal.
Post condición:	El cliente podrá gestionar las direcciones que hay creado.

10.12.11. Agregar una dirección.

Tabla 10.11.11: Caso de uso a detalle (Agregar direcciones)

N ^a CU: CU 0010.1	
Nombre:	Agregar una dirección.
Descripción	El cliente podrá agregar direcciones de compra a la aplicación.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de direcciones de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de direcciones. 2. El cliente selecciona la opción de “Agregar Dirección” del menú de opciones. 3. El sistema muestra una ventana con el formulario de direcciones de usuario. 4. El cliente visualiza el formulario. 5. El cliente llena los datos del formulario. 6. El cliente selecciona la opción “Agregar” del formulario. 7. El sistema valida que los datos que va a mandar el cliente sean correctos. 8. El sistema valida de manera correcta los datos. 9. El sistema guarda los datos de la dirección en la base de datos. 10. El sistema manda el mensaje “Dirección agregada”. 11. El sistema muestra la interfaz de direcciones. 12. El cliente visualiza el mensaje y la nueva dirección agregada en la tabla.
Flujo alternativo 1:	<ol style="list-style-type: none"> 7. El sistema valida que los datos que va a mandar el cliente sean correctos. 8. El sistema encuentra algún error en los datos mandados. 9. El sistema manda el mensaje de error en el campo que falta del formulario de direcciones. 10. El cliente visualiza el mensaje de error. 11. El cliente regresa al paso 4 del flujo principal.
Post condición:	Los datos de la nueva dirección son agregados en la base de datos.

10.12.12. Modificar una dirección.

Tabla 10.11.12: Caso de uso a detalle (Modificar direcciones)

N ^a CU: CU 0010.2	
Nombre:	Modificar una dirección.

Descripción	El cliente podrá modificar una dirección que haya agregado.
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de direcciones de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de direcciones. 2. El sistema busca las direcciones que tenga el usuario agregadas en la base de datos. 3. El sistema encuentra las direcciones del usuario. 4. El sistema muestra las direcciones del usuario en una tabla en la interfaz. 5. El cliente visualiza la lista de direcciones existentes. 6. El cliente selecciona la opción de “Modificar” de alguna dirección existente. 7. El sistema guarda los datos de la dirección seleccionada. 8. El sistema muestra una ventana con el formulario de direcciones de usuario con los datos de la dirección seleccionada. 9. El cliente visualiza el formulario. 10. El cliente modifica el dato que desee de la dirección. 11. El cliente selecciona la opción “Modificar” del formulario. 12. El sistema valida que los datos que va a mandar el cliente sean correctos. 13. El sistema valida de manera correcta los datos. 14. El sistema guarda los datos modificados de la dirección en la base de datos. 15. El sistema manda el mensaje “Dirección modificada”. 16. El sistema muestra la interfaz de direcciones. 17. El cliente visualiza el mensaje y los nuevos datos de la dirección en la tabla.
Flujo alternativo 1:	<ol style="list-style-type: none"> 12. El sistema valida que los datos que va a mandar el cliente sean correctos. 13. El sistema encuentra algún error en los datos mandados. 14. El sistema manda el mensaje de error en el campo que falta del formulario de direcciones. 15. El cliente visualiza el mensaje de error. 16. El cliente regresa al paso 4 del flujo principal.
Post condición:	Los datos modificados de la dirección seleccionada son guardados en la base de datos.

10.12.13. Eliminar una dirección.

Tabla 10.11.13: Caso de uso a detalle (Eliminar direcciones)

N ^a CU: CU 0010.2	
Nombre:	Eliminar una dirección.
Descripción	El cliente podrá eliminar una dirección que haya agregado..
Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de direcciones de la aplicación y debe haber iniciado sesión en la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de direcciones. 2. El sistema busca las direcciones que tenga el usuario agregadas en la base de datos. 3. El sistema encuentra las direcciones del usuario. 4. El sistema muestra las direcciones del usuario en una tabla en la interfaz. 5. El cliente visualiza la lista de direcciones existentes. 6. El cliente selecciona la opción de “Eliminar” de las opciones existentes. 7. El sistema guarda el ID del producto seleccionado. 8. El sistema busca en la base de datos la dirección con el ID guardado. 9. El sistema elimina la dirección encontrada. 10. El sistema manda el mensaje “Dirección eliminada”. 11. El sistema muestra la nueva tabla de datos con las direcciones existente. 12. El usuario visualiza los cambios realizados por el sistema.
Post condición:	La dirección seleccionada es eliminada de la base de datos del sistema.

10.12.14. Enviar mensajes.

Tabla 10.11.14: Caso de uso a detalle (Enviar Mensajes)

N ^a CU: CU 0011	
Nombre:	Enviar mensajes.
Descripción	El cliente podrá comunicarse con el administrador sin la necesidad de tener una cuenta creada en el sistema.

Usuario:	Cliente.
Precondición:	El cliente debe encontrarse en la página de inicio de la aplicación.
Flujo principal:	<ol style="list-style-type: none"> 1. El cliente visualiza la interfaz de inicio. 2. El cliente selecciona la opción “Contacto” del menú de opciones de la aplicación. 3. El sistema muestra la interfaz de contacto. 4. El cliente visualiza la interfaz. 5. El cliente visualiza el formulario de envío de mensajes. 6. El cliente llena los datos del formulario. 7. El cliente presión la opción “Enviar” del formulario. 8. El sistema valida que los datos que va a mandar el cliente sean correctos. 9. El sistema encuentra que los datos son correctos. 10. El sistema guarda el mensaje en la base de datos. 11. El sistema manda el mensaje “Mensaje Enviado”. 12. El sistema vacía el formulario de mensajes. 13. El cliente visualiza los cambios realizados por el sistema.
Flujo alternativo 1:	<ol style="list-style-type: none"> 8. El sistema valida que los datos que va a mandar el cliente sean correctos. 9. El sistema encuentra algún error en los datos mandados. 10. El sistema manda el mensaje de error en el campo que falta del formulario de direcciones. 11. El cliente visualiza el mensaje de error. 12. El cliente regresa al paso 4 del flujo principal.
Post condición:	El mensaje del usuario en guardado en la base de datos.

10.13. Diagramas de Casos de Uso y Diagramas de Actividades

10.13.1. Administrador

10.13.1.1. Información empresarial

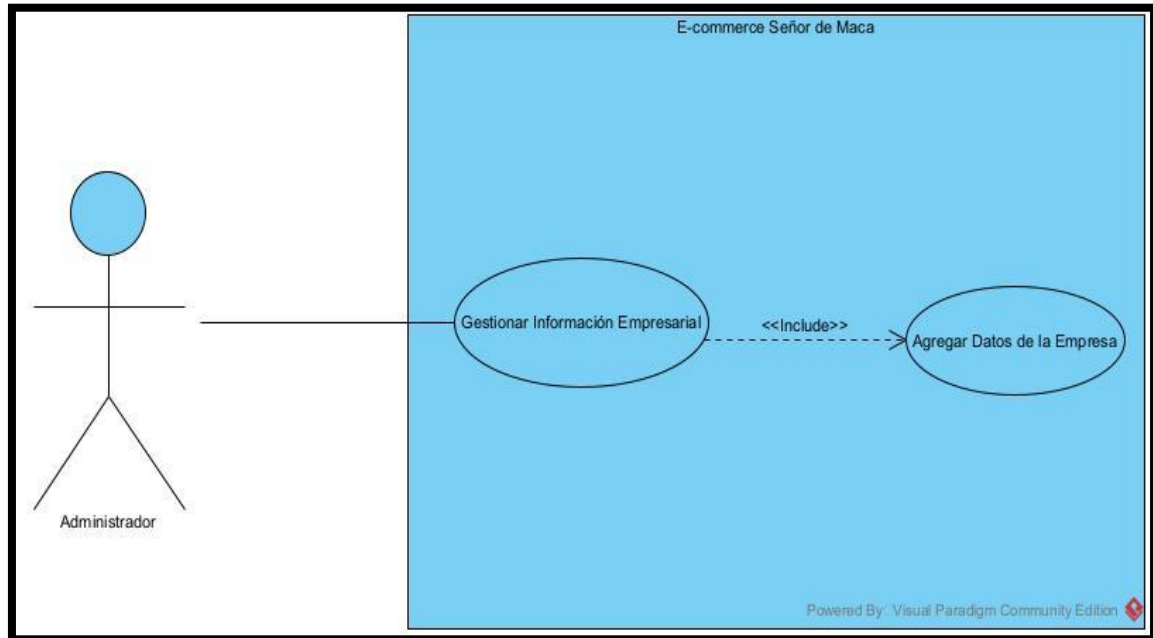


Figura 10.6: Diagrama de casos de Uso (Agregar Datos de la Empresa)

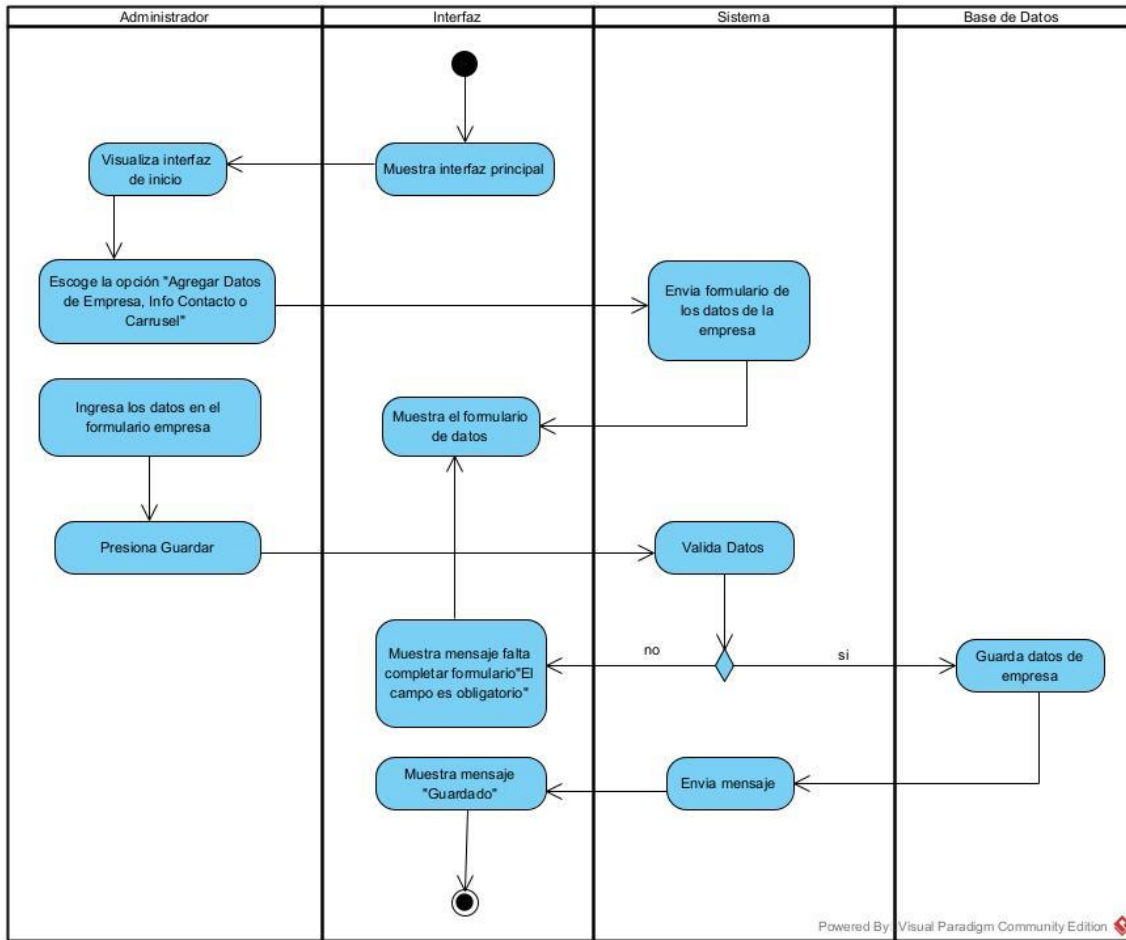


Figura 10.7: Diagrama de actividades: Agregar datos de la empresa.

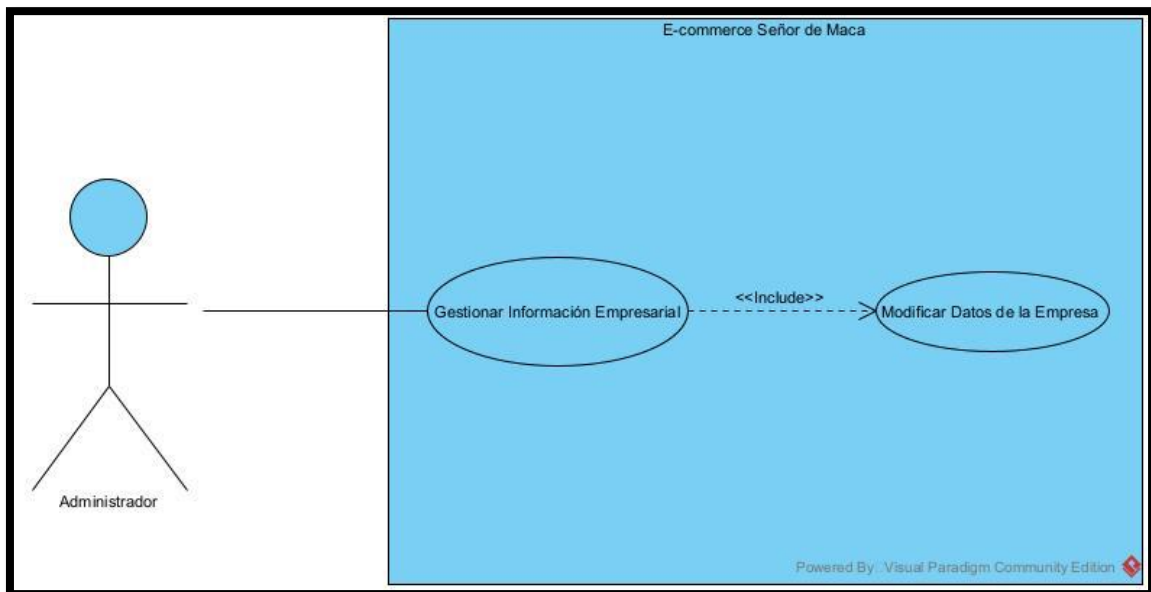


Figura 10.8: Diagrama de casos de Uso (Modificar Datos de la Empresa)

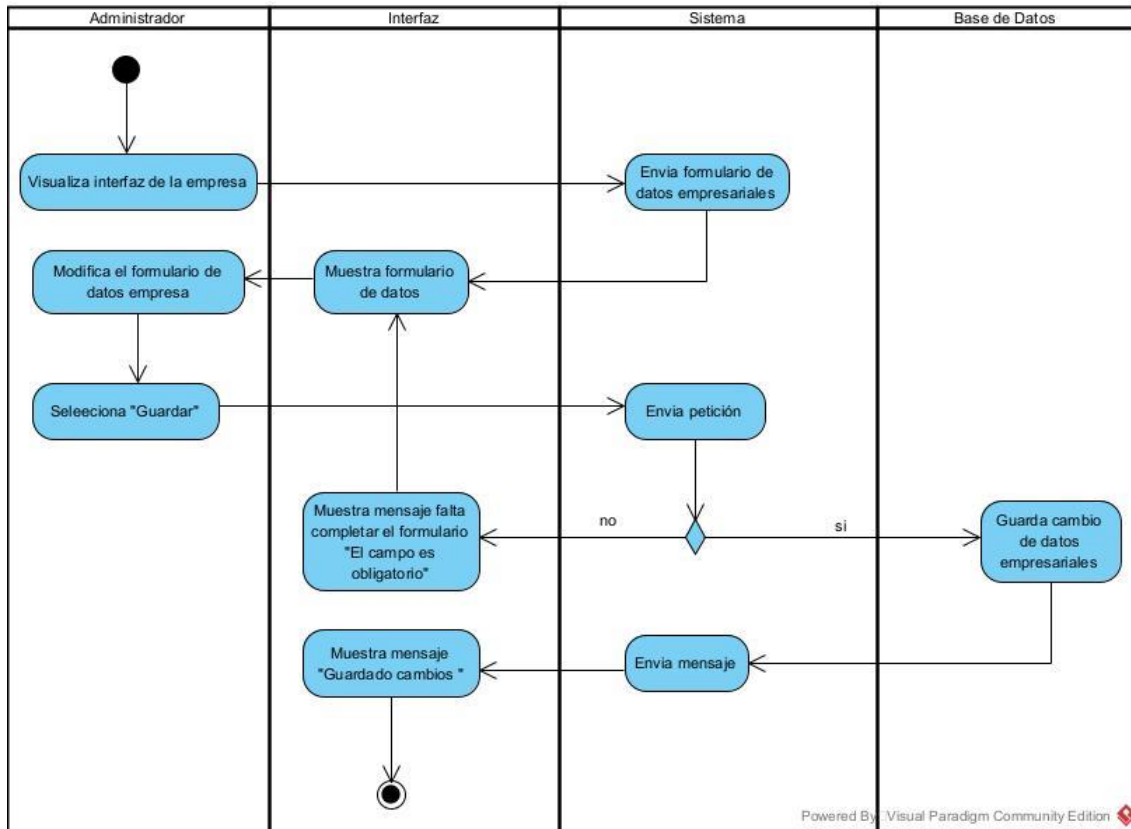


Figura 10.9: Diagrama de actividades: Modificar datos de la empresa.

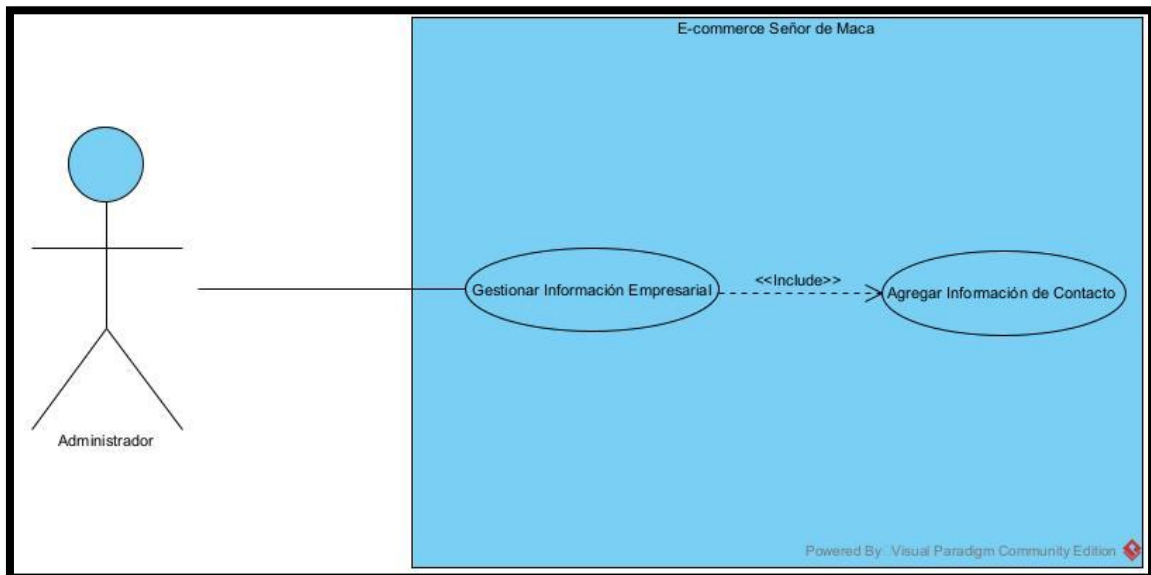


Figura 10.10: Diagrama de casos de Uso (Agregar Información de Contactos)

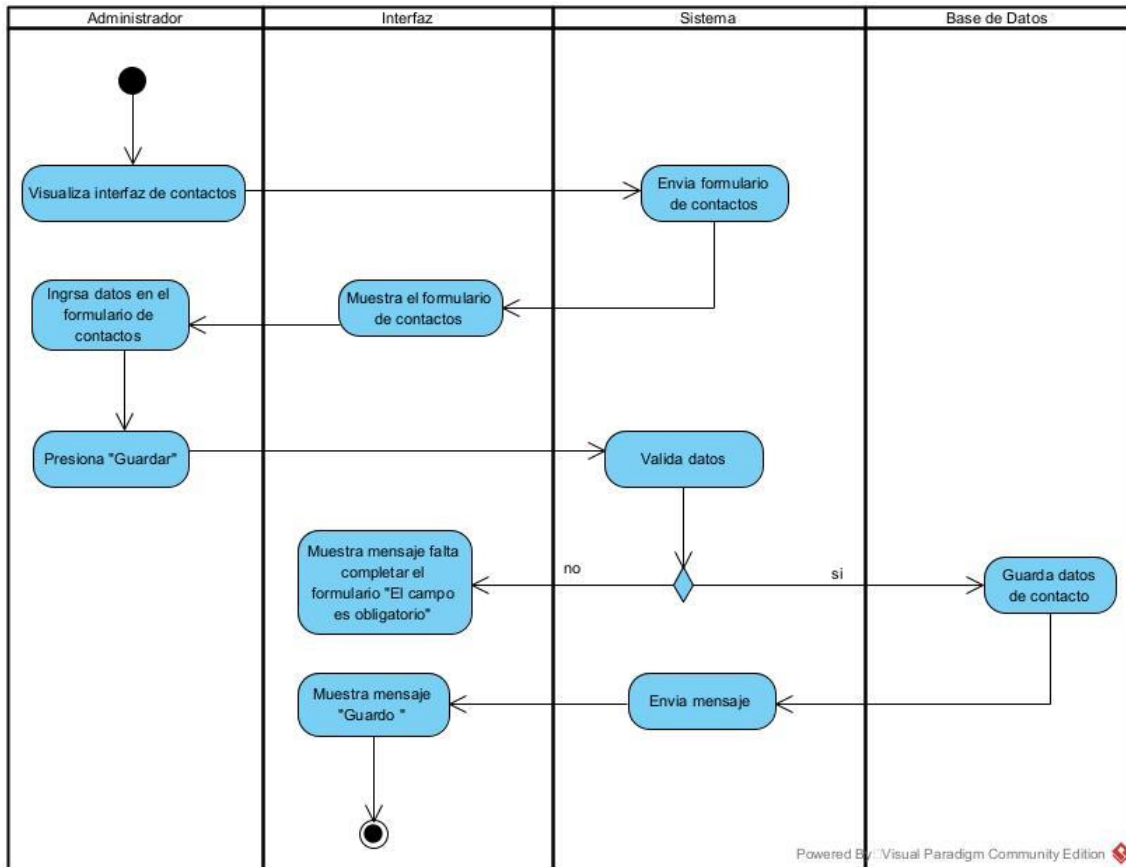


Figura 10.11: Diagrama de actividades: Agregar datos de contacto de la empresa.

