

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Departamento Computación
Ingeniería Informática

TRABAJO DE DIPLOMA

**"Sistema para la gestión de los planes y facturas
comerciales de la empresa provincial de Acopio
Villa Clara"**

Autor: Ernesto Moreira García

Tutores: Ing. Juan Carlos Ortega Camacho

Ing. Yoelvis González Pérez

Santa Clara , Cuba, junio 2019
Copyright©UCLV

UCLV
Universidad Central
"Marta Abreu" de Las Villas



MFC
Facultad de Matemática
Física y Computación

Department of Computation
Informatics Engineering

DIPLOMA THESIS

**System for the management of commercial plans
and invoices of the Acopio Villa Clara Company.**

Author: Ernesto Moreira García

Tutors: Ing. Juan Carlos Ortega Camacho

Ing. Yoelvis González Pérez

Santa Clara, Cuba, junio 2019
Copyright©UCLV

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente:

Atribución- No Comercial- Compartir Igual



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las Villas.

Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830

Teléfonos.: +53 01 42281503-14190



Hago constar que el presente trabajo fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

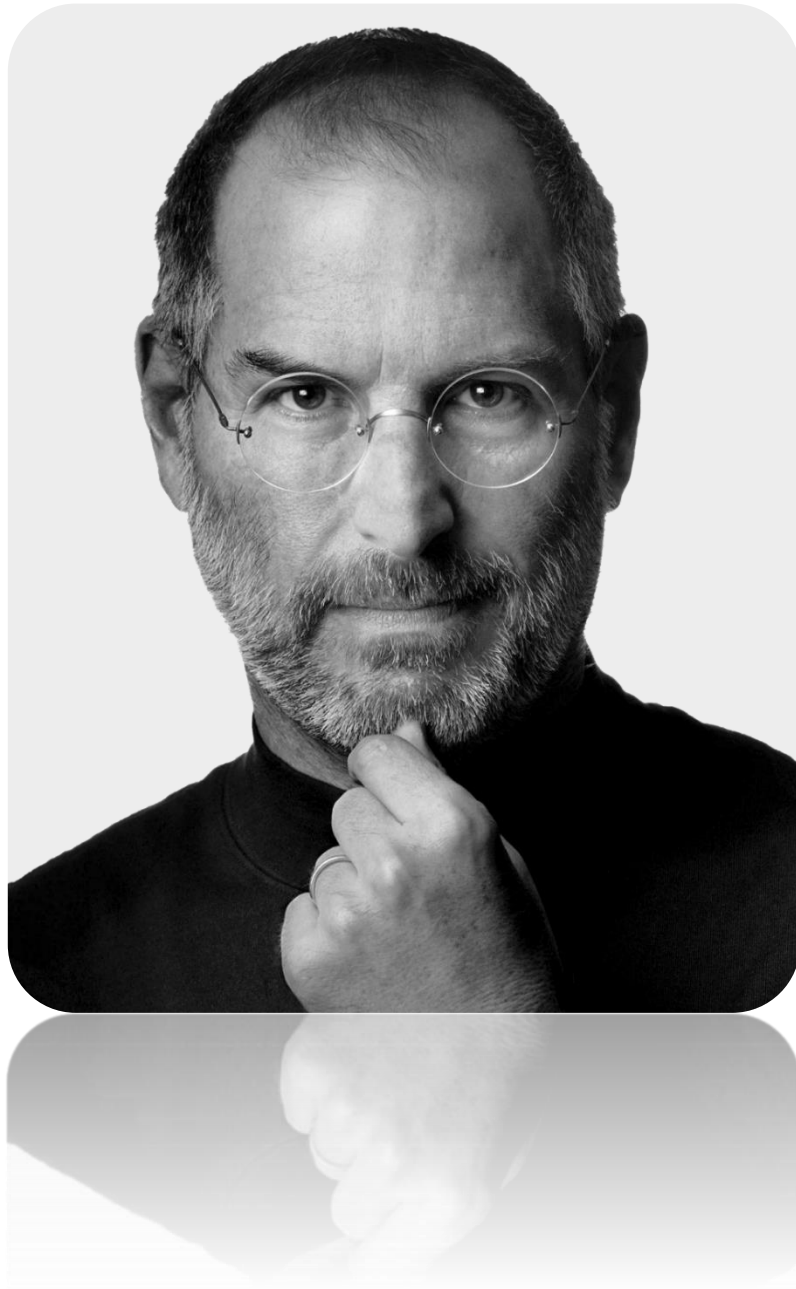
Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del tutor

Firma del jefe del Dpto



“La única manera de hacer un trabajo genial, es amar lo que haces”

Steve Jobs (1955-2011)

Dedicatoria

*A mi abuela Mercedes por ser el motivo de mi razón de ser, de mis logros y estar
a mi lado en todo momento a pesar de las dificultades.*

*A mis padres que son, mi base, mi apoyo en todo momento, mis consejeros y mis
ejemplos, sé que siempre se sentirán orgulloso de mi.*

Ernesto.

Agradecimientos

Agradecer es una tarea demasiado difícil, más aún cuando el espacio es limitado y tantas personas ayudaron a cumplir uno de los sueños más grandes de nuestra vida. A todas estas personas muchas gracias por estar ahí y por ayudarme siempre, pero no podía dejar de mencionar a:

- ❖ *A mi abuela Mercedes por dedicarme toda su vida., sin ella no fuera posible estar donde estoy.*
- ❖ *Mis padres, por ser el motor impulsor de mi vida y mi razón de ser, apoyarme y motivarme en todo momento.*
- ❖ *Mis hermanos, que son el tesoro más grande que me ha dado la vida.*
- ❖ *A todos mis amigos, que se han comportado como familia y eh contado con su apoyo en todo momento.*
- ❖ *Mis compañeros de aula, que ha estado para mí cada vez que los he necesitado y siempre me ofrecieron su tiempo y atención.*
- ❖ *A mis tutores, por su tiempo, ayuda y asesoramiento para poder sacar adelante el trabajo de diploma de manera satisfactoria.*
- ❖ *A la UCLV por darme la oportunidad de superarme.*

A todos...muchas gracias de corazón, Ernesto.

Resumen

El presente trabajo se realiza en la empresa comercializadora provincial Acopio perteneciente a la provincia de Villa Clara, que tiene como misión garantizar la logística al sector agropecuario en beneficio de un desarrollo prospero, sostenible y la seguridad alimentaria del país y principalmente de la provincia. Para llevar a cabo la comercialización de los productos agrícolas que maneja la entidad, surge la necesidad de desarrollar un sistema informático para informatizar la gestión de los planes y contratos comerciales vinculados a la empresa y para mantener un control en tiempo real de su estado. En el desarrollo del estudio se aplicaron técnicas para la obtención de la información de los procesos en cuestión tales como: revisión documental, entrevistas con personal especializado, observación directa y trabajo con expertos. Considerando el escenario actual se propone como solución a este problema realizar un estudio del negocio y del proceso existente y se realiza el diseño e implementación de un sistema de gestión y control que permita mejorar la calidad y la fluidez del proceso descrito, permitiendo realización de las actividades de compra y venta comercial con su facturación a través de sus Unidades Empresariales de Base municipales, así como la gestión de sus proveedores, destinos y productos con que cuenta la empresa, diseñándose e implementándose una base de datos para el almacenamiento de los datos.

Abstract

This work is carried out in the provincial trading company Acopio belonging to the province of Villa Clara, whose mission is to guarantee logistics to the agricultural sector for the benefit of a prosperous, sustainable development and food security of the country and mainly of the province. In order to carry out the commercialization of agricultural products handled by the entity, the need arises to develop a computer system to computerize the management of commercial plans and contracts linked to the company and to maintain a real-time control of their status. In the development of the study, techniques were applied to obtain the information of the processes in question, such as: documentary review, interviews with specialized personnel, direct observation and work with experts. Considering the current scenario, it is proposed as a solution to this problem to carry out a study of the business and the existing process and the design and implementation of a management and control system is carried out that allows to improve the quality and fluidity of the described process, allowing the performance of the commercial purchase and sale activities with their billing through their municipal Base Business Units, as well as the management of their suppliers, destinations and products that the company has, designing and implementing a database for data storage.

Contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN.....	6
¿Qué es la informatización de procesos en la sociedad?	6
Diferencias entre aplicación web y aplicación de escritorio.....	7
1.1 Tecnologías de programación en aplicaciones web.	10
1.1.1 Tecnologías del lado del cliente. Capa del navegador.....	10
1.1.2 Tecnologías del lado del servidor. Capa del servidor.....	13
1.2. Sistemas Gestores de Bases de Datos.....	14
1.3 Servidores web más utilizados en la actualidad.....	16
1.4 Marcos de trabajo PHP para desarrollo. ´	18
1.5 Lenguaje de Modelado.....	21
1.6 Herramientas CASE de desarrollo de software UML.	22
1.7 Metodologías de desarrollo de software.....	23
Conclusiones parciales del capítulo.....	26
CAPÍTULO 2: MODELADO DEL NEGOCIO Y REQUISITOS.....	28
2.1 Modelo de negocio.....	28
2.1.1 Reglas del negocio.....	29
2.1.2 Actores del negocio.....	29
2.1.3 Trabajadores del negocio.....	30
2.1.4 Diagrama de casos de uso del negocio.....	31
2.2 Actores del sistema.....	32
2.3 Definición de los requisitos:.....	32
2.3.1 Definición de los requisitos funcionales.....	33
2.3.2 Definición de los requisitos no funcionales.....	34
2.4 Casos de usos del sistema.....	35
2.5 Casos de Uso significativos del Sistema.....	38

2.5.1 Descripción de los Casos de Uso significativos del Sistema.	39
2.6 Estimación por puntos de casos de uso.....	42
Conclusiones parciales del capítulo:	42
CAPÍTULO 3: DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	44
3.1 Arquitectura del Software.	44
3.2 Patrón arquitectónico Modelo-Vista-Controlador (MVC).....	44
3.3 Diagrama de clases de diseño.....	47
3.4 Diagrama de secuencia.	49
3.5 Diseño y modelado de la base de datos	50
3.5.1 Modelo conceptual de la Base de Datos.....	51
3.5.2 Modelo entidad-relación de la Base de Datos.	51
3.5.3 Modelado lógico de la Base de Datos.....	52
3.6 Diagrama de despliegue.	52
3.7 Diagrama de despliegue y modelo de componentes.....	53
Conclusiones parciales.....	54
CONCLUSIONES	56
RECOMENDACIONES	57
REFERENCIAS BIBLIOGRÁFICAS	58
ANEXOS	60
Anexo 1.....	60
Anexo 2.....	68
Anexo 3.....	75

Índice de Tablas

Tabla 1: Atores del negocio.	30
Tabla 2: Trabajadores del negocio.	31
Tabla 3: Actores del sistema.	32
Tabla 4: Descripción Caso de Uso Gestionar facturas de compras.	42
Tabla 5: Descripción Caso de Uso Gestionar facturas de ventas.	78
Tabla 6: Descripción Caso de Uso Gestionar productos.	82
Tabla 7: Peso de actores sin ajustar.	69
Tabla 8: Peso de las transacciones.	70
Tabla 9: Peso de los factores de complejidad técnica.	71
Tabla 10: Escala de los factores de complejidad técnica.	71
Tabla 11: Peso de los factores ambientales.	73

Índice de Ilustraciones

Ilustración 1: Diagrama de casos de uso del negocio.....	31
Ilustración 2: Diagrama de Casos de Uso del Sistema 1.	37
Ilustración 3: Diagrama de Casos de Uso del Sistema 2.	38
Ilustración 4: Interrelación entre los elementos del patrón MVC.	46
Ilustración 5: Arquitectura del sistema.	47
Ilustración 6: Diagrama de clases de diseño.	48
Ilustración 7: Diagrama de clases de diseño para las clases entidad.	49
Ilustración 8: Diagrama de clases de diseño para el caso de uso Modificar Producto.....	50
Ilustración 9: Modelo conceptual de la Base de Datos.	51
Ilustración 10: Modelo entidad-relación de la Base de Datos.	52
Ilustración 11: Modelo lógico de la Base de Datos.....	52
Ilustración 12: Diagrama de despliegue.	53
Ilustración 13: Diagrama de despliegue y modelo de componentes.	54
Ilustración 14: Interfaz de login.	60
Ilustración 16: Interfaz Usuario, solo válida para el rol Dirección Provincial.	61
Ilustración 17: Interfaz Destinos para el rol Dirección Provincial.....	62
Ilustración 18: Interfaz Productos para rol Dirección Provincial.	63
Ilustración 19: Interfaz Proveedores para rol Dirección Provincial.....	64
Ilustración 20: Interfaz Compras para rol UEB.	65
Ilustración 21: Interfaz Compras para rol Dirección Provincial.	66
Ilustración 22: Interfaz Ventas para rol UEB.	67
Ilustración 23: Interfaz Ventas para rol Dirección Provincial.	68

Introducción

Las empresas en la actualidad se encuentran en un entorno cambiante en todos los ámbitos, tanto a nivel tecnológico, como de sistemas de gestión. Ello conlleva que han de hacer un esfuerzo importante para adaptarse lo más rápidamente posible a las nuevas situaciones para seguir siendo competitivas y eficientes en los mercados en los que se desenvuelven, sujetos inevitablemente al proceso de globalización, con sus ventajas, pero también con sus dificultades.

Estos y otros factores determinan que se estén produciendo modificaciones sustanciales en la cultura empresarial, pues en el mundo actual, con la globalización de los mercados, temas como productividad, comercialidad, competitividad y mejoramiento continuo cobran cada vez mayor relevancia, en este sentido; las empresas se preocupan por ofrecer a sus clientes mejores productos y servicios con el fin de permanecer en los exigentes mercados tanto nacionales como internacionales.

Cuba es uno de los países que se encuentran en vías de desarrollo, aun así, ha trabajado en el avance de las Tecnologías de la Información y las Comunicaciones (TICs). En este sentido el país se encuentra inmerso en un proceso de crecimiento y extensión, provocando de esta manera, el desarrollo paulatino de sistemas informáticos en las diferentes áreas de la nación.

La agricultura tiempo atrás, desde que la humanidad comenzó a desarrollarse en diferentes lugares del mundo, se ha constituido como uno de los ejes principales de nuestros pueblos. Frente a esta situación se han visto en la gran necesidad de crear un centro de acopio para eliminar a los intermediarios y que cada agricultor reciba un precio justo por sus productos ofertados.

Acopio es el sistema mediante el cual se adquieren productos agrícolas por compra directa o contratación y se distribuyen o comercializan en forma mayorista o minorista. El Sistema de Acopio de productos agrícolas, especialmente de viandas, hortalizas y granos, conocido a veces simplemente como “Acopio”, se inició como Empresas Provinciales de Acopio, subordinadas a los órganos del Poder Popular. En 1986, con la eliminación del Mercado Libre Campesino, estas

empresas se incorporaron al Ministerio de la Agricultura (MINAG) creándose la Unión Nacional de Acopio, no solo con la función de recibir las cantidades producidas por los productores, sino también su distribución, también se creó la Red de Mercados Agropecuarios Estatales.

A inicios de 2008 La Unión de Acopio era una de las Organizaciones Económicas del Ministerio de la Agricultura, contando con 14 Empresas Provinciales de Acopio. La representación estatal de las actividades de acopio y beneficio de productos agrícolas y forestales ha estado a cargo de la Dirección de Comercialización del MINAG, pero actualmente, con el proceso de separación de las funciones estatales y empresariales, la gestión de acopio pasa a ser una actividad atendida por Unidades Empresariales de Base (UEB) de carácter municipal y provincial.

La Empresa Provincial de Acopio Villa Clara se subordina a la Unión Nacional de Acopio (UNA) perteneciente al Ministerio de la Agricultura, es una entidad netamente comercializadora, cuenta con un total de 14 Unidades Empresariales de Base en la provincia que se subordinan a la Delegación del Ministerio de la Agricultura Provincial. Su **misión** es garantizar la logística al sector agropecuario en beneficio de un desarrollo próspero y sostenible y la seguridad alimentaria del país, con una óptima atención al cliente. Mientras que su **visión** es ser un grupo empresarial de éxito que comercializa y circula insumos y equipamiento de forma mayorista y minorista, presta servicios técnicos y especializados, contando con cuadros de alta cultura integral de dirección y colectivos laborales motivados y de alta profesionalidad.

La entidad cuenta con amplio flujo de trabajo en todas sus áreas de trabajo, con una gran red de trabajo que se encarga de llevar a cabo las actividades comerciales. Gran parte de estas actividades y procesos son realizadas de manera manual, resultando engorrosas y exhaustivas en las tareas del día a día para los trabajadores, por lo que se hace necesario la informatización de algunas tareas, para mejorar y facilitar tanto la experiencia laboral de los empleados, como la calidad de los servicios de la empresa, fomentando y haciendo énfasis en el uso de las tecnologías para su aprovechamiento.

Tanto el departamento de Comercial como el de Economía mantienen un estrecho vínculo en el proceso de comercialización de la entidad, estos son los encargados de trabajar directamente con la información relacionada con los planes y los contratos vinculados a los productos derivados de los proveedores de la empresa, así como su posterior salida a los diferentes mercados propios y otros sectores de consumo. Actualmente esta es una tarea un tanto complicada debido a las grandes cantidades contables que se deben llevar constantemente.

Para llevar a cabo este proceso la entidad cuenta con un balance de los productos agropecuarios, información a la cual tiene acceso una vez culminado cada mes el Departamento de Economía, lo que produce un desconocimiento en tiempo real del estado de los planes tanto de entrada como de salida de los productos, esto evita un mayor control y crea desinformación por parte de los departamentos encargados que se exponen a constantes riesgos de errores.

Derivándose de este escenario que el **problema de investigación** como lograr obtener la información del estado de los procesos comerciales de compra y venta en tiempo real en la empresa provincial Acopio Villa Clara.

Para dar solución a este problema, se traza como **objetivo general**: Implementar un sistema informático basado en una aplicación web, para la informatización de la organización y la gestión de los planes y contratos de la Empresa Provincial de Acopio Villa Clara, que permita además mantener su seguimiento y control en tiempo real por el personal del departamento Comercial de la empresa de la compra y venta de sus productos.

Para su cumplimiento se plantean como objetivos específicos:

1. Implementar una base de datos para almacenar la información del sistema.
2. Seleccionar las tecnologías a usar para desarrollar el sistema.
3. Elegir la metodología de software apropiada para estructurar, planificar y controlar el proceso de desarrollo.
4. Implementar la aplicación web siguiendo las buenas prácticas de programación.
5. Realizar la validación del sistema para el cumplimiento de los requisitos establecidos.

