



Facultad 2

Sistema para la distribución de productos en la Tienda Panamericana de la UCI

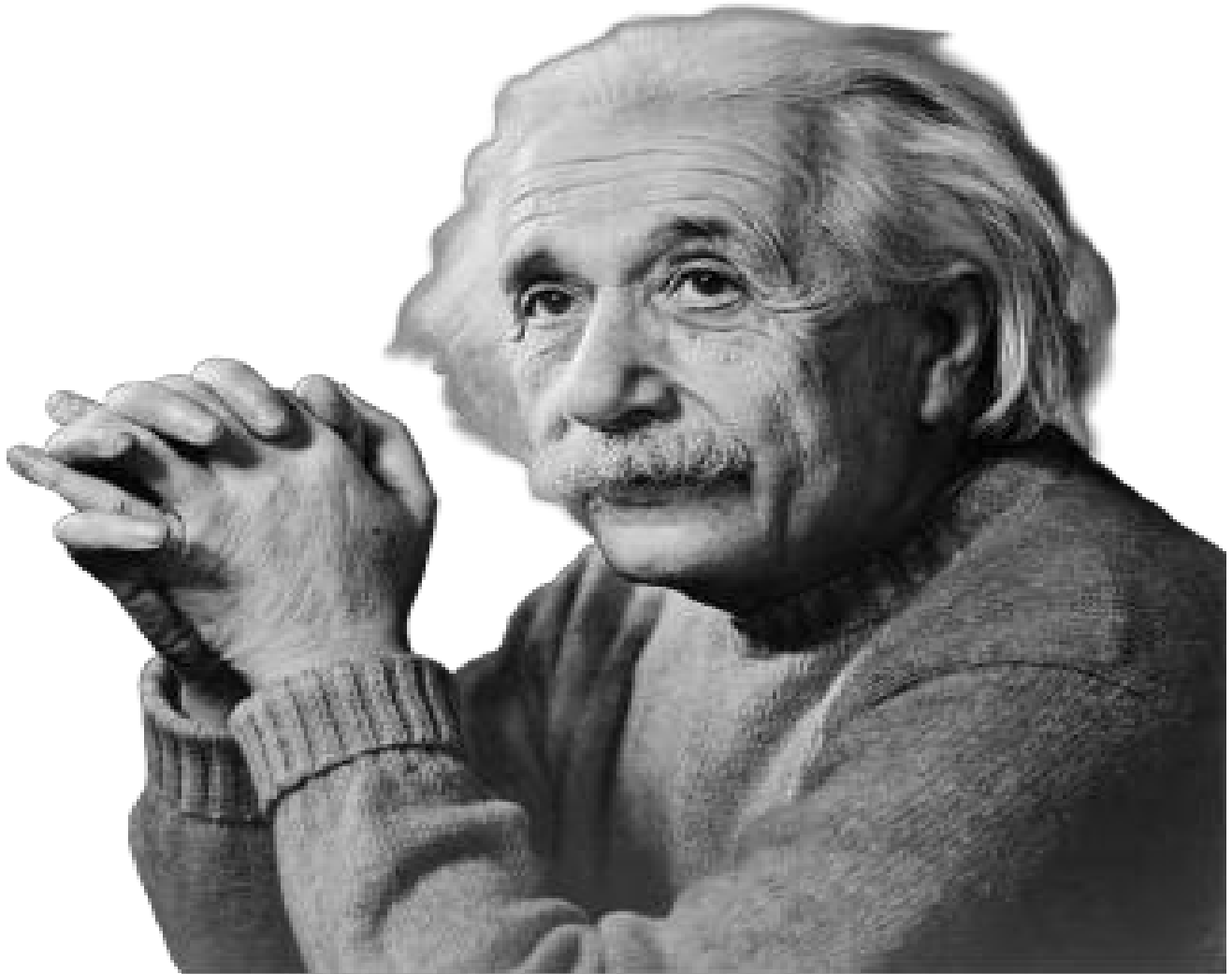
Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas.

Autora: Dorlenys Hernández Cabrera.

Tutor: Ing. Roberto Antonio Infante Milanés

La Habana, 29 de noviembre de 2023

“Año 65 de la Revolución”.



“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein

Declaración de autoría

Declaro ser autora de la presente tesis, con el título: ***“Sistema para el control de la distribución de productos en la Tienda Panamericana de la UCI”*** y concedo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo. Para que así conste firmo la presente a los 29 días del mes de noviembre del año 2023.

Firma del Autor

Dorlenys Hernandez Cabrera

Firma del Tutor

Ing. Roberto Antonio Infante Milanés

DATOS DE CONTACTO

AGRADECIMIENTOS

A mis padres y hermanas por el amor y la preocupación.

A una persona que sin ella no hubiera sido posible realizar esta investigación, mi novio Yordanis, gracias por ayudarme a tirar tanto código, por su paciencia con cada uno de mis berrinches, por ser como eres conmigo y por apoyarme en esta última etapa.

A mi suegros María Julia y Ponce, por la preocupación y apoyo siempre que lo necesité.

A todos mis compañeros de aula que transitaron conmigo estos maravillosos cinco años y quisiera mencionar a aquellos que han venido arrastrándome por una pierna para poder llegar hasta donde estoy:

Arazay, Lia, Talia , Alexander...espero y aparezcas en los agradecimientos de todos los graduados la verdad. De todos y cada uno me llevo una experiencia inolvidable.

A mi tutor por toda la ayuda brindada en estos meses de trabajo.

A todos y cada uno de mis profesores durante estos cinco años por sus enseñanzas.

A todas aquellas personas que alguna vez me preguntaron cómo me iba en la tesis y a los que me han ayudado de una forma u otra a realizar este sueño...Gracias!!!

DEDICATORIA

A mis padres y hermanas por brindarme el apoyo incondicional durante todo mi paso por la universidad, por dotarme de todos los recursos necesarios.

RESUMEN

La distribución de productos desempeña un papel fundamental en el éxito y la competitividad de las organizaciones. La forma en que los productos son distribuidos y llegan a los consumidores finales puede marcar la diferencia entre el crecimiento empresarial y el estancamiento. La Universidad de las Ciencias Informáticas (UCI) cuenta con una Tienda Panamericana en su campus, mediante esta tienda se le ofertan variados productos alimenticios, de aseo e higiene personal a los trabajadores de la institución. La investigación, cuyos resultados se describen en este informe, tuvo como propósito desarrollar un sistema para el control de la distribución de productos en la Tienda Panamericana de la UCI. El desarrollo del mismo fue guiado por la metodología de desarrollo de software XP, cumpliendo con cada una de sus fases y generando todos los artefactos. Para la implementación de este se utilizó como marco de trabajo web Laravel con su lenguaje de programación PHP. Para validar que la solución cumpliera con los requisitos definidos por el cliente, se aplicaron pruebas unitarias y pruebas de aceptación.

Palabras clave: Distribución, Distribución de productos, Tienda Panamericana.

ABSTRACT

The distribution of products plays a fundamental role in the success and competitiveness of organizations. The way products are distributed and reach end consumers can make a difference between business growth and stagnation. The University of Informatics Sciences (UCI) has a Panamerican Store on its campus, through which a variety of food, cleaning, and personal hygiene products are offered to the institution's employees. The research, described in this report, aimed to develop a system for controlling the distribution of products in the UCI's Panamerican Store. The development of the system was guided by the XP software development methodology, following each of its phases and generating all the artifacts. The Laravel web framework with its PHP programming language was used for the implementation. To validate that the solution met the requirements defined by the client, unit tests and acceptance tests were conducted.

Keywords: *System, Product Distribution, Methodology.*

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTOS TEÓRICOS SOBRE EL PROCESO DE DISTRIBUCIÓN DE PRODUCTOS.....	5
1.1 Conceptos asociados a la investigación.....	5
1.2 Estudio del estado del arte.....	5
1.2.1 Ámbito Internacional.....	6
1.3 Selección de la Metodología.....	7
1.3.1 Programación Extrema (Extreme Programming, XP):.....	8
1.3.2 Fundamentos de la metodología seleccionada.....	9
1.4 Herramientas y Tecnologías.....	10
1.4.1 Herramientas utilizadas.....	10
1.4.2 Lenguaje de modelado.....	11
1.4.3 Lenguajes del lado del servidor.....	11
1.4.3 Lenguajes del lado del cliente.....	11
1.4.4 Sistemas Gestor de Base de Datos (SGBD).....	12
1.4.4 Framework.....	13
Conclusiones del capítulo.....	14
CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN.....	16
2.1 Propuesta de solución.....	16
2.1.1 Fase de Exploración.....	17
2.1.2 Historias de Usuarios.....	18
2.1.3 Fase 1: Planificación.....	22
2.1.4 Estimación de esfuerzos.....	22
2.1.5 Plan de iteraciones.....	23
2.1.6 Plan de entrega.....	24
2.2 Fase 2: Diseño.....	24
2.2.1 Tarjetas CRC.....	24
2.2.2 Arquitectura de software.....	27
2.2.3 Patrón Arquitectónico.....	28
2.2.4 Patrones de Diseño.....	30
2.3 Modelo de datos.....	33
Conclusiones del capítulo.....	34
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS.....	35
3.1 Tareas de ingeniería.....	35
3.1.1 Iteración I.....	35
3.1.2 Iteración II.....	39
3.1.3 Iteración III.....	39
3.1.4 Diagrama de despliegue.....	40
3.2 Estrategia de pruebas.....	41
3.2.1 Pruebas unitarias.....	42

3.2.3 Pruebas de aceptación.....	44
3.2.4 Iteración I.....	44
3.2.5 Iteración II.....	47
3.2.6 Iteración III.....	49
3.2.7 Análisis de las pruebas de aceptación.....	50
Conclusiones del capítulo.....	51
CONCLUSIONES GENERALES.....	52
RECOMENDACIONES.....	53
REFERENCIAS BIBLIOGRÁFICAS.....	54
ANEXOS.....	57

ÍNDICE DE TABLAS

TABLA 1 HU1 AUTENTICAR USUARIOS.....	19
TABLA 2 HU2 GESTIONAR PRODUCTOS.....	19
TABLA 3 HU3 GESTIONAR ÁREAS.....	20
TABLA 4 HU4 GESTIONAR VENTAS.....	20
TABLA 5 HU5 VISUALIZAR ESTADÍSTICAS.....	20
TABLA 6 HU6 GENERAR LISTADO DE ASIGNACIONES.....	21
TABLA 7 ESTIMACIÓN DE ESFUERZO POR HISTORIA DE USUARIO.....	23
TABLA 8 PLAN DE DURACIÓN DE LAS ITERACIONES.....	23
TABLA 9 PLAN DE ENTREGA.....	24
TABLA10 TARJETA CRC 1.....	25
TABLA11 TARJETA CRC 2.....	25
TABLA12 TARJETA CRC 3.....	25
TABLA13 TARJETA CRC 4.....	26
TABLA14 TARJETA CRC 5.....	27
TABLA 15 TAREA DE INGENIERÍA #2.....	35
TABLA 16 TAREA DE INGENIERÍA #3.....	35
TABLA 17 TAREA DE INGENIERÍA #4.....	36
TABLA 18 TAREA DE INGENIERÍA #5.....	36
TABLA 19 TAREA DE INGENIERÍA #6.....	36
TABLA 20 TAREA DE INGENIERÍA #7.....	37
TABLA 21 TAREA DE INGENIERÍA #12.....	37
TABLA 22 TAREA DE INGENIERÍA #13.....	37
TABLA 23 TAREA DE INGENIERÍA #8.....	38
TABLA 24 TAREA DE INGENIERÍA #9.....	38
TABLA 25 TAREA DE INGENIERÍA #10.....	38
TABLA 26 TAREA DE INGENIERÍA #11.....	39