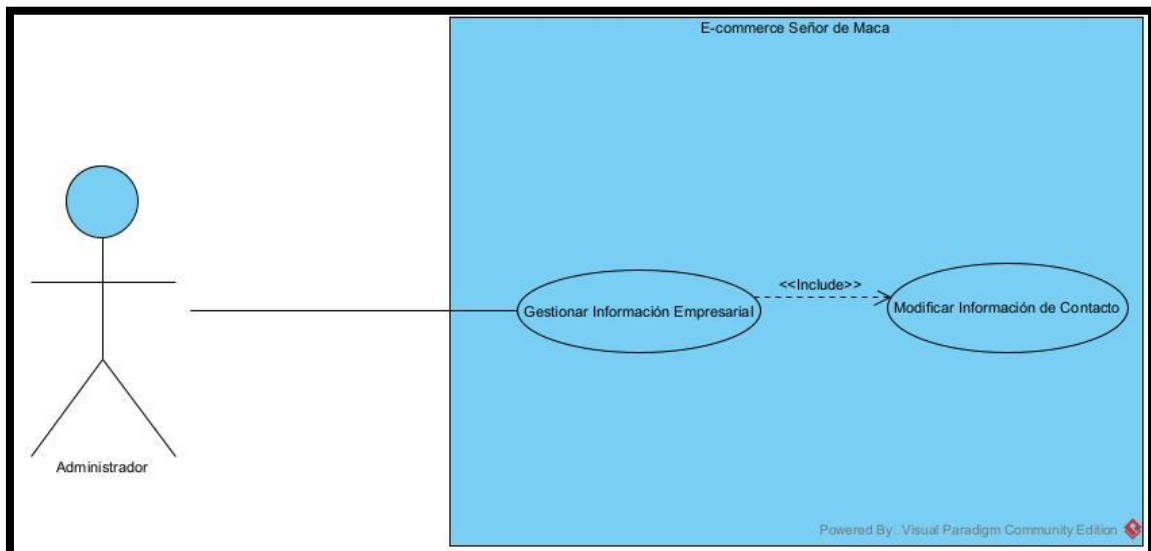


Figura 10.12: Modificar información de contacto.

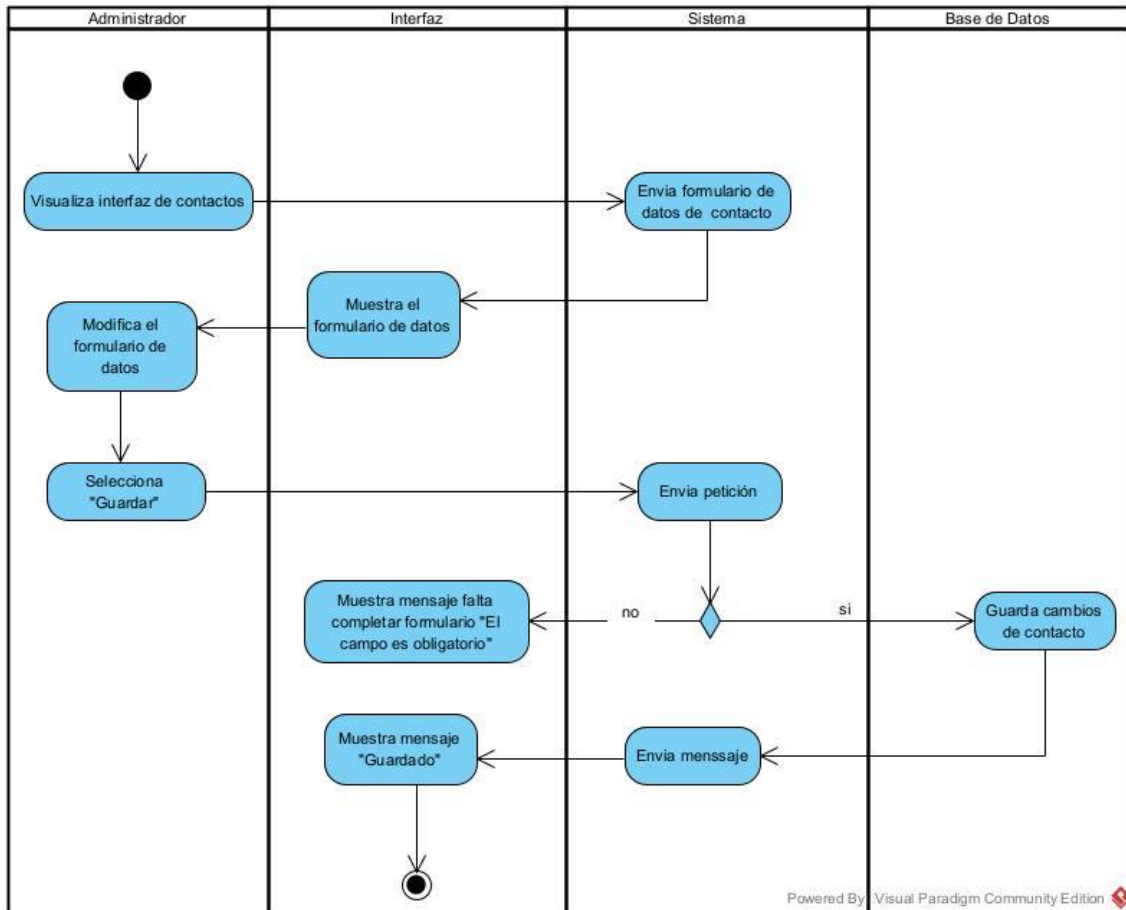


Figura 10.13: Diagrama de actividades: Modificar información de contacto.

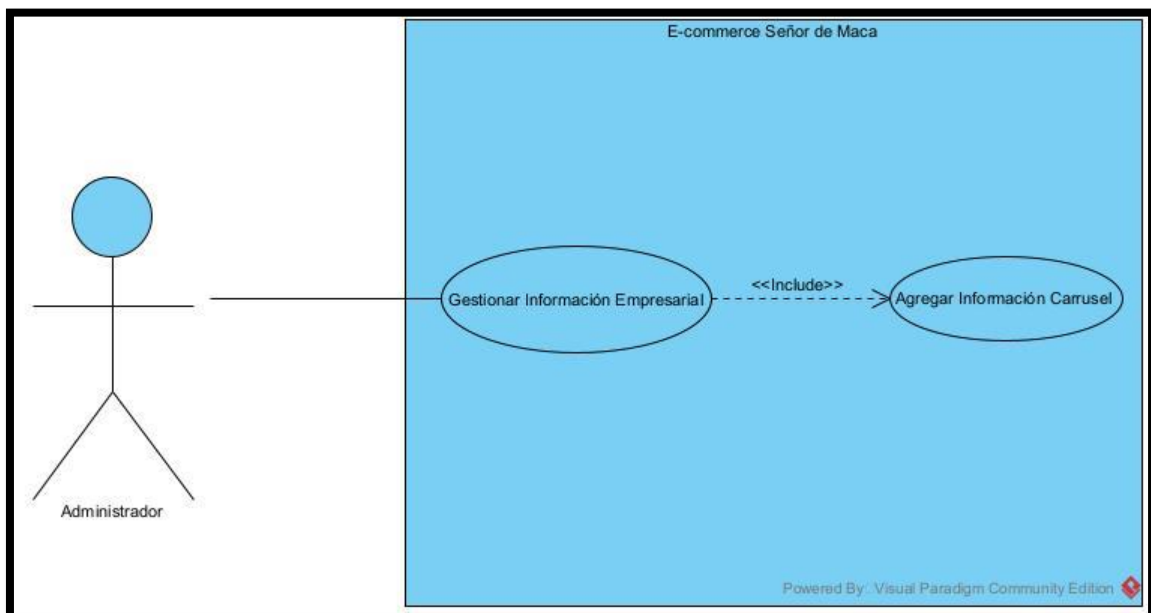


Figura 10.14: Agregar imágenes al Carrusel.

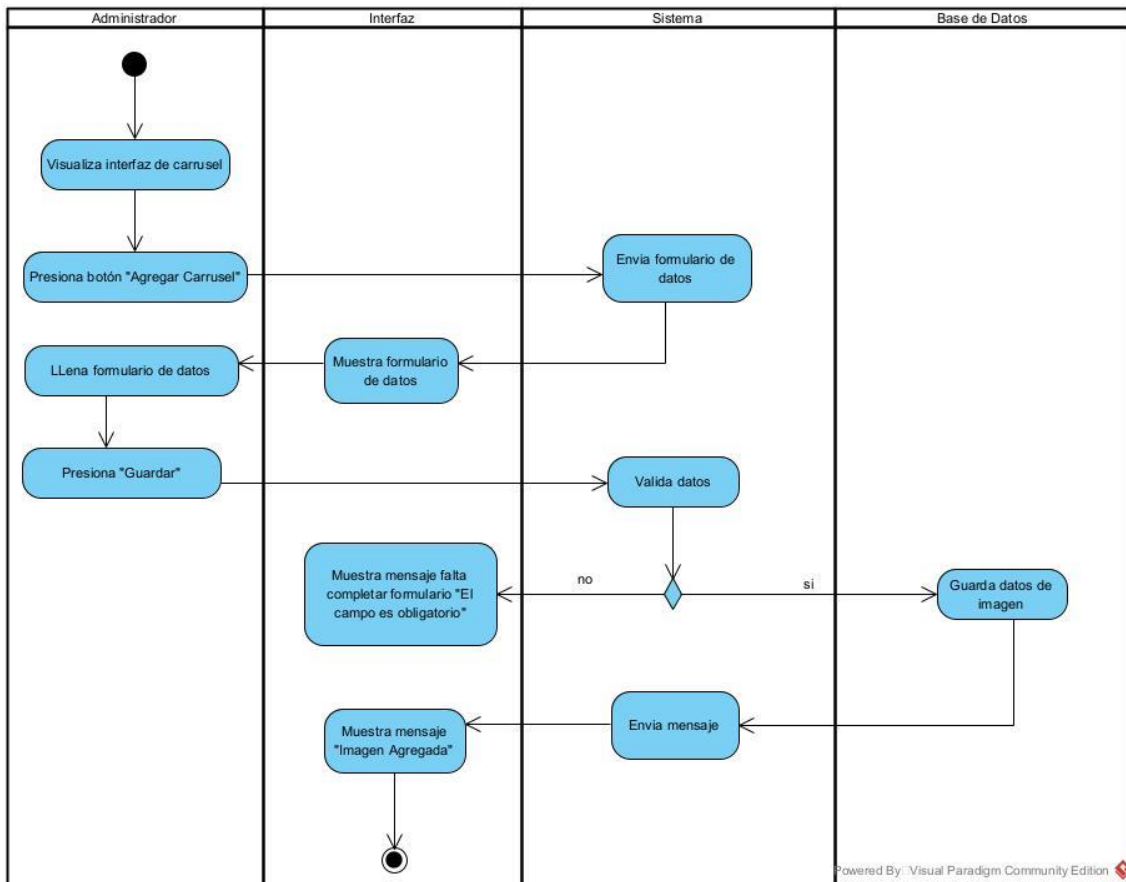


Figura 10.15: Diagrama de actividades: Agregar imágenes al carrusel.

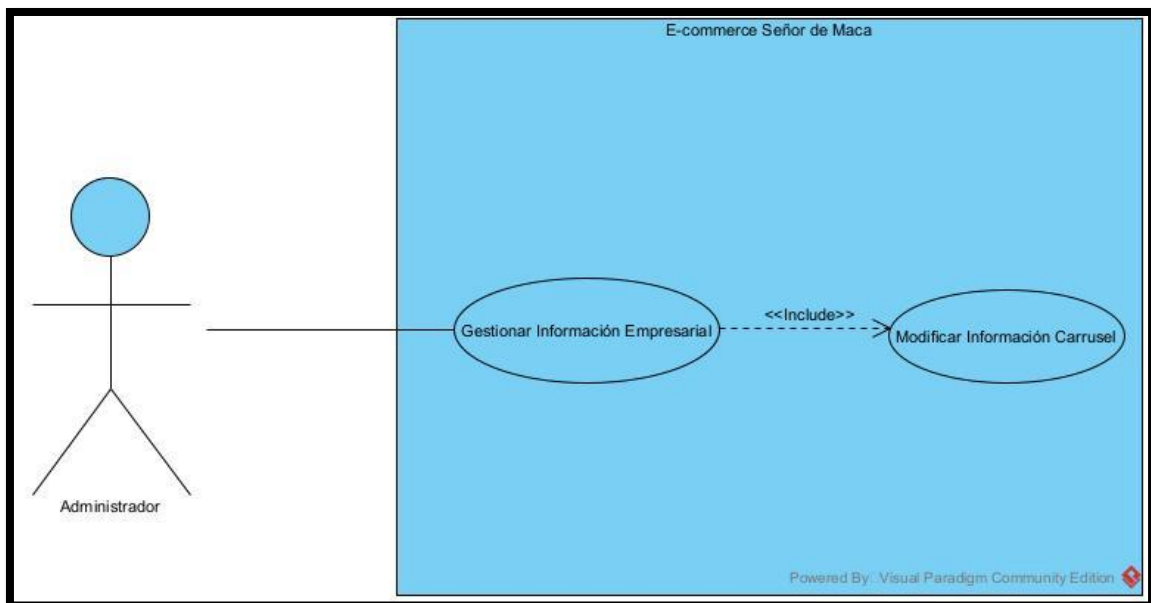


Figura 10.16: Modificar imágenes del carrusel.

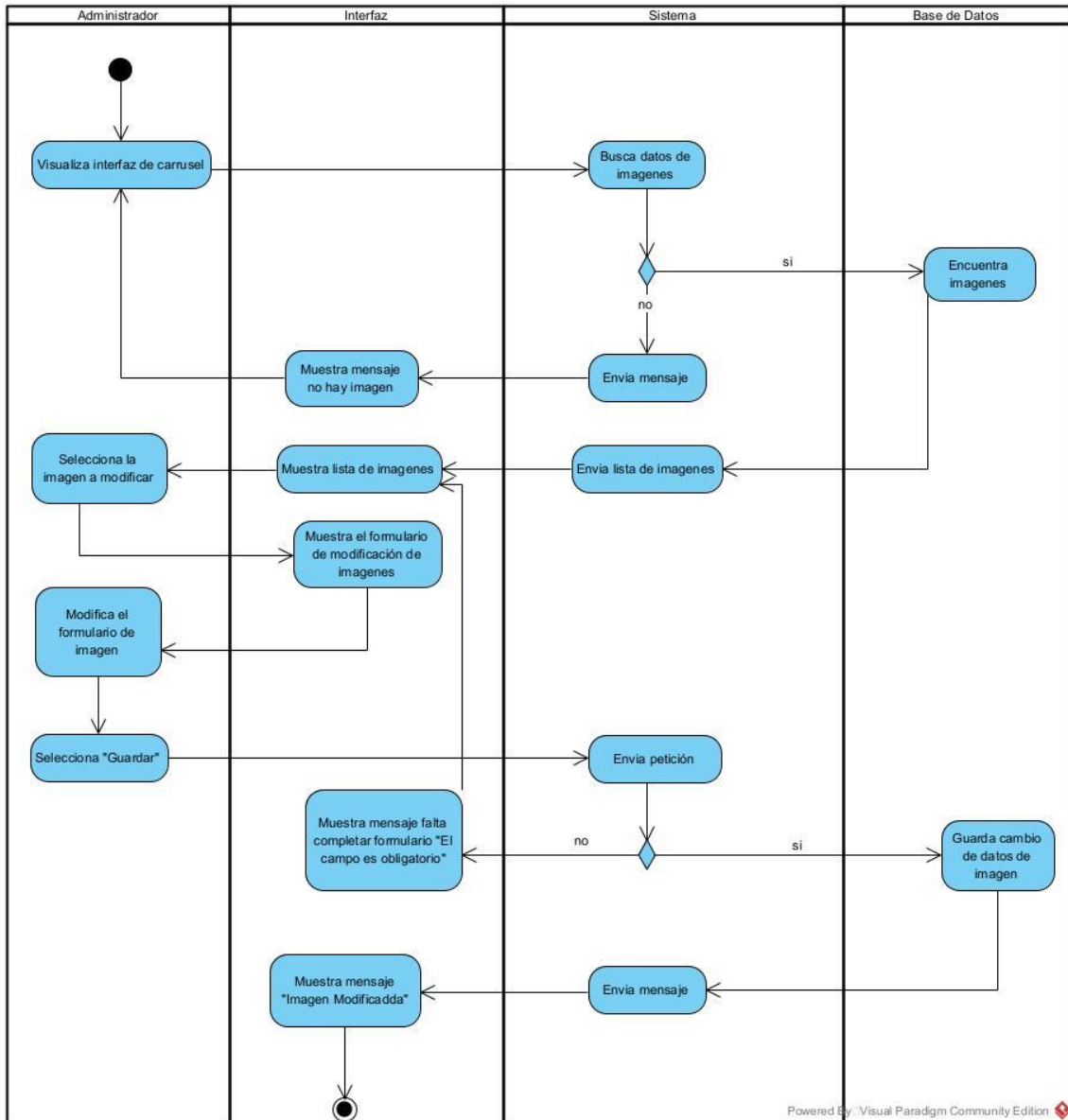


Figura 10.17: Diagrama de actividades: Modificar Imágenes del carrusel.

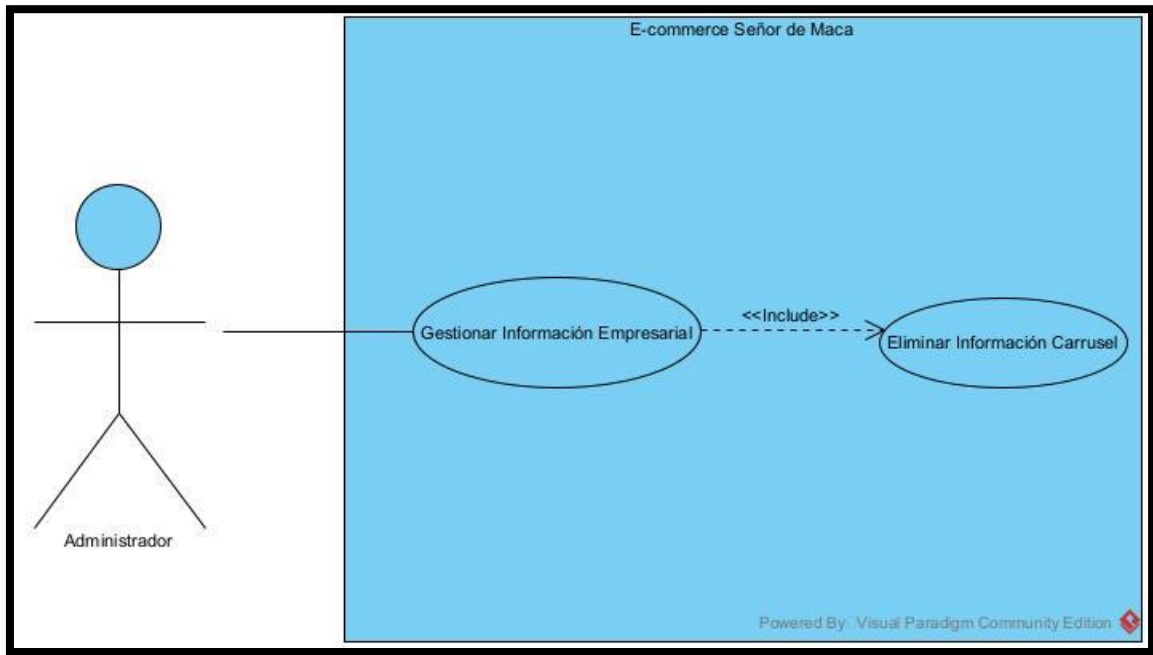


Figura 10.18: Eliminar imágenes del carrusel.

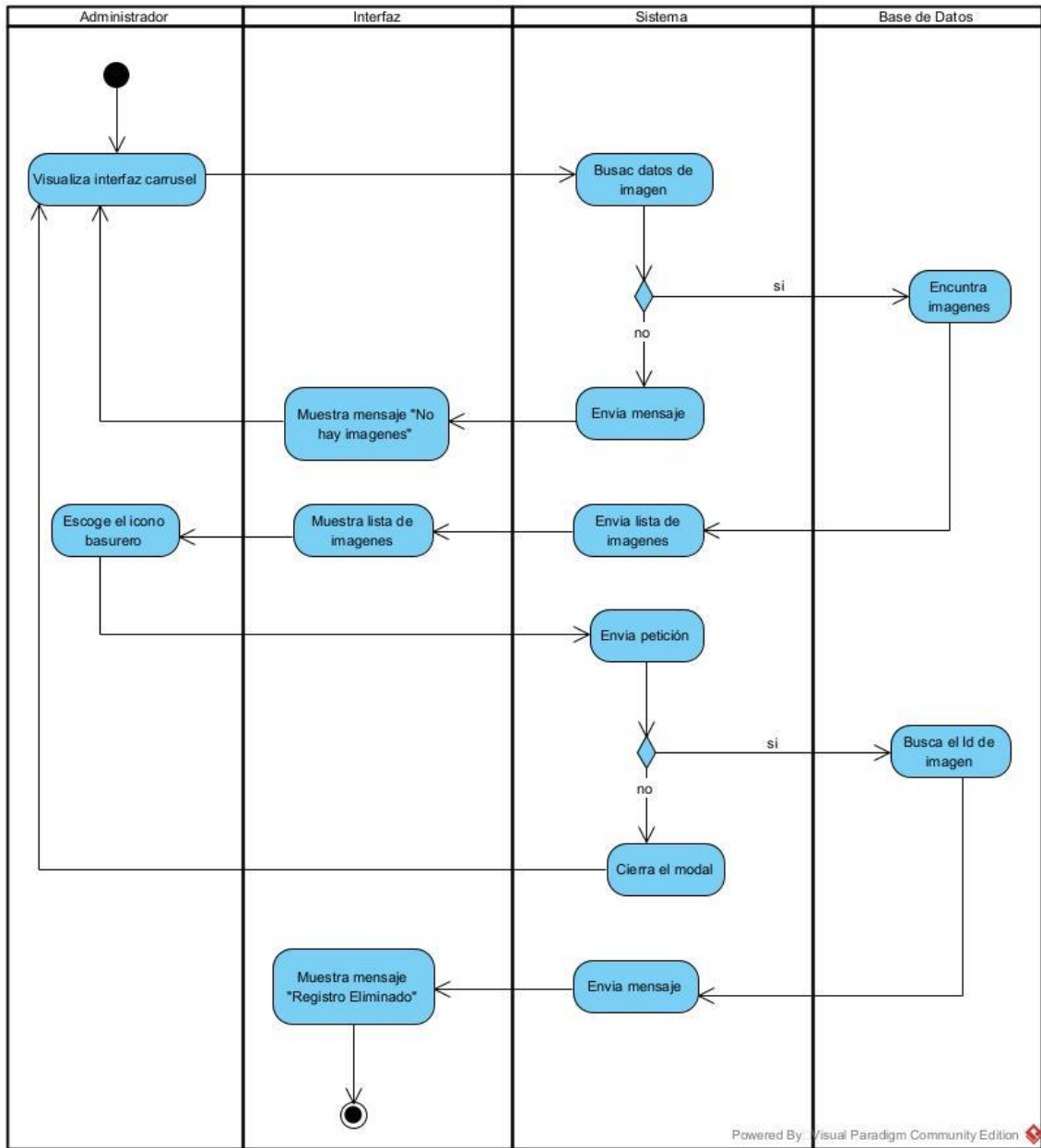


Figura 10.19: Diagrama de actividades: Eliminar imágenes del carrusel.