Preguntas de investigación:

1. ¿Cuáles son las actividades involucradas en los procesos de comercialización de la entidad?
2. ¿Qué solución, en cuanto a arquitectura y tecnología, sería la más idónea para dar solución al problema planteado?

La **justificación** de este trabajo se sustenta en la necesidad de implementar una aplicación para la gestión de los planes y contratos de la entidad Provincial de Acopio Villa Clara, debido a que los departamentos encargados de manejar esta información realizan un trabajo extenso y engorroso para la manipulación y control de los mismos a causa del amplio volumen de trabajo al que se enfrentan. La creación de este sistema permitirá la informatización de los servicios de la empresa, facilitando y mejorando la experiencia de trabajo del personal.

Se considera como **hipótesis** del trabajo, que el desarrollo de un entorno basado en una aplicación web facilita la organización, gestión y ofertas de toda la variedad de productos que maneja la entidad comercial Acopio Villa Clara, principalmente los Departamentos de Comercial y Contabilidad y Finanzas, llevando a un entorno web todo el trabajo que hasta el momento es llevado a cabo de manera manual.

Estructura del documento:

Capítulo 1. Fundamentación Teórica:

En este capítulo se detallan importantes aspectos a tener en cuenta para el desarrollo del trabajo. Se especifican y se analizan las posibles herramientas y metodologías de desarrollo de software a utilizar en la construcción del sistema, especificándose sus principales características y ventajas de su utilización.

Capítulo 2. Modelado del negocio y requisitos:

Este capítulo está dedicado a identificar las funcionalidades del sistema. Se definen los actores, los trabajadores y las entidades que se manipulan en el negocio, así como los casos de uso que se manejan, especificándose las relaciones que se establecen entre ellos. Se realiza, además, el

levantamiento de los requisitos, tanto los funcionales como los no funcionales, que sirven de entrada a la definición de los casos de uso del sistema, los que también serán presentados en este apartado junto con sus descripciones.

Capítulo 3: Descripción de la propuesta de solución.

En este capítulo se describe el diseño del sistema por medio de diagramas y modelos según lo propone la metodología de desarrollo seleccionada. Se muestran los diagramas de arquitectura del sistema, clases de diseño y clases persistentes con las que contará el sistema. Se exponen también los diferentes modelos de datos de la aplicación, especificándose las descripciones de sus tablas y relaciones entre sí.

Capítulo 1: Fundamentación teórica de la investigación.

El objetivo trazado en este capítulo es llevar a cabo la modelación conceptual y caracterización de las tecnologías de programación a utilizar en el cuerpo del trabajo, y realizar una descripción teórica de las herramientas que se utilicen para la confección de la aplicación web a desarrollar.

¿Qué es la informatización de procesos en la sociedad?

Comúnmente se observa, sin importar el tamaño y antigüedad de la organización, la existencia de empleados que pasan horas frente a pilas de papel o frente a su computadora, puliendo datos brutos que, una vez afinados servirán para tomar decisiones de corto o largo alcance.

Informatizar no solo sirve para un adecuado control, sino que también acelera los procesos de ingresos de mercadería, facturación y almacenamiento de documentos. Además, presenta innumerables ventajas al funcionamiento integral del negocio. En primer lugar, posibilita gestionar eficazmente y también facilita la obtención de distintos reportes de venta y compras, lo cual nos proporciona una información vital para las gestiones de información .(Cardozo, 2014)

Según (Oqotech, 2017) para lograr que la informatización de procesos sea exitosa, se deben establecer una serie de objetivos:

- La mejora del proceso a través del uso de las herramientas informáticas. Esto permite evitar los errores producidos por la carga manual de datos. Colabora con la implantación efectiva de los protocolos que hayan sido adoptados por la organización y permite definir controles automatizados de procesos. Además, permite tener un registro electrónico en tiempo real.
- Obtener datos. Gracias a las herramientas informáticas utilizadas, se pueden obtener datos en tiempo real.
- Generar información. Los datos recolectados son procesados de tal manera que facilitan la interpretación de sus resultados y la generación de indicadores.

Por lo que se puede decir que la informatización de la sociedad cubana es la utilización ordenada y masiva de las tecnologías de la información y las comunicaciones en todas las esferas de la sociedad cubana, en su esfuerzo por lograr cada vez más eficacia y eficiencia en todos los procesos y por consiguiente mayor generación de riqueza y aumento en la calidad de vida de los ciudadanos. La Informatización se basa en los principios y valores fundamentales de nuestro sistema social

Diferencias entre aplicación web y aplicación de escritorio.

Al buscar una solución de software para la gestión de la información de una empresa llegará el momento en el que se enfrente al dilema de elegir entre una aplicación web o de escritorio, para ello ha de ser necesario obtener un conocimiento previo para seleccionar la más idónea.

Aplicación de escritorio

Una aplicación de escritorio es aquella que se encuentra instalado en el ordenador o sistema de almacenamiento (USB) y se puede ejecutar sin internet en nuestro sistema operativo.

Será un programa el encargado de realizar la funcionalidad del software implementado que se instalará en cada puesto de trabajo y se conectará a través de la red con la base de datos. La principal ventaja de este sistema será la rapidez de uso ya que se podrá incorporar todos los controles de escritorio y todos los eventos asociados a ellos. Pueden ser más robustas y se puede hacer cualquier cosa que permita el software (cuestión de gráfica, control total de las entradas del usuario al momento de capturar)

Como principal desventaja se tiene la gestión de actualizaciones que nos obligará a actualizar todos los programas instalados en cada puesto de la empresa cada vez que se implementan evoluciones o corriamos fallos. Esto nos obligará a diseñar un sistema automático de gestión de actualizaciones ya que un usuario con un software obsoleto puede dañar la base de datos.

Otra desventaja importante es la escasa portabilidad ya que, si se implementa para un entorno Windows, solo en equipos de ese tipo funcionará y no podremos usarla en una *tablet* o un teléfono.

Aplicación web

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador, es un documento que contiene texto, imágenes, sonido, videos, además de enlaces (hipervínculos) a cualquier otro tipo de documento multimedia, permitiendo su visualización mediante un navegador web. En otras palabras, es una aplicación que se codifica en un lenguaje soportado por los navegadores web y se confía su ejecución al navegador.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizarlas y mantenerlas sin necesidad de distribuir e instalar software a miles de usuarios potenciales. Es importante mencionar que una página Web puede contener elementos que permiten una comunicación activa entre el usuario y la información, esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios y acceder a gestores de base de datos de todo tipo.(Mora, 2001).

Será un servidor el encargado de realizar la funcionalidad del sistema que hemos implementado a través de un programa que manejará el usuario con el navegador web (Internet Explorer, Firefox, Chrome, etc.) de su ordenador.

Por lo que se puede concluir que las aplicaciones web son una plataforma interactiva creada para que los usuarios realicen ciertas acciones. Su objetivo principal es facilitar el flujo de información, agilizar el trabajo y aumentar la efectividad dentro de los procesos habituales de una empresa.

La principal ventaja será la disponibilidad de la aplicación a través de dispositivos que tengan un navegador web: ordenadores, teléfonos móviles, *tablets* y otros dispositivos. De esta forma un escenario posible podría ser un comercial de una empresa que cierra un pedido en el domicilio de su cliente y a través de una tablet deja realizado el mismo y confirmado con el cliente un

plazo de entrega. En ese caso el equipo que tramite los pedidos ubicado en la empresa tendrá constancia del pedido en el momento y podrá tramitarlo rápidamente.

Otra ventaja muy importante será la gestión de actualizaciones, pues basta con actualizar la aplicación del servidor y todos los usuarios la tendrán en el momento. Sólo será necesario poner la aplicación en modo mantenimiento para que no haya ningún usuario conectado en ese momento (y no pierda datos) y realizar la mejora. Este tipo de actualizaciones puede hacerse en un horario fuera del tiempo de oficina de la empresa.

La interfaz de una aplicación web no es una desventaja frente a la interfaz de una aplicación de escritorio ya que actualmente los controles web cuentan con una funcionalidad y cercanía al usuario muy amplias. Si resulta una desventaja la necesidad de conexión a la red en todo momento, y por lo general su tiempo de respuesta es más lento en comparación con las aplicaciones de escritorio.

En definitiva, actualmente resulta más práctico y aconsejable el uso de aplicaciones web siempre que necesitemos un trabajo en diferentes ubicaciones.

¿Por qué realizar una aplicación web y no de escritorio?

Una vez vistas las ventajas que ofrecen las aplicaciones web se puede concluir que en la actualidad las empresas se apoyan cada vez más en adoptar un sistema que automatice sus procesos y mejore el tratamiento de sus productos, por lo que el desarrollo de una aplicación web está dejando de ser una alternativa para pasar a ser una necesidad casi esencial. Desarrollar este tipo de plataformas digitales es invertir en eficiencia, ya que los beneficios que puede brindar mejoran no solo los procesos sino los alcances funcionales y comerciales de la empresa.

El impacto de las aplicaciones Web sobre cómo operar un negocio, transmitir y recibir información es considerable. Las aplicaciones Web ofrecen la oportunidad de conectar a los usuarios entre sí y las empresas con sus clientes. Los desarrolladores de aplicaciones web de hoy están dando forma a un futuro digital, proporcionando incontables facilidades, y soluciones a procesos de la vida cotidiana de manera interactiva, intuitiva, práctica y elegante.

Analizados ambos tipos de aplicaciones se decidió implementar una aplicación web ya que no obligan a usar un determinado sistema operativo y no requieren instalación, estas solo necesitan que esté presente en cualquier computadora un navegador desde el cual acceder. Se puede configurar de manera segura dependiendo de los requisitos de la empresa, tampoco aparecerán problemas de incompatibilidad entre versiones, ya que todos trabajan con la misma. Definitivamente una Aplicación Web es más portable que una de escritorio y se puede acceder desde cualquier lugar, incluso es posible accederla mediante una conexión remota a través de un servidor, lo que es una característica deseable en el sistema que se quiere implementar en la empresa.

1.1 Tecnologías de programación en aplicaciones web.

Los lenguajes de programación Web han ido surgiendo según las necesidades de las plataformas, intentando facilitar el trabajo a los desarrolladores de aplicaciones. Se clasifican en lenguajes del lado cliente y lenguajes del lado servidor. Son muy utilizados lenguajes o arquitecturas que no son propiamente lenguajes de programación, como HTML (*HyperText Markup Language*) o XML (*Extensible Markup Language*). Se utilizan para servir los datos adecuados a las necesidades del usuario, en función de cómo hayan sido definidos por el dueño de la aplicación.

Los desarrolladores web generalmente utilizan lenguajes interpretados (scripts) en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios). Se han desarrollado tecnologías para coordinar estos lenguajes con las tecnologías en el lado del servidor.

1.1.1 Tecnologías del lado del cliente. Capa del navegador.

Son aquellos lenguajes que son asimilados directamente por el navegador y no necesitan tratamiento previo.

HTML HyperText Markup Language.

La información web publicada en Internet (*World Wide Web*) debe emplear un lenguaje que sea entendido universalmente. El lenguaje utilizado es el HTML, actualmente es el lenguaje hipertexto más aceptado a nivel mundial y por tanto uno de los más usados para la creación de documentos. El mismo es un lenguaje de marcado que se diseñó con el objetivo de estructurar documentos y mostrarlos en forma de hipertexto, brinda la información correspondiente relacionada con el contenido a mostrar en cada fragmento de información, y permite establecer relaciones unidireccionales entre documentos.

HTML cumple con dos objetivos fundamentales para el diseño y visualización de un documento digital:

- Organiza un documento en elementos lógicos, tales como: encabezado, párrafo.
- Define las operaciones tipográficas y las funciones que debe ejecutar un programa visualizador sobre dichos elementos.

Permite además a los autores:

- Diseñar formularios para la realización de transacciones con servicios remotos, para su uso en la búsqueda de información, hacer reservas, pedir productos, etc.
- Insertar hojas de cálculo, videoclips, clips de sonido, y otras aplicaciones directamente en sus documentos.

El lenguaje HTML basa su filosofía de desarrollo en el empleo de referencias. Para añadir un elemento externo a la página (multimedia), este no se incrusta directamente en el código de la página, sino que hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma estándar por cualquier navegador web actualizado. (Robbins, 2010)

CSS (*Cascading Stylesheets*)

Hojas de estilo en Cascada (CSS) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, así como interfaces de usuario para aplicaciones web.

Los estilos son reglas simples que normalmente requieren solo unas pocas líneas de código y pueden ser declarados en el mismo documento. No es estrictamente necesario obtener esta información de archivos externos, pero es una práctica recomendada. Cargar las reglas CSS desde un documento externo (otro archivo) nos permite organizar el documento principal, incrementar la velocidad de carga y aprovechar las nuevas características de HTML5. (Gauchat, 2012)

JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas

Características de JavaScript:

- Es simple, no hace falta tener conocimientos avanzados de programación para aprender a manejar JavaScript y es recomendado por muchos expertos a la hora de encontrar un lenguaje para comenzar a programar.
- Maneja objetos dentro de una página web y sobre estos objetos podemos definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas.
- Es dinámico, responde a eventos en tiempo real. Eventos que se producen al presionar un botón, pasar el puntero del mouse sobre un determinado texto, el simple hecho de cargar la página o caducar un lapso de tiempo.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes. Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. (Flanagan & Ferguson 2002)

1.1.2 Tecnologías del lado del servidor. Capa del servidor.

Son aquellos lenguajes que se ejecutan por el propio servidor y son enviados al cliente en un formato claro para él.

PHP *Hypertext Preprocessor*.

PHP es un acrónimo recursivo que significa *Hypertext Preprocessor*. Es un lenguaje de programación, interpretado, diseñado originalmente para la creación de páginas web dinámicas, se usa principalmente para programar las operaciones que se realizan del lado del servidor.

Dentro de sus principales características se encuentran:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutarlo y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en su sitio web oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web, es un producto de código abierto, lo que quiere decir que se puede acceder a su código. Se puede utilizar, modificar y redistribuir sin coste alguno. A diferencia de los productos comerciales y de código cerrado, si se desea modificar algo o agregar un elemento al programa, puede hacerse con total libertad. (Welling & Thomson, 2009)

1.2. Sistemas Gestores de Bases de Datos.

Actualmente existen varios gestores de bases de datos y MySQL está entre los más usados. Entre los diferentes gestores de bases de datos encontramos además PostgreSQL, Microsoft SQL Server y Oracle. A la hora de elegir un gestor u otro, hay que tener en cuenta las ventajas que aporta cada uno a la hora de crear, gestionar y administrar bases de datos, así como de qué forma almacena y busca la información.

Un Sistema Gestor de Base de Datos (SGBD) es un sistema que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.(Bertino & Martino, 1995).

PostgreSQL.

PostgreSQL es un potente sistema gestor que posee una gran escalabilidad, pues es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, soportando una mayor cantidad de peticiones simultáneas de manera correcta. Implementa el uso de deshacer los cambios, subconsultas y transacciones, haciendo su funcionamiento mucho más eficiente y tiene la capacidad de comprobar la integridad referencial, además de almacenar procedimientos en la propia base de datos. Es también, un sistema multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso.

Microsoft SQL Server.

Microsoft SQL Server es una plataforma de base de datos y análisis de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y

las aplicaciones de comercio electrónico. SQL Server ofrece la tecnología y las funciones con las que pueden contar las organizaciones. Microsoft SQL Server, al contrario de su más cercana competencia, no es multiplataforma, ya que sólo está disponible en Sistemas Operativos de Microsoft.

Oracle.

Oracle es un sistema gestor de base de datos relacional extremadamente potente y flexible. Está potencia y flexibilidad, sin embargo, implican también una cierta complejidad. Está orientado al acceso remoto y a Internet. Puede ser implementado en diferentes plataformas: UNIX, las distintas versiones Windows y Linux, aunque la plataforma para la que fue pensado inicialmente Oracle fue UNIX, este ha logrado adaptarse a otras como Windows desplazando del mercado a los gestores propios de este sistema operativo.

Oracle contiene una avanzada seguridad, los clientes pueden encriptar de manera transparente todos los datos de sus aplicaciones o solo algunas columnas específicas de la base de datos si así lo prefieren. Ofrece un rendimiento mucho mayor que cualquier otra plataforma de Base de Datos. Al permitirnos asignar nuestras propias zonas de memoria a sus datos y cualidades, podemos tener en todo momento controlados tanto el crecimiento como el rendimiento de los distintos esquemas que componen nuestra base, aunque, por otro lado, esto suponga un problema, ya que debemos estar pendientes en todo momento de su configuración para no sufrir fallos debido a algún problema de almacenamiento.

MySQL.

MySQL es un sistema gestor de bases de datos relacionales rápido, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, así como para la creación de cualquier otra solución que implique el almacenamiento de datos, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas también en esos lenguajes.

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el

programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente. Es uno de los sistemas gestores de bases de datos más utilizado en la actualidad, utilizado por grandes corporaciones como Yahoo!, Google, Motorola, entre otras.(Cobo & Gómez, 2005).

¿Por qué usar MySQL como Sistema Gestor de Base de Datos?

- Es una base de datos gratuita. Al ser de código abierto, no tiene coste, con el ahorro que eso conlleva.
- Es muy fácil de usar. Podemos empezar a usar la base de datos MySQL sabiendo unos pocos comandos.
- Es una base de datos muy rápida. Su rendimiento es estupendo sin añadirle ninguna funcionalidad avanzada.
- Utiliza varias capas de seguridad. Contraseñas encriptadas, derechos de acceso y privilegios para los usuarios.
- Pocos requisitos y uso eficiente de memoria. Tiene un bajo gasto de memoria y necesita pocos recursos de CPU o RAM.
- Es compatible con Linux y Windows.

La flexibilidad con todas las versiones de Linux, UNIX y Windows es otra de las ventajas principales de la base de datos MySQL. Permite personalizarla totalmente al ser de código abierto.

Por otro lado, el alto rendimiento que generan las bases de datos MySQL es muy alto, tanto si es un sistema de procesamiento de transacciones de alta velocidad, como un sitio web de alto volumen con servicio a mil millones de consultas al día.

1.3 Servidores web más utilizados en la actualidad.

Los servidores web son los pilares del *Hosting*, permiten a los clientes alquilar almacenamiento en servidores para alojar sus archivos. Un servidor HTTP, conocido como servidor web, es un

software que se instala, corre y procesa información desde el lado del server. El servidor web es el encargado de despachar archivos por los protocolos HTTP y HTTPS hacia un cliente que solicita dicha información.