TABLA 27 TAREA DE INGENIERÍA #1.....	39
TABLA 28 TAREA DE INGENIERÍA #1.....	39
TABLA 29 TAREA DE INGENIERÍA #1.....	40
TABLA 30 CASO DE PRUEBA DE ACEPTACIÓN 1.....	44
TABLA 31 CASO DE PRUEBA DE ACEPTACIÓN 2.....	45
TABLA 32 CASO DE PRUEBA DE ACEPTACIÓN 3.....	45
TABLA 33 CASO DE PRUEBA DE ACEPTACIÓN 4.....	46
TABLA 34 CASO DE PRUEBA DE ACEPTACIÓN 5.....	46
TABLA 35 CASO DE PRUEBA DE ACEPTACIÓN 6.....	46
TABLA 36 CASO DE PRUEBA DE ACEPTACIÓN 7.....	47
TABLA 37 CASO DE PRUEBA DE ACEPTACIÓN 8.....	47
TABLA 38 CASO DE PRUEBA DE ACEPTACIÓN 9.....	48
TABLA 39 CASO DE PRUEBA DE ACEPTACIÓN 10.....	48
TABLA 40 CASO DE PRUEBA DE ACEPTACIÓN 11.....	49
TABLA 41 CASO DE PRUEBA DE ACEPTACIÓN 12.....	49
TABLA 42 CASO DE PRUEBA DE ACEPTACIÓN 13.....	50

ÍNDICE DE FIGURAS

<i>FIGURA 1: INTERFAZ DE LA SECCIÓN ESTADÍSTICAS</i>	16
<i>FIGURA 2: MODELO VISTA CONTROLADOR</i>	28
<i>FIGURA 3:EVIDENCIA DEL PATRÓN EXPERTO</i>	30
<i>FIGURA 4:EVIDENCIA DEL PATRÓN CREADOR</i>	30
<i>FIGURA 5:EVIDENCIA DEL PATRÓN ALTA COHESIÓN</i>	31
<i>FIGURA 6:EVIDENCIA DEL PATRÓN CONTROLADOR</i>	31
<i>FIGURA 7:EVIDENCIA DEL PATRÓN SINGLETON</i>	31
<i>FIGURA 8:EVIDENCIA DEL PATRÓN OBSERVADOR</i>	32
<i>FIGURA 9:EVIDENCIA DEL PATRÓN FACTORY</i>	32
<i>FIGURA 10: MODELO DE DATOS</i>	33
<i>FIGURA 12 DIAGRAMA DE DESPLIEGUE</i>	40
<i>FIGURA 11 RESULTADOS DE LAS PRUEBAS UNITARIAS AUTOMATIZADAS</i>	42

OPINIÓN DEL(OS) TUTOR(ES)

<Contenido de la opinión de los tutores>

AVAL DEL CLIENTE

<Contenido del aval del cliente sobre la solución desarrollada>

INTRODUCCIÓN

La distribución de productos desempeña un papel fundamental en la cadena de suministro, asegurando la entrega eficiente y oportuna de bienes desde su lugar de producción hasta el consumidor final., asegurando que los bienes producidos lleguen a los consumidores finales de manera eficiente. Este proceso, que abarca actividades como almacenamiento, transporte y gestión de inventarios, no solo impulsa el crecimiento económico, sino que también genera empleo y mejora la competitividad de las empresas al reducir costos.

En Cuba, la distribución de productos presenta desafíos particulares debido a las características del sistema económico y las políticas gubernamentales que regulan el comercio. En el país, el gobierno desempeña un papel central en la planificación y control de la distribución de bienes y servicios, utilizando mecanismos como la asignación de cuotas, a través de las libretas de abastecimiento, y la distribución a través de tiendas estatales.

La Universidad de las Ciencias Informáticas (UCI) es una institución educativa, la cual cuenta con una Tienda Panamericana en su campus. Desempeña un papel vital al ofrecer una amplia gama de productos alimenticios, de aseo e higiene personal a los trabajadores de la institución. Sin embargo, esta tienda se enfrenta a desafíos y complejidades que afectan a la comunidad universitaria compuesta por estudiantes, trabajadores y sus familias. Uno de los principales problemas es la falta de regularidad en la frecuencia de suministro de productos. La incertidumbre en la llegada de los productos dificulta la gestión adecuada del inventario y limita la capacidad de satisfacer la demanda de los consumidores. Esta situación genera frustración y restringe la variedad de productos disponibles, afectando negativamente la experiencia de compra de la comunidad universitaria.

Otro aspecto problemático es la diferencia en el sistema de distribución. Mientras que en el resto del país se utiliza un sistema estándar y regulado, en la Universidad, se ha implementado la libreta de abastecimiento para que los miembros de la comunidad puedan realizar sus compras. Esto crea una disparidad significativa en términos de acceso y equidad para los miembros de la comunidad universitaria. Esta distribución diferenciada puede afectar la disponibilidad de productos y la experiencia de compra, generando descontento y dificultades para obtener los productos necesarios.

Actualmente y desde hace poco más de 3 años, la venta se realiza por áreas, a partir de la asignación que se realiza, a través del Buró Sindical. El proceso de organización de las ventas implica la

INTRODUCCIÓN

determinación de la cantidad de productos correspondientes a cada una de las áreas, teniendo en cuenta varios elementos, como la cantidad de trabajadores de cada área, el tipo y cantidad de producto, la correspondencia entre la composición del área y el tipo de producto, así como las cantidades asignadas a cada área en ventas anteriores del mismo tipo de producto. Toda esta información se recopila en Excel con los datos correspondientes a cada venta. Se hace necesario poder gestionar esto de manera más fácil, por lo que se considera que, informatizando el proceso, se contribuye a tal fin. Sobre todo, porque se podría lograr una mayor homogeneidad en las nomenclaturas, descripción y precios de los productos, teniendo en cuenta que los mismos varían para diferentes formatos y presentaciones de un mismo tipo de producto. A su vez, se permitiría visualizar de manera más fácil el histórico de las asignaciones por área, tipo de producto y fecha, lo que contribuiría a la toma de decisiones con las asignaciones y poder realizar los ajustes necesarios en el menor tiempo posible.

Teniendo en cuenta la situación descrita anteriormente, se plantea como **problema de la investigación**: ¿Cómo mejorar el proceso de asignación de productos a las áreas, para su comercialización a través de la Tienda Panamericana de la UCI?

La tipología de estudio que se sigue responde a una investigación cualitativa bajo los criterios de Carlos Álvarez de Zayas (Álvarez de Zayas, Carlos, 1995). De acuerdo a esta concepción se define como **objeto de estudio**: los procesos de distribución de productos.

El **objetivo general** del trabajo se limita a: Desarrollar un sistema informático para la organización y distribución de productos en la Tienda Panamericana de la UCI. Este objetivo incide sobre el **campo de acción**: la asignación de productos para la venta en la Tienda Panamericana de la UCI

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**

- Elaborar marco teórico a partir del estudio de diferentes sistemas utilizados para el control de distribución de productos.
- Analizar y elegir las herramientas y tecnologías para el desarrollo de la solución.
- Seleccionar y fundamentar las herramientas y lenguajes para el desarrollo de la solución.

- Analizar y diseñar la propuesta de solución para identificar los componentes de software que intervienen en su desarrollo y los artefactos de la ingeniería de software.

Para la realización exitosa de la investigación y para darle cumplimiento a los objetivos específicos planteados se ha de llevar a cabo las siguientes **tareas de la investigación**:

1. Analizar el estado actual de la planificación de asignaciones de productos en la UCI
2. Analizar los referentes actuales, de sistemas para la gestión de ventas de y distribución de productos
3. Analizar las metodologías, lenguajes y tecnologías de desarrollo que serán utilizadas para la implementación de la propuesta de solución.
4. Diseñar los artefactos ingenieriles de la propuesta de solución, de acuerdo a la metodología seleccionada.
5. Implementar la propuesta de solución
6. Diseñar los casos de pruebas.
7. Aplicar los casos de pruebas.

Para el desarrollo satisfactorio del presente trabajo de diploma se emplearon **métodos científicos** tanto teóricos como empíricos, los cuales se mencionan a continuación:

Como **métodos teóricos** se emplearon:

- **Análítico-Sintético**: empleado para el análisis, evaluación y selección de las técnicas a emplear en el desarrollo del sistema, de esta forma se pudo determinar los principios fundamentales que debe cumplir la propuesta de solución.
- **Histórico-Lógico**: se estudió la existencia de sistemas similares, permitiendo así profundizar en las particularidades de cada uno de ellos, con el fin de obtener experiencias propias para la futura implementación.
- **Modelción**: Se utiliza para generar los diagramas y tablas que representan el diseño y la

arquitectura del sistema.

Métodos empíricos

- **La observación:** este método permitió observar detalladamente el proceso actual de asignación de productos en la tienda, identificando los puntos críticos y las áreas que podrían mejorarse.
- **La entrevista:** permitió recopilar información necesaria de los encargados de la Tienda Panamericana y a los responsables del Buró Sindical para conocer su perspectiva sobre el proceso actual y su disposición a implementar un sistema informático.
- **Análisis documental:** mediante la selección y recolección de la documentación asociada al tema abordado, se hace uso de este método con el objetivo de extraer la información necesaria para lograr los resultados esperados.

El documento se encuentra estructurado de la siguiente manera: Resumen, introducción, 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas, y anexos.

Capítulo 1: Fundamentos teóricos de la investigación. En este capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se describen los conceptos fundamentales asociados al dominio del problema, además, se proporciona una descripción de las herramientas, tecnologías y metodología de desarrollo de software a utilizar para dar solución al problema planteado.

Capítulo 2: Planificación y diseño de la propuesta de solución. Se realiza una descripción de la propuesta de solución y se exponen los requisitos funcionales y no funcionales identificados.

Capítulo 3: Implementación y pruebas. Incluye toda la información relacionada con el desarrollo y la implementación de la solución propuesta. Se muestran los resultados obtenidos, así como la información adquirida mediante la realización de las pruebas software y su validación.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS SOBRE EL PROCESO DE DISTRIBUCIÓN DE PRODUCTOS

En el presente capítulo se abordan los temas relacionados con la fundamentación teórica de la investigación. Se detallan los conceptos asociados al dominio del problema, con los cuales se logró una mayor comprensión del objeto de estudio, por parte de los implicados en el desarrollo del sistema. Se realiza un análisis detallado de las tecnologías y herramientas a utilizar. Así como, un estudio del estado del arte del tema de investigación, analizando los sistemas o módulos existentes que son similares al que se va a desarrollar.

1.1 Conceptos asociados a la investigación

Después de una exhaustiva investigación acerca de los canales de distribución de productos, varios son los autores que lo abordan por ejemplo según *Philip Kotler* y *Gary Armstrong* un **canal de distribución** *“es un conjunto de organizaciones que dependen entre sí y que participan en el proceso de poner un producto o servicio a la disposición del consumidor o del usuario industrial.”* (1)

Según lo descrito por (Ballou, 2004), **la cadena de suministro** es un conjunto de actividades funcionales, que involucran transporte, control de inventarios, entre otras, que se repiten a lo largo del canal de flujo y a través de las cuales la materia prima se convierte en productos terminados y se agrega valor para el consumidor. (2)

Gestión de inventario: Cuando se habla sobre gestión de inventarios se refiere a un seguimiento profundo de los artículos o materiales que se almacenan, esto a través de ciertas actividades que logran proporcionar un amplio conocimiento en la administración adecuada del registro, compra y salida del inventario dentro de una empresa. (3)

1.2 Estudio del estado del arte

En una investigación el análisis de herramientas similares, permite la identificación de elementos que pueden ser de utilidad para el desarrollo adecuado de la misma. Como parte de la investigación se realizó un estudio de los sistemas informáticos para la evaluación de integralidad existentes, tanto en el país, como a nivel internacional, que pudieran ser la solución semejante al problema planteado.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

1.2.1 Ámbito Internacional

Zoho Inventory

Zoho Inventory es una herramienta que permite el seguimiento en tiempo real del inventario de una empresa, facilitando la gestión eficiente de los productos. Con una interfaz intuitiva, proporciona acceso a datos actualizados y ofrece opciones de autenticación de usuarios para controlar el acceso a la plataforma. Además, permite establecer permisos y roles de usuario. Con Zoho Inventory, se puede crear y mantener un catálogo de productos, mientras se rastrea el movimiento de cada artículo mediante el seguimiento de números de serie, y se administra múltiples almacenes. También se integra con canales de venta en línea populares, como Shopify y Amazon, simplificando la sincronización del inventario en diferentes plataformas. La herramienta ofrece funciones para gestionar pedidos, realizar seguimiento de envíos y controlar las compras a proveedores. Además, brinda herramientas de generación de informes y análisis para obtener una visión detallada del rendimiento del inventario.