10.13.2. Cliente

10.13.2.1. Compras

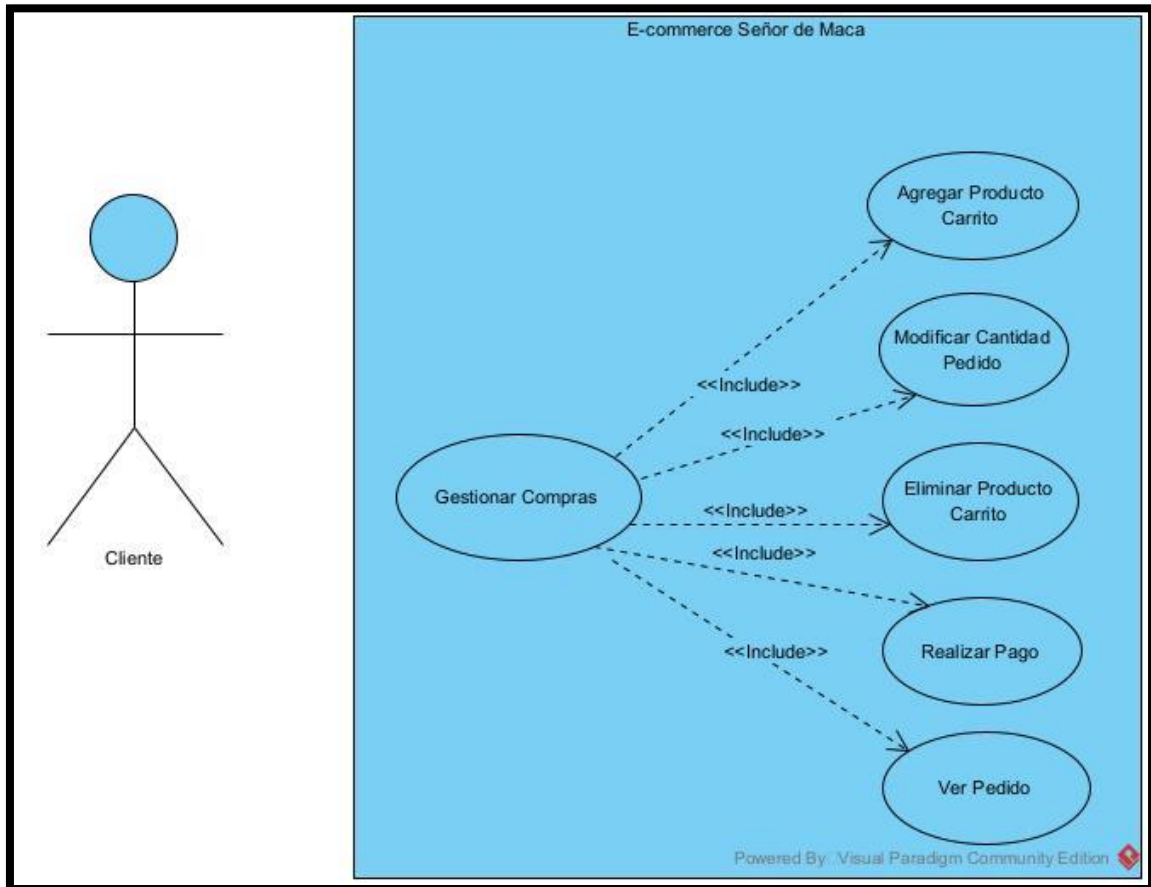


Figura 10.20: Diagrama de compras del cliente.

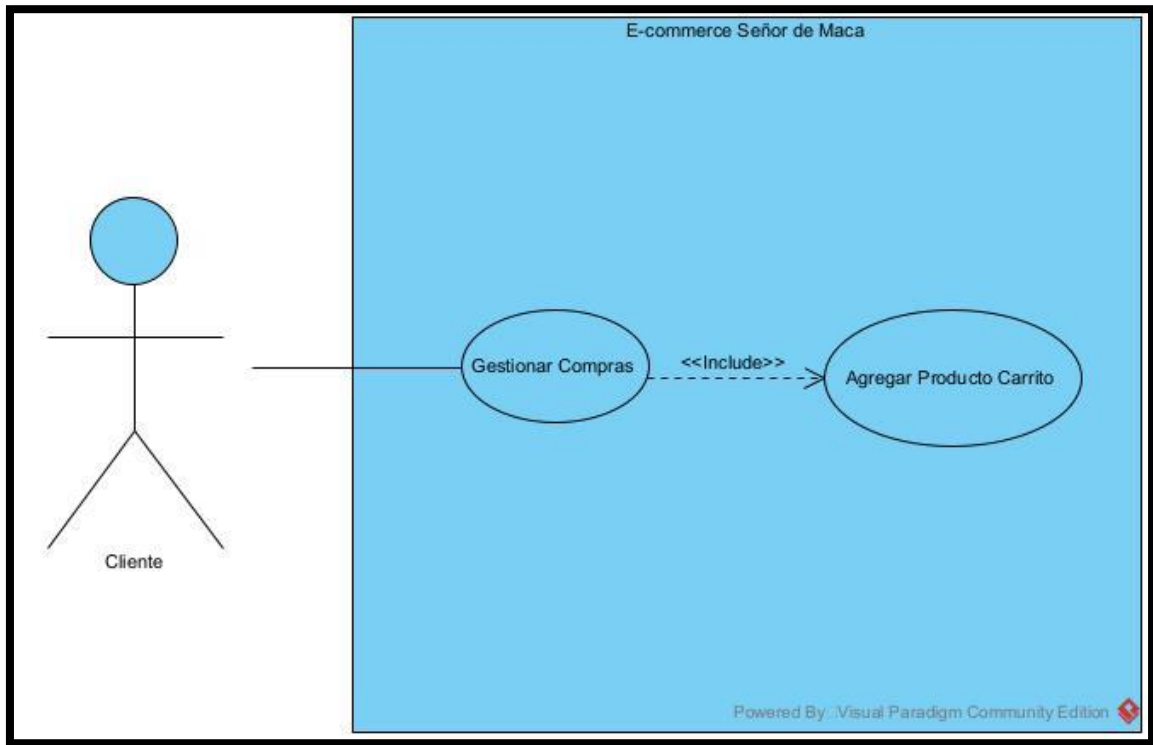


Figura 10.21: Agregar productos al carrito.

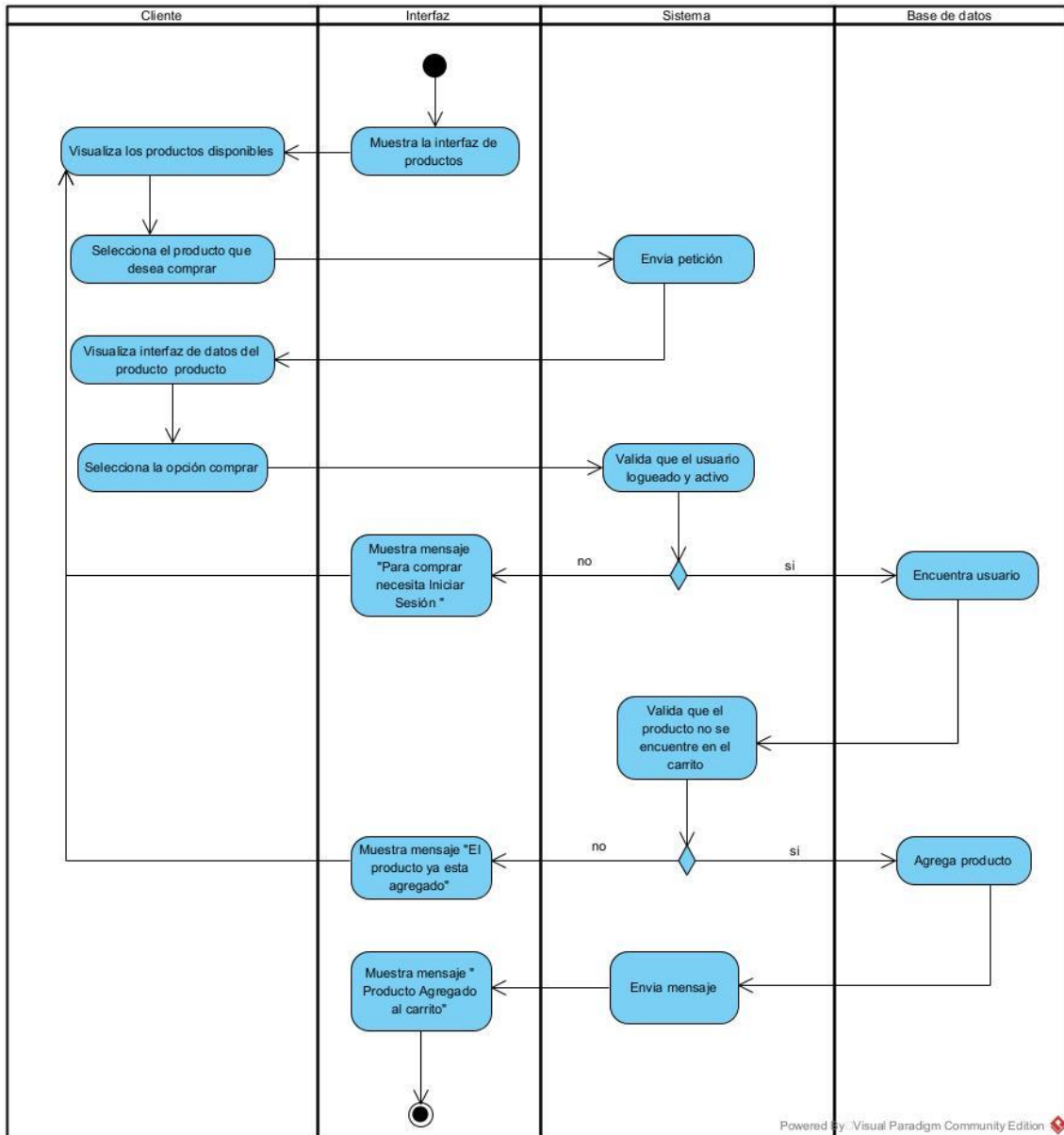


Figura 10.22: Diagrama de actividades: Agregar productos al carrito.

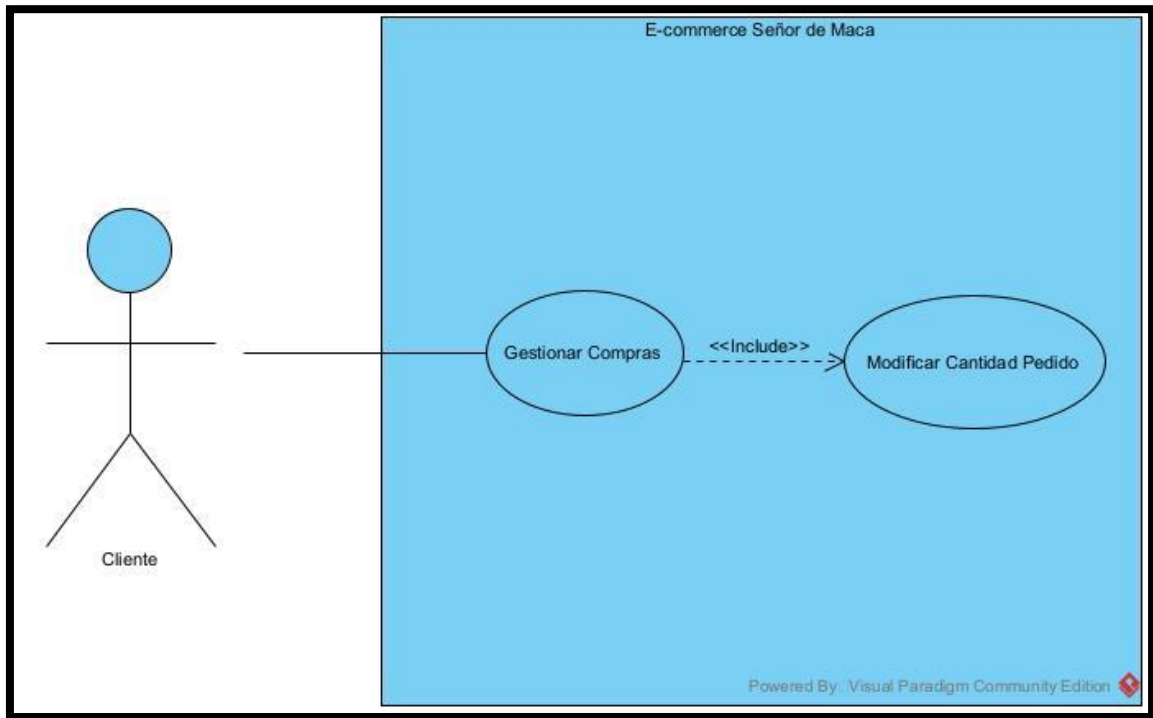


Figura 10.23: Modificar cantidad de productos pedidos.

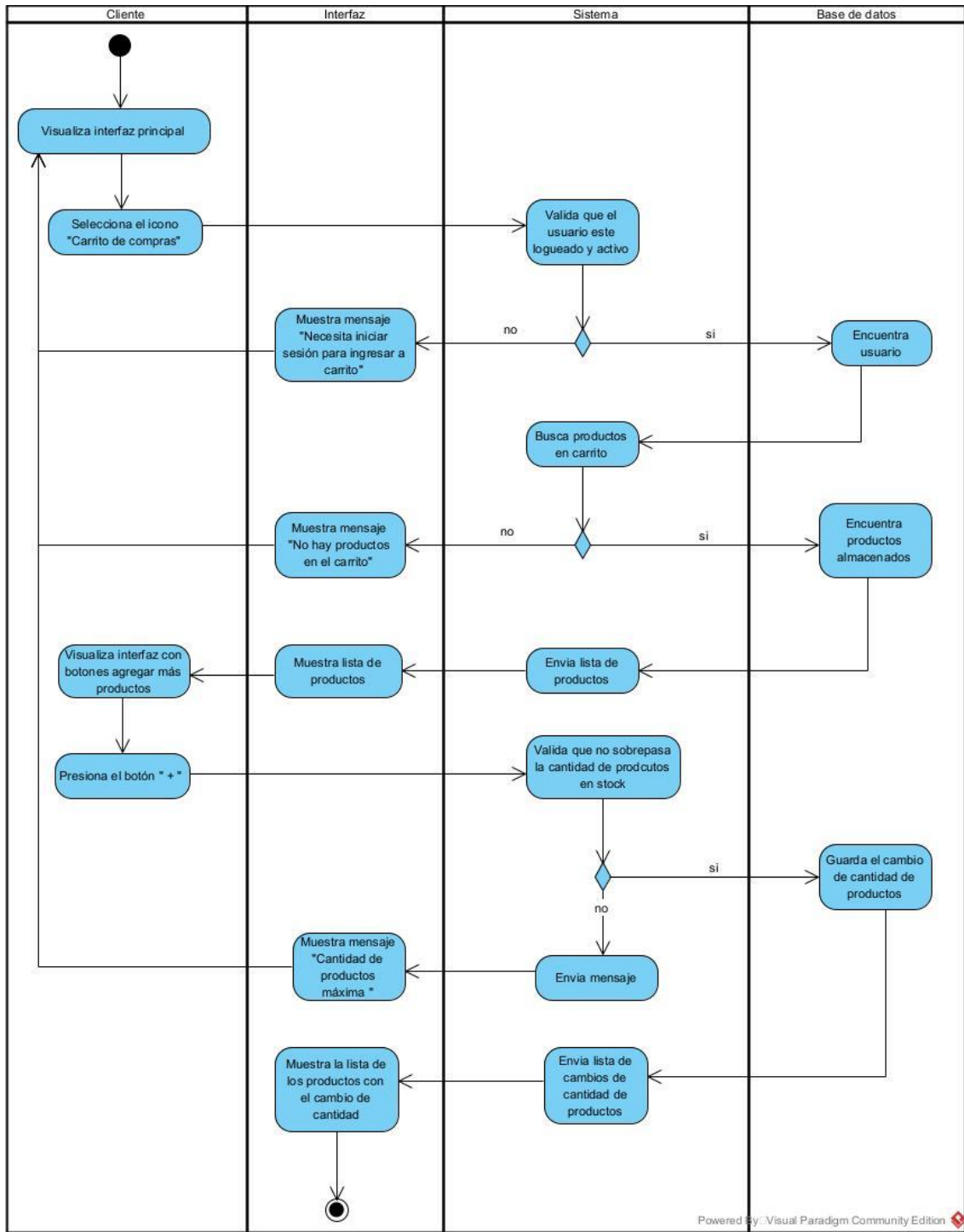


Figura 10.24: Diagrama de actividades: Modificar cantidad de productos pedidos.

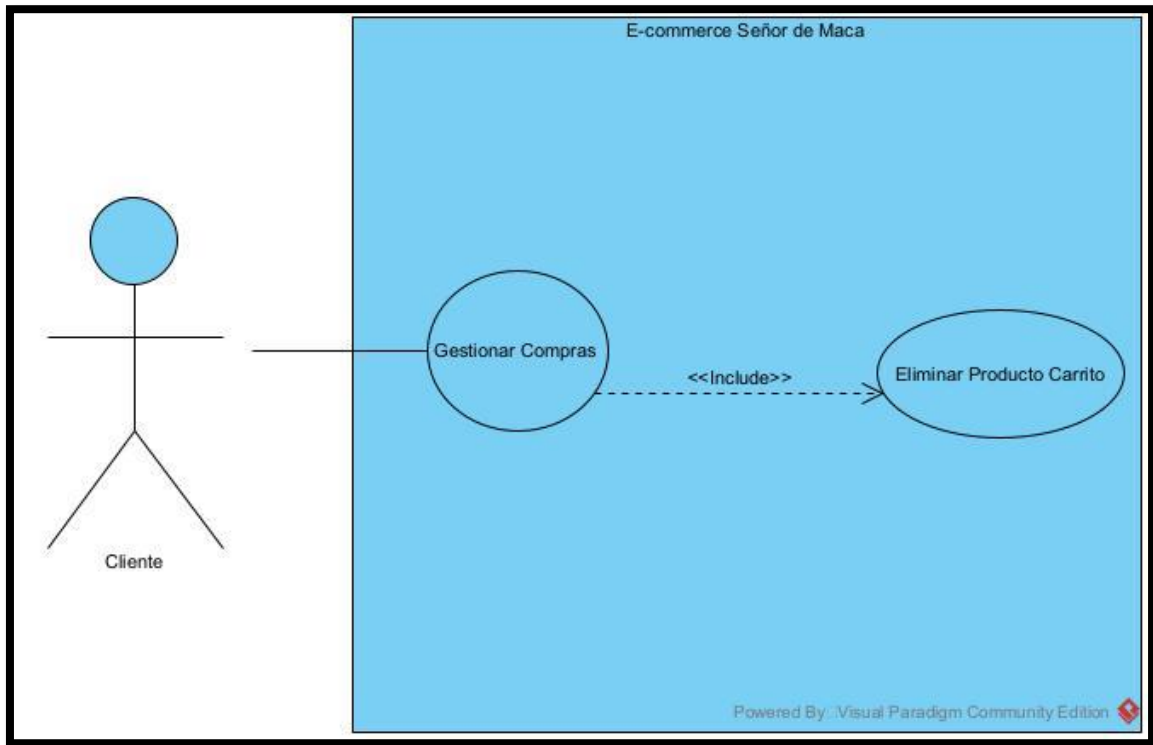


Figura 10.25: Eliminar productos del carrito

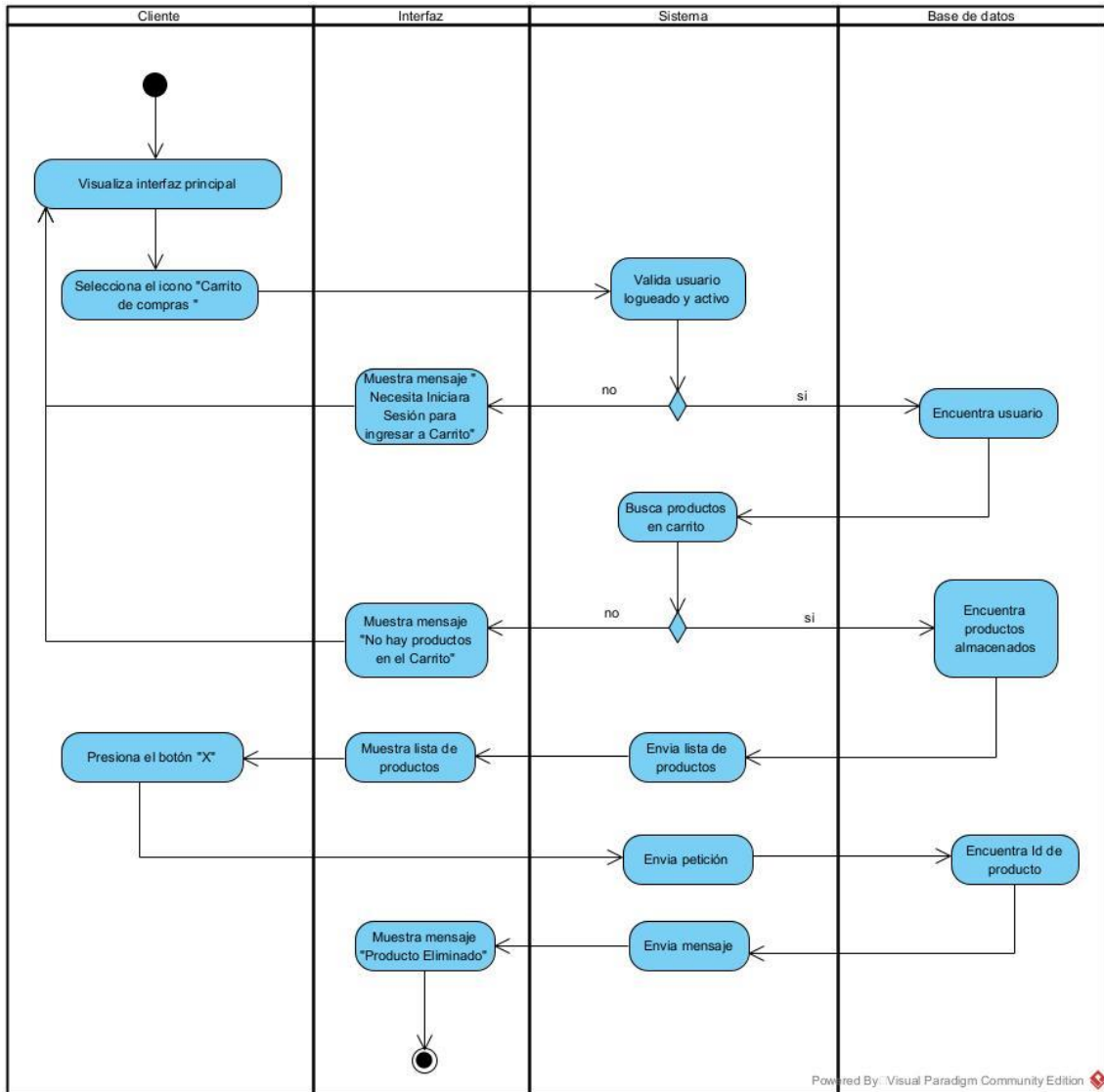


Figura 10.26: Diagrama de actividades: Eliminar productos del carrito.