Los servidores HTTP suelen almacenar por lo general páginas web, que incluyen archivos HTML, PHP, así como imágenes, videos y audios. Son los encargados de servir todo lo que se pide desde una computadora cuando alguien visita una web.

En resumen, un servidor web es un equipo informático que satisface peticiones de otras máquinas (clientes) de forma remota que trabaja usando el modelo cliente servidor.

Nginx y Apache son servidores web populares usados para enviar páginas web al navegador del usuario.

Servidor Nginx

Es un servidor web de open source y gratuito (aunque también existe una versión comercial) que se destaca por su alto rendimiento. Incluye además funciones como servidor proxy reverso HTTP, balanceador de carga, así como POP3 y IMAP. Está disponible para Windows y Linux.

Entre sus beneficios se encuentran una configuración simple, pero al mismo tiempo poderosa, permitiendo configurarlo para integrarse nativamente con casi cualquier tecnología y lenguaje de programación moderno. Es ideal para despachar archivos estáticos y también dinámicos.

También se destaca por consumir muy pocos recursos bajo entornos de muchas visitas simultáneas, ideal no solo para despachar visitas rápidamente, sino también para evitar agregar nuevo hardware cuando no es necesario realmente.

Servidor Apache

Apache es uno de los servidores web más populares que permite ejecutar un sitio web seguro sin demasiados problemas. Funciona excelentemente con muchos sistemas de administración de contenido como *Joomla* y *Drupal*, marcos web como *Django*, *Laravel* y lenguajes de programación. Esto lo convierte en una elección sólida para todo tipo de plataformas de hosting web.

Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix y Windows, es un software confiable y estable. Este es flexible debido a su estructura basada en módulos. Cuenta con una enorme comunidad y soporte fácilmente disponible en caso de cualquier problema.

Para el desarrollo y funcionamiento del sistema se utilizará el servidor web Apache, ya que estará apoyado en la herramienta XAMPP que es el entorno más popular de desarrollo PHP. Esta es una distribución de Apache, completamente gratuita y fácil de instalar que contiene los paquetes de MySQL, PHP además del servidor Apache.

1.4 Marcos de trabajo PHP para desarrollo. ´

Un framework o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.(Riehle, 2000).

Elegir el marco adecuado para su aplicación es uno de los aspectos más difíciles de las etapas iniciales del desarrollo del proyecto. Si bien los criterios generales para que un equipo de desarrollo elija el marco son: el costo del desarrollo, su experiencia con él, la popularidad del marco, hay varios otros factores como integraciones de terceros, implementación, pruebas y muchos más. El lenguaje de scripting PHP tiene una variedad de marcos con capacidades técnicas robustas como: *Laravel*, *Symfony*, *CodeIgniter*, *Yii 2*, *Phalcon*, *CakePHP*, *Zend*, *Slim*

y otros, en este caso nos centraremos en Symfony y Laravel, los más populares capaces de manejar el modelo-vista-control en PHP.

Symfony.

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente en PHP 5.3. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server.(Hernández, 2015)

Laravel.

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP. Su objetivo es desarrollar aplicaciones con código PHP de forma elegante y simple. Ofrece las funcionalidades necesarias para desarrollar aplicaciones modernas de manera fácil y segura. Está equipado con un montón de características interesantes, incluyendo enrutamiento , PHP nativo y muchos más. Construido con varios componentes de Symfony, Laravel ofrece a las aplicaciones web una muy útil base de código confiable y bien probado.

Laravel permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Cuenta con un potente motor de plantillas llamado *blade*. Intenta aprovechar lo mejor de otros *frameworks* y aprovechar las características de las últimas versiones de PHP. Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel depende también del desarrollo de sus

dependencias. Laravel propone usar rutas con *Closures*, en lugar de un Modelo-Vista-Controlador tradicional con el objetivo de hacer el código más claro. Aun así permite el uso de Modelo-Vista-Controlador tradicional.(Otwell, 2011).

¿Por qué usar Laravel como marco de trabajo?

Sin lugar a duda Laravel y Symfony son los marcos de trabajo PHP del tipo modelo-vista-controlador más populares a día de hoy, muy completos y muy potentes a la hora de desarrollar cualquier aplicación web. En este caso para la implementación de la aplicación web se ha decidido usar Laravel, en su última versión 8.5.0, justificado principalmente por los siguientes elementos:

1. Soporte MVC y enfoque orientado a objetos:

La primera y mayor ventaja de usar el marco de trabajo de Laravel es que sigue los patrones arquitectónicos basados en modelos, vistas y controladores, y tiene una sintaxis expresiva que lo hace orientado a objetos.

2. Autenticación y autorización incorporadas:

Laravel proporciona una configuración inmediata para el sistema de autenticación y autorización. En esta versión se utilizará Jetstream, una de las novedades incorporadas en la versión 8, que es un sistema recomendado por el framework para tratar la autenticación.

3. Consola Artisan:

Laravel tiene su propia interfaz de línea de comandos llamada Artisan, la cual es muy útil y muy poderosa y se utiliza para la publicación de paquetes, la gestión de migraciones de base de datos y la generación de código repetitivo para nuevos controladores, modelos y migraciones entre otras funcionalidades.

4. Mapeo objeto-relacional ORM Eloquent:

Eloquent es la implementación de ORM incorporada de Laravel. Este tiene el mejor mapeado relacional de objetos en comparación con los otros marcos que existen. Esta asignación

relacional de objetos le permite interactuar con los objetos de su base de datos y las relaciones de la base de datos utilizando una sintaxis expresiva.

5. Motor de plantillas:

Laravel viene con un motor de plantillas incorporado conocido como *Blade Template Engine*. Blade combina una o más plantillas con un modelo de datos para producir vistas resultantes, haciendo que las plantillas se conviertan en código PHP almacenado en caché para mejorar el rendimiento, además proporciona un conjunto de sus propias estructuras de control, como sentencias condicionales y bucles, que se asignan internamente a sus contrapartes de PHP.

1.5 Lenguaje de Modelado.

En cualquier proyecto de ingeniería ya sea en la construcción de un gran edificio, de un avión, o de una represa hidroeléctrica; en la implementación de un procesador de texto o de un software de comunicaciones para Internet, se requiere de etapas de modelado que permitan experimentar y visualizar el sistema que se construirá. Uniendo varios conceptos y teorías, se puede conceptualizar un lenguaje de modelado como una estandarización de notaciones y reglas, que permitan diagramar o graficar un sistema, o parte de él. La elección de un aceptado lenguaje de modelado es de vital importancia, pues un buen modelamiento del software que se va a desarrollar influye determinantemente en lograr una adecuada comunicación entre los desarrolladores y los clientes. (Booch, Rumbaugh, & Jacobson, 2006)

UML Unified Modeling Language.

El lenguaje unificado de modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. Es importante remarcar que UML es un "lenguaje de modelado" para especificar o

para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

El lenguaje unificado de modelado permite:

- Visualizar gráficamente un sistema de manera que otros puedan entenderlo.
- Especificar cuáles son las características de un sistema antes de su construcción.
- Construir sistemas diseñados a partir de modelos especificados.
- Documentar los elementos gráficos del sistema desarrollado para futuras revisiones.
- Realizar una verificación y validación del modelo realizado.

El modelo con UML es independiente del lenguaje de implementación, de tal forma que los diseños realizados se pueden implementar en cualquier lenguaje.(Fowler & Scott, 1999).

1.6 Herramientas CASE de desarrollo de software UML.

Se puede definir a las herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida del desarrollo de un Software. La utilización de una herramienta CASE garantiza una mejora en el proceso de desarrollo de software. Las metodologías incluyen gran cantidad de técnicas, y el esfuerzo de documentación es por lo general considerable. Por lo tanto, es difícil aplicar una metodología sin la ayuda de una herramienta. El énfasis en planificación, análisis y diseño promovido por una herramienta CASE tiene un fuerte impacto y recompensa en la mejora de la calidad del producto obtenido y en el aumento de la productividad en actividades de desarrollo y mantenimiento.

Visual Paradigm: es una herramienta CASE basada en el lenguaje UML que soporta todo el proceso de desarrollo propuesto por la metodología RUP para: negocio, análisis y diseño orientados a objetos, construcción, pruebas y despliegue.

Esta herramienta ayuda a los equipos de desarrollo de software a sobrepasar el proceso de

desarrollo de software que incluye modelación, construcción y despliegue, maximizando y acelerando las contribuciones tanto del equipo de desarrollo como individuales.(D. F. González, 2011).

1.7 Metodologías de desarrollo de software.

Lograr la construcción de un sistema informático eficiente, que cumpla con los requisitos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas.

Metodologías robustas:

Están guiadas por una fuerte planificación, centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan del proyecto definido en la fase inicial del mismo. En este grupo se encuentra el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés, Rational Unified Process) el cual se usa para llevar a cabo el análisis, diseño e implementación del software.(Pressman, 2010).

Proceso Unificado de Desarrollo RUP.

El proceso unificado de desarrollo es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar máximo provecho de los conceptos asociados a la orientación a objetos y modelado visual. Cuenta con las mejores prácticas del modelado de desarrollo de un software en particular:

- Desarrollo del software de forma iterativa.

- Manejo de requisitos.
- Utiliza arquitectura basada en componentes.
- Modela el software de forma visual, usando UML.
- Verifica la calidad del software.
- Controla los cambios.
- Dirige las tareas de cada desarrollador por separado y del equipo como un todo.

Es uno de los procesos más generales que existe, está enfocado en cualquier tipo de proyecto, aunque no sea de software, se basa en la documentación generada en cada una de las cuatro fases de desarrollo y los flujos de trabajo.

Utiliza el Lenguaje Unificado de Modelado para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial del Proceso Unificado, sus desarrollos fueron paralelos. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en tres frases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. Esto es lo que hace único al Proceso Unificado.

Esta metodología es la más adecuada para el desarrollo de un Sistema Informático porque permite mitigar en fases tempranas posibles riesgos en el proyecto, porque hace uso del modelado visual y de la orientación a objetos de una manera abarcadora y completa, y porque puede alcanzar un buen grado de certificación en el desarrollo del software se hace necesaria su utilización.(D. F. González, 2011).

Metodologías ágiles:

El desarrollo ágil de software envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

Se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento.

Proceso Unificado Ágil AUP

El Proceso Unificado Ágil de Scott Ambler, es una versión simplificada del Proceso Unificado Racional (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP.(Donatien, 2011)

Dentro de sus principales características se encuentran:

- El modelado agrupa los tres primeros flujos de RUP (Modelamiento del negocio, Requerimientos y Análisis del Diseño).
- Dispone de cuatro fases igual que RUP: Incepción o Creación, Elaboración, Construcción y Transición.
 1. Incepción: identificación del alcance y dimensión del proyecto, propuesta de la arquitectura.
 2. Elaboración: confirmación de la idoneidad de la arquitectura.
 3. Construcción: desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados.
 4. Transición: Validación e implementación del sistema.
- Abarca cuatro flujos de trabajo ingenieriles:
 1. El flujo de trabajo **Implementación** tiene como objetivo transformar el modelo del negocio en código ejecutable y realizar un nivel básico de las pruebas.
 2. El flujo de **Pruebas** tiene como objetivo realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema

funciona tal como está establecido, verificando que se cumplan los requerimientos.

3. El flujo de **Pruebas** tiene como objetivo realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, verificando que se cumplan los requerimientos.
4. Por ultimo dentro se tiene el flujo ingenieril el **Despliegue**, cuyo objetivo es el plan para la prestación del sistema y la ejecución de dicho plan, para que el sistema quede a disposición de los usuarios finales.

➤ Aplica técnicas ágiles que favorecen al desarrollo e implementación del sistema:

1. Desarrollo Dirigido por Pruebas.
2. Modelado Ágil.
3. Gestión de cambios Ágil.
4. Refactorización de Base de Datos para mejorar la productividad.

Una vez analizadas las principales características de la metodología de software Proceso Unificado Ágil (AUP) se considera que es la más apropiada, ya que es un marco de desarrollo de software iterativo e incremental. Es considerado como un proceso ceremonioso porque especifica muchas actividades y artefactos involucrados en el desarrollo de un proyecto de software. Es una metodología simple, todo es conciso y mantiene el foco en las actividades de alto valor.

Conclusiones parciales del capítulo.

En el transcurso del presente capítulo se llevó a cabo el estudio de los fundamentos teóricos que tienen como precedente el problema de investigación, haciendo énfasis en las tecnologías, herramientas y metodologías a emplear para el desarrollo del sistema a realizar, resultando como conclusiones:

- Se realizará una aplicación Web empleando Apache como servidor local de la aplicación y MySQL como gestor de bases de datos.

- Se trabajará sobre el marco de trabajo (*Framework*) Laravel basado en PHP como lenguaje de programación web.
- Se utilizará el Proceso Unificado Ágil (AUP) y algunos de los artefactos que ofrece como metodología de desarrollo de software para el análisis, diseño e implementación del sistema, a la par de UML como lenguaje de modelado.
- Como herramienta CASE de desarrollo de software ha sido seleccionada Visual Paradigm para realizar la modelación.

Capítulo 2: Modelado del negocio y requisitos.

En el presente capítulo se realizará un análisis y modelado completo del negocio, partiendo de los procesos actuales del negocio donde se van a identificar los actores y trabajadores que con él interactúan, además de los casos de usos principales y sus descripciones. Se presentarán distintos diagramas, como el modelo de casos de uso del negocio y del sistema, el modelo de objeto del negocio y el de análisis. Se realizará el levantamiento de los requisitos funcionales y no funcionales que llevan a la identificación de los casos de uso del sistema.

2.1 Modelo de negocio.

El modelado del negocio tiene como finalidad comprender la forma y el entorno en el cual se desarrolla el negocio, los rasgos que posee, así como las actividades más importantes dentro del mismo. En él se determinan los actores y trabajadores del negocio y su relación con los objetos del mismo.

El modelo de casos de uso del negocio es un modelo que describe los procesos del negocio (casos de uso del negocio) y su interacción con elementos externos (actores), este modelo describe las funciones que el negocio pretende realizar. Su objetivo es describir cómo el negocio es utilizado por sus clientes y socios, lo que implica la determinación de los actores y casos de uso del negocio.

Con este artefacto se pretende identificar los procesos en el negocio, definir las fronteras del negocio que van a modelarse, definir quiénes y qué interactuarán con el negocio, así como crear diagramas del modelo de casos de uso del negocio.

Modelado actual del negocio.

El negocio actual que requiere la solución informática a desarrollar en este trabajo comprende el desempeño de procesos relacionados con las actividades económicas y comerciales de la Empresa Acopio Villa Clara.

Esta entidad es la encargada de la comercialización de los productos agrícolas dentro de la provincia, apoyada en sus Unidades Empresariales de Base que están distribuidas por cada

municipio, trabajan mensualmente basados en sus planes de comercio, tanto de compra como de venta de los productos.

Las unidades municipales son las encargadas de elaborar los planes de entrada (compra) según sus proveedores, una vez realizados se envían hacia la entidad provincial. Mientras que los planes de salida (venta) se realizan a partir de los diferentes destinos a los cuales se les proveen distintos tipos de productos. Toda esta información se obtiene de manera mensual, a través de los balances que realiza la empresa para evaluar el cumplimiento de trabajo trazado.

Independientemente de los planes, en el proceso se gestionan las facturas, que son el resultado de las acciones comerciales que realiza la entidad, dígame compra o venta, estas operaciones son llevadas a cabo por el Departamento de Economía.

2.1.1 Reglas del negocio.

Las reglas del negocio describen las características del dominio en el que se encuadra la organización, pueden ser requisitos funcionales, restringir los existentes o definir cálculos particulares. Si las reglas del negocio no se satisfacen, el sistema puede no trabajar de forma satisfactoria.

A continuación, se describen las reglas del negocio correspondientes al negocio actual:

RN1: <Toda actividad comercial debe estar constituida únicamente por los productos comercializables que se encuentran en el sistema>

RN2: <Cada UEB solo puede gestionar únicamente sus facturas propias, incapacitándose cualquier acción sobre una factura no correspondiente a su unidad >

2.1.2 Actores del negocio.

Los actores del negocio deben representarse por una calle, de tal manera que el caso de uso del negocio podrá reflejar la interacción del actor o los actores del negocio con los trabajadores del negocio en la realización del caso de uso del negocio correspondiente.(Jacobson, Spence, & Bittner, 2013).

En la siguiente tabla se muestran los actores del negocio actual.

<u>Actores del negocio</u>	<u>Descripción</u>
Proveedor	Representa a una persona, grupo de personas o entidad que realizan la venta de productos producidos a Acopio. Puede ser de carácter estatal o particular.
Comprador	Representan las entidades que realizan la compra de productos a la empresa.

Tabla 1: Atores del negocio.

2.1.3 Trabajadores del negocio.

Trabajador del negocio conocido también como actor interno del negocio, representa a una persona o a un grupo de personas que tienen relación directa con el proceso empresarial, su definición depende del caso de uso de negocio que se esté analizando. (Menéndez Mueras, 2005).

La siguiente tabla muestra los trabajadores del negocio actual.

<u>Trabajadores del negocio</u>	<u>Descripción</u>
Comercial	Trabajador de la entidad encargado de llevar a cabo los procesos comerciales, lleva a cabo la relación de intercambio de la organización con el mercado.
Economista	Estos ofrecen asesoría financiera y contable en los procesos comerciales de la empresa, participan en la planificación comercial.

Contador	Realiza informes financieros a través de la revisión de los libros contables, estados financieros, análisis de gastos e ingresos y la realización de balances.
-----------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 2: Trabajadores del negocio.

2.1.4 Diagrama de casos de uso del negocio.

Describe los procesos de un negocio (casos de uso del negocio) y su interacción con elementos externos (actores), tales como socios y clientes. Representa, además, las funciones que el negocio pretende realizar y su objetivo básico es detallar cómo el negocio es utilizado por sus clientes y socios.

En la siguiente figura se muestra el diagrama de casos de uso del negocio.