ShipBob:

ShipBob es una aplicación que se enfoca en la gestión de la cadena de suministro y la distribución de productos. Proporciona herramientas para el seguimiento de inventario, la gestión de pedidos, la optimización de envíos y la integración con múltiples canales de venta. La plataforma de ShipBob combina tecnología avanzada y una red de centros de cumplimiento estratégicamente ubicados para optimizar los procesos logísticos. Ofrece un panel de control intuitivo que brinda información detallada sobre las métricas de cumplimiento, envíos, inventario y más. Esto permite a las empresas tomar decisiones informadas y optimizar sus operaciones.

Resumen del análisis de la soluciones existentes

A partir del análisis de los sistemas informáticos homólogos antes descrito se puede arribar a la conclusión que tienen ciertas similitudes con el sistema a desarrollar. Ambas soluciones ofrecen funcionalidades relevantes para la gestión de productos, áreas y ventas, permiten la manipulación de catálogos, seguimiento de inventario, administración de pedidos y sincronización con canales de venta en línea. Además, brindan herramientas de generación de informes y análisis para obtener una visión detallada del rendimiento del inventario. Por lo que se desarrollará un nuevo sistema teniendo en cuenta estas propuestas identificadas.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

1.3 Selección de la Metodología

Una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo han evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo. (4)

En estos últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo, las llamadas *Metodologías Tradicionales o Métodos Pesados* y las *Metodologías Ágiles o los Métodos Ligeros*. Mientras los Métodos Pesados intentan conseguir el objetivo común, por medio de orden y documentación, los Métodos Ligeros tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

Las **metodologías tradicionales** centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control del mismo, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo, definido todo esto, en la fase inicial del desarrollo del proyecto. Estas metodologías imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente. Para ello, se hace énfasis en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar. Otra de las características importantes dentro de este enfoque, son los altos costes al implementar un cambio y la falta de flexibilidad en proyectos donde el entorno es volátil (Maida, Esteban Grabiél Pacienza y Julian 2015).

Una **metodología de desarrollo ágil** generalmente es un proceso Incremental (entregas frecuentes con ciclos rápidos), también Cooperativo (clientes y desarrolladores trabajan constantemente con una comunicación muy fina y constante), Sencillo (el método es fácil de aprender y modificar para el

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

equipo) y finalmente Adaptativo (capaz de permitir cambios de último momento). Las metodologías ágiles proporcionan una serie de pautas y principios junto a técnicas pragmáticas que hacen que la entrega del proyecto sea menos complicada y más satisfactoria tanto para los clientes como para los equipos de trabajo, evitando de esta manera los caminos burocráticos de las metodologías tradicionales, generando poca documentación y no haciendo uso de métodos formales. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan (Maida, Esteban Grabiél Pacienzia y Julian 2015).

1.3.1 Programación Extrema (Extreme Programming, XP):

La programación extrema es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. Al igual que otras metodologías ágiles, la programación extrema es una metodología de desarrollo de software dividida en sprints de trabajo. Los marcos ágiles siguen un proceso iterativo, en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima. El diseño de la programación extrema permite a los desarrolladores responder a las solicitudes de los clientes, adaptarse y realizar cambios en tiempo real. Sin embargo, la programación extrema es mucho más disciplinada; realiza revisiones de código frecuentes y pruebas unitarias para realizar cambios rápidamente. Además es muy creativa y colaborativa, ya que promueve el trabajo en equipo durante todas las etapas de desarrollo. (5)

Permite una adaptabilidad a entorno de desarrollo, por lo que su versión más simplificada, consta de 4 fases:

- **Planificación:** consiste en realizar una recopilación de todos los requerimientos del proyecto, se crean las Historias de Usuario (HU), el plan de iteraciones y el plan de entregas.
- **Diseño:** se basa en conseguir diseños simples y sencillos, con el objetivo de procurar hacerlo todo lo menos complicado posible para el cliente o usuario, se crean las tarjetas CRC, las cuales definen una clase expresando sus funcionalidades y las otras clases con las que colabora.
- **Desarrollo:** consiste en establecer una buena comunicación entre el equipo y el cliente, para

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

que los desarrolladores puedan codificar todo lo necesario para el proyecto que se requiere. Además, se definen las tareas de ingeniería para que los desarrolladores tengan una guía para implementar todas las HU.

- **Pruebas:** consiste en comprobar el funcionamiento de la codificación que se haya implementado, garantizando la evaluación de las distintas tareas en las que han sido divididas las HU.

1.3.2 Fundamentos de la metodología seleccionada

Los autores de la presente investigación deciden emplear Programación Extrema (XP) como metodología de desarrollo de software, pues se adecua a los requerimientos exigidos, dado que se trata de un proyecto de corta duración, su equipo de desarrollo es pequeño y el cliente forma parte del mismo, lo que contribuye a fomentar las relaciones interpersonales y el buen clima de trabajo.

Esta es una metodología que consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo.

Por otro lado, tiene un desarrollo iterativo e incremental. Se ha tenido en cuenta que XP define la constante comunicación y la retroalimentación, teniendo como uno de sus fines fundamentales la construcción de un producto que vaya en línea con los requerimientos del cliente y las necesidades del proyecto. La aplicación será desarrollada por un programador, no estrictamente coincidiendo con la característica de programación en parejas que propone la misma, pero basado en el trabajo entre el cliente y el programador.

1.4 Lenguajes empleados

Las tecnologías que están disponibles para la web pueden ser agrupadas en dos grandes categorías, según la propia arquitectura de una aplicación web. **Tecnologías del lado del cliente:** disponibles para los clientes web. **Tecnologías del lado del servidor:** son las que están disponibles en el servidor, que es el encargado de almacenar la lógica para construir las páginas web dinámicas.

1.4.1 Herramientas utilizadas

Microsoft Visual Studio Code

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. (6) **V.1.82**

Visual Paradigm

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Proporciona a los desarrolladores de software la plataforma de desarrollo de vanguardia para construir aplicaciones de calidad más rápidas, mejores y más baratas. Facilita una excelente interoperabilidad con otras herramientas CASE y la mayoría de los IDEs líderes que sobresalen todo su proceso de desarrollo de Model-Code-Deploy en esta solución única (7) **v.16.2**

PHPMyAdmin

Es una herramienta de administración de bases de datos MySQL ampliamente utilizada en entornos de desarrollo web porque proporciona una interfaz gráfica intuitiva que facilita la gestión de bases de datos, tablas, consultas y otros aspectos relacionados con MySQL. Con phpMyAdmin, los usuarios pueden administrar y manipular fácilmente su base de datos sin necesidad de conocimientos avanzados de MySQL o de comandos SQL. (14)

1.4.2 Lenguaje de modelado

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene. No es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos. (8)

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

1.4.3 Lenguajes del lado del servidor.

Los lenguajes del lado del servidor son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Estos son empleados para el diseño de páginas webs dinámicas donde los más utilizados para el desarrollo son: ASP, PHP y PERL.

El lenguaje de Hypertext Pre-processor (Pre-Procesador de Hipertextos) **PHP** es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto. Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes: no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas. (9) v.8.2

1.4.3 Lenguajes del lado del cliente.

Los lenguajes de lado cliente (entre los cuales no sólo se encuentra el HTML sino también el Java y el JavaScript los cuales son simplemente incluidos en el código HTML) son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento.

El lenguaje HyperText Markup Language (Lenguaje de Marcas de Hipertexto) **HTML** es el lenguaje con el que se define el contenido de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc... Es muy fácil de aprender, lo que permite que cualquier persona, aunque no haya programado en la vida, pueda enfrentarse a la tarea de crear una web. (10) v.5

JAVASCRIPT

Es un lenguaje de programación que se puede aplicar a un documento HTML y usarse para crear interactividad dinámica en los sitios web. Fue inventado por Brendan Eich, cofundador del proyecto Mozilla, Mozilla Foundation y la Corporación Mozilla. Por sí solo es bastante compacto aunque muy flexible, y los desarrolladores han escrito gran cantidad de herramientas encima del núcleo del lenguaje JavaScript, desbloqueando una gran cantidad de funcionalidad adicional con un mínimo esfuerzo. No requiere compilación, y sirve principalmente para mejorar la gestión de la interfaz cliente/servidor como, por ejemplo, un script JavaScript insertado en un documento HTML permite

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

reconocer y tratar localmente los eventos generados por el usuario. (12)

Las hojas de estilo en cascada o (Cascading Style Sheets) **CSS** son un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML. Se le denomina Hojas de Estilos en Cascada porque las características se aplican de arriba a abajo mediante reglas que poseen un esquema prioritario. Esta especificación es un lenguaje de diseño gráfico que se escribe dentro del código HTML del sitio web y, permite crear páginas de una manera más exacta y aplicarles estilos (colores, márgenes, formas, tipos de letras, etc.) por lo que se tiene mayor control de los resultados finales. (15) v.3

1.4.4 Sistemas Gestor de Base de Datos (SGBD).

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos y ofrecen a los usuarios unos mecanismos para poder acceder a ellos. Además, se encargan de cosas menos visibles, como asegurarse que los datos no se pierden, están bien enlazados los unos con los otros, mantener la seguridad de la información y la protección de acceso a los usuarios que tienen permisos. Los SGBD más populares son MySQL, Oracle, PostgreSQL, SQL Server, SQLite, MariaDB. Todos ellos son sistemas gestores de bases de datos relacionales, que son aquellas a las que se accede y administra mediante el lenguaje SQL. Pero también hay otros SGBD como MongoDB que son "no relacionales" y que no soportan SQL. (11)

MySQL

Es un sistema de administración de bases de datos relacionales de código abierto desarrollado por Oracle. Se considera como la base de datos de código abierto más utilizada en el mundo. Soporta una amplia gama de tipos de datos, lo que permite tener una gran versatilidad en cuanto a las situaciones, industrias o casos de uso donde puede implementarse una base de datos MySQL. Puede emplearse para la industria financiera, al manejar datos con mucha precisión; por otro lado, también puede utilizarse en ámbitos de geolocalización por sus datos de tipo espacial. (13) v.5

1.4.4 Framework

Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

organización y desarrollo de software. Utilizar frameworks puede simplificar (y mucho) una tarea o proceso, de ahí que se trate de una de las herramientas habituales que manejan los Digital Workers, porque les ayuda a ser más ágiles y productivos. Generalmente, los frameworks son usados por programadores porque permiten acelerar el trabajo y favorecer que este sea colaborativo, reducir errores y obtener un resultado de más calidad. (16)

Laravel

Laravel es un framework PHP gratis y de código abierto que brinda un conjunto de herramientas y recursos para crear aplicaciones modernas. Posee un ecosistema integral que combina funciones integradas y una variedad de paquetes y extensiones compatibles.