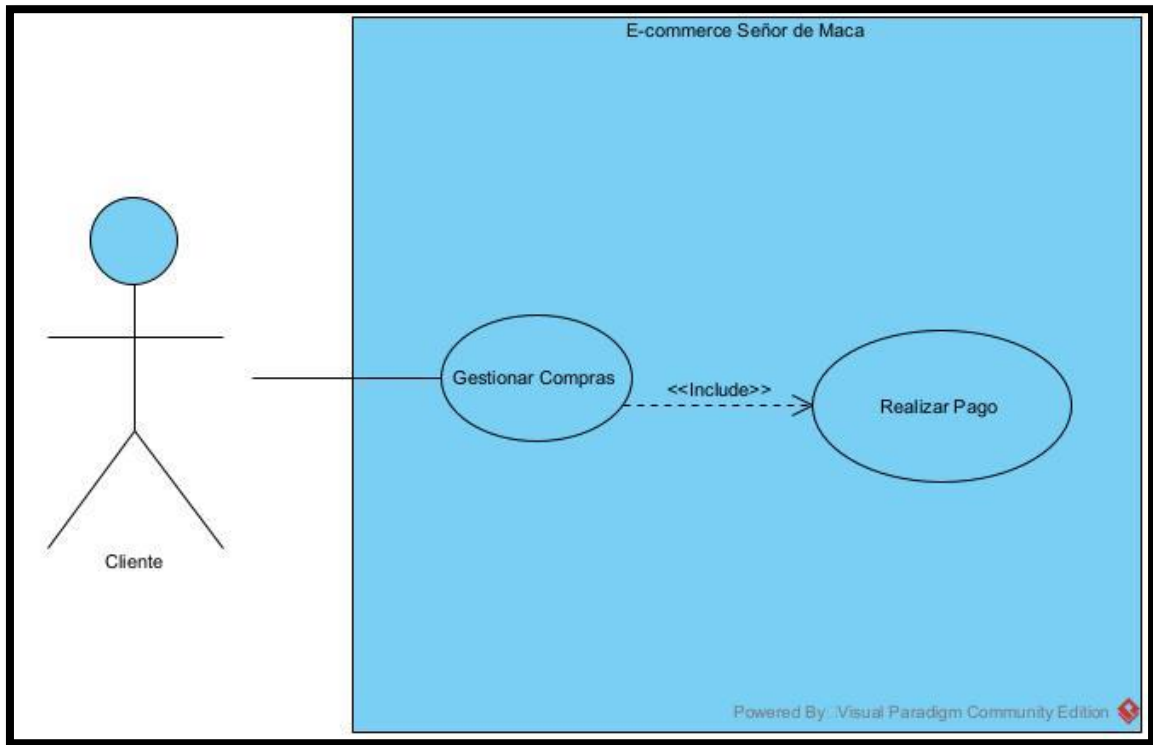


Figura 10.27: Realizar pago.

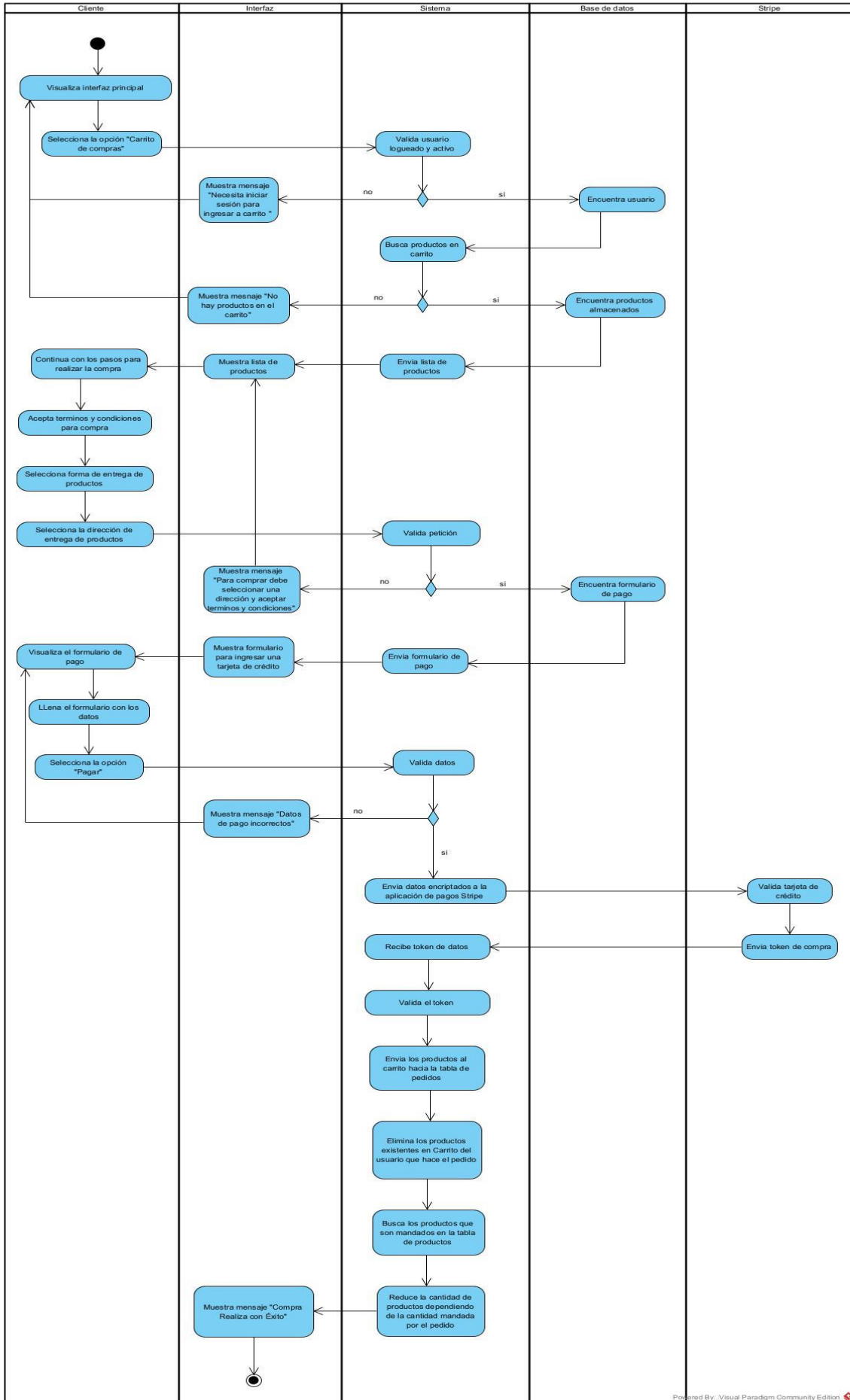


Figura 10.28: Diagrama de actividades: Realizar pago.

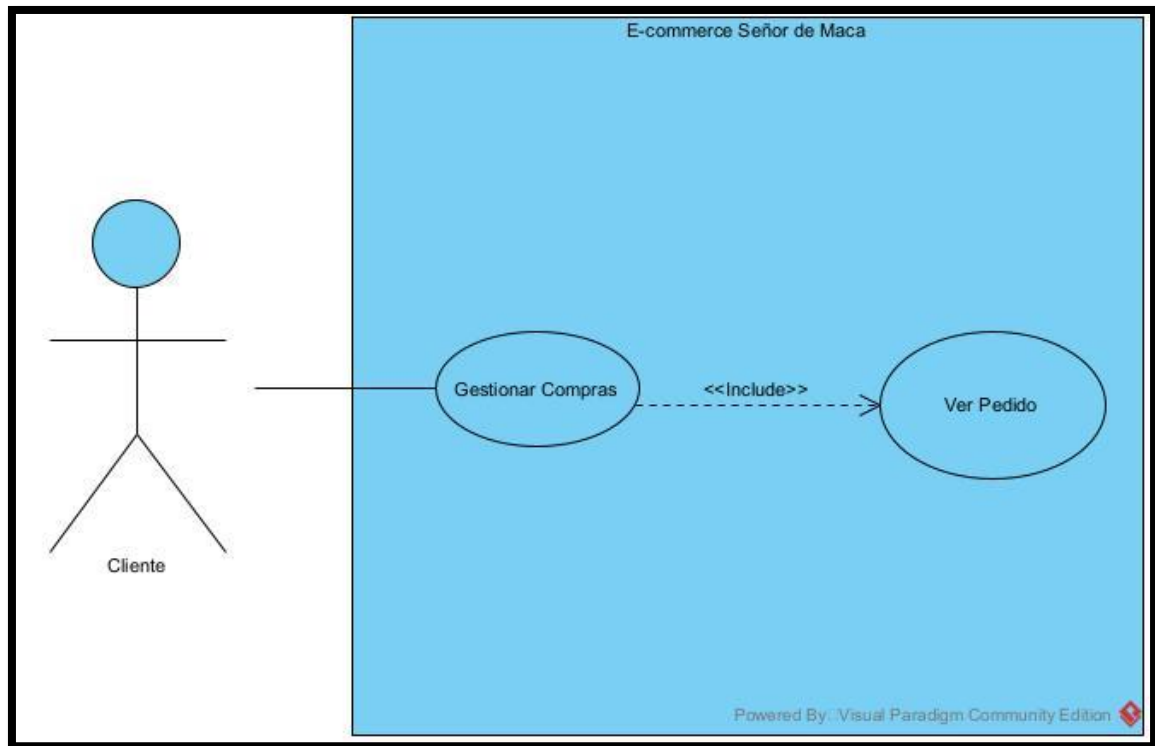


Figura 10.29: Ver pedidos

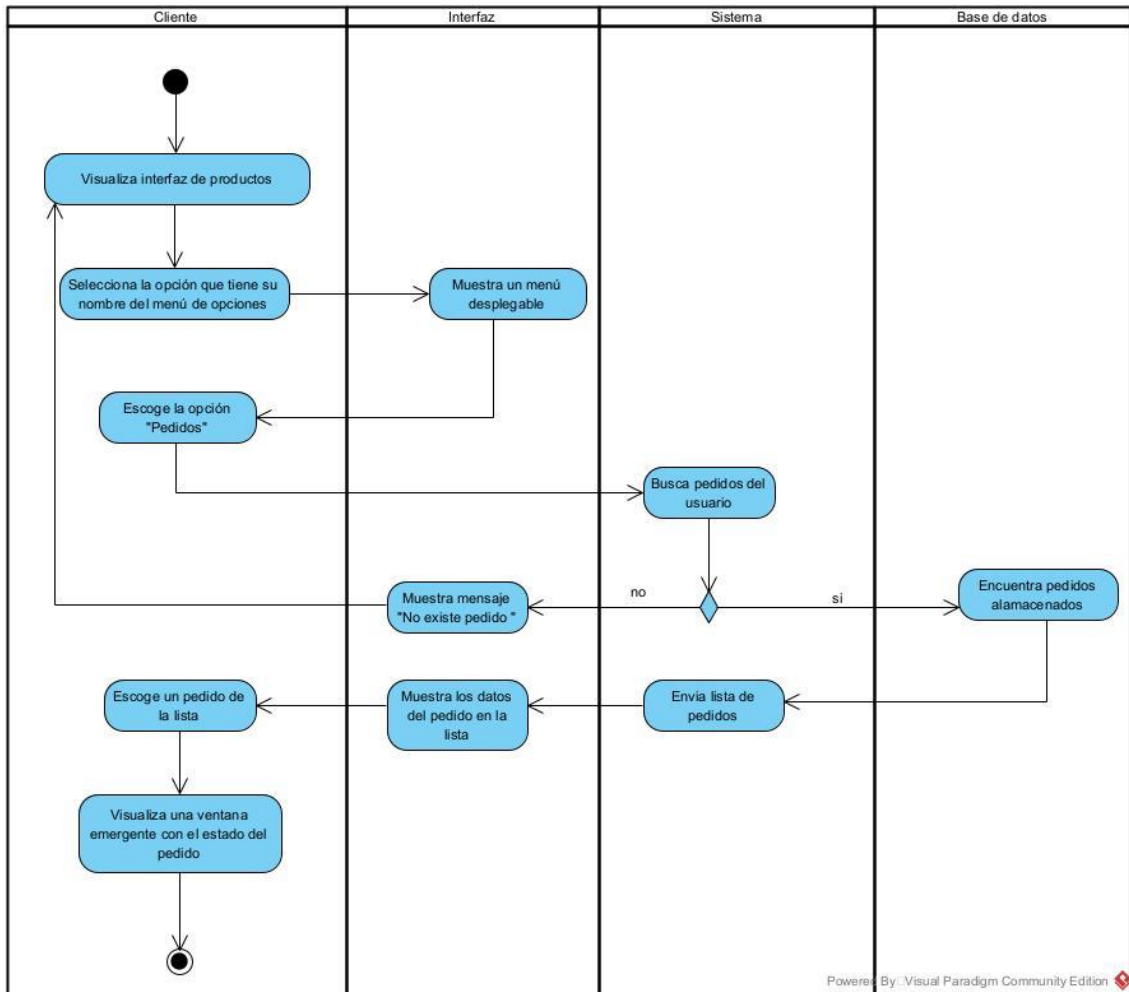


Figura 10.30: Diagrama de actividades: Ver pedidos.

10.13.3. Mensaje

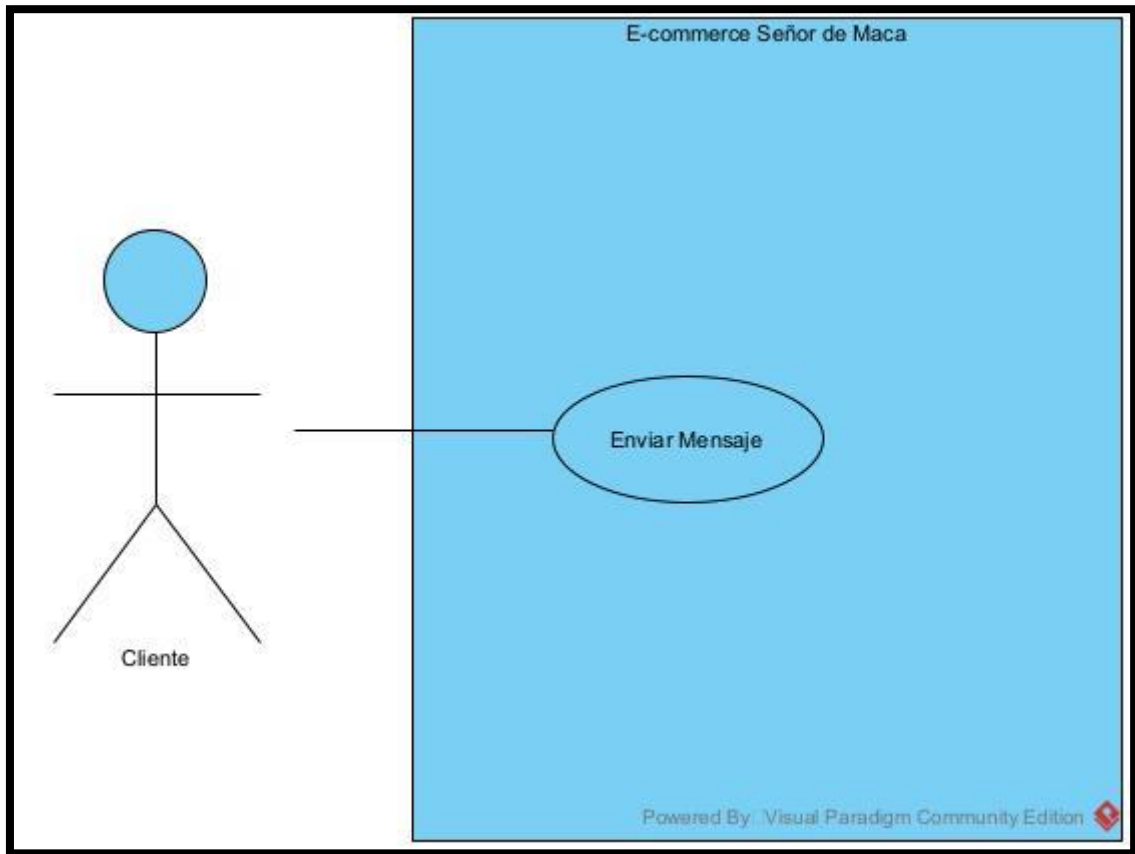


Figura 10.31: Enviar mensajes

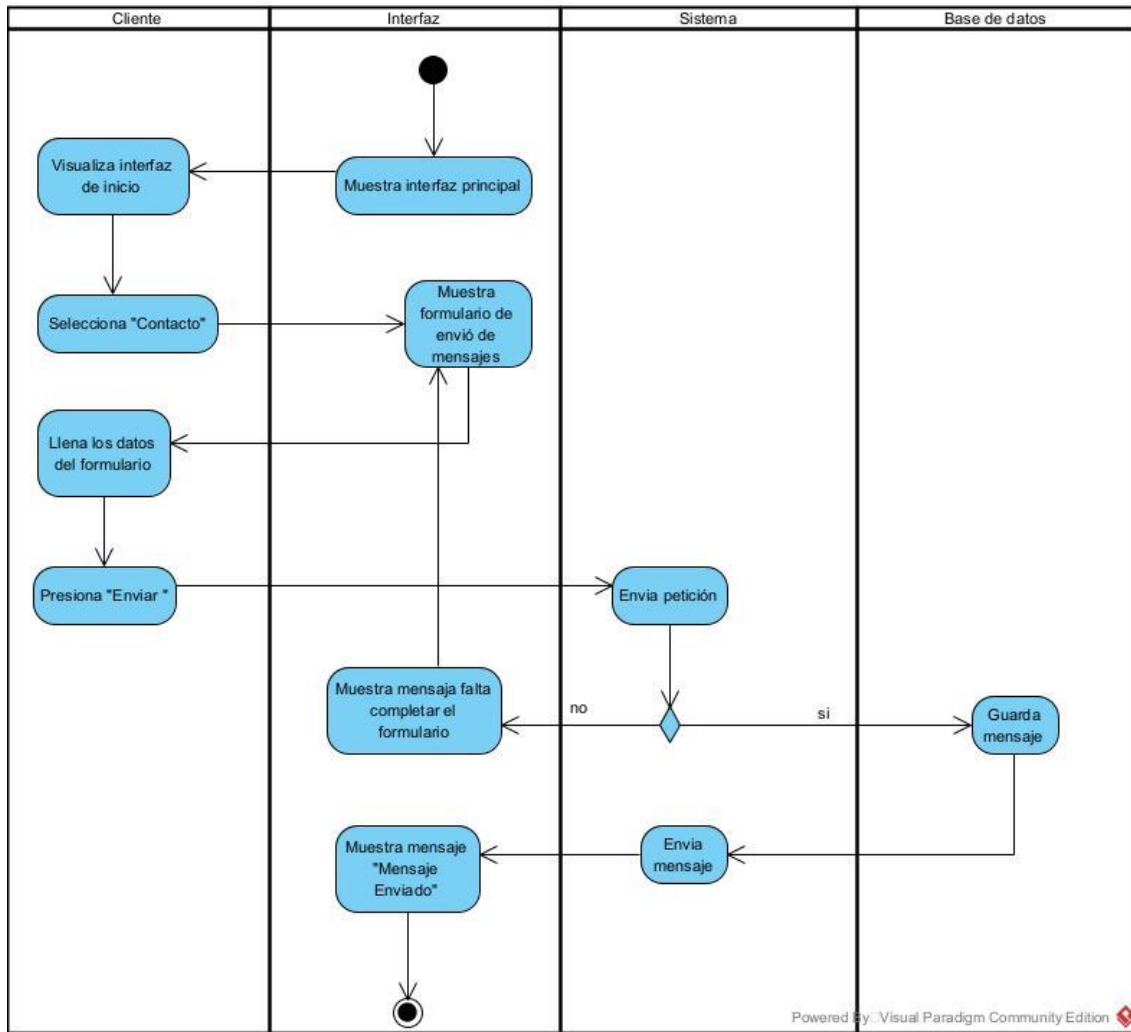


Figura 10.32: Diagrama de actividades: Enviar mensajes.

10.14. MANUAL DE USUARIO

ANEXO F: Manual de Usuario

- 1) Ingresamos a la url <https://main.d2a4kc3lt00ynz.amplifyapp.com/> en nuestro navegador de preferencia, se mostrará una pantalla de inicio como la que se muestra a continuación.