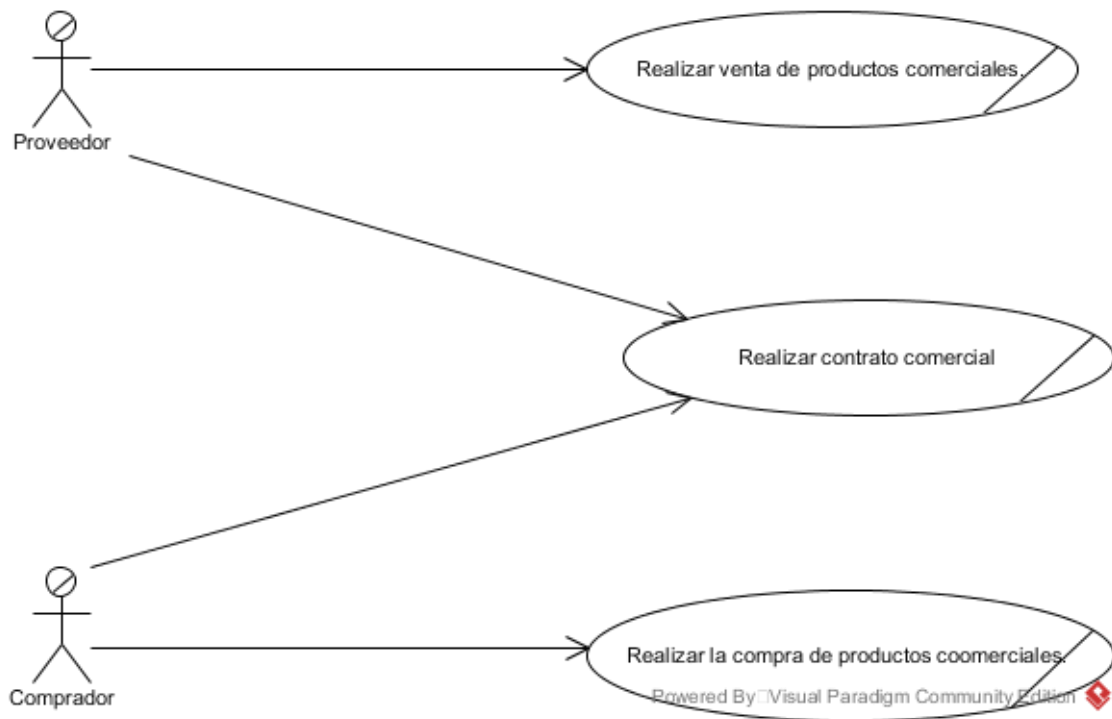


Ilustración 1: Diagrama de casos de uso del negocio

2.2 Actores del sistema.

Los actores del sistema representan a una persona, conjunto de personas, hardware, software y a cualquier componente que interactúa con la solución informática basado en una necesidad.

En la siguiente tabla se muestran los actores del sistema.

<u>Actores del sistema</u>	<u>Descripción</u>
Económico Dirección Provincial VC	Encargado de manejar los usuarios que accedan al sistema, así como sus roles y otorgar los permisos. Es el encargado de supervisar las facturas, los planes y reportes elaborados por las unidades empresariales de base, y mantener el control sobre el estado y cumplimiento de los mismos. Gestiona los proveedores, los destinos y los productos.
Económico Unidad Empresarial de Base UEB	Los encargados de realizar el acopio y la venta de los servicios, realizando la facturación y a su vez generando los reportes que serían controlados y supervisados por la dirección provincial. Llevan a cabo las actividades comerciales.

Tabla 3: Actores del sistema.

2.3 Definición de los requisitos:

El profesor académico de origen británico Ian Sommerville presenta una definición acerca de lo que es un requisito: “Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de este. En el otro extremo, es una definición detallada y formal de una función del sistema.”

La especificación de los requisitos de software es una descripción completa del comportamiento del sistema que se va a desarrollar. Incluye un conjunto de casos de uso que describe todas las interacciones que tendrán los usuarios con el software.

2.3.1 Definición de los requisitos funcionales.

Los requisitos funcionales son las capacidades o funciones que el sistema debe cumplir, estos pueden ser frases muy generales sobre lo que el sistema debería hacer. Para lograr el cumplimiento de los objetivos presentados, las funcionalidades que el sistema debe cumplir son:

Rf 1: Autenticar usuarios.

Rf 2: Gestionar facturas de ventas.

- Rf 2.1: Crear facturas de ventas.
- Rf 2.2: Eliminar facturas de ventas.
- Rf 2.3: Listar facturas de ventas.

Rf 3: Gestionar facturas de compra.

- Rf 3.1: Crear facturas de compra.
- Rf 3.2: Eliminar facturas de compra.
- Rf 3.3: Listar facturas de compra.

Rf 4: Gestionar proveedores.

- Rf 4.1: Insertar proveedor.
- Rf 4.2: Editar proveedor.
- Rf 4.3: Eliminar proveedor.
- Rf 4.4: Listar proveedores.

Rf 5: Gestionar destinos.

- Rf 4.1: Insertar destinos.
- Rf 4.2: Editar destinos.

- Rf 4.3: Eliminar destinos.
- Rf 4.4: Listar destinos.

Rf 6: Gestionar productos.

- Rf 6.1: Insertar productos.
- Rf 6.2: Editar productos.
- Rf 6.3: Eliminar productos.
- Rf 6.4: Listar productos.

Rf 7: Gestionar Control de usuarios.

- Rf 7.1: Insertar usuario Dirección General.
- Rf 7.2: Insertar usuario UEB municipal.
- Rf 7.3: Editar usuario Dirección General.
- Rf 7.4: Editar usuario UEB municipal.
- Rf 7.5: Eliminar usuario Dirección General.
- Rf 7.6: Eliminar usuario UEB municipal.
- Rf 7.7: Listar usuarios.

Rf 8: Generar reportes de ventas.

Rf 9: Generar reportes de compras.

Rf 10: Exportar reportes.

2.3.2 Definición de los requisitos no funcionales.

Los requisitos no funcionales son restricciones que afectan a los servicios funcionales del sistema, propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Pueden ser más críticos que los funcionales permitiendo que puedan marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. A continuación, se dan a conocer los requisitos no funcionales del sistema.

➤ Rendimiento:

RNF 1: El tiempo de respuesta al ejecutar una acción no debe ser mayor que dos segundos, al igual que la velocidad de procesamiento de la información.

RNF 2: Debe ser eficiente a la hora de gestionar las acciones, permitiendo alcanzar los resultados deseados sin hacer un uso extensivo de la navegación por el sitio.

➤ Software:

RNF 3: Se requiere MySQL como servidor de base de datos.

RNF 4: Sistema Operativo Windows 2000 o superior.

RNF 5: Uso de navegador Ej. Chrome, Mozilla Firefox, Microsoft Edge, Safari, entre otros.

➤ Hardware

RNF 6: Las computadoras que utilizarán el software a desarrollar deberán tener 512MB de Memoria tipo RAM y 1.0 GHz de velocidad del microprocesador para su correcto funcionamiento.

➤ Portabilidad:

RNF 7: El sistema debe ser multiplataforma. Teniendo no solo la posibilidad de ejecutarse en Windows, sino también sobre Linux o Unix.

➤ Seguridad:

RNF 8: Garantizar que la información solo sea editada únicamente por las personas con los permisos para ello.

RNF 9: Debe mantenerse la consistencia de los datos en correspondencia con la realidad.

➤ Interfaz:

RNF 10: Mostrar una interfaz sencilla e intuitiva para los usuarios, de manera que no presenten problemas a la hora de realizar alguna funcionalidad.

2.4 Casos de usos del sistema.

Un caso de uso describe una interacción entre el sistema y una agente externo que se denomina actor, un caso de uso capta siempre una función visible para el usuario, logra un objetivo concreto y específico para el usuario y puede ser algo simple o algo complejo (Drake, 2008).

Casos de uso del sistema:

CU 1: Gestionar Usuarios.

CU 2: Gestionar Productos.

CU 3: Gestionar Proveedores.

CU 4: Gestionar Destinos.

CU 5: Gestionar Facturas de Compra.

CU 6: Gestionar Facturas de Venta.

CU 7: Generar Reportes de Ventas.

CU 8: Generar Reportes de Compras.

En las siguientes figuras se muestran los diagramas de casos de uso del sistema:

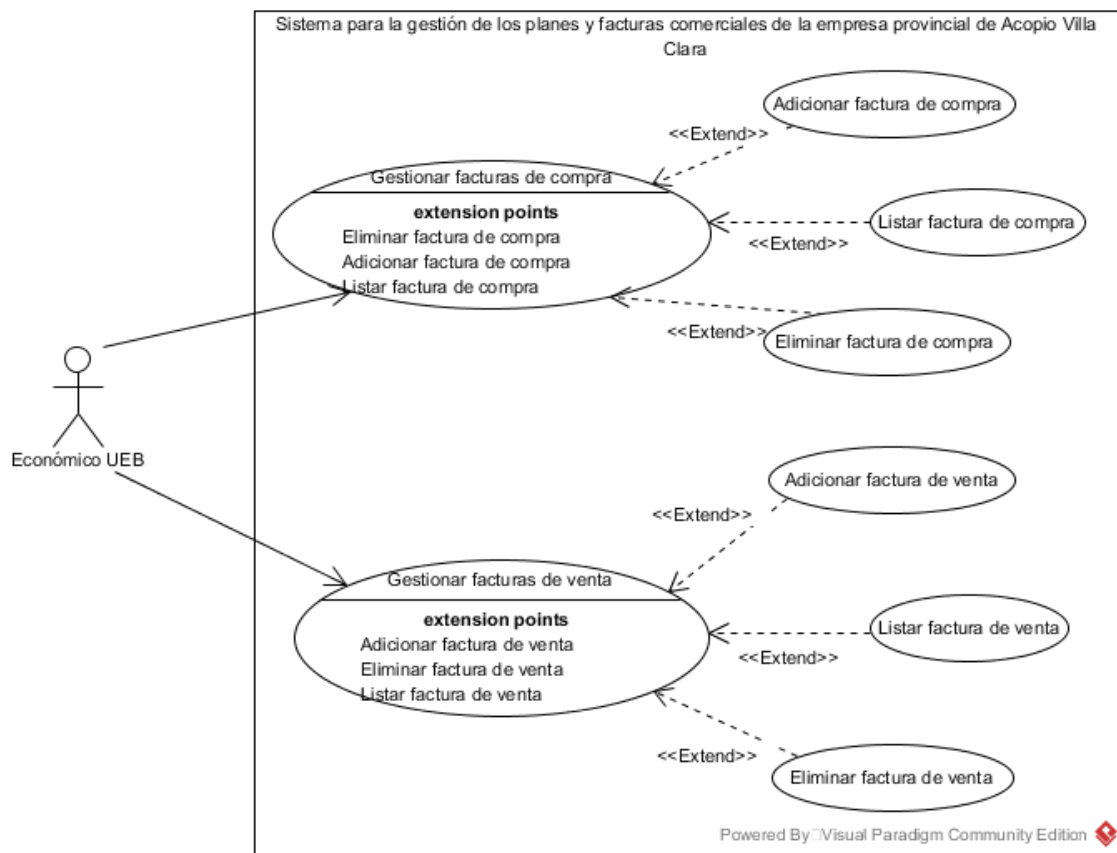


Ilustración 2: Diagrama de Casos de Uso del Sistema 1.

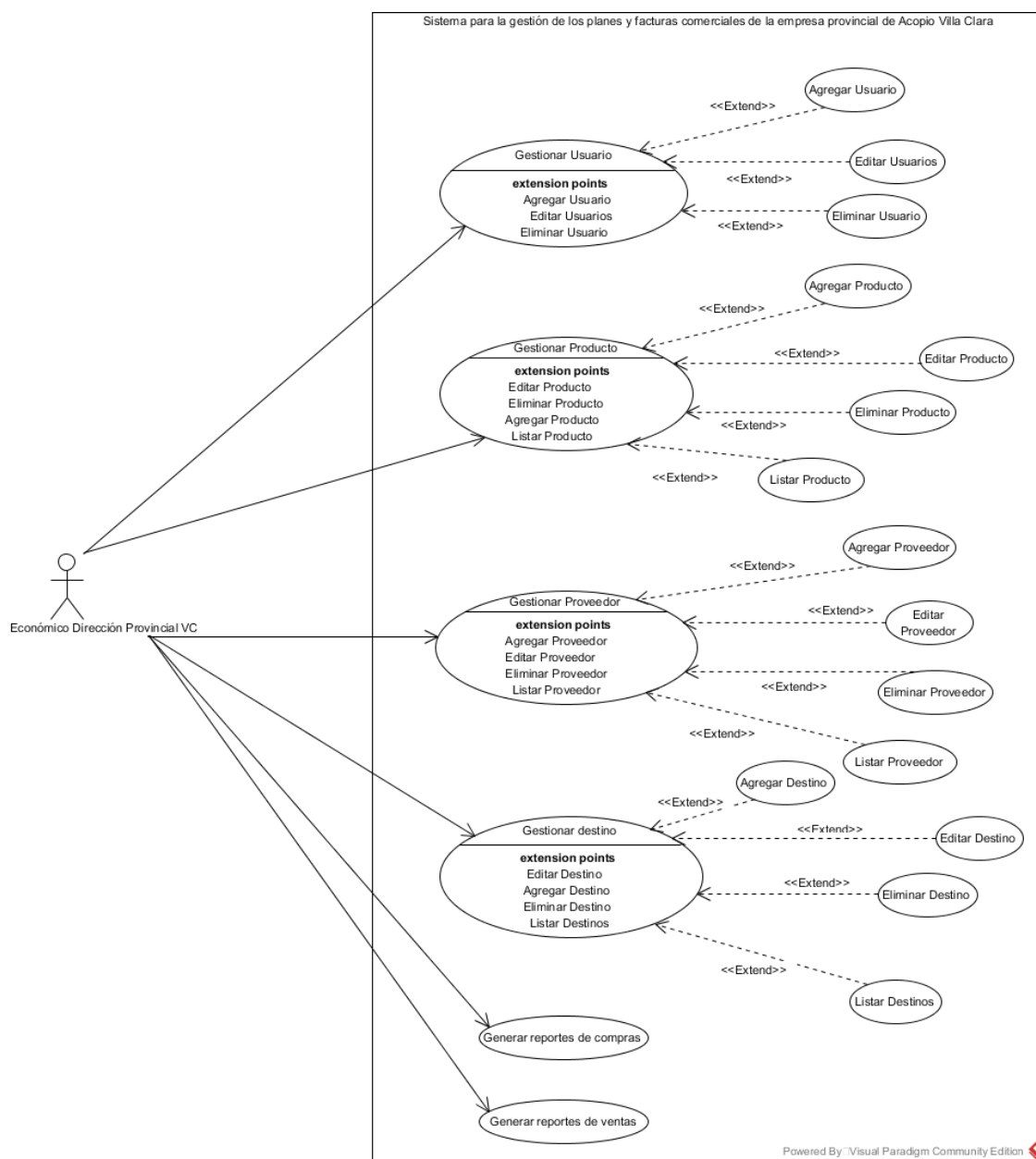


Ilustración 3: Diagrama de Casos de Uso del Sistema 2.

2.5 Casos de Uso significativos del Sistema.

El sistema presenta los siguientes casos de uso significativos:

Gestionar facturas de compra: Este caso de uso permite que la UEB una vez realizada una compra a un proveedor realice la factura derivada de la misma, con sus datos pertinentes y la agregue al sistema, de manera que cuando en la Dirección Provincial se realice el reporte de las compras quede registrada la información de dicha actividad.

Gestionar facturas de venta: Este caso de uso permite que la UEB una vez realizada una venta a un destino realice la factura derivada de la misma, con todos sus datos agregados al sistema y por consiguiente a la base de datos para su posterior consulta o acceso. Ver [Anexo 3](#).

Gestionar los Productos: En este caso de uso la Dirección Provincial es la encargada de incorporar los productos que se comercializan en la empresa, modificarlos o eliminar aquellos que en algún momento dejen de ser comercializados. Todos los productos existentes son listados para facilitar el control de la información sobre los mismos. Ver [Anexo 3](#).

2.5.1 Descripción de los Casos de Uso significativos del Sistema.

En las siguientes tablas se muestran las descripciones de los casos de uso más significativos:

Gestionar facturas de compra.

Caso de Uso del Sistema	Gestionar las facturas de compra.
Actor	UEB
Propósito	Realizar las facturas de las compras que se realizan.
Resumen	Este caso de uso inicia una vez que una UEB, accede al sistema, en la sección de facturas de compra donde va a encontrar un listado con sus facturas existentes, procedente de alguna actividad de compra, luego puede proceder a agregar una nueva

	factura o en algún caso puntual eliminar alguna.
Responsabilidad	Gestionar las facturas de compra.
Casos de uso asociados.	Gestionar Productos. Gestionar Proveedores.
Precondiciones	Antes de iniciar el caso de uso del sistema, la UEB debe tener la aplicación en pleno funcionamiento y la conexión segura y estable.

Interfaz Gestionar Facturas de Compra

The screenshot displays the 'Gestionar Facturas de Compra' interface. On the left, a sidebar contains navigation links: Inicio, Destinos, Productos, Proveedores, Compras (highlighted), and Ventas. The main area is titled 'Compras' and includes a 'Factura de compra +' button. Below this is a table with columns: Contrato, Productos, Usuario, Proveedor, Fecha, and Acción. The table lists 8 purchase contracts. A search bar and a 'Mostrar 10 elementos' dropdown are also present.