Este framework de PHP creció en popularidad rápidamente en los últimos años, y muchos desarrolladores lo adoptaron como su framework de trabajo favorito para lograr un proceso de desarrollo optimizado. Para entender mejor qué es Laravel, citaremos una frase de su página web: *“Laravel es un marco de aplicación web con una sintaxis expresiva y elegante. Ya sentamos las bases, liberándote para crear sin preocuparte por las cosas pequeñas”*. (17) v.10

Características

- Motor de plantillas ligeras e integradas, que permiten crear diseños increíbles utilizando siempre contenido dinámico. Además, proporciona multitud de widgets que incorporan CSS, y JS con estructuras súper robustas. Las plantillas de Laravel están muy bien diseñadas y el resultado es a la vez simple y complejo.
- Compatible con arquitectura MVC: Laravel es un framework compatible con patrones de arquitectura MVC. Esto garantiza la separación de la lógica empresarial por un lado y las capas de presentación por el otro. Este sistema es fácilmente escalable y ofrece multitud de funciones ya integradas.
- Una de las características que más apreciamos los desarrolladores es la incorporación de una herramienta de línea de comandos (llamada Artisan) que ayuda a automatizar las tediosas tareas de programación repetitivas.
- Laravel ofrece sus propias bibliotecas modulares de manera preinstalada en el sistema. Esto no es algo que suela ocurrir en otros framework de PHP.

CAPÍTULO 1: FUNDAMENTOS TEÓRICOS DE LA INVESTIGACIÓN

- Está pensado para resolver problemas concretos que son de mucha actualidad. Esto es una gran ventaja. Digamos que se centra en lo importante. (18)

Se decidió usar el framework Laravel ya que incluye herramientas que facilitan la construcción de aplicaciones web, haciendo de este proceso algo mucho más rápido y dando como resultado un código bien estructurado y fácil de mantener. Hoy en día, existen muchísimos sitios web creados con esta tecnología, incluyendo grandes empresas como Disney, Twitch, The New York Times, entre otros.

Conclusiones del capítulo

- La definición de los principales conceptos asociados al dominio de la presente investigación, permitió sentar las bases teóricas para el desarrollo del sistema.
- El análisis del estado actual de los sistemas informáticos para la gestión de ventas de productos, permitió constatar que ninguno se ajusta a las necesidades de la gestión de venta de producto a través de la Tienda Panamericana de la UCI, por lo que es necesario la implementación de la propuesta de solución de la presente investigación.
- En este capítulo se realizó un estudio profundo de las herramientas, el lenguaje y la metodología que se utilizaron para darle cumplimiento al desarrollo de la aplicación.
- Se escogió a Visual Studio Code como entorno de desarrollo integrado, debido a que posee una amplia variedad de extensiones desarrolladas por la comunidad.
- Se optó como sistema gestor de base de datos a MySQL ya que es conocido por su capacidad para manejar grandes volúmenes de datos y su rendimiento eficiente.
- La metodología de desarrollo ágil es la más indicada para ser utilizada, esto ayudará a que el proyecto sea terminado de forma más rápida para luego poder darle uso, específicamente la metodología ágil XP.

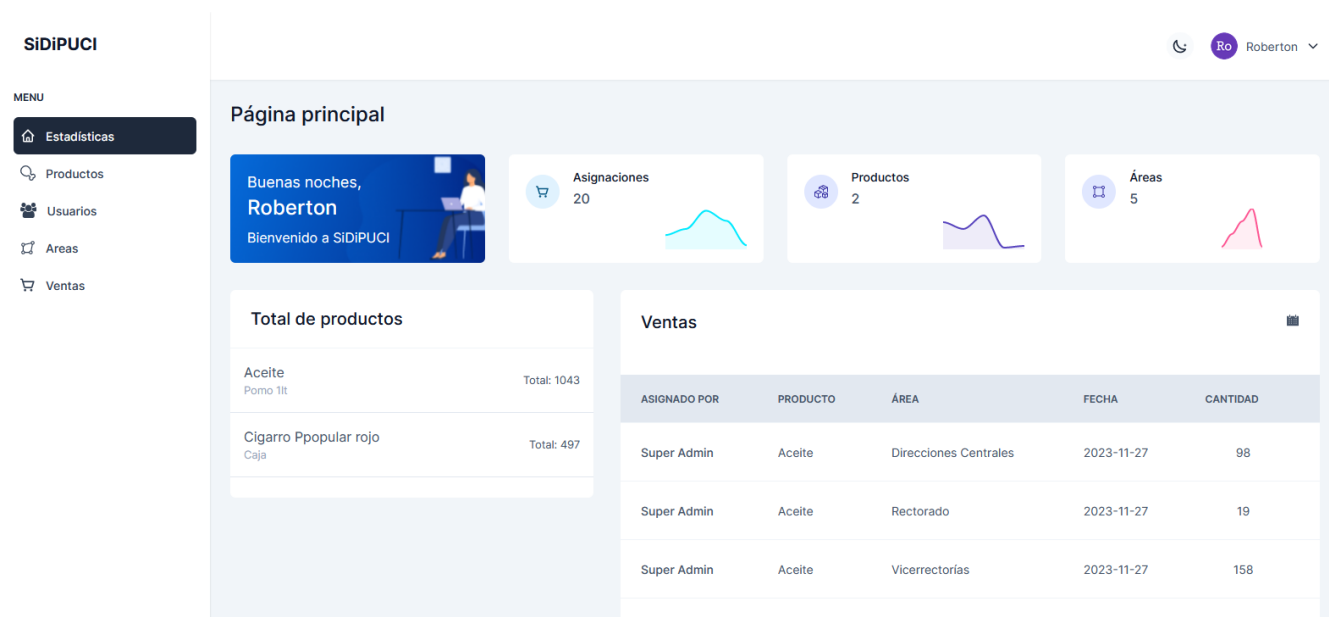
CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En este capítulo se abordan los elementos asociados a las fases de Planificación y Diseño en correspondencia con la metodología de desarrollo de software empleada. Se describe la propuesta de solución, a partir de las Historias de Usuario (HU) como artefacto para la descripción de las funcionalidades del sistema. Se muestran el plan de iteraciones, de entrega y los elementos relacionados con la arquitectura del sistema.

2.1 Propuesta de solución

La presente propuesta tiene como objetivo desarrollar una plataforma web que permita gestionar de manera más eficiente la asignación de los productos que llegan a la Tienda Panamericana de la UCI.

Figura 1: Interfaz de la sección Estadísticas



La plataforma proporciona una interfaz intuitiva y fácil de usar, brindando a los usuarios las herramientas necesarias para agregar, editar y eliminar productos, así como asignarlos a las diferentes áreas de la organización. Contará con un sistema de autenticación para garantizar la seguridad y privacidad de la información. Los usuarios deberán iniciar sesión con credenciales válidas antes de acceder a las funcionalidades de la plataforma. Una vez autenticado el usuario, será redirigido a una página principal que contendrá las secciones de productos, áreas y asignaciones. En esta página, el usuario podrá acceder y gestionar de manera eficiente toda la información relacionada con estos elementos claves:

- La sección de productos permitirá al usuario listar, modificar, agregar y eliminar productos. Al seleccionar la opción de productos, se mostrará una lista completa de los productos existentes, donde se podrán visualizar detalles como el tipo de producto, el precio y el formato. Además, se proporcionará la funcionalidad para editar la información de un producto específico, así como agregar nuevos productos mediante un formulario intuitivo. También se brindará la opción de eliminar productos de la lista, garantizando un control total sobre los elementos presentes en el inventario.
- La sección de áreas ofrecerá las funcionalidades necesarias para gestionar y administrar las diferentes áreas de la organización. Al seleccionar esta opción, se presentará una lista de las áreas existentes, donde se podrá visualizar información como el nombre y la descripción de cada área. El usuario podrá modificar los datos de un área específica, actualizar su nombre o descripción según sea necesario. Asimismo, se dispondrá de la capacidad de agregar nuevas áreas mediante un formulario sencillo y eliminar áreas que ya no sean relevantes.
- La sección de asignaciones permitirá al usuario asignar productos a las áreas correspondientes, garantizando una distribución adecuada de las responsabilidades. Al acceder a esta sección, se mostrará una lista de todas las asignaciones realizadas para cada producto. De esta manera, el usuario podrá tener una visión general de qué áreas están a cargo de la gestión de cada producto en particular. Además, se proporcionará la funcionalidad para eliminar asignaciones existentes y editar asignaciones previamente realizadas.
- La sección de estadísticas permitirá al usuario visualizar el total de productos globales asignados. Se visualizan además los productos asignados a cada área, permitiéndole filtrar por fechas.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

2.1.1 Fase de Exploración

Dando comienzo a las fases de la metodología de desarrollo XP está la fase de exploración. En esta se define el alcance general del proyecto. Donde el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”. Los programadores estiman los tiempos de desarrollo en base a esta información. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado.

2.1.2 Historias de Usuarios.

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia.

(19)

En el proceso de desarrollo del sistema se definió para mantener la organización, el nivel de prioridad de cada uno de los requisitos obtenidos en la entrevista con el cliente. Los niveles se dividen en:

- **Alta:** Funcionalidades que son sumamente primordiales en el sistema, que por su función no deberían faltar en el editor, es decir, el sistema depende de estas para su correcto funcionamiento.
- **Media:** Cuando el cliente cree que son necesarias, no son muy primordiales, pero otras dependen de estas para poder ejecutarse.
- **Bajo:** Funcionalidades que su existencia en el sistema no es significativa, es decir, este puede funcionar correctamente sin estas.

El riesgo en desarrollo se determinó mediante los siguientes indicadores:

- **Alto:** Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.

- **Medio:** Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
- **Bajo:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que conlleven perjuicios para el desarrollo del proyecto.

Se definieron las siguientes HU:

1. Autenticar usuarios
2. Gestionar productos
3. Gestionar áreas
4. Gestionar ventas
5. Visualizar estadísticas
6. Generar listado de asignación

A continuación se detallan las mismas:

Tabla 1 HU1 Autenticar usuarios

Historia de Usuario	
Número: 1	Usuario: Usuario del sistema
Nombre: Autenticar usuarios	Programador: Dorlenys Hernández
Prioridad en Negocio: Media	Riesgo en Desarrollo: medio
Puntos Estimados: 0.5	Iteración: 2
Descripción: Como usuario Quiero registrarme en el sistema Para poder acceder y utilizar todas las funciones disponibles	

Tabla 2 HU2 Gestionar productos

Historia de Usuario	
Número: 2	Usuario: Usuario

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Nombre: Gestionar productos	Programador: Dorlenys Hernández
Prioridad en Negocio: Alta	Riesgo en Desarrollo: medio
Puntos Estimados: 0.5	Iteración: 1
Descripción: Como usuario Quiero gestionar productos Para adicionar, modificar, eliminar y ver una lista completa de los productos existentes	

Tabla 3 HU3 Gestionar áreas

Historia de Usuario	
Número: 3	Usuario: Usuario
Nombre: Gestionar áreas	Programador: Dorlenys Hernández
Prioridad en Negocio: Alta	Riesgo en Desarrollo: medio
Puntos Estimados: 0.5	Iteración: 1
Descripción: Como usuario Quiero Gestionar áreas Para adicionar, modificar, eliminar y ver una lista completa de las áreas existentes	

Tabla 4 HU4 Gestionar ventas

Historia de Usuario	
Número: 4	Usuario: Usuario
Nombre: Gestionar ventas	Programador: Dorlenys Hernández
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos Estimados: 0.5	Iteración: 1

Descripción: Como usuario Quiero Gestionar ventas Para adicionar, modificar, eliminar y ver una lista completa de las ventas existentes.

Tabla 5 HU5 Visualizar estadísticas

Historia de Usuario	
Número: 5	Usuario: Usuario
Nombre: Visualizar estadísticas	Programador: Dorlenys Hernández
Prioridad en Negocio: Bajo	Riesgo en Desarrollo: Bajo
Puntos Estimados: 0.5	Iteración: 3
Descripción: Como usuario Quiero Visualizar estadísticas Para mostrar la cantidad de asignaciones, productos y áreas en el sistema.	