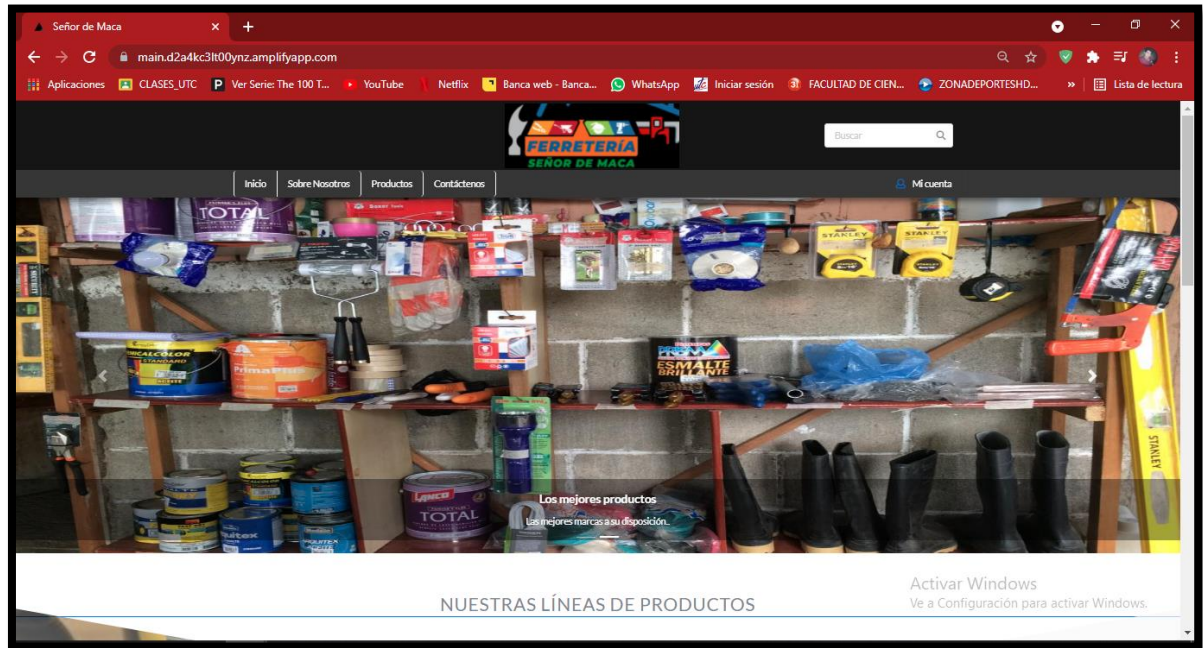


Figura 10.33. Página de inicio

En la página de inicio se podrá visualizar un menú superior por el cual se podrá navegar por los distintos módulos que cuenta la página web, entre ellos tenemos: Inicio, Sobre Nosotros, Productos, Contáctenos, Mi cuenta.

2) MÓDULO PRODUCTOS

Una vez que ingresamos al módulo productos se nos desplegará una pantalla la cual muestra las categorías de los productos que oferta la empresa.

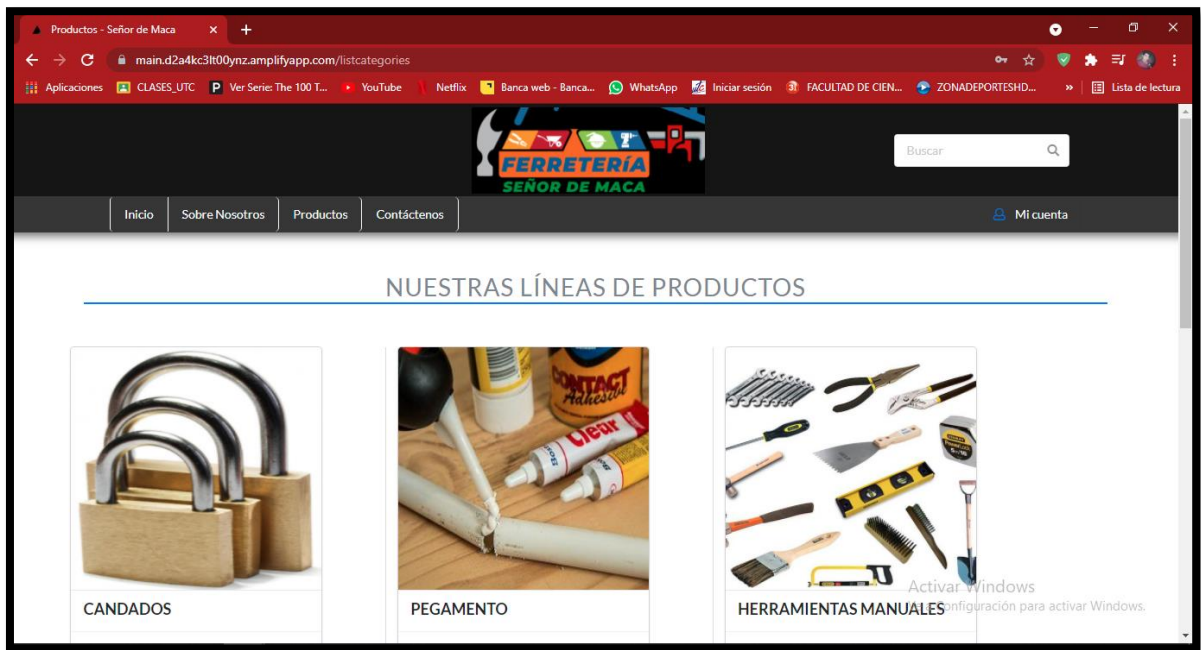
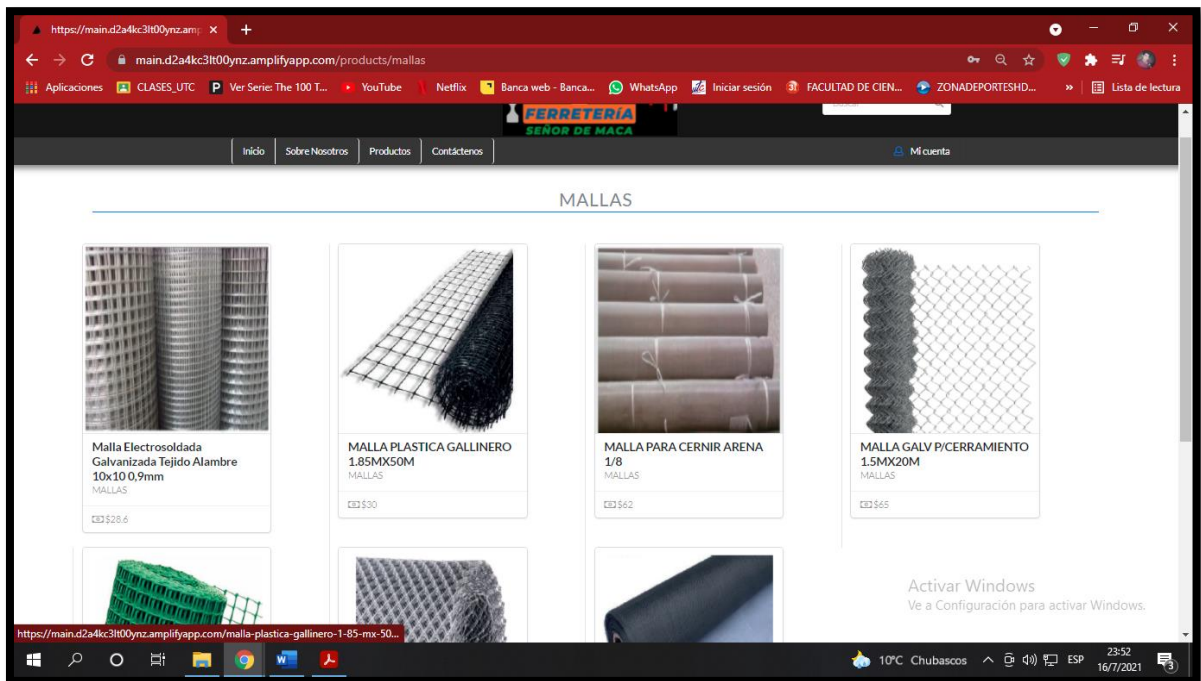


Figura 10.34. Tienda

Seleccionamos una categoría y se visualizan los productos de la misma.



Al seleccionar el producto se podrá visualizar los detalles, precio y si fuera el caso el descuento del mismo.

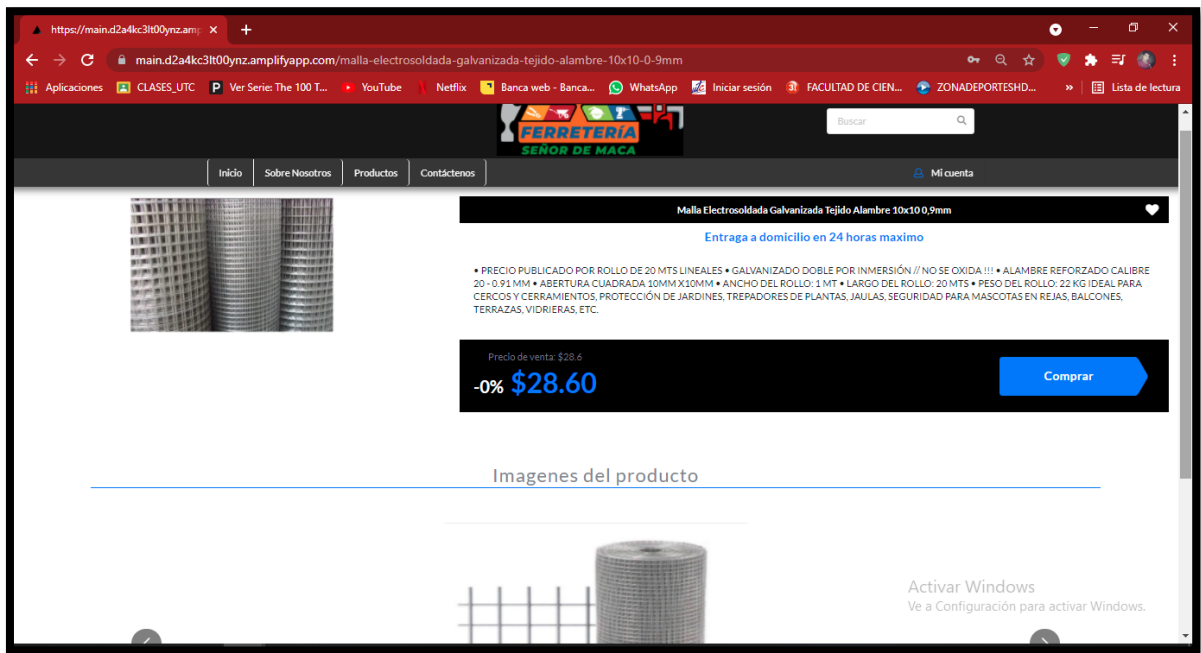


Figura 10.35. Detalles del producto.

3) REGISTRO O INICIO DE SESIÓN

Para continuar con la compra el usuario tendrá que iniciar sesión con su respectivo usuario y contraseña deberá dirigirse a la opción Mi Cuenta la cual se encuentra en el menú superior derecho de la página web, una vez ingresado se le presentará una pantalla similar a la siguiente:

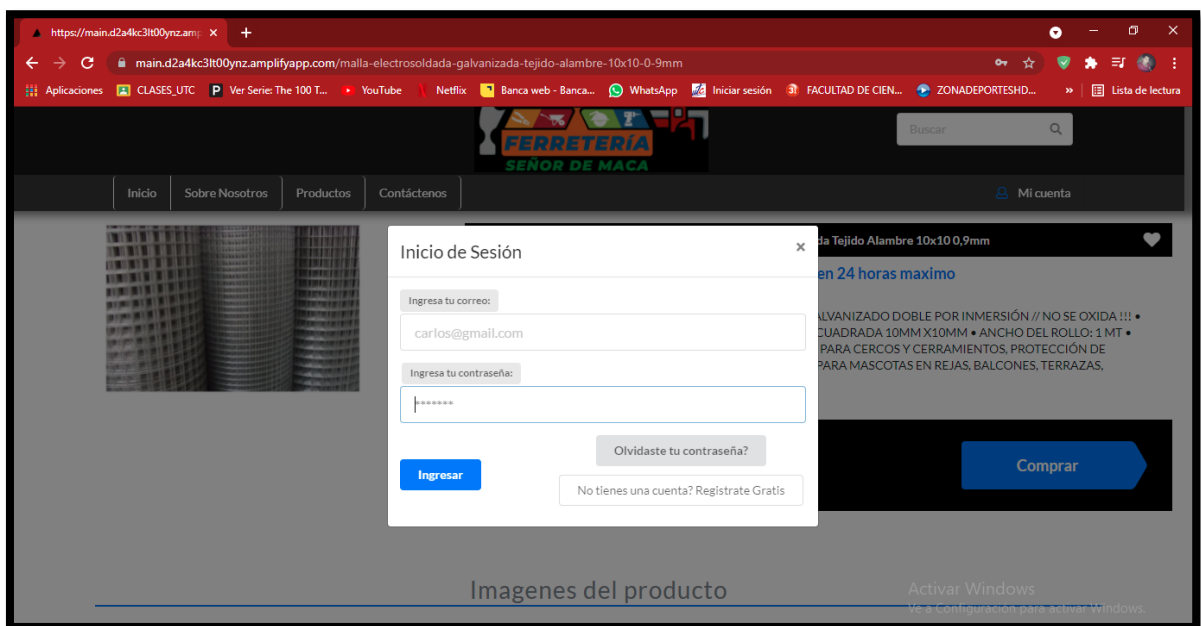


Figura 10.36. Mi cuenta

Una vez que haya ingresado a este punto deberá ingresar su correo y contraseña, si usted no se encuentra registrado seleccione la opción “¿No tienes cuenta? Regístrate gratis” y llene el formulario.

The image shows a web browser window with a registration form titled "Registro de Usuario" overlaid on a product page. The form fields are filled with the following information:

- Ingres tu nombre: Carlos
- Ingres tu apellido: Galeas
- Ingres tu UserName: Carlos20
- Ingres tu correo: carlos@gmail.com
- Ingres una contraseña: *****

At the bottom of the form, there is a blue "Registrar" button and a link that says "Tienes una cuenta? Logueate". The background shows a product page for "Malla Tejido Alambre 10x10 0,9mm" with a "Comprar" button.

Figura 10.37. Formulario de registro

Usted deberá llenar todos los campos solicitados, es importante saber que los campos deben llenarse conforme a los formatos establecidos ejemplo. Número de teléfono deberá contener solo números, el correo electrónico deberá contener una estructura @ejemplo.com y deberá ingresar una contraseña única para mantener su cuenta segura, para verificar sus datos deberá validar su cuenta mediante un correo electrónico el cual se le enviará con un código único para la validación.

En esta sección deberá validar y activar su cuenta para lo cual deberá ingresar el correo registrado y el código único el cual lo encontrará en su bandeja de entrada de su correo electrónico.

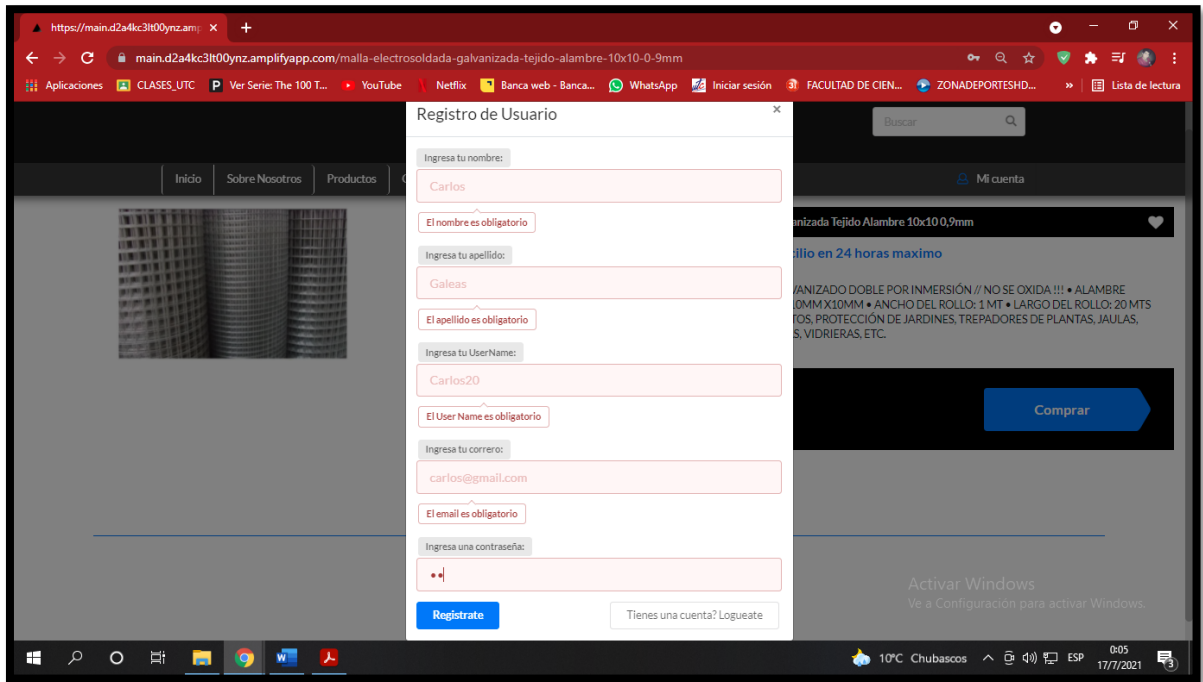
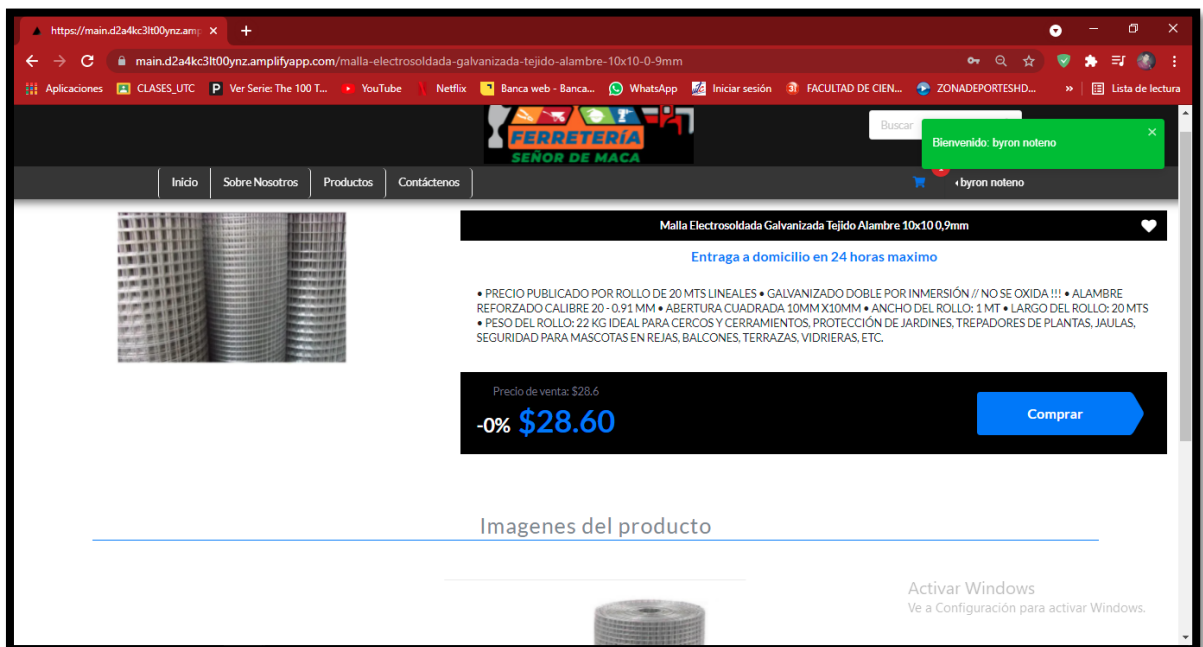


Figura 10.38. Formulario de validación

Una vez registrado el usuario podrá iniciar sesión de forma correcta con su usuario y contraseña.



4) REALIZAR UNA COMPRA

Una vez registrado e iniciado sesión podrá terminar el proceso de compra, el usuario deberá señalar el producto y este se agrega al carrito de compras.

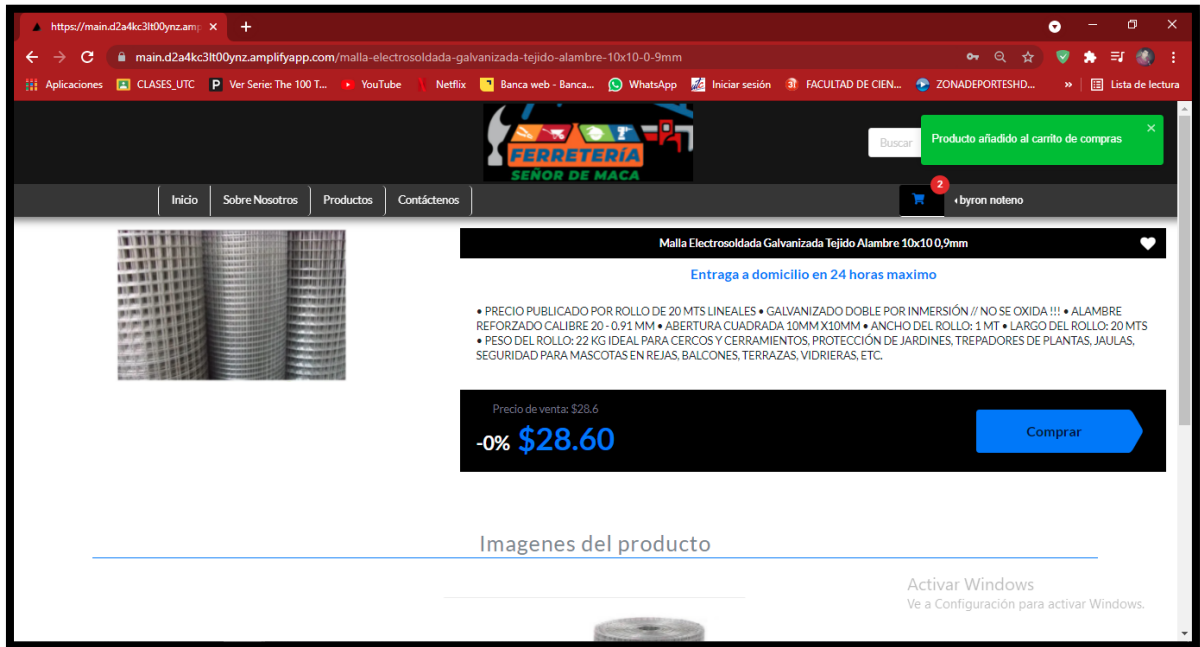


Figura 10.39. Agregar producto al carrito de compras

Una vez que el cliente selecciona todos los productos que desea comprar deberá dirigirse al carrito de compras, aquí se visualizará la lista de los productos que haya seleccionado y escoge la cantidad de cada uno de ellos.