Contrato	Productos	Usuario	Proveedor	Fecha	Acción
07904a7c-19a7-360f-a599-aa800f970d74	0	UEB Remedios	Calixto Garcia	2016-10-19	[Iconos]
61c5e52d-401f-339c-a48b-2ab941ff6042	0	UEB Remedios	Jesus Menendez	1989-09-08	[Iconos]
663b4ca3-db2d-3294-9266-c77e61a3f514	0	UEB Remedios	Jesus Menendez	1995-12-07	[Iconos]
684910a1-ea80-318d-b9d4-67085dd6c932	0	UEB Remedios	Jose Marti	1982-08-07	[Iconos]
919374a4-bbdd-327d-a346-abf9afefbf52	0	UEB Remedios	Henry Reeve	2002-02-21	[Iconos]
a32144ee-24ca-3927-a6b8-8f23ed643b1b	0	UEB Remedios	Jesus Menendez	2002-10-09	[Iconos]
a747597b-e37f-372a-b688-4292bbb15148	0	UEB Remedios	Jesus Menendez	1990-05-31	[Iconos]
ade74c4d-2b29-35a1-b672-0ea68d518285	0	UEB Remedios	Ernesto Che Guevara	1989-11-18	[Iconos]

Flujo normal de eventos. Sección A: Nueva Factura

Acción de Actor	Respuesta del Sistema
1. El usuario (UEB) selecciona Adicionar factura	2. El sistema muestra la información referente con los campos que deben ser

3. La UEB llena los campos con los datos correspondientes a dicha actividad.	<p>llenados para realizar la factura por parte de la UEB.</p> <p>4. El sistema valida y envía el registro guardado a la base de datos.</p> <p>5. El sistema notifica de la correcta creación de la factura.</p>
Flujo alternativo 1	
<p>3. La UEB llena los campos con los datos y presiona adicionar.</p> <p>5. Verifica todos los campos, corrige y luego actualiza para proceder a guardar.</p>	4. El sistema notifica de algún error por datos mal introducidos el algún campo.
<u>Flujo normal de eventos. Sección B: Eliminar Factura.</u>	
Acción del actor.	Respuesta del sistema
<p>1. El usuario (UEB) se dirige hacia el botón Eliminar y hace clic.</p> <p>3. La UEB hace clic en el botón Aceptar.</p>	<p>2. El sistema muestra un cartel de confirmación de eliminación.</p> <p>4. El sistema procede a eliminar la factura con todos sus datos, y actualiza la acción en la base de datos.</p> <p>5. Muestra mensaje de éxito de eliminación.</p>

Post condiciones.	La base de datos debe quedar actualizada de acuerdo a las operaciones realizadas en ella.
-------------------	-------------------------------------------------------------------------------------------

Tabla 4: Descripción Caso de Uso Gestionar facturas de compras.

2.6 Estimación por puntos de casos de uso.

La estimación de costo y esfuerzo del software nunca será una ciencia exacta, demasiadas variables (humanas, ambientales, técnicas, políticas) pueden afectar el costo final del software y el esfuerzo aplicado para su desarrollo, Sin embargo, la estimación del proyecto de software puede transformarse de un arte oscuro a una serie de pasos sistemáticos que proporcionen estimaciones con riesgo aceptable.(Pressman, 2010).

Puntos de caso de uso es un método de estimación de esfuerzo para proyectos de software, a partir de sus casos de uso. El método utiliza los actores y los casos de uso relevantes para calcular el esfuerzo que significará desarrollarlos.

Una vez realizado el método se concluyó que el tiempo de duración estimado es de aproximadamente 23 semanas con una sola persona en el proyecto. Para observar el desarrollo del método ver [Anexo2](#).

Conclusiones parciales del capítulo:

- Con la realización de este capítulo se ha comprendido mejor el modelo del sistema a realizar y las restricciones planteadas por los clientes que tendrá dicho sistema.
- Se realizó el modelo de negocio que incluyó el diagrama de casos de uso del negocio, en el cual se representaron los actores del negocio y su relación con los casos de uso del negocio.
- Se identificaron las características que tendrá el sistema, derivadas estas de los requisitos funcionales y no funcionales planteados por los clientes.
- Identificadas dichas características se realizó el modelo de casos de uso del sistema en el cual se representaron los actores y su relación entre los casos de uso del sistema.

- Se realizó una descripción textual de los casos de uso del sistema significativos detallando en ellas las características que va a tener cada proceso automatizado.
- Se realizó la estimación por puntos de casos de uso lo cual permitió obtener una estimación del tiempo total de desarrollo de la aplicación.

Capítulo 3: Descripción de la propuesta de solución.

En el presente capítulo se realizará una descripción de la solución propuesta para el desarrollo del sistema, resaltando los aspectos más relevantes para realizar la gestión de los planes y facturas comerciales de la empresa. Se hace una descripción de la arquitectura del sistema utilizada para la implementación, se diseñará el modelado físico y conceptual de la base de datos a utilizar, el cual refleja las tablas de la Base de Datos con sus relaciones para almacenar los datos del sistema y se definirán los diagramas de clases y secuencia de los casos de uso arquitectónicamente significativos.

3.1 Arquitectura del Software.

En el libro "An introduction to Software Architecture", (Garlan & Shaw, 1994) definen que la Arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema".

Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software, define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales.

3.2 Patrón arquitectónico Modelo-Vista-Controlador (MVC).

La arquitectura del sistema a desarrollar será descrita a través del patrón arquitectónico Modelo-Vista-Controlador (MVC), que implementa el framework de desarrollo Laravel, este separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define

componentes para la representación de la información, y por otro lado para la interacción del usuario.

De manera genérica, los componentes del patrón MVC se podrían definir como la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

Vista: La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

Controlador: El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

(Y. F. R. Y. D. González, 2012)

En la siguiente figura se muestra una representación esquemática del patrón arquitectónico de diseño modelo-vista-controlador.

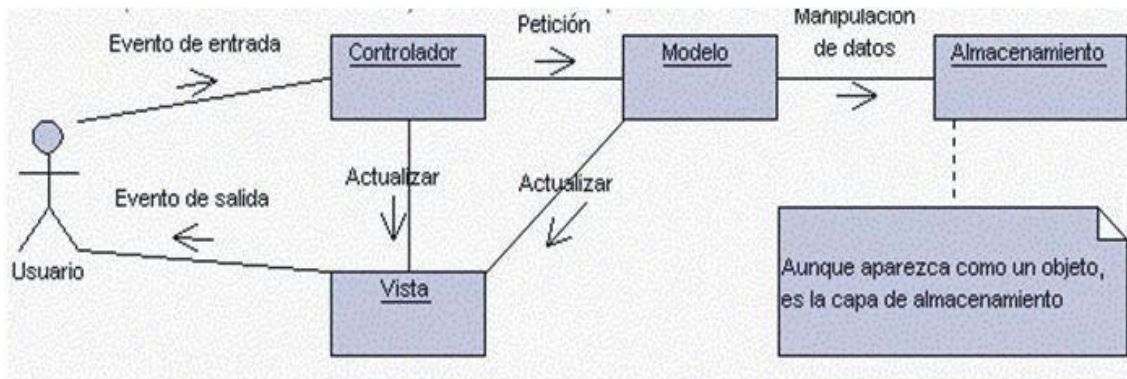


Ilustración 4: Interrelación entre los elementos del patrón MVC.

En la figura anterior se representa el flujo de control implementado por el patrón MVC el cual tiene como secuencia:

1. El usuario interactúa con la interfaz de usuario de la aplicación.
2. El controlador recibe una notificación de la petición realizada por el usuario y gestiona dicho evento.
3. El controlador accede al modelo, lo actualiza o modifica en correspondencia a la petición enviada por el usuario.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo.
5. La interfaz de usuario espera nuevas interacciones del usuario, y así se da comienzo al ciclo nuevamente.

En la siguiente se muestra la arquitectura del sistema:

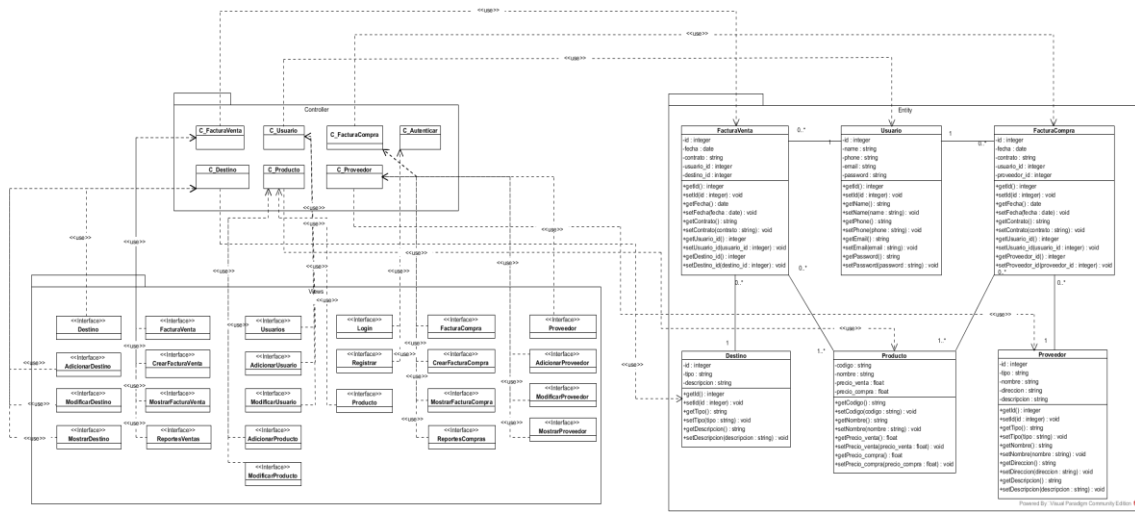


Ilustración 5: Arquitectura del sistema.

3.3 Diagrama de clases de diseño.

El diagrama de clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. En UML, una clase representa un objeto o un conjunto de objetos que comparte una estructura y un comportamiento comunes. Se representan con un rectángulo que incluye filas del nombre de la clase, sus atributos y sus operaciones. Este tiene como objetivo comunicar el diseño de un programa orientado a objetos.

En la siguiente figura se muestra el diagrama de clases de diseño para el sistema a implementar:

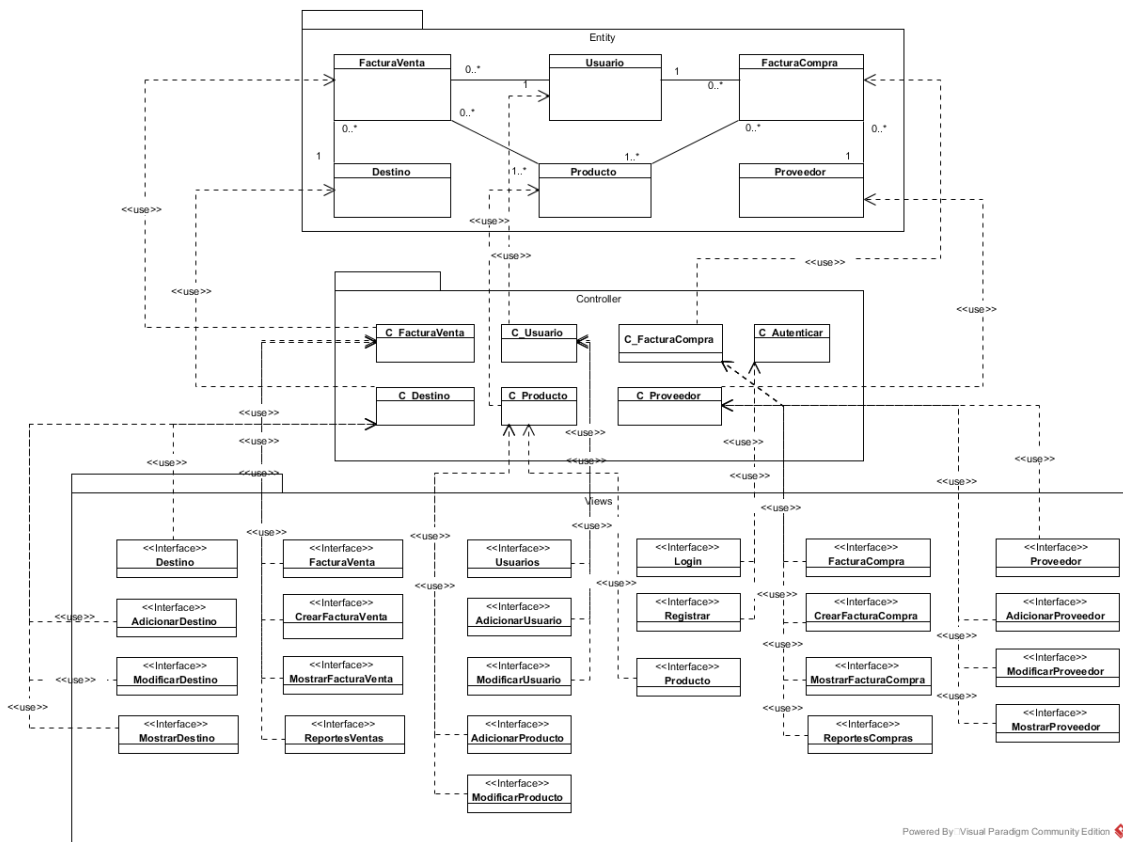


Ilustración 6: Diagrama de clases de diseño.

En la siguiente figura se muestra la vista del diagrama de clases de diseño para las clases de almacenamiento de datos:

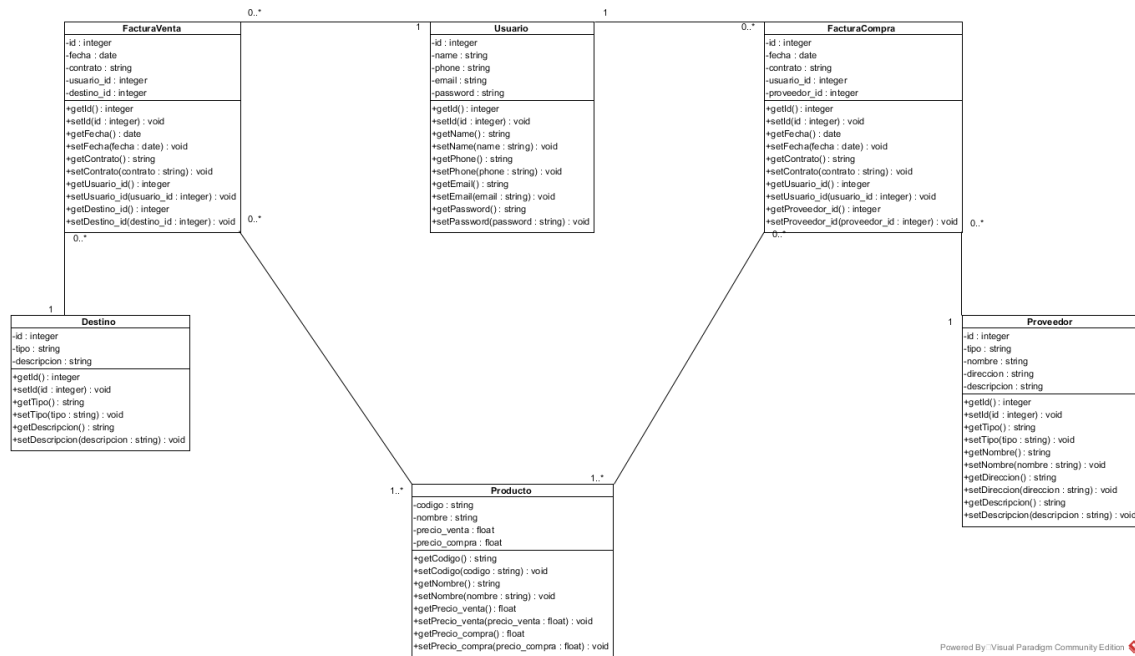


Ilustración 7: Diagrama de clases de diseño para las clases entidad.

En la figura anterior se muestra el diagrama de clases de diseño, el cual es una representación de las clases entidad persistentes dentro de la aplicación, estas son las clases utilizadas para el almacenamiento de los datos de la aplicación, así como sus relaciones entre si y la cardinalidad de las mismas.

3.4 Diagrama de secuencia.

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista “negocio” del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

La siguiente figura muestra el diagrama de secuencia para el caso de uso Modificar productos.

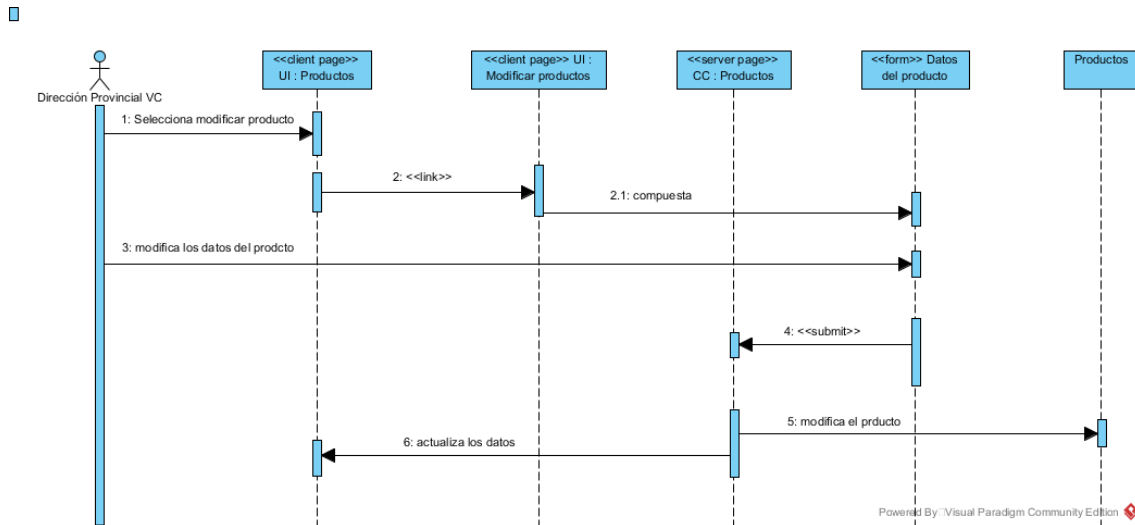


Ilustración 8: Diagrama de clases de diseño para el caso de uso Modificar Producto.

En el diagrama anterior se observa el diagrama de secuencia para el caso de uso modificar productos. Este proceso tiene comienzo cuando el usuario “Dirección Provincial” con el permiso para realizar dicha actividad presiona la opción de modificar en un producto determinado dentro de la propia interfaz de producto, luego aparece una ventana con los datos del producto seleccionado en sus respectivos campos, el usuario realiza las modificaciones deseadas y presiona guardar. Si los datos entrados están correctos se guardan las modificaciones realizadas en la base de datos y queda actualizada la misma.

3.5 Diseño y modelado de la base de datos

El modelado de datos es el proceso de visualización y representación de datos para su almacenamiento en un almacén de datos. El modelado en si puede incluir diagramas, símbolos o textos para representar los datos y la forma en que se interrelacionan. Debido a la estructura que el modelado de datos impone a los datos, el proceso de modelado aumenta posteriormente la coherencia en la denominación, las reglas, la semántica y la seguridad, al tiempo que mejora el análisis de los datos. El objetivo es ilustrar los tipos de datos utilizados y almacenados en el sistema, las relaciones entre estos tipos de datos, las formas en las que pueden agruparse y organizarse y sus formas y atributos.

El diseño de una base de datos es de suma importancia ya que de ello dependerá que los datos estén correctamente actualizados y la información siempre sea exacta. Si hacemos un buen diseño de base de datos podremos obtener reportes efectivos y eficientes.