Tabla 6 HU6 Generar listado de asignaciones

Historia de Usuario	
Número: 6	Usuario: Usuario
Nombre: Generar listado de asignaciones	Programador: Dorlenys Hernández
Prioridad en Negocio: Bajo	Riesgo en Desarrollo: Bajo
Puntos Estimados: 0.5	Iteración: 3
Descripción: Como usuario Quiero Generar listado de asignaciones Para mostrar la cantidad de productos a las diferentes áreas.	

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Requisitos no funcionales

Los requisitos no funcionales, a menudo abreviados como RNF, complementan los requisitos funcionales al especificar cómo un sistema de software debe realizar ciertas funciones. Definen las cualidades, características y limitaciones del sistema más que sus características específicas. En esencia, los requisitos no funcionales establecen los estándares para el rendimiento, la seguridad y la usabilidad del sistema.

Para este trabajo se definieron los siguientes RNF:

RNF1 Usabilidad

- RNF1.1 El sistema debe presentar un acceso fácil e interfaz amigable con informaciones fáciles de entender, para facilitar el uso del mismo.

RNF2 Seguridad

- RNF2.1 La autenticación de usuarios debe ser segura y proteger la información confidencial.
- RNF2.2 Validación de credenciales en el lado del cliente para evitar posibles ataques al sistema y la entrada de personal sin autorización a la aplicación.

RNF3 Rendimiento

- RNF3.1 El sistema debe contar con una velocidad óptima de carga y actualización de los datos de la página, no sobrepasando los 5 segundos.
- RNF3.2 El sistema debe responder de manera rápida al iniciar sesión.

RNF4 Portabilidad

- RNF4.1 El sistema deberá ser multiplataforma y permitir que se acceda a él desde cualquier navegador. La interfaz debe adaptarse a cualquier resolución de pantalla.

2.1.3 Fase 1: Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, posteriormente los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. (20) La estimación de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de

programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. (20).

2.1.4 Estimación de esfuerzos

En este momento el equipo de desarrollo realiza una estimación del esfuerzo que requerirá la implementación de cada HU para brindar al cliente un valor aproximado de la duración del proyecto, Los puntos estimados están vistos por semanas, de modo que 1 punto =1 semana, 0.25 puntos = 2 días (aproximadamente)

A continuación, se muestra la estimación realizada:

Tabla 7 Estimación de esfuerzo por Historia de Usuario.

Historia de Usuario	Puntos estimados/semanas
Autenticar usuarios	1
Gestionar productos	1
Gestionar áreas	1
Gestionar ventas	1
Visualizar estadísticas	0.5
Generar listado de asignaciones	0.5

2.1.5 Plan de iteraciones

Las estimaciones realizadas permitieron confeccionar una evaluación puntual del tiempo de implementación de cada historia de usuario para la posterior elaboración del plan de iteración. Una vez realizadas las estimaciones fue preciso construir un plan de iteración donde se pueden agrupar estas historias y dar su cumplimiento de manera paulatina.

A continuación se detalla el plan de iteraciones:

Tabla 8 Plan de duración de las Iteraciones

Nro. Iteración	Historia de usuario	Esfuerzo estimado (semanas)
----------------	---------------------	-----------------------------

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

1	Gestionar productos	3 semanas
	Gestionar áreas	
	Gestionar ventas	
2	Autenticar usuarios	1 semana
3	Visualizar estadísticas	1 semana
	Generar listado de asignaciones	

2.1.6 Plan de entrega

En el momento en que culmina el proceso de elaboración de las UH, se inicia el proceso de creación de un plan de entrega. El cual tiene como objetivo fundamental la obtención por parte de los programadores de una estimación detallada del período de tiempo que deben tener en cuenta para la implementación.

Tabla 9 Plan de entrega

Iteraciones	Duración	Fecha de Inicio	Fecha de Fin
Iteración No. 1	3 semanas	25-10-2023	15-11-2023
Iteración No. 2	1 semana	16-11-2023	23-11-2023
Iteración No. 3	1 semana	24-11-2023	31-12-2023

2.2 Fase 2: Diseño

En esta fase de la metodología xp se realizan códigos sencillos para probar la efectividad del programa, la idea es que el código que se cree tenga suficiente capacidad para cubrir con la historia del usuario, es decir, sus necesidades. El diseño del sistema xp debe ser flexible e integrado para que al momento de ejecutar la prueba pueda pasar a la siguiente fase. Al mismo tiempo, en el diseño de software se originan tarjetas CRC (Clase – Responsabilidad – Colaboración), las cuales describen las

clases que se usaron para la programación de la historia del cliente. (21)

2.2.1 Tarjetas CRC

La utilización de tarjetas CRC (Class-Responsibility-Collaboration) es una técnica de diseño orientado a objetos. El objetivo de la misma es hacer, mediante tarjetas, un inventario de las clases que se van a necesitar para implementar el sistema y la forma en que van a interactuar, de esta forma se pretende facilitar el análisis y discusión de las mismas por parte de varios actores del equipo de proyecto con el objeto de que el diseño sea lo más simple posible verificando las especificaciones del sistema. (22)

Tabla10 Tarjeta CRC 1

Tarjeta CRC	
Clase AuthenticatedSessionController	
Responsabilidad: se encarga de autenticar al administrador mediante el método store()	Colaboración: User.

Tabla11 Tarjeta CRC 2

Tarjeta CRC	
Clase HomeController	
Responsabilidad: se encarga de mostrar las estadísticas globales de la página mediante el método index()	Colaboración: Product, Area, User, AreaProduct.

Tabla12 Tarjeta CRC 3

Tarjeta CRC	
Clase UserController	
Responsabilidad: - Se encarga de renderizar la vista	Colaboración user: User.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

que muestra todos los usuarios de la página index() - Se encarga de crear usuarios a través del método store() - Se encarga de eliminar usuarios mediante el método destroy() - Se encarga de actualizar usuarios mediante el método update() - Se encarga de renderizar la vista de edición de usuarios mediante el método edit() - Se encarga de renderizar la vista de creación de usuarios mediante el método create()	

Tabla13 Tarjeta CRC 4

Tarjeta CRC	
Clase ProductController	
Responsabilidad: - Se encarga de renderizar la vista que muestra todos los usuarios de la página mediante el método index() - Se encarga de crear productos a través del método store() - Se encarga de eliminar productos mediante el método destroy() - Se encarga de actualizar productos mediante el método	Colaboración: Product.

update() - Se encarga de renderizar la vista de edición de productos mediante el método edit() - Se encarga de renderizar la vista de creación de productos mediante el método create()	

Tabla14 Tarjeta CRC 5

Tarjeta CRC	
Clase AreaController	
Responsabilidad: - Se encarga de renderizar la vista que muestra todas las áreas de la página mediante el método index() - Se encarga de crear áreas a través del método store() - Se encarga de eliminar áreas mediante el método destroy() - Se encarga de actualizar áreas mediante el método update() - Se encarga de renderizar la vista de edición de áreas mediante el método edit() - Se encarga de renderizar la vista de creación de áreas mediante el método create()	Colaboración: Area, AreaProduct, Product.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

2.2.2 Arquitectura de software

La arquitectura del software define la estructura que debe de tener un software, las piezas que debemos construir y el modo en el que se deben de juntar y trabajar entre ellas. Se define a alto nivel mediante una serie de patrones y abstracciones que seguir para el desarrollo del software y para la interacción entre sus diversas piezas.(23)

En una arquitectura de software confluyen tres elementos fundamentales:

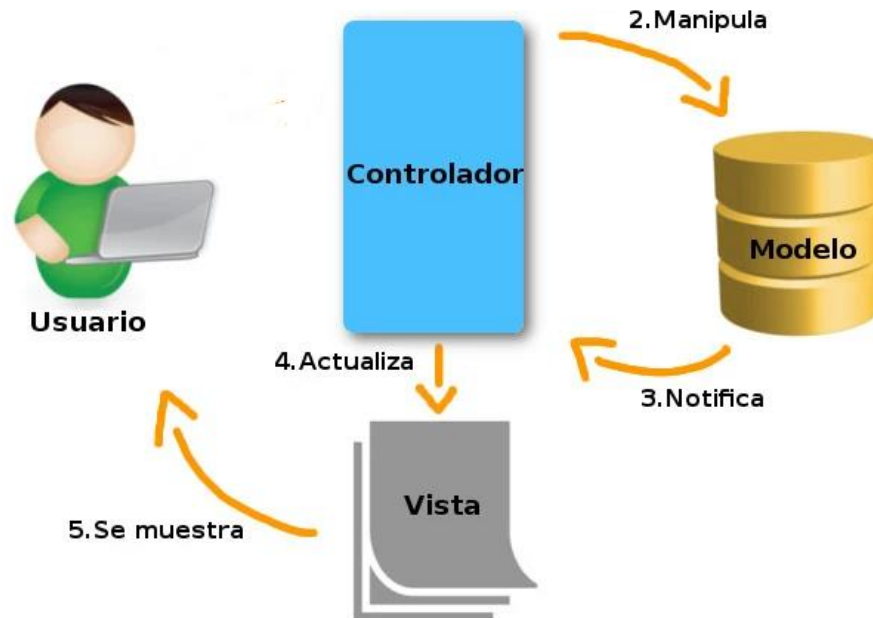
- Los modelos que definen la estructura, topología y dinámica del sistema.
- La trazabilidad o correspondencia de dichos modelos con los requisitos o necesidades establecidas en el contexto que va a operar la solución.
- Las reglas, los principios y justificaciones que rigen la arquitectura y que sustentan las decisiones que se tomaron.

2.2.3 Patrón Arquitectónico

El **patrón Modelo-Vista-Controlador (MVC)** es un patrón de arquitectura de software que, utilizando 3 componentes (Vistas, Modelos y Controladores) separa la lógica de la aplicación de la lógica de la vista en una aplicación. Es una arquitectura importante puesto que se utiliza tanto en componentes gráficos básicos hasta sistemas empresariales; la mayoría de los frameworks modernos utilizan MVC (o alguna adaptación del MVC) para la arquitectura, entre ellos podemos mencionar a Ruby on Rails, Django, AngularJS y muchos otros más. (24)

Se decidió utilizar este patrón para el desarrollo del sistema en cuestión debido a que con él la aplicación se puede desarrollar rápidamente, de forma modular y mantenible. Separar las funciones de la aplicación en modelos, vistas y controladores hace que la aplicación sea muy ligera. El diseño modular permite a los diseñadores y a los desarrolladores trabajar conjuntamente, así como realizar rápidamente el prototipo. Esta separación también permite hacer cambios en una parte de la aplicación sin que las demás se vean afectadas.

Figura 2: Modelo Vista Controlador



Vista: parte de la aplicación que contiene la interfaz gráfica. Cada elemento gráfico que interactúa con el usuario es la vista y su función es la de ser la capa que obtiene información sobre lo que quiere el usuario, a lo cual vamos a llamar eventos, la vista muestra siempre la información del modelo.

Modelo: tiene la responsabilidad de relacionar los datos con los que la aplicación va a operar. A esto se le llama lógica de negocio, es decir, es la parte de la aplicación que se encarga de mapear las actividades del mundo real a la forma a la que se va a modificar la información.

Controlador: responde a eventos o acciones que realice el usuario a través de la vista para poder solicitar una operación de información, también es responsable de mostrarle al usuario la vista adecuada dependiendo de la solicitud recibida, por lo cual es el vínculo que une al modelo con la vista.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En resumen, la vista muestra el modelo al usuario, este activa eventos en las vistas los cuales son recibidos por el controlador. El controlador manipula al modelo dependiendo de la solicitud. El modelo devuelve un conjunto de datos acordes a la solicitud realizada por el controlador, para que finalmente el controlador muestre una nueva vista al usuario con la información actualizada del modelo.