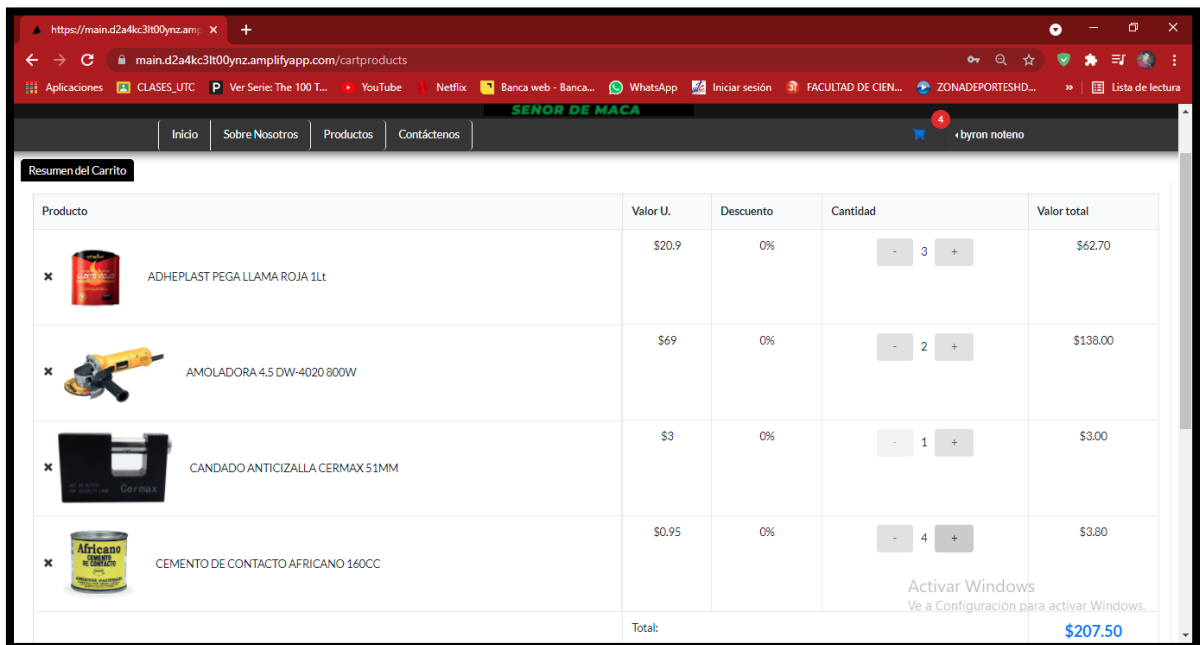


Figura 10.40. Modificar productos del carrito de compras

5) FORMA DE PAGO

El cliente deberá seleccionar la opción “Aceptar Términos y Condiciones” y la forma de entrega del producto posteriormente deberá seleccionar la forma de pago de los productos.

a. Depósito o transferencia bancaria

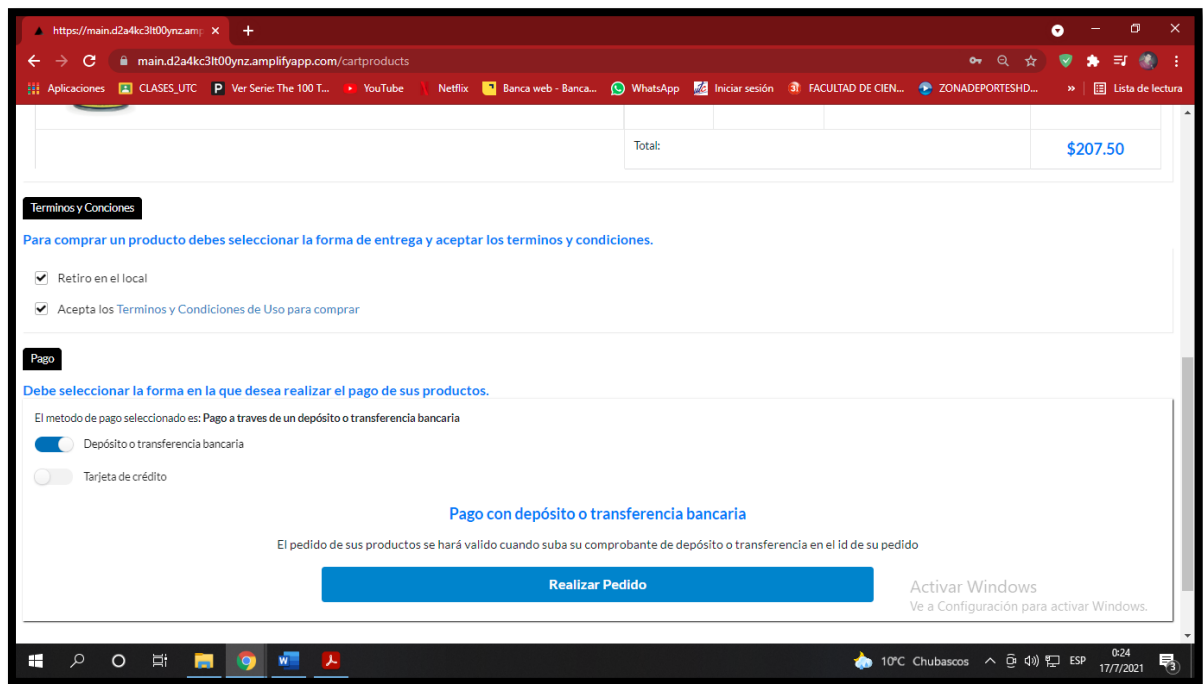


Figura 10.41: Opción de pago: Depósito o transferencia bancaria

El cliente selecciona la pestaña “Realizar Pedido”. Se dirige a la interfaz para validar el pedido en donde se detallan los productos a comprar.

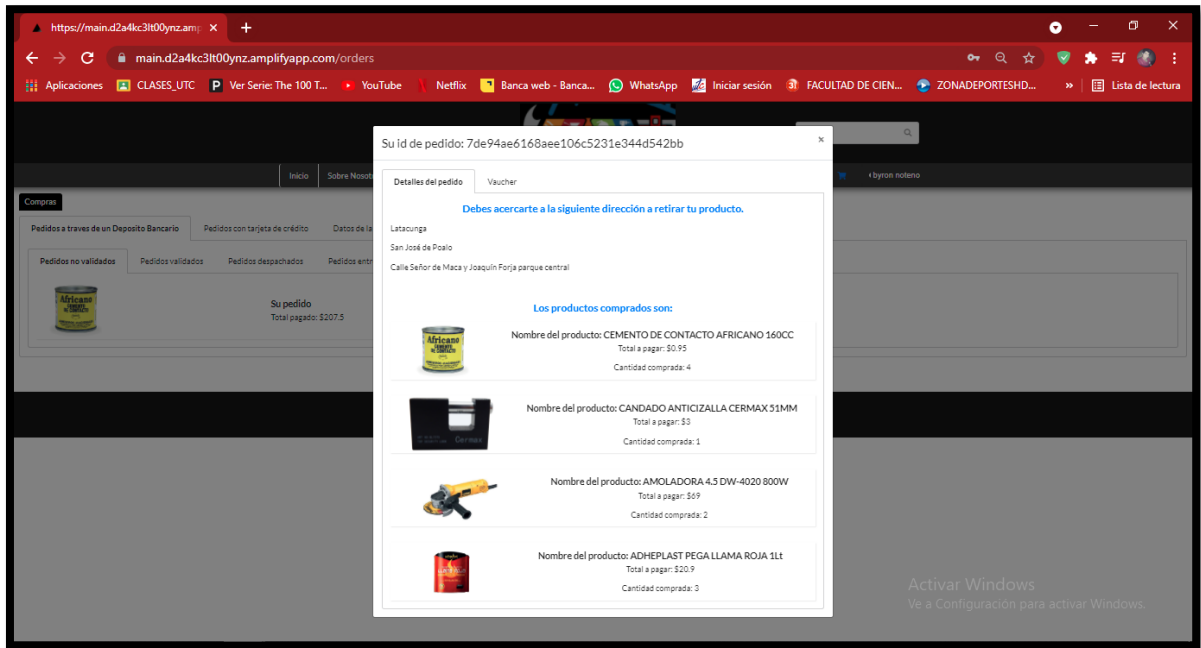


Figura 10.42. Ver detalles del pedido

Para validar el pedido debe dirigirse a la pestaña “Voucher” y subir el comprobante de transferencia o depósito.

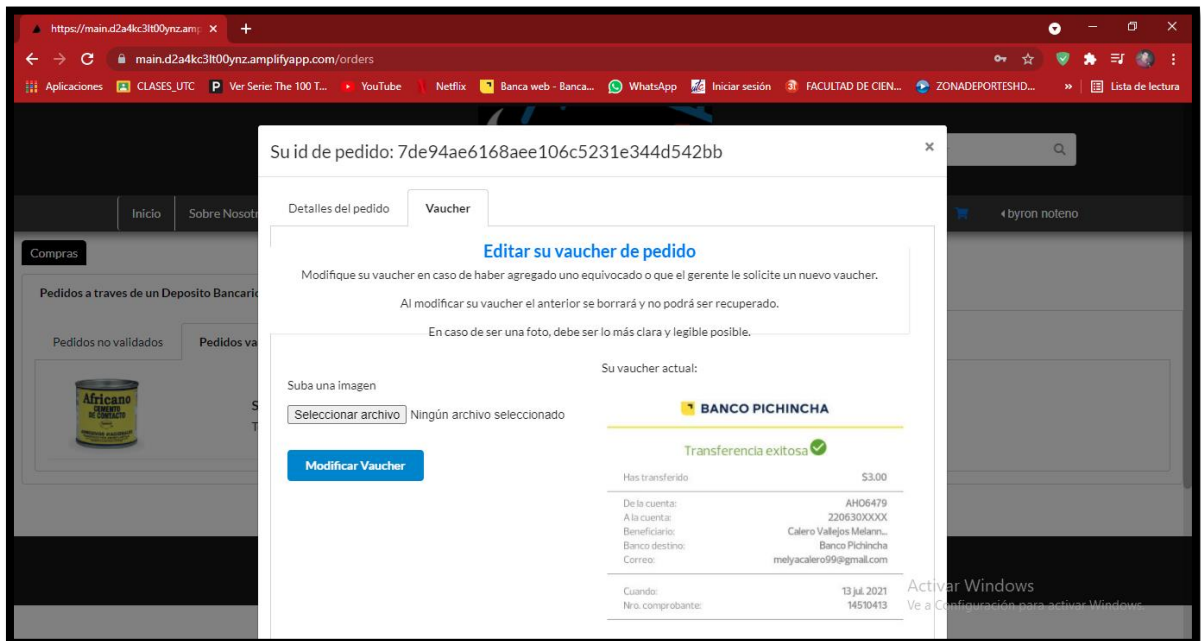


Figura 10.43. Agregar un voucher de pedido

b. Pago con tarjeta de crédito

The screenshot shows a web browser window with the URL `https://main.d2a4kc3t00ynz.amplifyapp.com/cartproducts`. The page is titled "Terminos y Condiciones" and contains the following elements:

- A heading "Terminos y Condiciones" with a sub-heading "Para comprar un producto debes seleccionar la forma de entrega y aceptar los terminos y condiciones."
- Two checked checkboxes: "Retiro en el local" and "Acepta los Terminos y Condiciones de Uso para comprar".
- A heading "Pago" with a sub-heading "Debe seleccionar la forma en la que desea realizar el pago de sus productos."
- A message: "El metodo de pago seleccionado es: Pago a traves de una tarjeta de crédito".
- Two radio buttons: "Depósito o transferencia bancaria" (unchecked) and "Tarjeta de crédito" (checked).
- A blue button labeled "Pago con tarjeta de crédito".
- A form field for "Número de tarjeta" with a "MM / AA CVC" label.
- A blue button labeled "Pagar".
- A Windows watermark: "Activar Windows Ver Configuración para activar Windows."

Figura 10.44. Pago con tarjeta de crédito.

El cliente ingresa los datos de la tarjeta para poder continuar con la compra y selecciona “Pagar”.

The screenshot shows a web browser window with the URL `https://main.d2a4kc3t00ynz.amplifyapp.com/orders`. The page is titled "Compras" and contains the following elements:

- A navigation bar with links: "Inicio", "Sobre Nosotros", "Productos", "Contáctenos".
- A search bar with the text "Buscar".
- A user profile icon labeled "byron noteno".
- A heading "Compras" with a sub-heading "Pedidos a traves de un Deposito Bancario" and "Pedidos con tarjeta de crédito".
- A sub-heading "Datos de la cuenta de deposito".
- A table with columns: "Pedidos no despachados", "Pedidos despachados", "Pedidos entregados".
- A row in the "Pedidos despachados" column with the following data:
 - Image of a roll of material.
 - Text: "Su pedido", "Total pagado: \$29.35".
 - Date and time: "17/07/2021 - 0:39".
 - A blue eye icon.
- A footer with the text: "©2021 Señor de Maca S.A. Todos los derechos reservados".
- A Windows watermark: "Activar Windows Ver Configuración para activar Windows."

Figura 10.45. Compra exitosa con tarjeta de crédito.

Una vez realizada la compra las notificaciones del estado del pedido llegarán a su correo.



Figura 10.46. Notificación al realizar el pedido

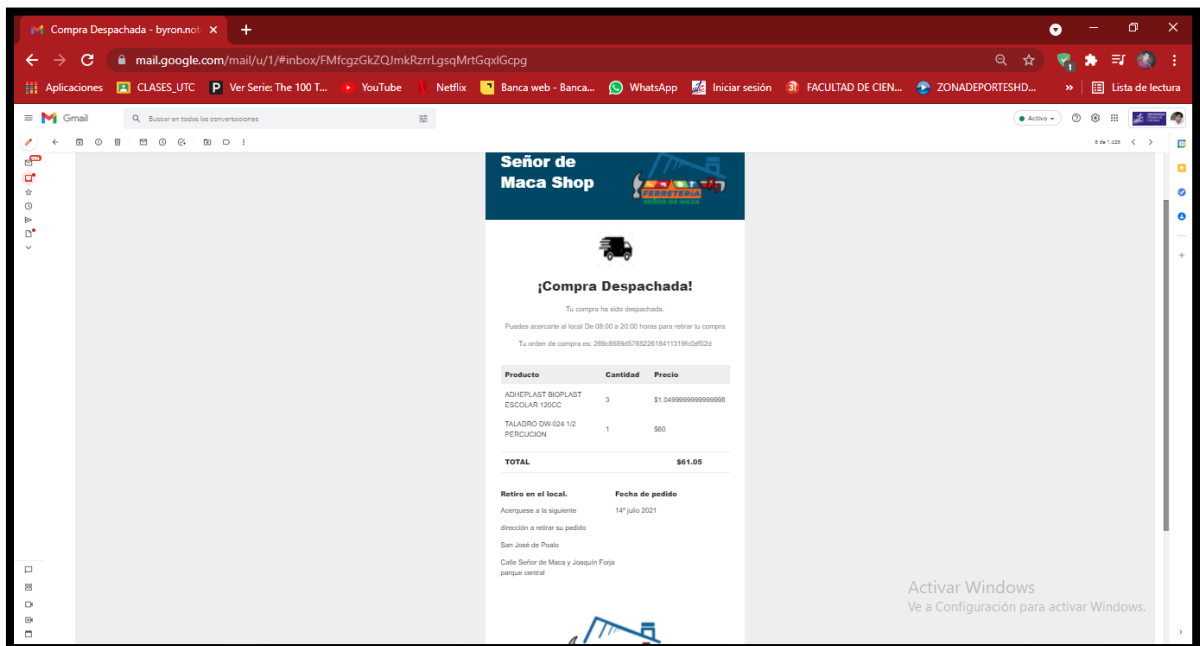


Figura 10.47. Notificación al despachar el pedido

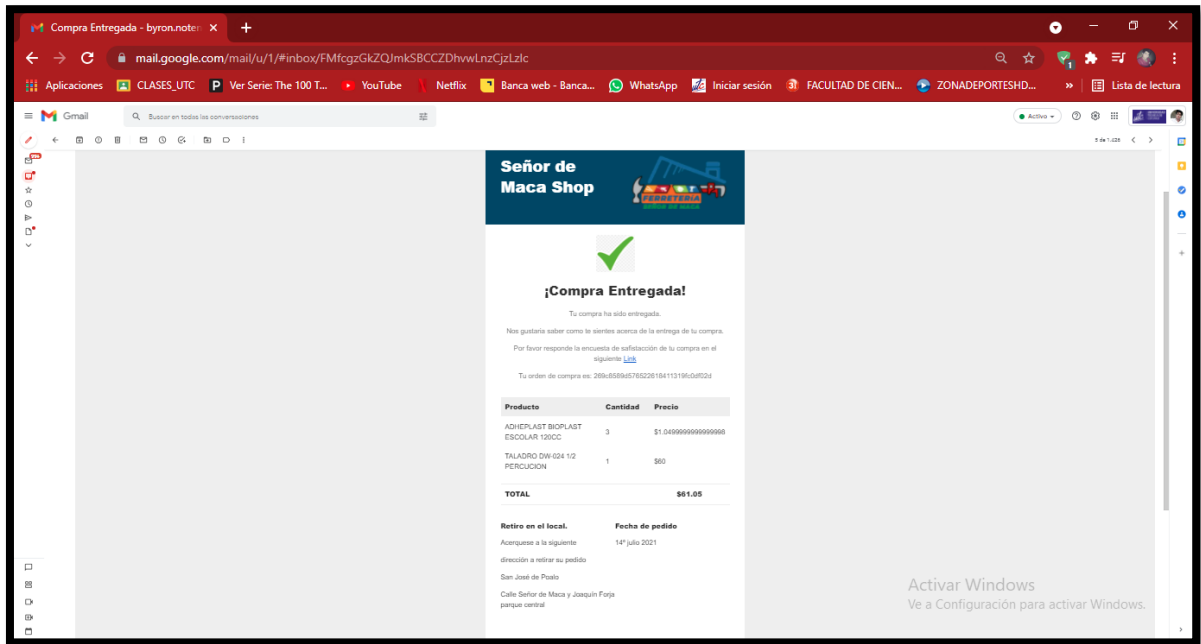


Figura 10.48. Notificación al entregar el pedido

- **Mi cuenta.** - Todos sus datos se encuentran en esta sección para que los pueda modificar en caso de ser necesarios.

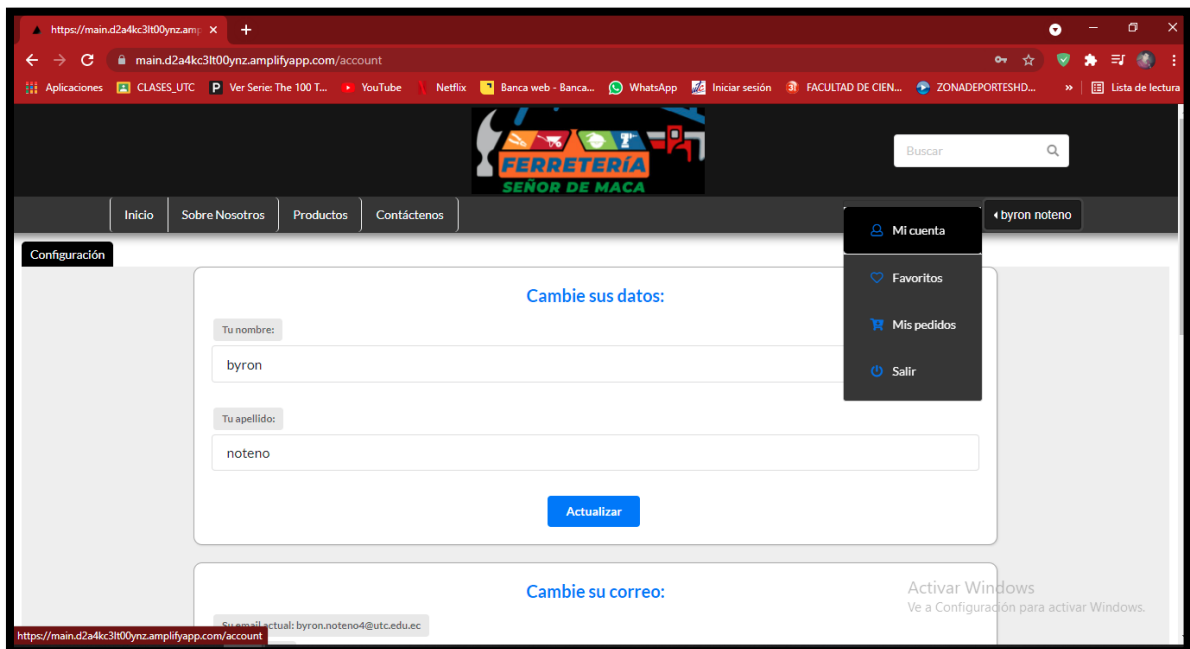


Figura 10.49. Gestión de datos personales

- **Cerrar sesión.** - Para mantener su cuenta segura es aconsejable cerrar sesión al terminar sus compras.
- **Comprar.** - Podrá realizar una nueva compra cuando lo necesite.

10.15. Análisis de las encuestas

ANEXO G: Análisis de las Encuestas y Entrevista

Pregunta 1: ¿Conoce Ud. la empresa “Señor De Maca”?

Tabla 10.15.1. Análisis pregunta 1

OPCIÓN	ENCUESTADOS	%
Si	35	87%
No	5	13%
TOTAL	40	100%

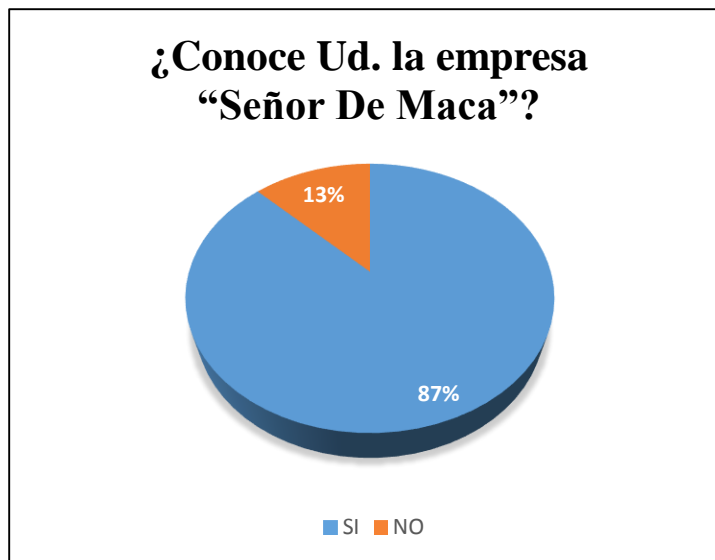


Figura 10.50: Pregunta 1

ANÁLISIS

Como se puede observar el 87% de los encuestados han afirmado que conocen la empresa no obstante el faltante correspondiente al 13% han manifestado su respuesta de manera negativa, al ser positiva la mayoría se corresponde a continuar con el proceso.

Pregunta 2: ¿Cómo adquiere sus productos?

Tabla 10.15.2: Análisis pregunta 2

OPCIÓN	ENCUESTADOS	%
Personalmente	25	62%
Encomienda	15	38%
En línea	0	0%
TOTAL	40	100%

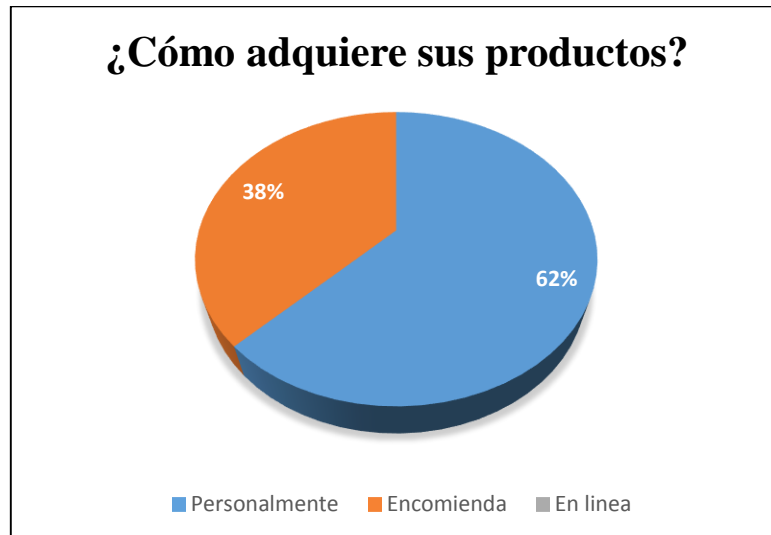


Figura 10.51: Pregunta 2

ANÁLISIS

El resultado a esta interrogante nos arroja un 62% de los encuestados que prefieren adquirir sus productos de manera presencial dejando un 38% que optan por la vía de la encomienda, dejando nula la preferencia de en línea.