Al diseñar una base de datos determinamos las tablas y los campos que darán forma a nuestra base de datos. El hecho de tomarse el tiempo necesario para identificar, organizar y relacionar la información evitara problemas posteriores. Es por ello que para el diseño es necesario conocer la problemática y todo el contexto sobre la información que se almacenará en nuestro repositorio de datos, se debe determinar la finalidad de la base de datos y en base a ello reunir toda la información que será registrada.

3.5.1 Modelo conceptual de la Base de Datos.

Un modelo conceptual es una representación de un sistema, hecho de la composición de conceptos que se utilizan para ayudar a las personas a conocer, comprender o simular un tema que representa el modelo, incluye las entidades importantes y las relaciones entre ellos. También es un conjunto de conceptos. Algunos modelos son objetos físicos.

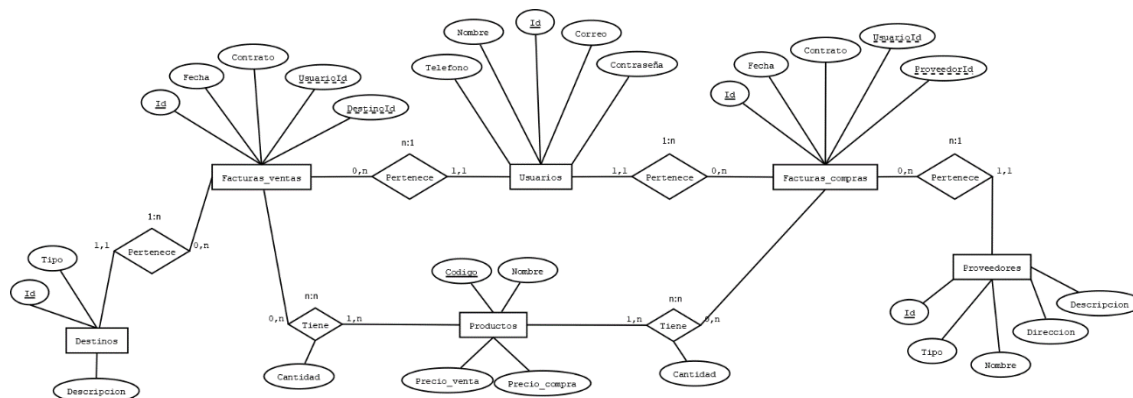


Ilustración 9: Modelo conceptual de la Base de Datos.

3.5.2 Modelo entidad-relación de la Base de Datos.

Un diagrama entidad-relación, también conocido como modelo entidad relación, es un tipo de diagrama de flujo que ilustra como las entidades, se relacionan entre si dentro de un sistema.

En la siguiente figura se muestra el modelo entidad-relación de los datos del sistema.

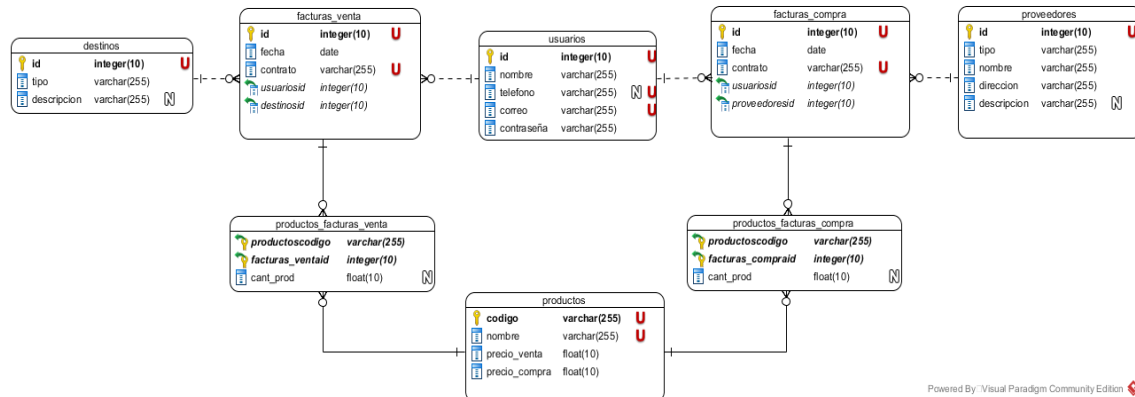


Ilustración 10: Modelo entidad-relación de la Base de Datos.

3.5.3 Modelado lógico de la Base de Datos.

Un modelo de datos lógicos describe los datos con el mayor detalle posible, independientemente de cómo se implementarán físicamente en la base de datos. Estos incluyen todas las entidades y relaciones entre ellas, y todos los atributos para cada entidad están especificados.

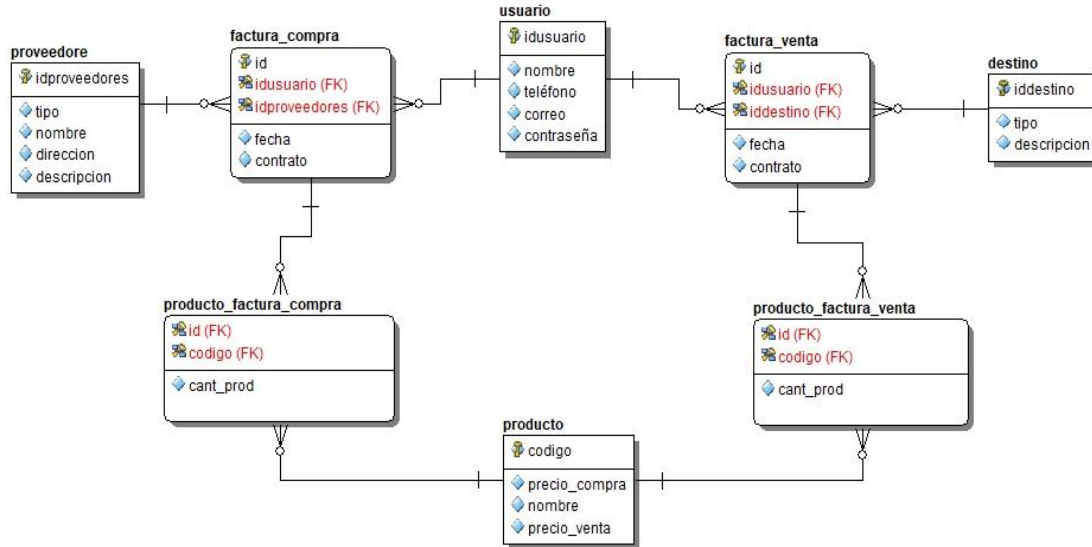


Ilustración 11: Modelo lógico de la Base de Datos.

3.6 Diagrama de despliegue.

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar la disposición física de los artefactos software en nodos. Estos son los complementos de los diagramas de componentes que, unidos, proveen la vista de implementación del sistema, de manera general muestran el ambiente de computación.

En la siguiente figura se muestra el diagrama de despliegue:

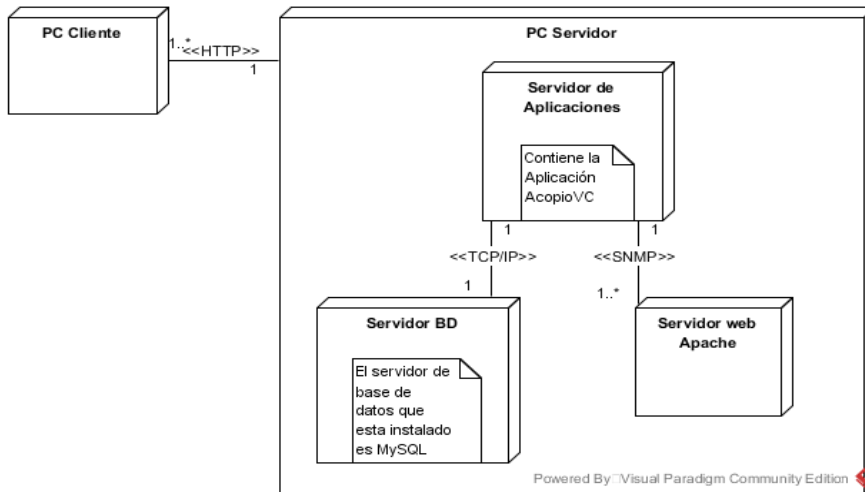


Ilustración 12: Diagrama de despliegue.

3.7 Diagrama de despliegue y modelo de componentes.

En este diagrama se describe la topología del sistema, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos, este representa a los nodos y sus relaciones. Un diagrama de componentes representa las dependencias entre componentes de software, dígame componentes de código fuente y componentes ejecutables.

En la siguiente figura se muestra el diagrama de despliegue y el modelo de componentes:

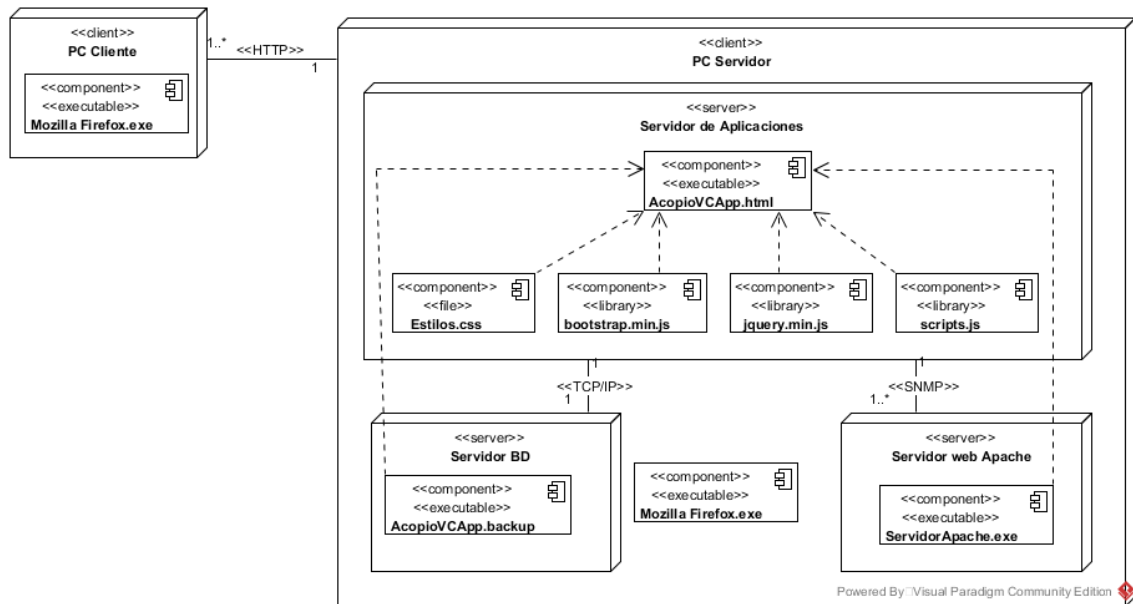


Ilustración 13: Diagrama de despliegue y modelo de componentes.

En el diagrama de despliegue y el modelo de componente anterior se visualiza la presencia de dos nodos clientes relacionados entre sí, el primero hace referencia a las PC de los usuarios que disponen de su navegador para acceder a la aplicación, mientras que el otro contiene un nodo servidor con los diferentes componentes relacionados con la aplicación además de contar también con un componente navegador. La estructura de esta representación se debe a que la empresa Acopio Villa Clara se cuenta con servidor de aplicaciones en una maquina cliente independiente y las demás máquinas acceden a él.

Conclusiones parciales.

- En el presente capítulo se obtuvo un modelado de las clases del sistema con sus principales características utilizando la notación UML para ayudar a la interpretación y comprensión de la estructura.
- Como arquitectura del sistema se utilizó el patrón arquitectónico y de diseño modelo-vista-controlador el cual ofrece el framework Laravel para desarrollo de aplicaciones

web el cual separa los datos, la lógica y la interfaz para una mejor organización a la hora de la implementación.

- Se realizó el diseño y modelado de la base de datos, utilizando como herramientas para el modelado el Visual Paradigm y ER/Studio Embarcadero para representar cada una de las tablas y sus relaciones entre ellas.
- Se realizó el diagrama de despliegue junto con el modelo de componentes para ver las disposición, relación y dependencias entre los componentes de software ubicados en sus respectivos nodos.

Conclusiones

A raíz del desarrollo del presente trabajo se puede arribar a las siguientes conclusiones:

- ✓ Se implementó el Proceso Unificado Ágil como metodología de software para llevar a cabo el desarrollo del software, utilizando los flujos de trabajo y artefactos que proporciona.
- ✓ Se desarrolló una arquitectura de sistema basada en el patrón arquitectónico de diseño modelo-vista-controlador que facilitó la separación de estos componentes para obtener una mejor estructura y organización con el código fuente.
- ✓ Se diseñó una base de datos que permitió el almacenamiento y gestión de la información de las actividades comerciales de compra y venta de la empresa Acopio.
- ✓ Se concibió una interfaz intuitiva y sencilla para proporcionar una mejor interacción y fácil manejo de los usuarios con el sistema.
- ✓ Se llevó a cabo un plan de pruebas para verificar y asegurar el cumplimiento de manera satisfactoria de los requisitos funcionales y estructurales del software, culminando con las pruebas de aceptación por parte del cliente final.

Recomendaciones

A partir de los estudios realizados y con el fin de darle continuidad al presente trabajo, se recomienda:

- ✓ Continuar dándole soporte al sistema para la implementación de futuros módulos con el objetivo de llevar todos los procesos comerciales que realiza la empresa hacia una plataforma común.

Referencias Bibliográficas

- Bertino, E., & Martino, L. (1995). *Sistemas de bases de datos orientadas a objetos*: Díaz de Santos.
- Booch, G., Rumbaugh, J., & Jacobson, I. (2006). *El lenguaje unificado de modelado*.
- Cardozo, L. (2014). SUIZO ARGENTINO. *Informatizacion de procesos. Herramientas de desarrollo*. Retrieved from <https://suizoargentina.com/informatizacion-de-procesos-herramienta-de-desarrollo/>
- Cobo, Á., & Gómez, P. (2005). *PHP y MySQL-tecnologias para el desarrollo de aplicaciones web*: Ediciones Días de Santos.
- Donatien, A. R. (2011). *Descripción de la Metodología de Desarrollo de Software Agile Unified Process (AUP)*. Retrieved from
- Drake, J. M. (2008). *Análisis de requisitos y especificación de una aplicación*.
- Flanagan, D., & Ferguson, P. (2002). *JavaScript: The Definitive Guide*.
- Fowler, M., & Scott, K. (1999). *UML Gota a Gota*. Retrieved from
- Garlan, D., & Shaw, M. (1994). *An introduction to Software Architecture*: School of Computer Science, Carnegie Mellon University.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*.
- González, D. F. (2011). *Análisis y Diseño del Sistema para la Gestión de las Aplicaciones Web en la Delegación Provincial del MININT*. Retrieved from Santa Clara:
- González, Y. F. R. Y. D. (2012). Patrón Modelo-Vista-Controlador. *Revista Telemática*, Vol. 11. No. 1.
- Hernández, M. T. (2015). *Fymfony Framework. Desarrollo Rápido de Aplicaciones Web*: IT Campus Academe.
- Jacobson, I., Spence, I., & Bittner, K. (2013). *CASOS DE USO 2.0 La guía para ser exitoso con los casos de uso*.

- Menéndez Mueras, R. (2005). *Construcción de software orientado a objetos con el proceso unificado y el UML, un punto de vista práctico.*: Universidad Continental.
- Mora, S. L. (2001). *Programación en Internet: Clientes Web*: Editorial Club Universitario.
- Oqotech (Producer). (2017, Noviembre 23). Oqotech. *Guía práctica para la informatización de procesos de negocio*. Retrieved from <https://www.oqotech.com/blog/e-books/guia-practica-informatizacion-de-procesos-de-negocio/>
- Otwell, T. (Producer). (2011). Laravel Retrieved from <http://www.laravel.com>
- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico*.
- Riehle, D. (2000). *Framework Design: A Role Modeling Approach*.
- Robbins, J. N. (2010). *HTML5 Pocket Reference*.
- Welling, L., & Thomson, L. (2009). *Desarrollo web con PHP y MySQL*: ANAYA MULTIMEDIA.

Anexos

Anexo 1.

Manual de usuario de la aplicación con las principales interfaces de interacción y descripción de los componentes:

Interfaz para realizar el login para acceder al sistema, para ello es necesario escribir las credenciales correo electrónico y contraseña.

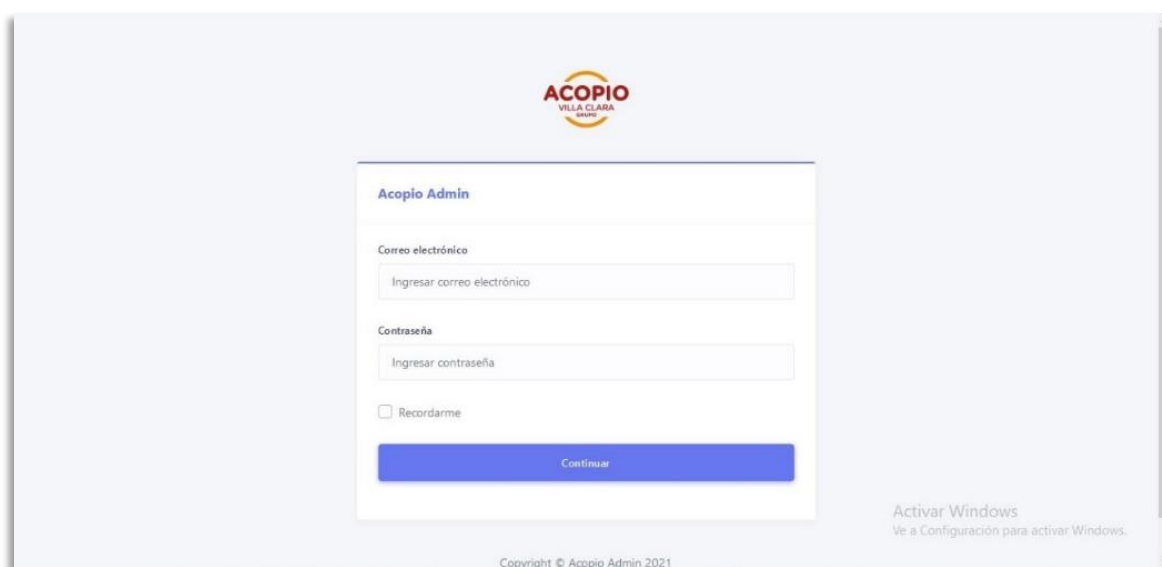


Ilustración 14: Interfaz de login.

Interfaz de Usuario:

Únicamente disponible para el rol de Dirección Provincial que es el único con el privilegio de gestionar los usuarios que pueden acceder al sistema.