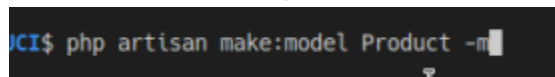
2.2.4 Patrones de Diseño

Los patrones de diseño son una técnica para resolver problemas comunes en el desarrollo web y otros ámbitos referentes al diseño de interacción o interfaces. Establece una relación entre un determinado contexto, un problema y una solución. Para el diseño de software, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución (25).

Patrones GRASP usados:

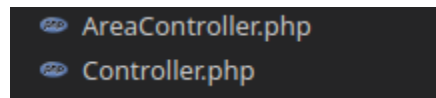
Experto: El patrón Experto se usa con frecuencia en la asignación de responsabilidades, es un principio básico que se utiliza continuamente en el diseño de objetos. El denominado experto expresa por lo general la intuición común de que los objetos realizan las acciones relacionadas con la información que poseen.

Figura 3:Evidencia del Patrón Experto



Creador: Se basa principalmente en la asignación de responsabilidades relacionadas con la creación de objetos. Es el encargado de crear las instancias de los objetos para cada una de las funcionalidades de la aplicación. El patrón se evidencia en las clases controladoras.

Figura 4:Evidencia del Patrón Creador.



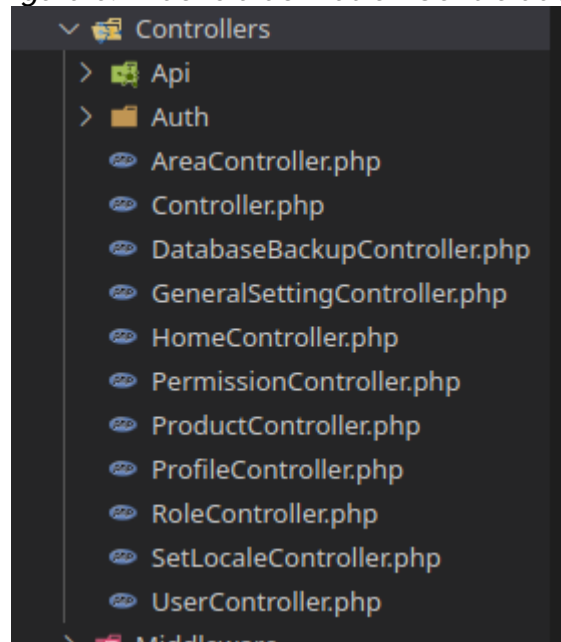
Alta Cohesión: Este patrón conduce a la asignación de responsabilidades relacionadas con un número relativamente pequeño. Para ello propone asignar a las clases las responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad.

Figura 5:Evidencia del Patrón Alta Cohesión.

```
public function update(Request $request, Area $area)
{
    $area->update($request->only('name'));
    return redirect()->route('areas.index')->with('message', 'Area actualizada existosamente');
}
```

Controlador: Este patrón define quién debería encargarse de atender un evento del módulo de gestión de perfil de usuario. La solución consiste en asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controladora.

Figura 6:Evidencia del Patrón Controlador.



CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

Patrones GOF utilizados:

Singleton En ocasiones es necesario mantener una visibilidad global a una única instancia de una clase de tal forma que sea accedida en cualquier parte de la aplicación, este patrón garantiza el comportamiento descrito anteriormente. La conexión a la base de datos (BD) es un ejemplo de la aplicación de este patrón.

Figura 7:Evidencia del Patrón Singleton.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=cukiGestionTienda
DB_USERNAME=root
DB_PASSWORD=
```

Observador Un efecto muy frecuente en aquellos sistemas que se fragmentan en un conjunto de clases que cooperan, es la necesidad de mantener la consistencia entre los distintos objetos interrelacionados. Para no recurrir a soluciones fuertemente acopladas (que reducen la posibilidad de reutilización), este patrón define una dependencia “uno a muchos” entre objetos, para que, cuando uno de ellos cambie su estado, los que dependan de él sean avisados y puedan actualizarse convenientemente.

Figura 8:Evidencia del Patrón Observador.

```
9 | $table->foreign(PermissionRegistrar::$pivotPermission)
10 |     ->references('id') // permission id
11 |     ->on($tableNames['permissions'])
12 |     ->onDelete('cascade');
```

Patrón Factory: El patrón de diseño Factory nos ayuda a crear instancias de otras clases, con la finalidad de ocultar la complejidad que se requiere para crearlas.

Figura 9:Evidencia del Patrón Factory.

```
use Illuminate\Database\Eloquent\Factories\Factory;
use Illuminate\Support\Str;
use Spatie\Permission\Models\Role;

/**
 * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
 */
class UserFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'name' => fake()->name(),
            'email' => fake()->unique()->safeEmail(),
            'email_verified_at' => now(),
        ];
    }
}
```

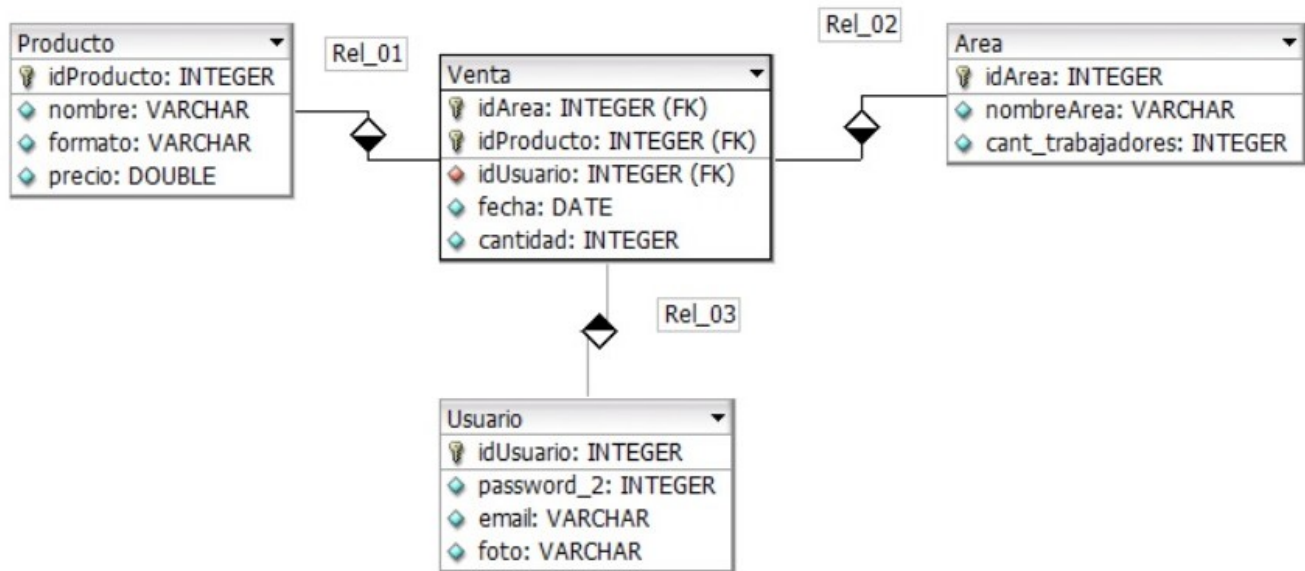
2.3 Modelo de datos

El modelado de datos es el proceso de diagramación de los flujos de datos. Al crear la estructura de una base de datos nueva o alternativa, el diseñador comienza con un diagrama de flujo de los datos por dentro y fuera de la base de datos. Este diagrama se usa para definir los formatos y estructuras de los datos y las funciones de gestión de la base de datos, a fin de dar un soporte eficiente al flujo de datos. Una vez creada e implementada la base de datos, el modelo de datos es la documentación y justificación de por qué existe la base de datos y cómo se diseñaron los flujos. (26)

A continuación, se muestra un segmento del diagrama de base de datos que genera el sistema a implementar.

Figura 10: Modelo de Datos.

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN



Conclusiones del capítulo

En este capítulo se realizó el análisis y diseño de la aplicación, arribando a las siguientes conclusiones:

- El levantamiento de requisitos permitió determinar las funcionalidades que debe cumplir el sistema.
- Los artefactos generados durante el análisis y diseño de la solución contribuyeron a un mejor entendimiento del sistema para dar paso a la implementación de la solución propuesta.
- La generación de las tarjetas CRC permitieron identificar las clases del sistema a desarrollar y la relación entre ellas.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

En este capítulo se desarrollan las dos últimas fases que plantea la metodología XP: Desarrollo y Pruebas. En el caso de la primera se da inicio con las tareas de ingeniería, donde cada HU se convierte en tareas a cumplir por los programadores. En la última fase se realizan pruebas de aceptación y unitarias para verificar el correcto funcionamiento del sistema que se obtuvo como resultado de la presente investigación.

3.1 Tareas de ingeniería

Tienen como objetivo definir cada una de las actividades que dan cumplimiento a las HU, de forma tal que se entienda lo que el sistema tiene que hacer y facilite su construcción. Pueden estar descritas por un lenguaje técnico y no ser necesariamente entendibles por el cliente. Cada HU como funcionalidad de la aplicación está compuesta por una o varias tareas de ingeniería, éstas no son más que pasos lógicos a seguir por el programador para realizar la implementación de una HU. A continuación, se detallan para cada una de las iteraciones las tareas a desarrollar por cada HU (28)

3.1.1 Iteración I

Dentro de la primera iteración se encuentran las funcionalidades relacionadas a las HU 2, 3, 4. Para ello se realizaron las siguientes tareas y el resto se encuentran en los anexos

Tabla 15 Tarea de ingeniería #2

Tarea de ingeniería	
Número de la tarea: 2	Numero de historia de usuario: HU_2
Nombre de la Tarea: Listar producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 25/10/2023	Fecha de fin: 26/10/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios ver una lista completa de todos los productos disponibles	

Tabla 16 Tarea de ingeniería #3

Tarea de ingeniería	
Número de la tarea: 3	Numero de historia de usuario: HU_4

Nombre de la Tarea: Agregar producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 27/10/2023	Fecha de fin: 28/10/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios crear un nuevo producto ingresando su nombre, descripción y precio.	

Tabla 17 Tarea de ingeniería #4

Tarea de ingeniería	
Número de la tarea: 4	Numero de historia de usuario: HU_10
Nombre de la Tarea: Asignar producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 29/10/2023	Fecha de fin: 30/10/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar un producto existente y asignarlo a un área	

Tabla 18 Tarea de ingeniería #5

Tarea de ingeniería	
Número de la tarea: 5	Numero de historia de usuario: HU_11