Pregunta 3: ¿Le gustaría facilitar sus compras?

Tabla 10.15.3: Análisis pregunta 3

OPCIÓN	ENCUESTADOS	%
Si	40	100%
No	0	0%
TOTAL	40	100%



Figura 10.52: Pregunta 3

ANÁLISIS

De acuerdo con el resultado se muestra de manera unánime la afirmación o aceptación del 100% de encuestados en poder facilitar el proceso de sus compras.

Pregunta 4: ¿Con que frecuencia Ud. utiliza páginas web o aplicaciones móviles en su vida diaria?

Tabla 10.15.4: Análisis pregunta 4

OPCIÓN	ENCUESTADOS	%
Siempre	19	47%
Casi siempre	7	18%
A veces	10	25%
Casi nunca	4	10%
Nunca	0	0%
TOTAL	40	100%

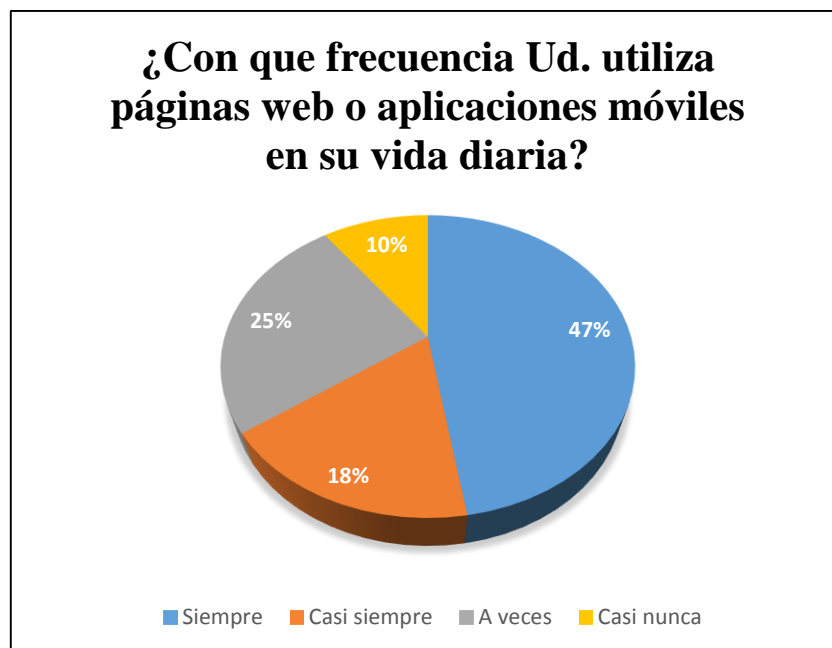


Figura 10.53: Pregunta 4

ANÁLISIS

El resultado de las encuestas realizadas a 40 personas, el 47% mencionan que siempre utilizan páginas web o aplicaciones móviles, en cambio un 25% a veces lo hacen, así también un 18% manipulan de manera regular “casi siempre”, dejando un 10% que su uso es de forma irregular “casi nunca”.

Pregunta 5: ¿Usted cree que la tecnología facilita las actividades de las personas?

Tabla 10.15.5: Análisis pregunta 5

OPCIÓN	ENCUESTADOS	%
Si	30	75%
No	1	2%
Tal vez	9	23%
TOTAL	40	100%

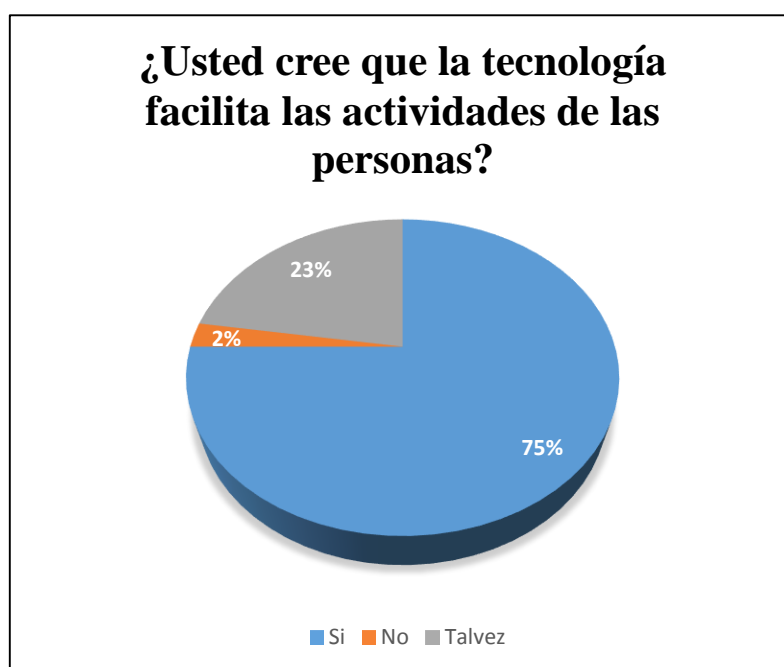


Figura 10.54: Pregunta 5

ANÁLISIS

Como resultado a esta incógnita se obtiene un 75% correspondiente a 30 encuestados que están de acuerdo en que la tecnología facilita las actividades, no obstante, un 23% de encuestados se muestran dudosos, dejando al final un 2% que están en completo desacuerdo.

Pregunta 6: ¿Conoce algún sistema de compras?

Tabla 10.15.6: Análisis pregunta 6

OPCIÓN	ENCUESTADOS	%
Si	22	55%
No	18	45%
TOTAL	40	100%

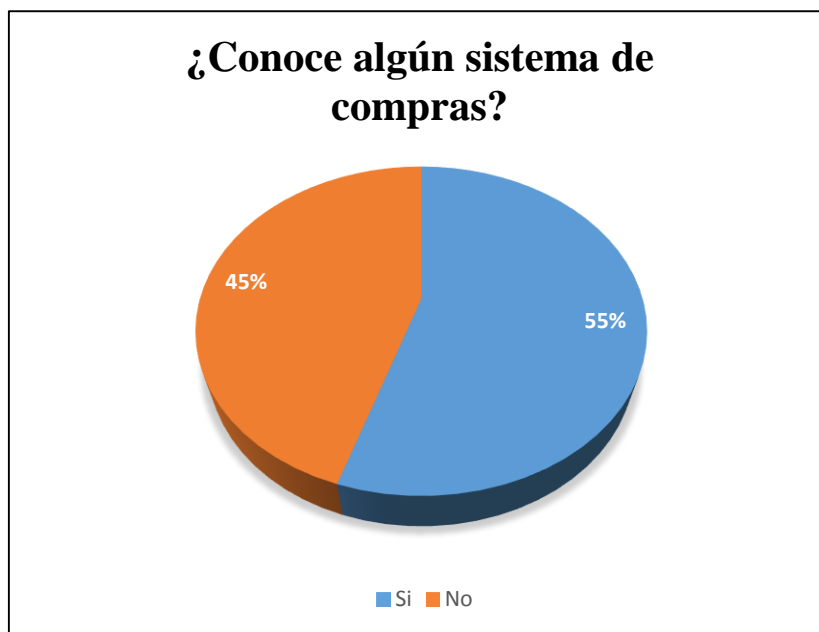


Figura 10.55: Pregunta 6

ANÁLISIS

Con un pequeño margen de diferencia se puede observar que un 55% de los encuestados conocen algún sistema de compras, sin embargo, de manera negativa hubo un 45% que tiene un desconocimiento sobre el tema.

Pregunta 7: ¿Le gustaría realizar sus comprar en línea?

Tabla 10.15.7: Análisis pregunta 7

OPCIÓN	ENCUESTADOS	%
Si	25	62%
No	2	5%
Tal vez	13	33%
TOTAL	40	100%

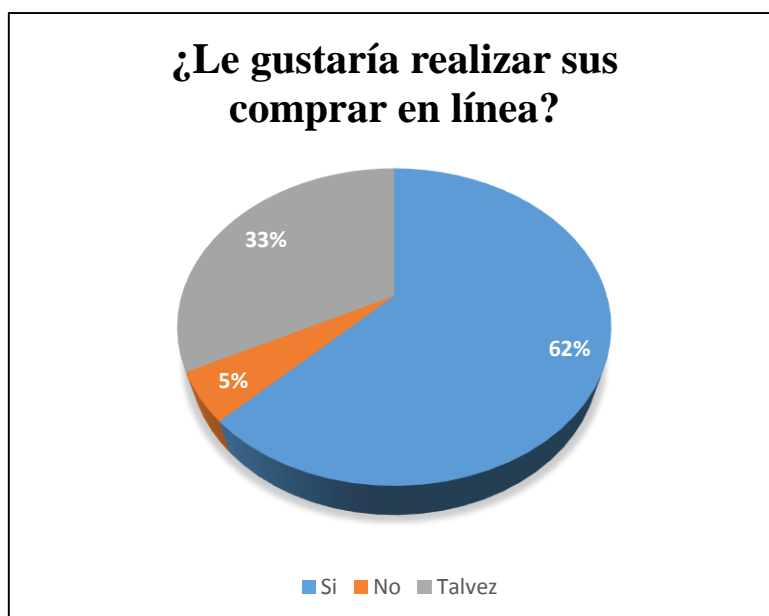


Figura 10.56: Pregunta 7

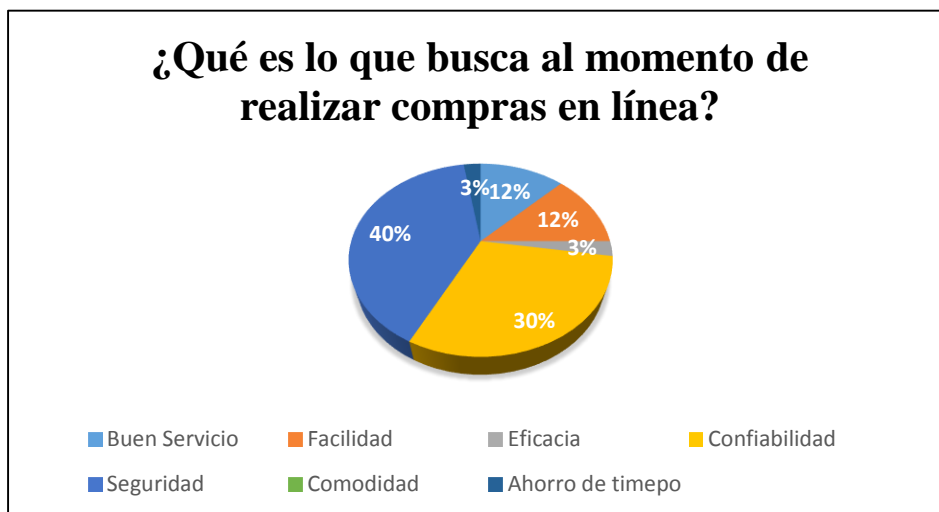
ANÁLISIS

Se logra observar que existe una gran parte de los encuestados que les gustaría realizar sus compras en línea correspondientes al 62%, así también un 33% se manifiestan de manera insegura y un 5% lo desaprueban.

Pregunta 8: ¿Qué es lo que busca al momento de realizar compras en línea?

Tabla 10.15.8: Análisis pregunta 8

OPCIÓN	ENCUESTADOS	%
Buen Servicio	5	12%
Facilidad	5	12%
Eficacia	1	3%
Confiabilidad	12	30%
Seguridad	16	40%
Comodidad	0	0%
Ahorro de tiempo	1	3%
TOTAL	40	98%



Figura

10.57: Pregunta 8

ANÁLISIS

En esta interrogante se muestra un pequeño margen de diferencia con respecto a lo que se busca al momento de realizar las compras en línea dejando un 40% que escogen seguridad y un 30% por confiabilidad, sin embargo, un empate con el 12% que estarían optando por facilidad y buen servicio, y así mismo un 3% de igual manera empate que

preferieron ahorro de tiempo y eficacia, demostrando así que existe una división sobre el tema.

Pregunta 9: ¿Cree usted que un sistema de compras online ayuda a los ciudadanos?

Tabla 10.15.9: Análisis pregunta 9

OPCIÓN	ENCUESTADOS	%
Si	40	100%
No	0	0%
TOTAL	40	100%



Figura 10.58: Pregunta 9

ANÁLISIS

De acuerdo a los encuestados el 100% manifiesta su conformidad y apoyo a que un sistema de compras en línea ayudaría a los ciudadanos, demostrando de esta manera que nuestro proyecto es factible.

Pregunta 10: ¿Utilizaría la aplicación web y móvil de la empresa “Señor De Maca” para adquirir sus productos?

Tabla 10.15.10: Análisis pregunta 10

OPCIÓN	ENCUESTADOS	%
Si	29	72%
No	0	0%
Tal vez	11	28%
TOTAL	40	100%

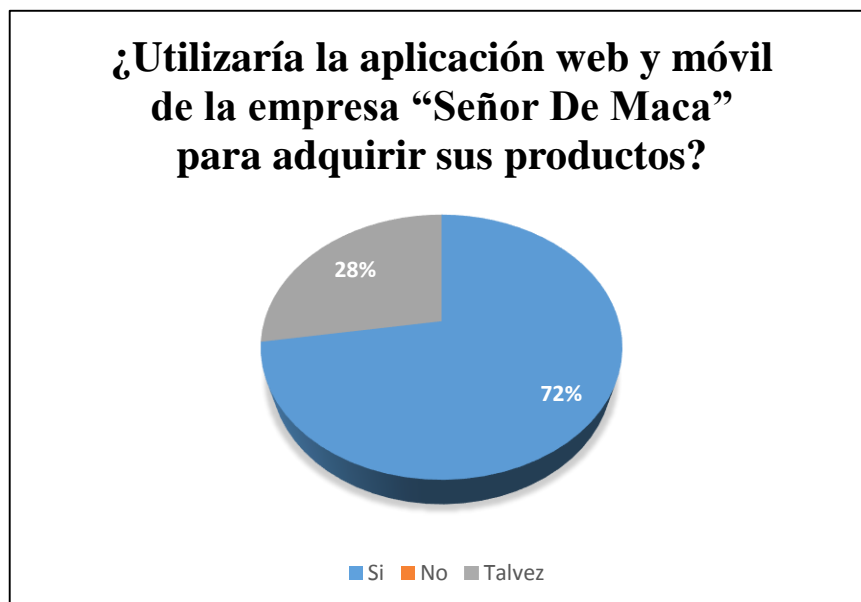


Figura 10.59: Pregunta 10

ANÁLISIS

Como se puede observar el 72% de los encuestados estarían dispuestos en usar la aplicación de la empresa “Señor De Maca” para realizar sus compras, no obstante, se aprecia que existe un 28% que se manifiestan de carácter dudoso.

10.16. IMÁGENES RELEVANTES DEL DOCUMENTO

ANEXO H: Imágenes relevantes de la aplicación

10.16.1. Funcionalidades de la aplicación web

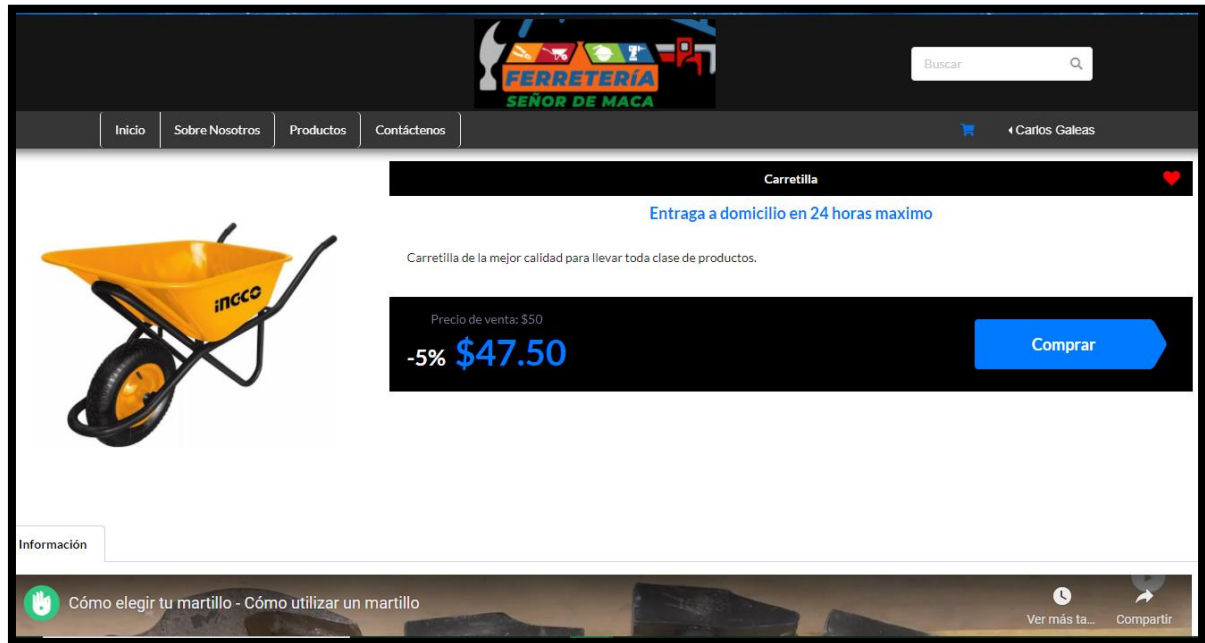


Figura 10.60: Interfaz de Producto



Figura 10.61: Interfaz de categorías de productos

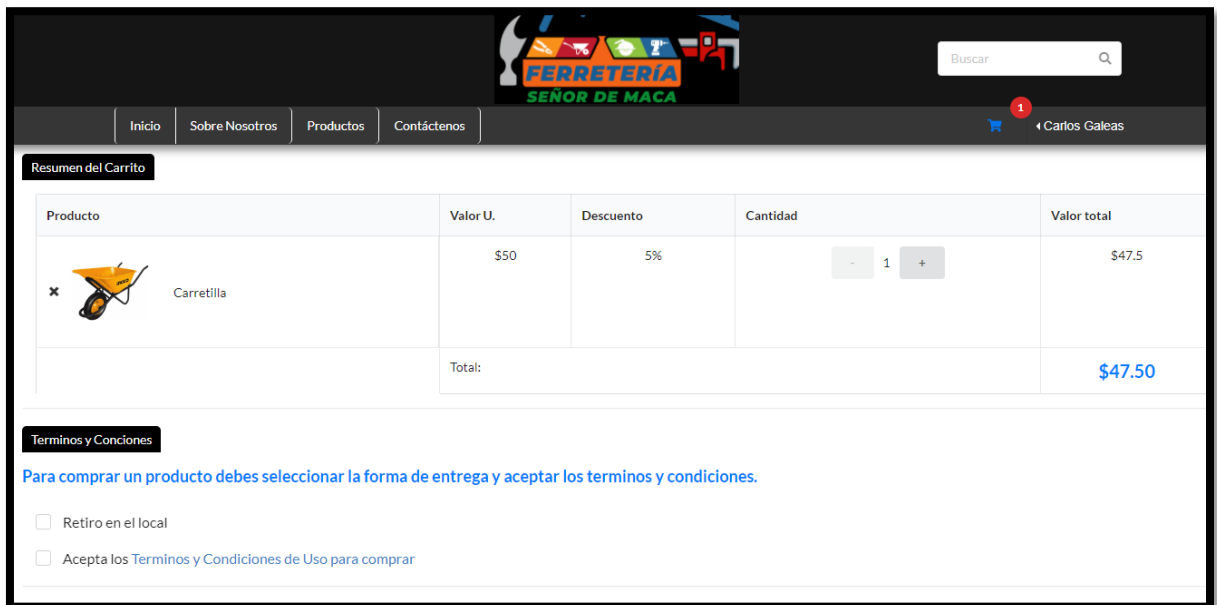


Figura 10.62: Interfaz de carrito de compras

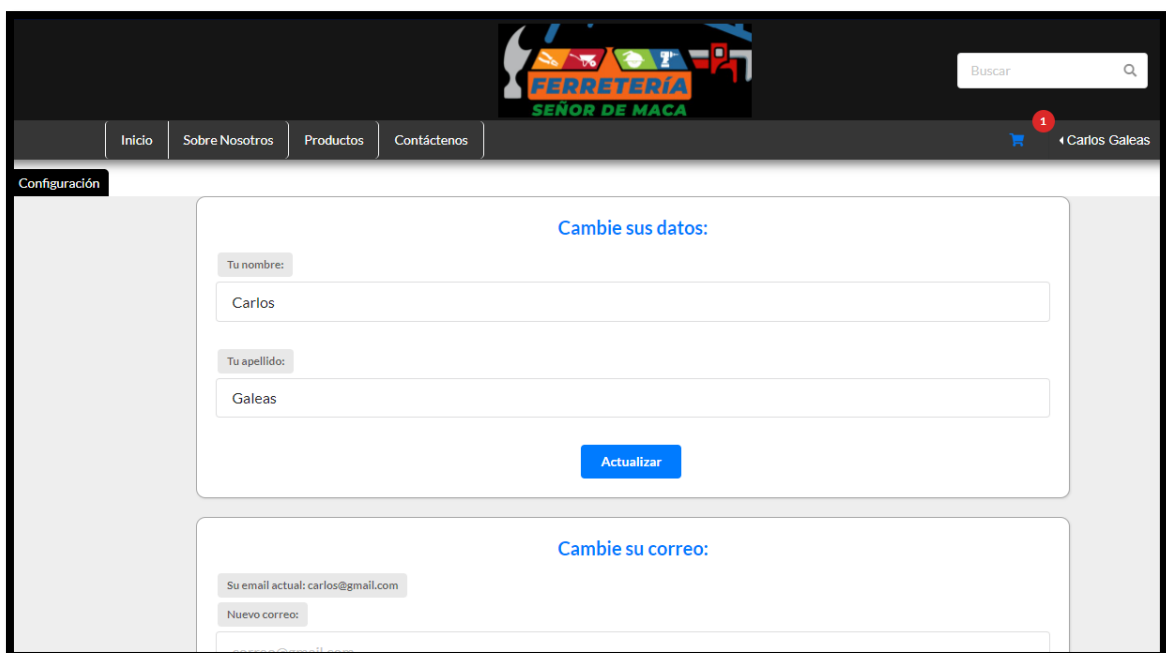


Figura 10.63: Interfaz de datos del cliente

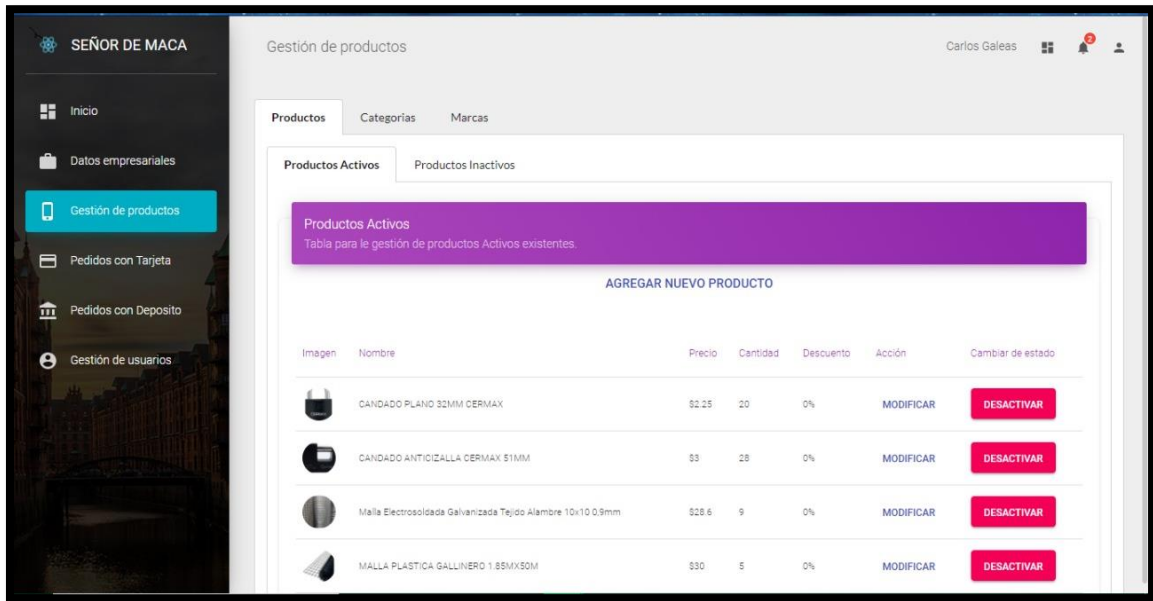


Figura 10.64: Interfaz productos Administrador (Productos)