1. Selección Usuarios en el menú lateral.
2. Botón para mostrar detalles del usuario seleccionado.
3. Botón para editar Usuario.
4. Botón para eliminar usuario.
5. Campo para filtrar Usuario por datos.

6. Botón para agregar un nuevo Usuario.

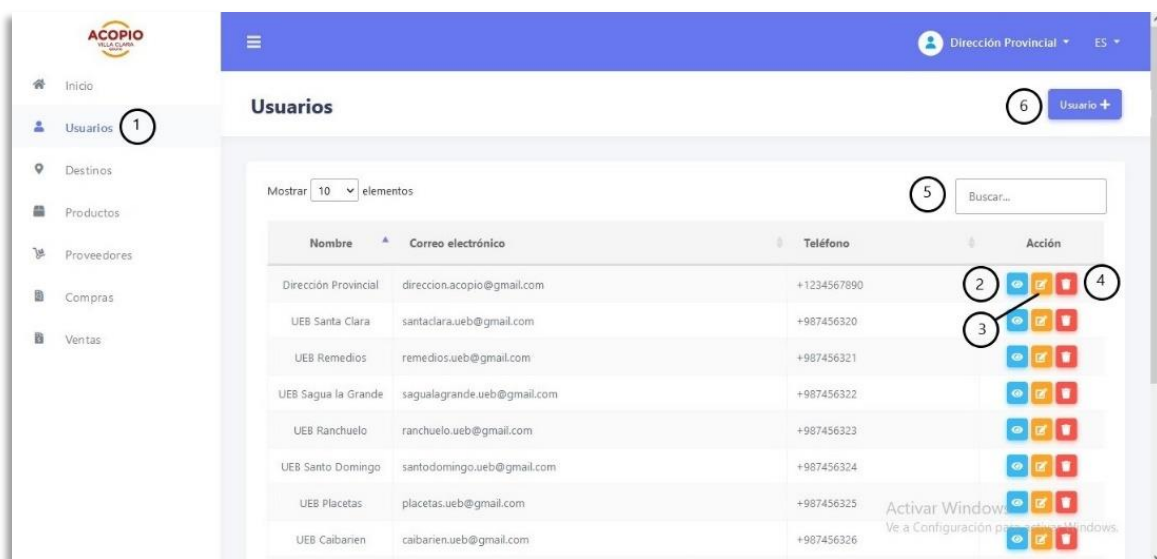


Ilustración 15: Interfaz Usuario, solo válida para el rol Dirección Provincial.

Interfaz Destinos para rol Dirección Provincial:

1. Selección Destinos en el menú lateral.
2. Botón para mostrar detalles del Destino seleccionado.
3. Botón para editar Destino.
4. Botón para eliminar Destino.
5. Campo para filtrar Destino por datos.
6. Botón para agregar un nuevo Destino.

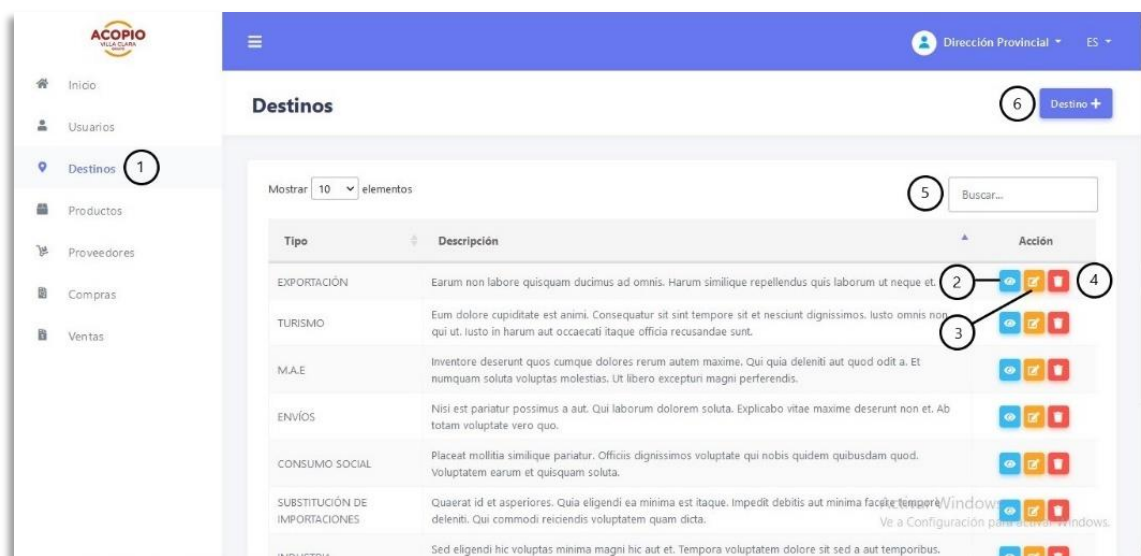


Ilustración 16: Interfaz Destinos para el rol Dirección Provincial.

Interfaz Productos para rol Dirección Provincial:

1. Selección Productos en el menú lateral.
2. Botón para mostrar detalles del Producto seleccionado.
3. Botón para editar Producto.
4. Botón para eliminar Producto.
5. Campo para filtrar Producto por datos.
6. Botón para agregar un nuevo Producto.

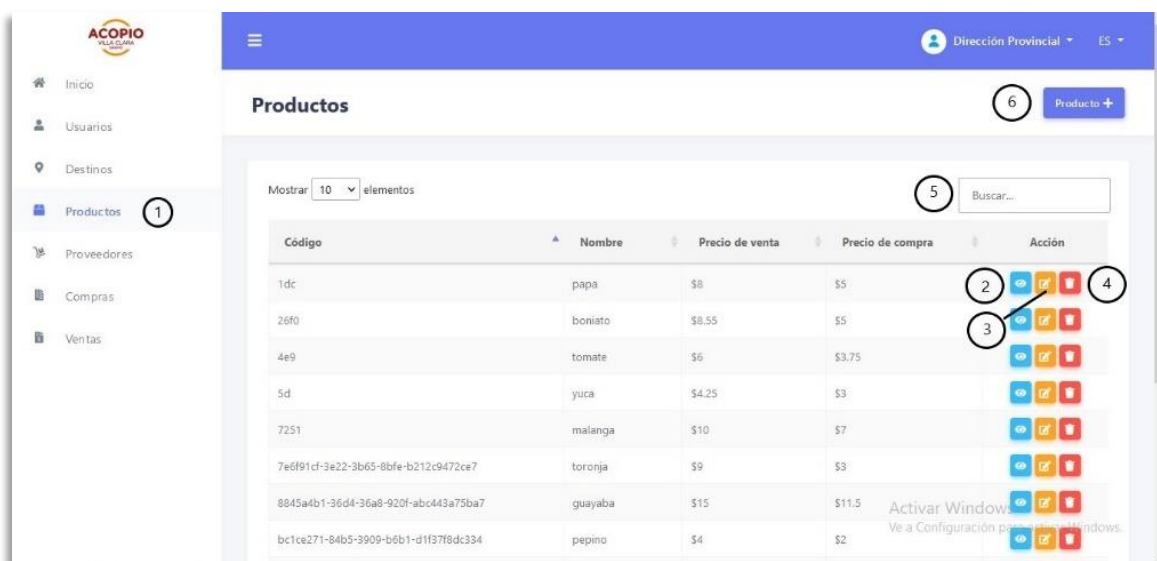


Ilustración 17: Interfaz Productos para rol Dirección Provincial.

Interfaz Proveedores para rol Dirección Provincial:

1. Selección Proveedores en el menú lateral.
2. Botón para mostrar detalles del Proveedor seleccionado.
3. Botón para editar Proveedor.
4. Botón para eliminar Proveedor.
5. Campo para filtrar Proveedor por datos.
6. Botón para agregar un nuevo Proveedor.

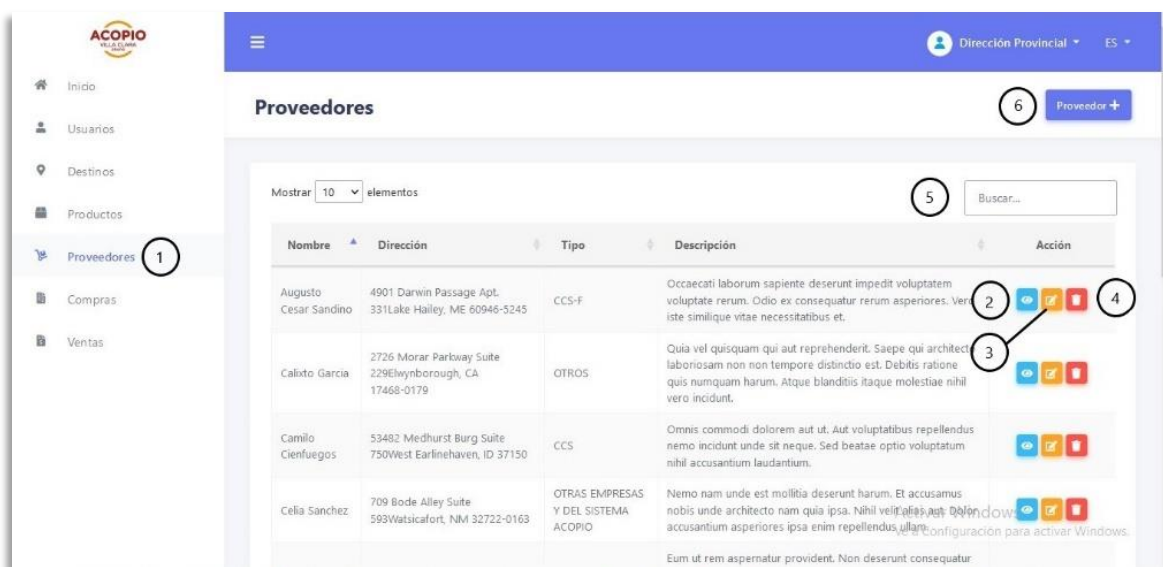


Ilustración 18: Interfaz Proveedores para rol Dirección Provincial.

Interfaz Compras para rol UEB:

El rol UEB es el encargado de realizar las compras en el sistema mediante las facturas de compra.

1. Selección Compras en el menú lateral.
2. Botón para mostrar detalles de la Compra seleccionada.
3. Botón para eliminar Compra.
4. Campo para filtrar Compra por datos.
5. Botón para agregar una nueva Compra.

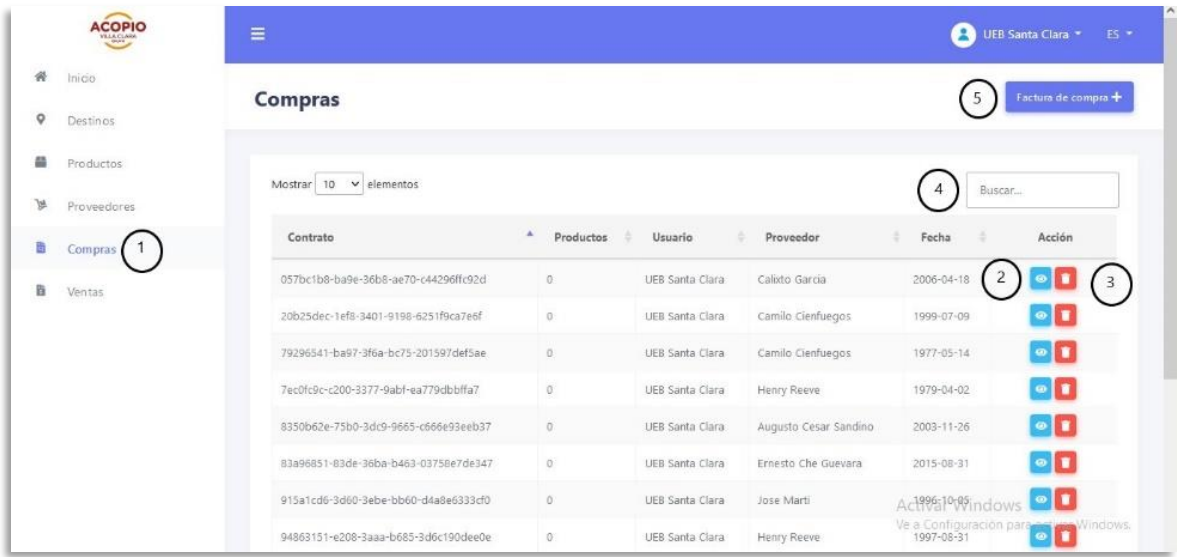


Ilustración 19: Interfaz Compras para rol UEB.

Interfaz Compras para rol Dirección Provincial:

1. Selección Compras en el menú lateral.
2. Botón para mostrar detalles de la Compra seleccionada.
3. Campo para filtrar Compra por datos.
4. Botón para generar reporte de Compras.

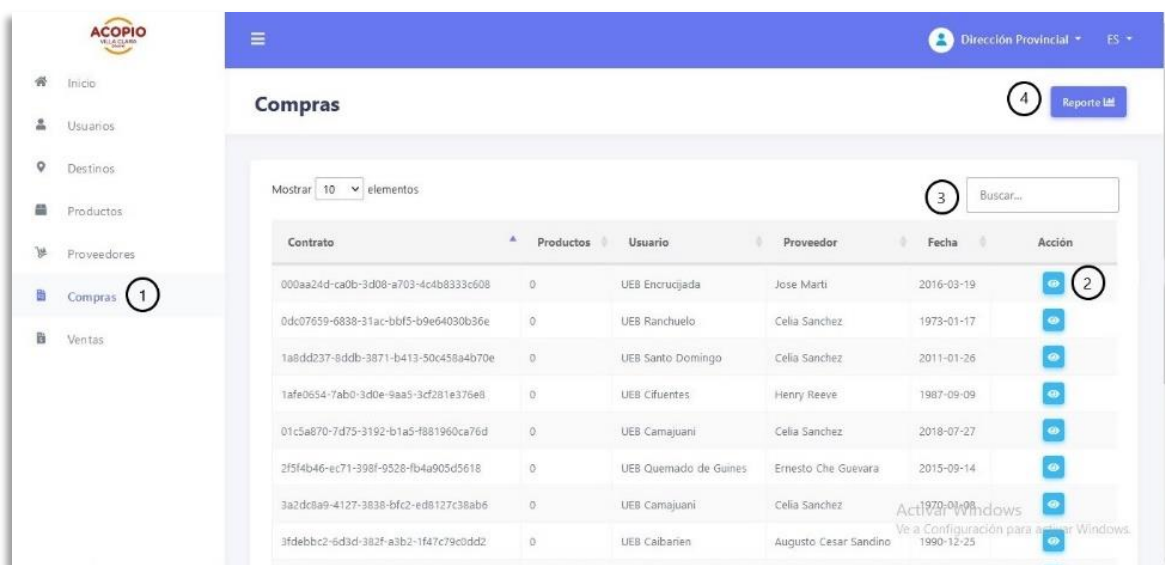


Ilustración 20: Interfaz Compras para rol Dirección Provincial.

Interfaz Ventas para rol UEB:

El rol UEB es el encargado de realizar las ventas en el sistema mediante las facturas de venta.

1. Selección Ventas en el menú lateral.
2. Botón para mostrar detalles de la Venta seleccionada.
3. Botón para eliminar Venta.
4. Campo para filtrar Venta por datos.
5. Botón para agregar una nueva Venta.

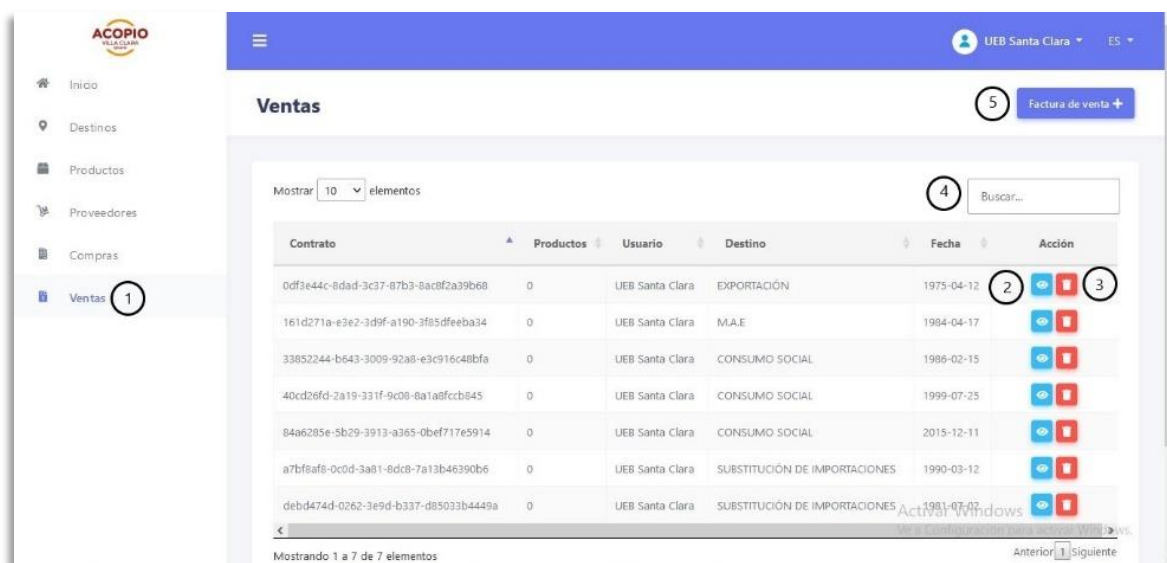


Ilustración 21: Interfaz Ventas para rol UEB.