Nombre de la Tarea: Eliminar asignación de un producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 31/10/2023	Fecha de fin: 1/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar un producto que está asignado a una categoría y eliminar dicha asignación.	

Tabla 19 Tarea de ingeniería #6

Tarea de ingeniería	
Número de la tarea: 6	Numero de historia de usuario: HU_12
Nombre de la Tarea: Listar todas las asignaciones de un producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5

Fecha Inicio: 2/11/2023	Fecha de fin: 3/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar un producto y ver una lista de todas las asignaciones	

Tabla 20 Tarea de ingeniería #7

Tarea de ingeniería	
Número de la tarea: 7	Numero de historia de usuario: HU_3
Nombre de la Tarea: Modificar producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 4/11/2023	Fecha de fin: 5/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: El sistema debe permitir al usuario autenticado modificar los detalles de un producto existente	

Tabla 21 Tarea de ingeniería #12

Tarea de ingeniería	
Número de la tarea: 12	Numero de historia de usuario: HU_7
Nombre de la Tarea: Modificar área	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 6/11/2023	Fecha de fin: 7/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar un área específica y realizar cambios en su nombre.	

Tabla 22 Tarea de ingeniería #13

Tarea de ingeniería	
Número de la tarea: 13	Numero de historia de usuario: HU_9
Nombre de la Tarea: Eliminar área	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5

Fecha Inicio: 8/11/2023	Fecha de fin: 9/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar un área existente y eliminarla de forma permanente	

Tabla 23 Tarea de ingeniería #8

Tarea de ingeniería	
Número de la tarea: 8	Numero de historia de usuario: HU_5
Nombre de la Tarea: Eliminar producto	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 10/11/2023	Fecha de fin: 11/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar un producto existente y eliminarlo de forma permanente	

Tabla 24 Tarea de ingeniería #9

Tarea de ingeniería	
Número de la tarea: 9	Numero de historia de usuario: HU_6
Nombre de la Tarea: Listar área	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 12/11/2023	Fecha de fin: 13/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios ver una lista completa de todas las áreas disponibles	

Tabla 25 Tarea de ingeniería #10

Tarea de ingeniería	
Número de la tarea: 10	Numero de historia de usuario: HU_8
Nombre de la Tarea: Agregar área	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.25
Fecha Inicio: 13/11/2023	Fecha de fin: 14/11/2023
Programador Responsable Dorlenys Hernández	

Descripción: Esta funcionalidad permitirá a los usuarios crear un nueva área ingresando su nombre

Tabla 26 Tarea de ingeniería #11

Tarea de ingeniería	
Número de la tarea: 11	Numero de historia de usuario: HU_13
Nombre de la Tarea: Editar Asignación	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 14/11/2023	Fecha de fin: 15/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: Esta funcionalidad permitirá a los usuarios seleccionar una asignación específica y modificar los detalles de dicha asignación.	

3.1.2 Iteración II

Dentro de la segunda iteración se encuentran las funcionalidades relacionadas a la historia de usuario 1.

Tabla 27 Tarea de ingeniería #1

Tarea de ingeniería	
Número de la tarea: 1	Numero de historia de usuario: HU_1
Nombre de la Tarea: Autenticar usuarios	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Fecha Inicio: 17/11/2023	Fecha de fin: 18/11/2023
Programador Responsable Dorlenys Hernández	
Descripción: : El usuario debe introducir sus datos y en caso de ser correctos, podrá acceder a las acciones, y en caso contrario no lo deja acceder.	

3.1.3 Iteración III

Dentro de la tercera iteración se encuentran las funcionalidades relacionadas a las historias de usuario 5, 6.

Tabla 28 Tarea de ingeniería #1

Tarea de ingeniería

Número de la tarea: 1	Numero de historia de usuario: HU_1
Nombre de la Tarea: Visualizar estadísticas	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 20/11/2023	Fecha de fin: 21/11/2023
Programador Responsable Dorlenys Hernández	
Descripción:	

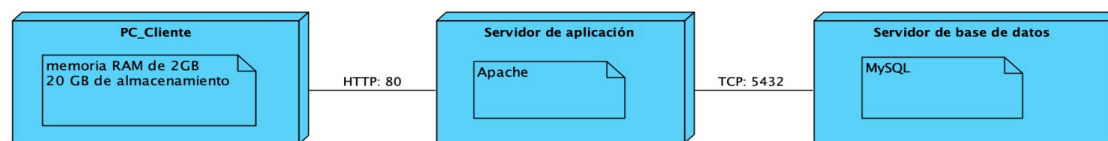
Tabla 29 Tarea de ingeniería #1

Tarea de ingeniería	
Número de la tarea: 1	Numero de historia de usuario: HU_1
Nombre de la Tarea: Generar listado de asignación	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Fecha Inicio: 22/11/2023	Fecha de fin: 23/11/2023
Programador Responsable Dorlenys Hernández	
Descripción:	

3.1.4 Diagrama de despliegue

A continuación, se muestra el diagrama de despliegue correspondiente a la propuesta solución desarrollada

Figura 12 Diagrama de Despliegue



PC-Cliente: Es el nodo que representa la estación de trabajo que permite al usuario mediante el

protocolo HTTP y el puerto 8080, acceder a la plataforma.

Servidor de aplicación web: Es la estación de trabajo que hospeda el código fuente de la aplicación y que le brinda al usuario las interfaces para realizar los procesos del sistema. Esta estación se comunica con el servidor de base de datos donde se almacenan los datos de la aplicación realizando la comunicación mediante el protocolo TCP (Transmission Control Protocol).

Servidor de BD(PostgreSQL): Este servidor es el encargado del almacenamiento de los datos del sistema. Se comunica con el servidor de aplicaciones del sistema mediante el protocolo TCP:5432, posibilitando el acceso mediante el usuario con privilegios para las operaciones determinadas a realizarse en el mismo.

HTTP: Protocolo de transferencia de hipertexto seguro ("Hypertext Transfer Protocol") es un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML, es el que impulsa todo internet. Los navegadores web utilizan este protocolo para solicitar páginas web a los servidores. El servidor devuelve todos los datos necesarios en código HTML para que puedan mostrarse en el navegador, en este caso por el puerto 8080

TCP: Protocolo de la capa de transporte (Transmission Control Protocol) es un protocolo de comunicación utilizado en redes de computadoras para garantizar la transmisión confiable de paquetes de datos en este caso usando TCP 5432

3.2 Estrategia de pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas, pues anima a probar constantemente, o tanto como sea posible. Permitiendo aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección .

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática. Y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (25)

3.2.1 Pruebas unitarias

Las pruebas unitarias de *software*, también conocidas como unit testing, incluyen un conjunto de características y propiedades que permiten su funcionamiento, de modo que una de las principales

metas de este tipo de pruebas es que permiten garantizar que cada una de las unidades de *software* analizadas se encuentran funcionando de la forma que deberían e independientemente. Cabe destacar que las pruebas unitarias de *software* funcionan mediante el mecanismo de aislar una parte determinada del código fuente, con el objetivo de asegurar su funcionamiento; es decir, son pruebas pequeñas que validan el comportamiento de un fragmento del código. (27)

Para la realización de dichas pruebas se utilizó el marco de trabajo Laravel, haciendo uso de su librería PHPUnit, en el siguiente código se refleja un segmento realizado a las funcionalidades.

A continuación, se muestra la realización de estas pruebas:

Figura 11 Resultados de las pruebas unitarias automatizadas