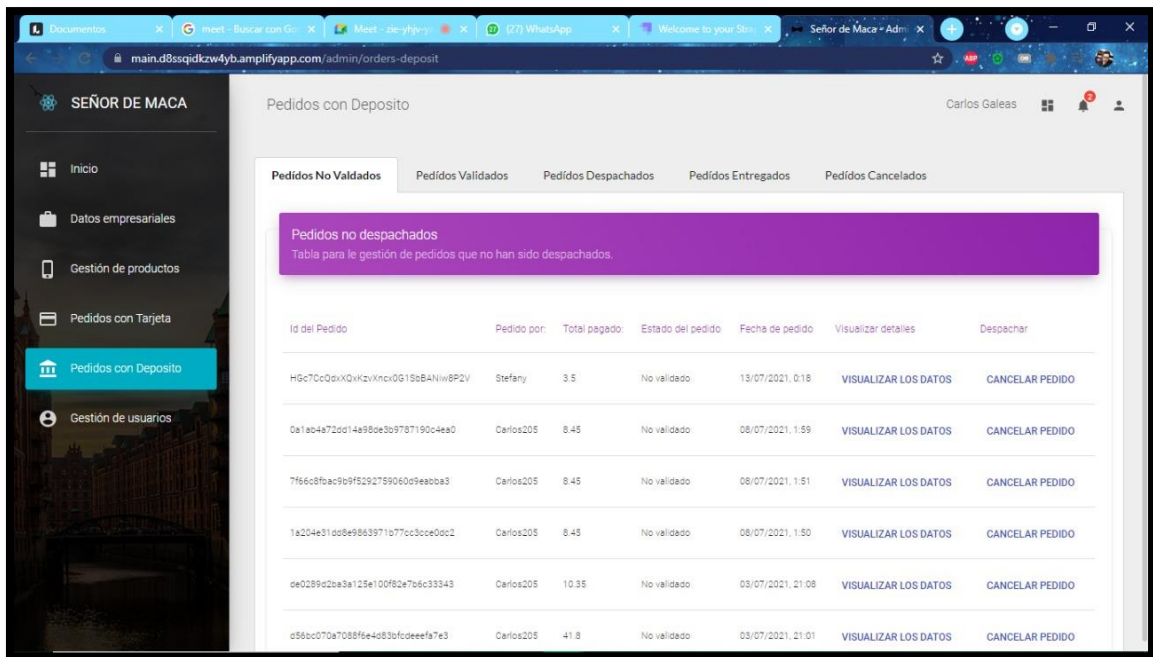


Figura 10.65: Interfaz productos Administrador (Pedidos)

10.16.2. Código de la aplicación.

```
const [loading, setLoading] = useState(false);

const { login } = useAuth();

const initialValues = {
  identifier: '',
  password: ''
};

const validations = () => ({
  identifier: Yup.string().email().required('El email es obligatorio'),
  password: Yup.string().required('La contraseña es obligatorio')
});

const {handleSubmit, handleChange, errors, setErrors, values} = useFormik({
  initialValues,
  validationSchema: Yup.object(validations()),
  onSubmit: async(formData) =>{
    setLoading(true);
    const response = await loginUser(formData);
    if(response?.jwt){
      login(response.jwt);
      toast.success(`Bienvenido: ${response.user.name} ${response.user.lastname}`);
      setShow(false);
    }else {
      toast.error(`Error de registro: ${response.message[0].messages[0].message}`);
    }
    setLoading(false);
  }
});
```

Figura 10.66: Código para validar datos de los formularios de la aplicación

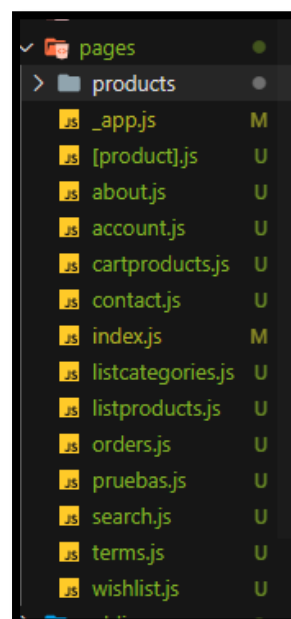


Figura 10.67: Rutas establecidas como páginas de la aplicación

```

export const registerUser = async(formData) =>{
  try {
    const url = `${BASE_URL}/auth/local/register`
    const params = {
      method: 'POST',
      headers: {
        "Content-Type": "application/json"
      },
      body: JSON.stringify(formData)
    }

    const response = await fetch(url, params);
    const result = await response.json();
    return result;

  } catch (error) {
    return error;
  }
}

```

Figura 10.68: Api para el registro de un nuevo cliente

```

const authFetch = async(url, params, Logout) => {
  const token = getToken();

  if( ! token ){
    logout();
  } else {
    if( checkToken(token) ){
      logout();
    } else {
      const paramsTemp = {
        ...params,
        headers: {
          ...params?.headers,
          Authorization: `Bearer ${token}`
        },
      };
      try {
        const response = await fetch(url, paramsTemp);
        const result = await response.json();
        return result;
      } catch (error) {
        return error;
      }
    }
  }
}

export default authFetch;

```

Figura 10.69: Api para las acciones generales de usuarios registrados

```

const handleSubmit = async(e) => {
  e.preventDefault();
  setLoading(true);

  if(!stripe || !elements) return;

  const cardElement = elements.getElement(CardElement);
  const result = await stripe.createToken(cardElement);

  if(result.error) {
    toast.error(result.error.message);
  } else {
    const response = await paymentCart(
      result.token,
      productsData,
      auth.idUser,
      address.id,
      logout
    );
    if(size(response) > 0) {
      toast.success('Compra realizada');
      removeAllProductCartApi(auth.idUser, products, logout);
      updateAmountProductBySell(productsData, logout);
      setReloadCart(true);
      setReload(true);
      router.push('/');
    } else{
      toast.error('Error al realizar el pedido');
    }
  }
  setLoading(false);
}

```

Figura 10.70: Formulario y función que valida las compras realizadas

10.16.3. Imágenes metodología Mobile-D

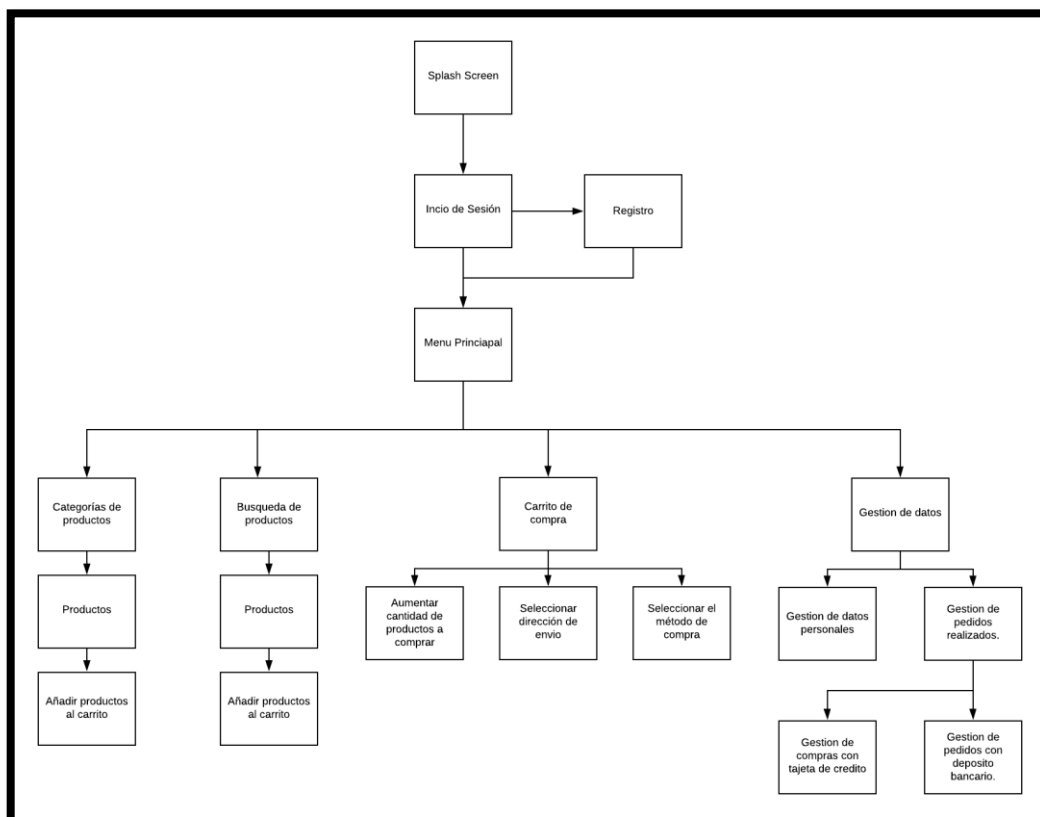


Figura 10.71 Diagrama de navegación de la aplicación



Figura 10.72 Inicio de sesión móvil del cliente.



Figura 10.73 Interfaz móvil de datos del producto.

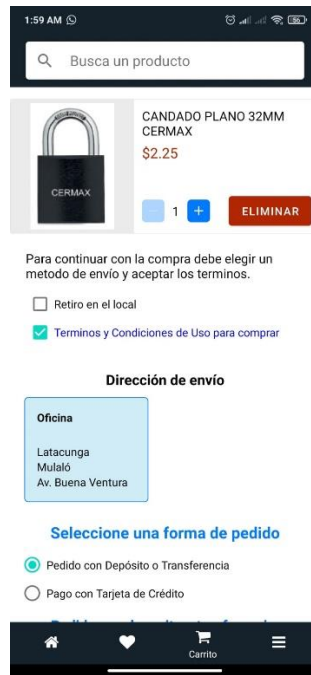


Figura 10.74 Interfaz móvil del carrito de compras del cliente.

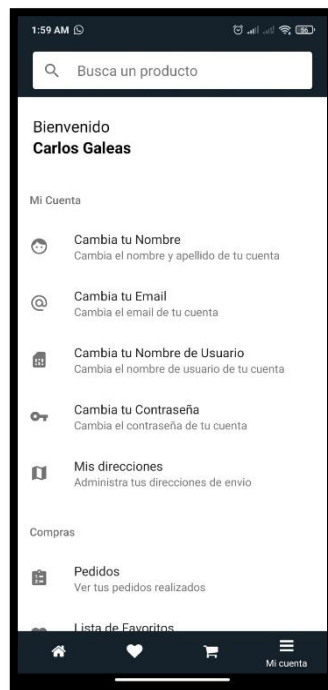


Figura 10.75 Interfaz móvil de datos del cliente



Figura 10.76 Interfaz móvil de pedidos del cliente.

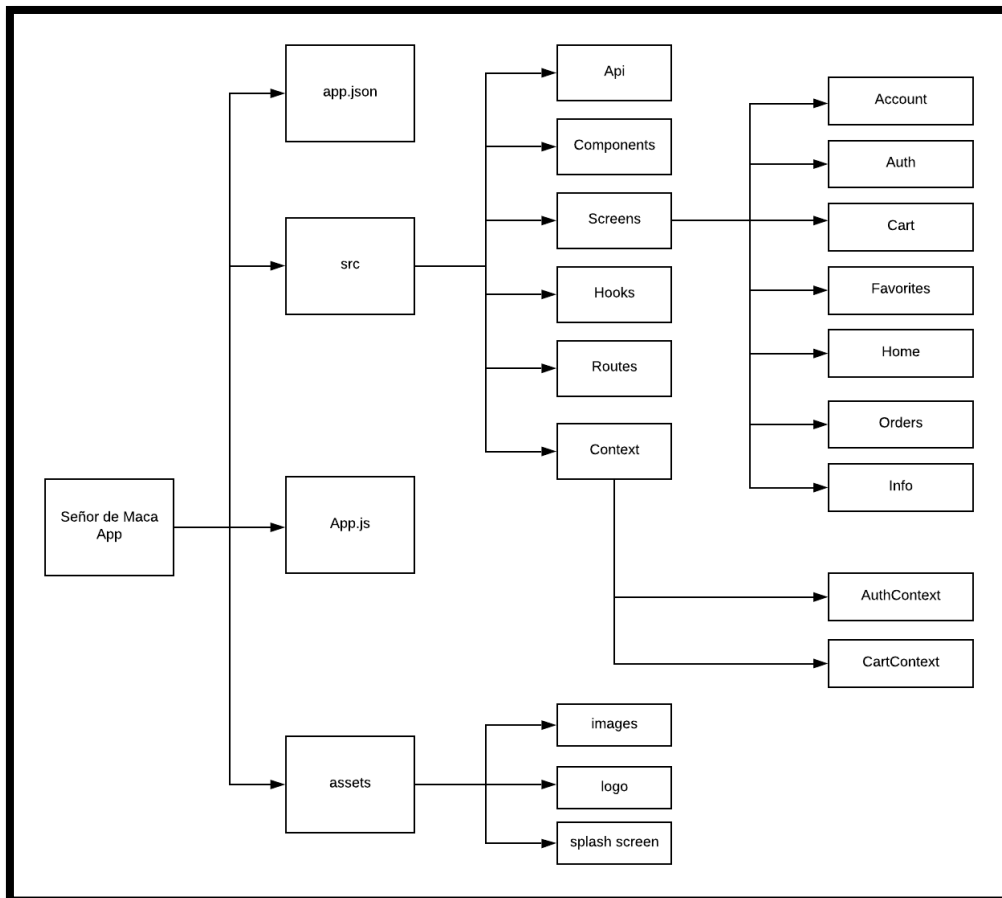


Figura 10.77 Estructura de los directorios de la aplicación.

10.16.4. Imágenes de la configuración del servidor AWS

```
...or create a new repository on the command line

echo "# test-project" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/CarlosGaleas20/test-project.git
git push -u origin main

...or push an existing repository from the command line

git remote add origin https://github.com/CarlosGaleas20/test-project.git
git branch -M main
git push -u origin main

...or import code from another repository
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.
Import code
```

Figura 10.78 Códigos para cargar una aplicación en Github.

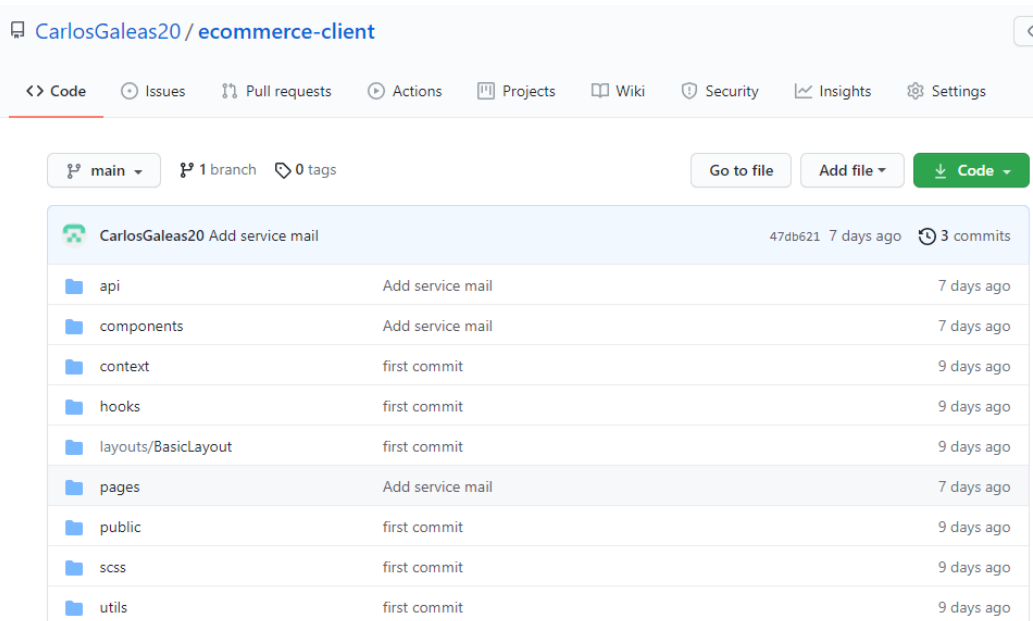


Figura 10.79 Aplicación en Github.


```

Build and test settings
We've auto-detected your app's build settings. Please ensure your build command and output folder (baseDirectory) are correctly detected.
1 | version: 1
2 | frontend:
3 |   phases:
4 |     preBuild:
5 |       commands:
6 |         - npm ci
7 |     build:
8 |       commands:
9 |         - npm run build
10 |   artifacts:
11 |     baseDirectory: build
12 |     files:
13 |       - '**/*'
14 |   cache:
15 |     paths:
16 |       - node_modules/**/*
17

```

Figura 10.82 Código para construir la aplicación en el Host de Amplify.


main		
Continuous deploys set up (Edit)		
		
https://main...amplifyapp.com		
Last deployment	Last commit	Previews
14/7/2021 5:35:40	Add service mail 47db621 GitHub - main	Disabled

Figura 10.83 Aplicación construida con éxito en Amplify.

Domain management [Add domain](#)

Use your own custom domain with free HTTPS to provide a secure, friendly URL for your app. Register your domain on Amazon Route53 for a one-click setup, or connect any domain registered on a 3rd party provider.

amplifyapp.com	
Domain	Status
amplifyapp.com	Available

URL	Branch	Redirects to
https://main.d8ssqidkzw4yb.amplifyapp.com	main	-

Figura 10.84 Dominio predeterminado de Amplify.

Add domain

Domain
Enter the name of your root domain (eg. yourdomain.com)

Q senordemacaadmin.com X Configure domain

Subdomains
Configure subdomains for your app.

https://senordemacaadmin.com main Exclude root

https:// www .senordemacaadmin.com main Remove

Add

Figura 10.85 Configuración de re direccionamiento del dominio.

Custom domain: senordemacaadmin.com

Domain: senordemacaadmin.com | Status: SSL creation

Status details

SSL creation — SSL configuration — Domain activation

We are issuing an SSL certificate to secure traffic to your custom domain, please wait.

URL	Branch	Redirects to
https://senordemacaadmin.com	main	https://www.senordemacaadmin.com
https://www.senordemacaadmin.com	main	-

Figura 10.86 Configuración automática de Amplify en el dominio seleccionado.

Domain management

Use your own custom domain with free HTTPS to provide a secure, friendly URL for your app. Register your domain on Amazon setup, or connect any domain registered on a 3rd party provider.

Custom domain: senordemaca.com

Domain: senordemaca.com | Status: Available

URL	Branch	Redirects to
https://senordemaca.com	main	https://www.senordemaca.com
https://www.senordemaca.com	main	-

Figura 10.87 Resultado final de la configuración del host.

Elegir un nombre de dominio

Disponibilidad de 'senordemaca.com'

Nombre de dominio	Estado	Precio /1 año	Acción
senordemaca.com	✓ Disponible	12,00 \$	<input type="button" value="Añadir al carro"/>

Sugerencias de dominios relacionados

Nombre de dominio	Estado	Precio /1 año	Acción
elsenordemaca.com	✓ Disponible	12,00 \$	<input type="button" value="Añadir al carro"/>
gruposenordemaca.com	✓ Disponible	12,00 \$	<input type="button" value="Añadir al carro"/>
senordemaca.net	✓ Disponible	11,00 \$	<input type="button" value="Añadir al carro"/>

Figura 10.88 Verificación de disponibilidad del dominio.

Carro de la compra

Cuotas fijas

senordemaca.com

Registrarse durante año **12,00 \$**

senordemacaadmin.com

Registrarse durante año **12,00 \$**

SUBTOTAL **24,00 \$**

Figura 10.89 Carrito de compra AWS.

Solicitudes pendientes

Nombre de dominio	Estado	Marca temporal
senordemaca.com	Registro de dominios en curso	20 julio, p.m. 23:13 UTC-5
senordemacaadmin.com	Registro de dominios en curso	20 julio, p.m. 23:13 UTC-5

Figura 10.90 Dominios pendientes en AWS.

Dominios registrados

[Registrar un dominio](#)
[Transferir dominios](#)
[Informe de facturación de dominios](#)

Mostrando de 1 a 2

Nombre de dominio	Protección de la privacidad	Fecha de vencimiento	Renovación automática	Bloqueo de transferencia
senordemaca.com	Todos los contactos	20 julio, 2022	✓	✓
senordemacaadmin.com	Todos los contactos	20 julio, 2022	✓	✓

Figura 10.91 Dominios aprobados en AWS