Interfaz Ventas para rol Dirección Provincial:

1. Selección Ventas en el menú lateral.
2. Botón para mostrar detalles de la Venta seleccionada.
3. Campo para filtrar Venta por datos.
4. Botón para generar reporte de Compras

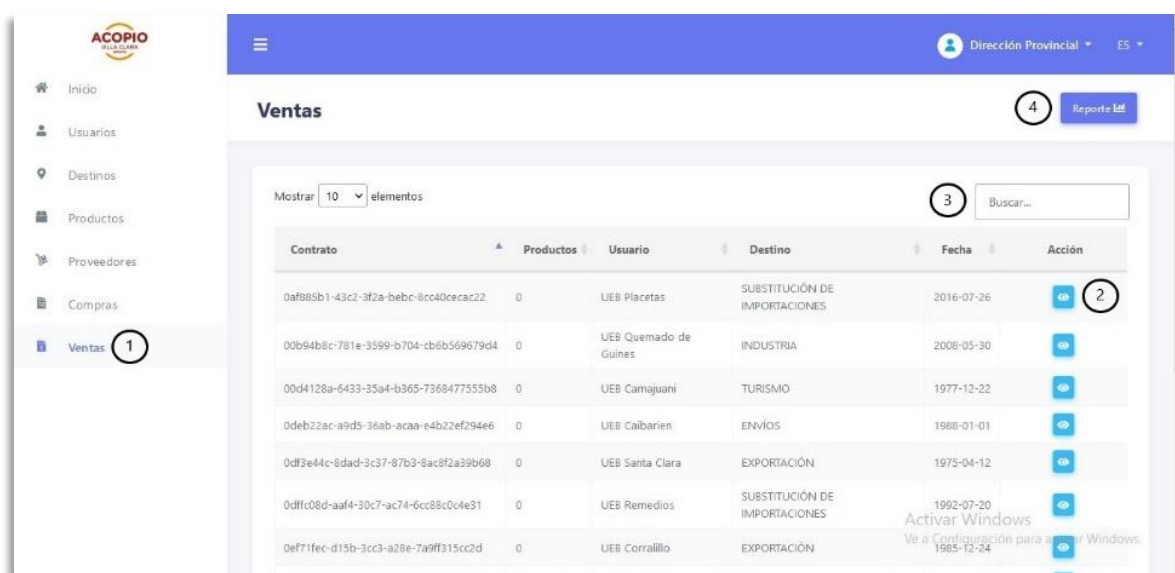


Ilustración 22: Interfaz Ventas para rol Dirección Provincial.

Anexo 2.

Desarrollo del método de estimación por casos de uso.

El método de puntos por casos de uso consta de cuatro etapas, en las que se desarrollan los siguientes cálculos:

1. Factor de peso de los actores sin ajustar (UAW).
2. Factor de peso de los casos de uso sin ajustar (UUCW)
3. Puntos de caso de uso ajustados (UCP)
4. Esfuerzo horas-hombre.

Puntos de caso de uso sin ajustar (UUCP)

Al inicio de un proyecto de software, cuando apenas se conocen los casos de uso y sus actores asociados, se puede proyectar una breve descripción de cada caso de uso, en el cual se describe de forma breve la funcionalidad que éste debe brindar.

El UUCP son los puntos de casos de uso sin ajustar, esto nos puede servir para tener una idea un poco más precisa de la dificultad de los casos de uso e interfaces, tomando en cuenta los pesos de los actores (UAW) y los pesos de los casos de uso (UUCW). $UUCP = UAW + UUCW$

Estas siglas significan:

- UUCP: Puntos de casos de uso sin ajustar.
- UAW: Factor de peso de los actores sin ajustar.
- UUCW: Factor de peso de los casos de uso sin ajustar.

Aplicando el análisis de puntos de función a estos casos de uso, se puede obtener una estimación trivial del tamaño y a partir de ella una estimación del esfuerzo.

A continuación, se realiza la estimación por puntos de casos de uso:

1. Clasificación de los actores:

Tipo de actor	Descripción	Factor
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API).	1
Medio	Otro sistema interactuando a través de un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).	3

Tabla 5: Peso de actores sin ajustar.

Fórmula: $UAW = \text{Sum}(\text{cantidadDeUnTipoDeActor} * \text{Factor})$

Se tienen dos actores de un tipo complejo con un factor de 3.

$$UAW = 2 * 3 = 6$$

2. Clasificación de los casos de uso:

Para determinar el nivel de complejidad realizada mediante el método basado en transacciones. Toma en cuenta el número de transacciones que se pueden realizar en un caso de uso y lo evalúa según la siguiente tabla:

Tipo de caso de uso	Descripción	Factor
Simple	3 transacciones o menos	5
Medio	4 a 7 transacciones	10
Complejo	Más de 7 transacciones	15

Tabla 6: Peso de las transacciones.

Fórmula: $UUCW = \text{Sum}(\text{cantidadDeUnTipoDeCasoUso} * \text{Factor})$

Se tienen 8 casos de uso de tipo simple con un factor de 5.

$$UUCW = 8 * 5 = 40$$

3. Calcular puntos de caso de uso sin ajustar:

Fórmula: $UUCP = UAW + UUCW$

$$UUCP = 6 + 40 = 46$$

Puntos de caso de uso ajustados (UCP)

Para esto se utilizan las siglas UCP y se obtiene al multiplicar el UUCP el TCF y el EF quedando la operación de la siguiente forma:

$$UCP = UUCP \times TCF \times EF$$

Estas siglas significan:

- UCP: Puntos de casos de uso ajustados.
- UUCP: Puntos de casos de uso sin ajustar.
- TCF: Factores técnicos.
- EF: Factores ambientales.

Determinar los factores de complejidad técnica:

Este se compone de 13 puntos que evalúan la complejidad de los módulos del sistema que se desarrolla, cada uno de estos factores tienen un peso definido con los cuales se obtendrá puntos ponderados por cada uno de ellos.

Factor	Descripción	Peso
T1	Sistema distribuido.	2
T2	Objetivos de performance o tiempo de respuesta.	1
T3	Eficiencia del usuario final.	1
T4	Procesamiento interno complejo.	1
T5	El código debe ser reutilizable.	1
T6	Facilidad de instalación.	0.5
T7	Facilidad de uso.	0.5
T8	Portabilidad.	2
T9	Facilidad de cambio.	1
T10	Concurrencia.	1
T11	Incluye objetivos especiales de seguridad.	1
T12	Provee acceso directo a terceras partes.	1
T13	Se requiere facilidades especiales de entrenamiento a usuario.	1

Tabla 7: Peso de los factores de complejidad técnica.

Cada uno de estos puntos se debe evaluar según la siguiente escala:

Descripción	Valor
Irrelevante	De 0 a 2.
Medio	De 3 a 4.
Esencial	5

Tabla 8: Escala de los factores de complejidad técnica.

Las fórmulas para este punto son:

- $TFactor = \text{Sum}(\text{Valor} * \text{Peso})$
- $TCF = 0.6 + (0.01 * TFactor)$

$$TFactor = \text{Sum}(\text{Valor} * \text{Peso})$$

$$TFactor = [(2*0) + (1*1) + (1*1) + (1*1) + (4*1) + (2*0.5) + (2*0.5) + (1*2) + (3*1) + (0*1) + (5*1) + (0*1) + (1*1)]$$

$$TFactor = (0+1+1+1+4+1+1+2+3+0+5+0+1)$$

$$TFactor = 20$$

$$TCF = 0.6 + (0.01 * TFactor)$$

$$TCF = 0.6 + (0.01 * 20)$$

$$TCF = 0.6 + 0.20$$

$$TCF = 0.80$$

Determinar los factores ambientales:

Los factores sobre los cuales se realiza la evaluación son 8 puntos, que están relacionados con las habilidades y experiencia del grupo de personas involucradas con el desarrollo del proyecto.

Factor	Descripción	Peso
E1	Familiaridad con el modelo de proyecto utilizado.	1.5
E2	Experiencia en la aplicación.	0.5
E3	Experiencia en orientación a objetos.	1
E4	Capacidad del analista líder.	0.5
E5	Motivación.	1
E6	Estabilidad de los requerimientos	2
E7	Personal part-time	-1
E8	Dificultad del lenguaje de programación	-1

Tabla 9: Peso de los factores ambientales.

Cada uno de estos factores se debe calificar con un valor de 0 a 5.

Las fórmulas para este punto son:

- $E_{Factor} = \text{Sum (Valor * Peso)}$
- $EF = 1.4 + (-0.03 * E_{Factor})$

$$E_{Factor} = \text{Sum (Valor * Peso)}$$

$$E_{Factor} = [(4*1.5) + (2*0.5) + (3*1) + (4*0.5) + (5*1) + (3*2) + (-1*2) + (-1*4)]$$

$$E_{Factor} = (6 + 1 + 3 + 2 + 5 + 6 - 2 - 4)$$

$$E_{Factor} = 17$$

$$EF = 1.4 + (-0.03 * E_{Factor})$$

$$EF = 1.4 + (-0.03 * 17)$$

$$EF = 1.4 - 0.51$$

$$EF = 0.89$$

Calcular los puntos de casos de uso ajustados:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 46 * 0.80 * 0.89$$

$$UCP = 32.752$$

Esfuerzo horas-hombre (E).

Este cálculo se realiza con el fin de tener una aproximación del esfuerzo, pensando solo en el desarrollo según las funcionalidades de los casos de uso y está basado en los factores ambientales.

Se debe contar la cantidad de factores ambientales del E1 al E6 que tienen una puntuación menor a 3, también contar la cantidad de estos mismos del E7 y E8 que son mayores que 3. Para evaluar el resultado o la cantidad total según la siguiente tabla:

Horas-Persona (CF)	Descripción
20	Si el valor es ≤ 2
28	Si el valor es ≤ 4
36	Si el valor es ≥ 5

El esfuerzo en horas-persona viene dado por:

$$E = \text{UCP} \times \text{CF}$$

E: Esfuerzo estimado en horas-persona.

UCP: Puntos de Casos de Uso ajustados.

CF: Horas-Persona.

Calcular esfuerzo estimado:

$$E = \text{UCP} \times \text{CF}$$

$$E = 32.752 \times 20$$

$$E = 655.04 \text{ h/h}$$

E representa una parte del total del esfuerzo de todo el proyecto, asignándole un 60%. Este 60% se refiere al esfuerzo total para el desarrollo de las funcionalidades especificadas en los Casos de Uso.

En la siguiente tabla se detallan la distribución en porcentaje, para el esfuerzo total en el desarrollo del proyecto:

Actividad	Porcentaje
Análisis	10%

Diseño	20%
Programación	60%
Pruebas	10%

Calcular el esfuerzo total del proyecto:

$$\frac{655.04}{x} = \frac{60}{100}$$

$$X = 1091.7$$

El esfuerzo total del proyecto es de 1091.7 h/hombres.

Se trabaja de lunes a sábado un total de 8 horas al día.

$$8 * 6 = 48 \text{ horas.}$$

$$1091.7 / 48 = 22.74$$

El tiempo de duración estimado es de aproximadamente 23 semanas con una sola persona en el proyecto.

Anexo 3.

Descripción de casos de uso significativos.

Gestionar facturas de venta.

Caso de Uso del Sistema	Gestionar las facturas de compra.
Actor	UEB
Propósito	Realizar las facturas de las ventas que se realizan.
Resumen	Este caso de uso inicia una vez que una UEB, accede al sistema en la sección de

	facturas de venta, donde va a encontrar un listado con sus facturas existentes procedente de alguna actividad de venta, luego puede proceder a agregar una nueva factura o en algún caso puntual eliminar alguna.
Responsabilidad	Gestionar las facturas de venta.
Casos de uso asociados.	Gestionar Productos. Gestionar Destinos.
Precondiciones	Antes de iniciar el caso de uso del sistema, la UEB debe tener la aplicación en pleno funcionamiento y la conexión segura y estable.

Interfaz Gestionar Facturas de Venta

Contrato	Productos	Usuario	Destino	Fecha	Acción
0df3e44c-8dad-3c37-87b3-8ac8f2a39b68	0	UEB Santa Clara	EXPORTACIÓN	1975-04-12	[Icono]
161d271a-e3e2-3d9f-a190-3f85dfee8a34	0	UEB Santa Clara	M.A.E	1984-04-17	[Icono]
33852244-b643-3009-92a8-e3c916c48bfa	0	UEB Santa Clara	CONSUMO SOCIAL	1986-02-15	[Icono]
40cd26fd-2a19-331f-9c08-8a1a8fcb845	0	UEB Santa Clara	CONSUMO SOCIAL	1999-07-25	[Icono]
84a6285e-5b29-3913-a365-0bef717e5914	0	UEB Santa Clara	CONSUMO SOCIAL	2015-12-11	[Icono]
a7bf8af8-0c0d-3a81-8dc8-7a13b46390b6	0	UEB Santa Clara	SUBSTITUCIÓN DE IMPORTACIONES	1990-03-12	[Icono]
debd474d-0262-3e9d-b337-d85033b4449a	0	UEB Santa Clara	SUBSTITUCIÓN DE IMPORTACIONES	1991-07-02	[Icono]

Flujo normal de eventos. Sección A: Nueva Factura

Acción de Actor	Respuesta del Sistema
<p>1. El usuario (UEB) selecciona Adicionar factura</p> <p>3. La UEB llena los campos con los datos correspondientes a dicha actividad.</p>	<p>2. El sistema muestra la información referente con los campos que deben ser llenados para realizar la factura por parte de la UEB.</p> <p>4. El sistema valida y envía el registro guardado a la base de datos.</p> <p>5. El sistema notifica de la correcta creación de la factura.</p>
Flujo alternativo 1	
<p>3. La UEB llena los campos con los datos y presiona adicionar.</p> <p>5. Verifica todos los campos, corrige y luego actualiza para proceder a guardar.</p>	<p>4. El sistema notifica de algún error por datos mal introducidos el algún campo.</p>
<u>Flujo normal de eventos. Sección B: Eliminar Factura.</u>	
Acción del actor.	Respuesta del sistema
<p>1. El usuario (UEB) se dirige hacia el botón Eliminar y hace clic.</p> <p>3. La UEB hace clic en el botón Aceptar.</p>	<p>2. El sistema muestra un cartel de confirmación de eliminación.</p>

	<p>4. El sistema procede a eliminar la factura con todos sus datos, y actualiza la acción en la base de datos.</p> <p>5. Muestra mensaje de éxito de eliminación.</p>
Post condiciones.	La base de datos debe quedar actualizada de acuerdo a las operaciones realizadas en ella.
Flujo normal de eventos. Sección C: Mostrar Factura.	
Acción del actor.	Respuesta del sistema
1. El usuario se dirige a las acciones disponibles sobre la factura de venta y selecciona Mostrar.	2. El sistema muestra una vista con todos los datos de la factura seleccionada.

Tabla 10: Descripción Caso de Uso Gestionar facturas de ventas.

Gestionar Productos

Caso de Uso del Sistema	Gestionar Productos.
Actor	Dirección Provincial VC
Propósito	Gestionar los productos que se comercializan en la entidad.
Resumen	Inicia cuando la Dirección Provincial accede a la sección de Productos, donde le aparece un listado con todos los productos y posteriormente puede adicionar,

	modificar o eliminar un producto, y finaliza cuando el usuario termine de realizar todas sus acciones con los productos.
Responsabilidades	Gestionar los productos comercializados por la entidad.
Casos de uso asociados	No tiene.
Precondiciones	Antes de iniciar este caso de uso del sistema la Dirección Provincial debe tener el sistema en pleno funcionamiento y una conexión estable y segura.

Interfaz Gestionar Productos

The screenshot displays the 'Interfaz Gestionar Productos' web application. On the left, a sidebar contains navigation links: Inicio, Usuarios, Destinos, Productos (highlighted), Proveedores, Compras, and Ventas. The main area is titled 'Productos' and features a 'Producto +' button. Below this is a table with columns: Código, Nombre, Precio de venta, Precio de compra, and Acción. The table lists several products with their respective prices. A search bar and a 'Mostrar 10 elementos' dropdown are also present.

Código	Nombre	Precio de venta	Precio de compra	Acción
1dc	papa	\$8	\$5	[Iconos]
26f0	boniato	\$8.55	\$5	[Iconos]
4e9	tomate	\$6	\$3.75	[Iconos]
5d	yuca	\$4.25	\$3	[Iconos]
7251	malanga	\$10	\$7	[Iconos]
7e6f91cf-3e22-3b65-8bfe-b212c9472ce7	toronja	\$9	\$3	[Iconos]
8845a4b1-36d4-36a8-920f-abc443a75ba7	guayaba	\$15	\$11.5	[Iconos]
bc1ce271-84b5-3909-b6b1-d1f37f8dc334	pepino	\$4	\$2	[Iconos]

Flujo normal de eventos. Sección A: Nuevo Producto.

Acción del actor	Respuesta del sistema
------------------	-----------------------

1. Seleccionar Adicionar	2. El sistema muestra la información referente con los campos que tienen que ser llenados por el usuario.
3. El usuario llena los campos y presiona adicionar	4. El sistema valida y envía el registro guardado a la base de datos. 5. El sistema notifica la inserción de un nuevo producto de manera correcta.
Flujo alternativo 1	
3. El usuario llena los campos con los datos y presiona adicionar.	4. El sistema notifica de algún error por datos mal introducidos.
5. El usuario verifica todos los campos y actualiza.	
<u>Flujo normal de los eventos. Sección B: Editar Producto</u>	
Acción del actor	Respuesta del sistema
1. El usuario se dirige a las acciones disponibles sobre el producto y selecciona Editar.	2. El sistema carga el producto y carga sus datos en un formulario.
3. El usuario realiza los cambios que desea hacer sobre el producto seleccionado y actualiza.	

	<p>4. El sistema actualiza el producto en la base de datos con sus nuevos cambios.</p> <p>5. Se muestra el mensaje de éxito.</p>
Flujo alternativo 1	
<p>3. El usuario realiza los cambios que desea hacer sobre el producto seleccionado y actualiza.</p> <p>5. El usuario verifica que los campos hayan quedado llenos y actualiza.</p>	<p>4. El sistema muestra mensaje de error donde indica que algún campo requerido ha quedado vacío.</p>
Flujo alternativo 2	
<p>3. El usuario realiza los cambios que desea hacer sobre el producto seleccionado y actualiza.</p> <p>5. El usuario verifica que los campos estén correctamente actualizados y actualiza.</p>	<p>4. El sistema muestra mensaje de error donde indica que algún campo contiene datos incorrectos.</p>
<u>Flujo normal de eventos. Sección C: Eliminar Producto</u>	
Acción del actor	Respuesta del sistema
<p>1. El usuario se dirige a las acciones disponibles sobre el producto y selecciona Eliminar.</p>	

3. El usuario hace clic en el botón aceptar.	<p>2. El sistema muestra un cartel de confirmación.</p> <p>4. El sistema elimina los datos del producto de la base de datos.</p> <p>5. Muestra el mensaje de éxito.</p>
Post condiciones	La base de datos debe quedar actualizada de acuerdo a las operaciones realizadas en el sistema.
<u>Flujo normal de eventos. Sección D: Mostrar Producto</u>	
Acción de actor	Respuesta del sistema
1. El usuario se dirige a las acciones disponibles sobre el producto y selecciona Mostrar.	2. El sistema muestra una vista con todos los datos del producto seleccionado.

Tabla 11: Descripción Caso de Uso Gestionar productos.