```
PASS Tests\Unit\AppTest
  app

PASS Tests\Feature\AreaTest
  user can not get areas
  user can get areas
  user can delete areas
  user can create areas
  user can update areas

PASS Tests\Feature\Auth\AuthenticationTest
  login screen can be rendered
  users can authenticate using the login screen
  users can not authenticate with invalid password

FAIL Tests\Feature\ProductTest
  user can not get products
  user can get products
  user can delete products
  user can create products
  user can update products
  user can asign product
  asign product with not numeric amount

---

• Tests\Feature\ProductTest > asign product with not numeric amount
Expected response status code [302] but received 500.
Failed asserting that 302 is identical to 500.
```

3.2.3 Pruebas de aceptación

Las pruebas de aceptación, también llamadas pruebas funcionales son supervisadas por el cliente basándose en los requerimientos tomados de las historias de usuario. En todas las iteraciones, cada una de las historias de usuario seleccionadas por el cliente deberá tener una o más pruebas de aceptación, de las cuales deberán determinar los casos de prueba e identificar los errores que serán corregidos. Las pruebas de aceptación son pruebas de caja negra, que representan un resultado esperado de determinada transacción con el sistema. Para que una historia de usuario se considere aprobada, deberá pasar todas las pruebas de aceptación elaboradas para dicha historia. Es importante resaltar la diferencia entre las pruebas de aceptación y las unitarias en lo que al papel del usuario se refiere. Mientras que en las pruebas de aceptación juega un papel muy importante seleccionando los casos de prueba para cada historia de usuario e identificando los resultados esperados, en las segundas no tiene ninguna intervención por ser de competencia del equipo de programadores (Luis Miguel Echeverry Tabón, y otros, 2007).

Las pruebas de aceptación tienen más peso que las unitarias ya que constituyen un indicador de la satisfacción del cliente con la solución además de marcar el final de una iteración y el comienzo de la siguiente. Se recomienda que el cliente sea quien diseñe estas pruebas o que al menos participe de manera activa en el proceso.

Para la primera iteración, se definieron un total de 6 de pruebas de aceptación. Todas enfocadas a evaluar la implementación de algunas funcionalidades del sistema. A continuación se muestran algunos casos de pruebas de la primera iteración, los demás se encuentran en los anexos.

3.2.4 Iteración I

Tabla 30 Caso de prueba de aceptación 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: HU_1
Nombre Autenticar usuarios	
Descripción: Prueba de la funcionalidad de Autenticar usuario.	
Condiciones de ejecución: El usuario debe estar en la pestaña de autenticación	
Pasos de ejecución: 1. Introducir correo y contraseña. 2. Presionar el botón "Entrar". 3. Aparece una notificación de: "Bienvenido"	

Resultados esperados: El usuario accede al sistema
Evaluación: Satisfactoria

Tabla 31 Caso de prueba de aceptación 2

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: HU_2
Nombre Listar producto	
Descripción: El sistema debe permitir listar los productos.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de ejecución: El usuario selecciona el botón mostrar productos y se le muestra una tabla con todos los productos listados.	
Resultados esperados: Se listan los productos.	
Evaluación: Satisfactoria	

Tabla 32 Caso de prueba de aceptación 3

Caso de prueba de aceptación	
Código: HU2_P3	Historia de usuario: HU_2
Nombre Agregar producto.	
Descripción: el sistema debe permitir agregar un nuevo producto en caso de haber llenado correctamente todos los campos, en caso contrario deberá mostrar una notificación de campos requeridos.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de ejecución: El usuario selecciona la opción Agregar producto y se le muestra un formulario con los campos a llenar, una vez llena los campos selecciona el botón agregar y recibe un mensaje de notificación indicando que se ha agregado de manera exitosa.	
Resultados esperados: Se agrega un producto al Sistema.	
Evaluación: Satisfactoria	

Tabla 33 Caso de prueba de aceptación 4

Caso de prueba de aceptación

Código: HU4_P4	Historia de usuario: HU_4
Nombre Asignar producto.	
Descripción: El sistema debe permitir asignar productos a cada una de las áreas.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: El usuario selecciona la opción "Asignar producto" en la sección producto, introduce la cantidad de productos que va a asignar y luego toca el botón asignar productos para completar la asignación.	
Resultados esperados: Se asignan los productos	
Evaluación: No Satisfactoria	

Tabla 34 Caso de prueba de aceptación 5

Caso de prueba de aceptación	
Código: HU4_P5	Historia de usuario: HU_4
Nombre Eliminar asignación de un producto.	
Descripción: El sistema debe permitir eliminar la asignación de un producto previamente asignado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: El usuario selecciona la opción "Eliminar asignación" en el menú de Asignaciones. Se muestra una confirmación de eliminación para la asignación seleccionada. El usuario confirma la eliminación de la asignación.	
Resultados esperados: El sistema elimina la asignación de producto seleccionada y muestra un mensaje de confirmación de la eliminación.	
Evaluación: Satisfactoria	

Tabla 35 Caso de prueba de aceptación 6

Caso de prueba de aceptación	
Código: HU4_P6	Historia de usuario: HU_4
Nombre Listar todas las asignaciones de un producto.	
Descripción: El sistema debe permitir al usuario listar todas las asignaciones existentes	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: El usuario selecciona el producto específico para el cual desea ver las asignaciones. Se muestra una lista de todas las asignaciones existentes para el producto	

seleccionado.
Resultados esperados: El sistema muestra una lista de todas las asignaciones de productos.
Evaluación: Satisfactoria

3.2.5 Iteración II

Para la segunda iteración, se definieron un total de 5 de pruebas de aceptación. Todas enfocadas a evaluar la implementación de algunas funcionalidades del sistema.

Tabla 36 Caso de prueba de aceptación 7

Caso de prueba de aceptación	
Código: HU2_P7	Historia de usuario: HU_2
Nombre Modificar producto.	
Descripción: El sistema debe permitir al usuario autenticado modificar los detalles de un producto existente	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: El usuario selecciona la opción "Modificar producto" en el menú. Se muestra un formulario con los detalles actuales del producto. El usuario realiza las modificaciones necesarias en los campos del formulario, y por último selecciona el botón "Guardar" para guardar los cambios realizados.	
Resultados esperados: El sistema guarda los cambios y actualiza los detalles del producto en la base de datos, mostrando un mensaje de confirmación de la modificación.	
Evaluación: Satisfactoria	

Tabla 37 Caso de prueba de aceptación 8

Caso de prueba de aceptación	
Código: HU2_P8	Historia de usuario: HU_2
Nombre Eliminar producto.	
Descripción: El sistema debe permitir al usuario autenticado eliminar un producto existente.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de ejecución: El usuario selecciona la opción "Eliminar producto" en el menú. Selecciona el producto específico que desea eliminar. Se muestra una confirmación de eliminación para el producto	

seleccionado. El usuario confirma la eliminación del producto.
Resultados esperados: El sistema elimina el producto seleccionado de la base de datos y muestra un mensaje de confirmación de la eliminación.
Evaluación: Satisfactoria

Tabla 38 Caso de prueba de aceptación 9

Caso de prueba de aceptación	
Código: HU3_P9	Historia de usuario: HU_3
Nombre Listar área.	
Descripción: El sistema debe permitir al usuario autenticado listar todas las áreas existentes.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: El usuario selecciona la opción "Listar áreas" en el menú. Se muestra una lista de todas las áreas existentes en el sistema.	
Resultados esperados: El sistema muestra una lista de todas las áreas existentes.	
Evaluación: Satisfactoria	

Tabla 39 Caso de prueba de aceptación 10

Caso de prueba de aceptación	
Código: HU3_P10	Historia de usuario: HU_3
Nombre Agregar área.	
Descripción: El sistema debe permitir al usuario autenticado agregar una nueva área.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de ejecución: El usuario selecciona la opción "Agregar área" en el menú. Se muestra un formulario vacío para ingresar el nombre del área. El usuario completa los campos requeridos en el formulario. El usuario selecciona el botón "Guardar" para agregar el área.	
Resultados esperados: El sistema guarda el área en la base de datos y muestra un mensaje de confirmación de que el área se ha agregado correctamente.	
Evaluación: Satisfactoria	

Tabla 40 Caso de prueba de aceptación 11

Caso de prueba de aceptación

Código: HU4_P11	Historia de usuario: HU_4
Nombre Editar Asignación.	
Descripción: El sistema debe permitir al usuario autenticado editar los detalles de una asignación existente.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de ejecución: El usuario selecciona la opción "Editar Asignación" en el menú. Se muestra un formulario con los detalles actuales de la asignación. El usuario realiza las modificaciones necesarias en los campos relevantes del formulario y por último selecciona el botón "Guardar" para guardar los cambios realizados.	
Resultados esperados: El sistema guarda los cambios y actualiza los detalles de la asignación en la base de datos, mostrando un mensaje de confirmación de la edición.	
Evaluación: Satisfactoria	

3.2.6 Iteración III

Para la tercera iteración, se definieron un total de 2 de pruebas de aceptación. Todas enfocadas a evaluar la implementación de algunas funcionalidades del sistema.

Tabla 41 Caso de prueba de aceptación 12

Caso de prueba de aceptación	
Código: HU3_P12	Historia de usuario: HU_3
Nombre Modificar área.	
Descripción: El sistema debe permitir al usuario autenticado modificar los detalles de un área existente.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema.	
Pasos de ejecución: El usuario selecciona la opción "Modificar área" en el menú. Se muestra un formulario con los detalles actuales del área. El usuario realiza las modificaciones necesarias en los campos relevantes del formulario, y selecciona el botón "Guardar" para guardar los cambios realizados	
Resultados esperados: El sistema guarda los cambios y actualiza los detalles del área en la base de datos, mostrando un mensaje de confirmación de la modificación.	
Evaluación: Satisfactoria	

Tabla 42 Caso de prueba de aceptación 13

Caso de prueba de aceptación	
Código: HU3_P13	Historia de usuario: HU_3
Nombre Eliminar área.	
Descripción: El sistema debe permitir al usuario autenticado eliminar un área existente	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema	
Pasos de ejecución: El usuario selecciona la opción "Eliminar área" en el menú. Se muestra una confirmación de eliminación para el área seleccionada. El usuario confirma la eliminación del área	
Resultados esperados: El sistema elimina el área seleccionada de la base de datos y muestra un mensaje de confirmación de la eliminación.	
Evaluación: Satisfactoria	

3.2.7 Análisis de las pruebas de aceptación

Se desarrollaron un total de trece casos de pruebas de aceptación, en la que se obtuvieron 12 resultados satisfactorios, se obtuvo 1 resultado no satisfactorio en cuanto a la funcionalidad Asignar producto, se detectó que no asigna productos con un valor no numérico, y no redirige. Estas pruebas fueron realizadas de forma organizada, por cada iteración definida.

En la primera iteración se realizó un total de 6 pruebas de aceptación, donde 5 fueron satisfactorias y solo una falló.

En la segunda iteración se realizó un total de 5 pruebas de aceptación, donde se obtuvo resultados satisfactorios en todas.

En la tercera iteración Se realizó un total de 2 pruebas de aceptación, donde se obtuvo resultados satisfactorios en todas.

Conclusiones del capítulo

- El plan de entrega fue cumplido acorde al tiempo planificado para el desarrollo de las tareas por cada historia de usuario.
- Las pruebas de aceptación permitieron que el cliente comprobara el estado de cada una de las funcionalidades desarrolladas en el sistema.

- Las pruebas al software permitieron comprobar la calidad del sistema propuesto.
- Las tareas de ingeniería ayudaron al equipo de proyecto a lograr identificar, controlar, rastrear los requisitos y los cambios en cualquier etapa mientras se desarrollaba el proyecto.
- Las realizaciones de las pruebas de aceptación aprobaron el correcto funcionamiento y el cumplimiento de los requisitos de software definidos para la aplicación.

CONCLUSIONES GENERALES

- El análisis realizado del estado del arte, fundamentó la necesidad de llevar a cabo el desarrollo de un sistema web que contribuye al proceso de distribución de productos en la Tienda Panamericana de la UCI.
- El uso de la metodología XP como guía del proceso de desarrollo de software, así como las herramientas y tecnologías tales como el framework Laravel, y el lenguaje de programación PHP facilitó el correcto desarrollo de la propuesta solución.
- Los requisitos del sistema, la arquitectura modelo vista controlador y el uso de los patrones de diseño, permitieron obtener una aplicación web optima que cumple con las expectativas del cliente.

RECOMENDACIONES

- Incorporar al sistema funcionalidades que permitan visualizar el comportamiento de los históricos de las asignaciones y ventas a partir de graficas como histogramas de forma tal que se contribuya con la toma de decisiones
- Actualmente en el sistema, se le asignan productos a todas a las áreas y se propone agregar una función para asignar productos a un grupo de áreas en específico.

REFERENCIAS BIBLIOGRÁFICAS

1. Ceupe. «Política de distribución de marketing mix: estrategias y factores que afectan en su selección», 25 de noviembre de 2023. <https://www.ceupe.com/blog/politica-de-distribucion-de-marketing-mix.html>.
2. Ballou, Ronald H. *Logística: administración de la cadena de suministro*. Pearson Educación, 2004.
3. Christopher, Martin. *Logistics & Supply Chain Management*. 4. ed. Harlow: Financial Times, Prentice Hall, 2011.
4. Gabriel, Esteban. «Metodologías de desarrollo de software», s. f.
5. Asana. «¿Qué es la programación extrema (XP)? [2022]». Asana. Accedido 25 de noviembre de 2023. <https://asana.com/es/resources/extreme-programming-xp>.
6. OpenWebinars.net. «Qué es Visual Studio Code y qué ventajas ofrece», 22 de julio de 2022. <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>.
7. Hernández González, Anaisa. *Aplicación del Proceso Unificado de Desarrollo a proyectos de software*, 2004.
8. Lucidchart. «Qué es el lenguaje unificado de modelado (UML)». Accedido 25 de noviembre de 2023. <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>
9. Rock Content - ES. «PHP: ¿qué es, para qué sirve y cuáles son sus características?», 9 de marzo de 2020. <https://rockcontent.com/es/blog/php/>.
10. DesarrolloWeb.com. «Qué es HTML». Accedido 25 de noviembre de 2023. <https://desarrolloweb.com/articulos/que-es-html.html>.
11. DesarrolloWeb.com. «Qué es SQL». Accedido 25 de noviembre de 2023. <https://desarrolloweb.com/articulos/262.php>.
12. «Fundamentos de JavaScript - Aprende desarrollo web | MDN», 18 de julio de 2023. https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/JavaScript_basics.
13. Londoño, Pablo. «Qué es MySQL, para qué sirve y características principales». Accedido 25 de noviembre de 2023. <https://blog.hubspot.es/website/que-es-mysql>.
14. GoDaddy, Equipo de Contenidos de. «Qué es phpMyAdmin y cómo usarlo». *Blog* (blog), 31 de julio de 2023. <https://es.godaddy.com/blog/que-es-phpmyadmin/>.
15. Rock Content - ES. «¿Qué es CSS y por qué es importante en los sitios web?», 12 de junio de 2019. <https://rockcontent.com/es/blog/que-es-css/>.
16. UNIR FP. «Framework: qué es, para qué sirve y algunos ejemplos». Accedido 25 de noviembre de 2023. <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>.

17. UNIR FP. «Framework: qué es, para qué sirve y algunos ejemplos». Accedido 25 de noviembre de 2023. <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>.
18. «Qué es Laravel y que características ofrece - SURÁTICA Software». Accedido 26 de noviembre de 2023. <https://www.suratica.es/que-es-laravel/#que-es-laravel>.
19. Gabriel, Esteban. «Metodologías de desarrollo de software», s. f.
20. «[PDF] Metodología Ágil de Desarrollo de Software XP - Free Download PDF». Accedido 26 de noviembre de 2023. <https://silo.tips/download/metodologia-agil-de-desarrollo-de-software-xp>.
21. BetsilleCamejo2021Uajmq. «Metodología xp | Desarrolla tu proyecto escuchando a tus clientes». Negocios y Empresa, 3 de junio de 2022. <https://negociosyempresa.com/metodologia-xp-que-es/>.
22. Jummp. «Desarrollo de software. Tarjetas CRC», 10 de enero de 2012. <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>.
23. Zúñiga, Fernán García de. «Arquitectura de software: descubre en qué consiste | Arsys». Blog de arsys.es, 3 de febrero de 2020. <https://www.arsys.es/blog/arquitectura-software>.
24. CódigoFacilito. «MVC (Model, View, Controller) explicado.» Accedido 26 de noviembre de 2023. <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>.
25. Pressman, Roger S. *Software Engineering: A Practitioner's Approach*. Palgrave Macmillan, 2005.
26. SAP. «¿Qué es el modelado de datos? | Definición, importancia y tipos | SAP Insights». Accedido 26 de noviembre de 2023. <https://www.sap.com/latinamerica/products/technology-platform/datasphere/what-is-data-modeling.html>.
27. Team, KeepCoding. «¿Qué son las pruebas unitarias de software?», 1 de agosto de 2022. <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>.
28. Wallace, Doug, Isobel Raggett, y Joel Aufgang. *Extreme Programming for Web Projects*. Addison-Wesley Professional, 2003.

ANEXOS