

**Universidad de Matanzas
Facultad de Ciencias Técnicas
Departamento de Informática**



**TÍTULO: HERRAMIENTA WEB PARA LA IMPLEMENTACIÓN DEL COMERCIO
ELECTRÓNICO MAYORISTA EN LA SUCURSAL CIMEX-MATANZAS**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO
INFORMÁTICO**

AUTOR: David Chávez Armenteros

TUTOR: MsC. Orlenys Machin Díaz

Matanzas, 2023

Resumen

El avance de las tecnologías de la información y comunicación en nuestro país ha ampliado significativamente las posibilidades para llevar a cabo tareas de gran impacto en la economía nacional de manera más eficiente, aprovechando las herramientas modernas que estas tecnologías ofrecen. Este estudio se centra en aprovechar estas ventajas al desarrollar una herramienta web para facilitar el comercio electrónico mayorista en la sucursal CIMEX Matanzas, con el objetivo de mejorar la eficiencia en la gestión del inventario de la entidad. La aplicación permite mostrar la gama de productos disponibles, ofrece información detallada sobre los mismos y permite a los clientes con contratos realizar reservas, además ofrece a los administradores un apartado para la gestión del inventario. Para lograr estos objetivos, se emplean métodos científicos tanto teóricos como empíricos para recopilar información. El desarrollo se lleva a cabo utilizando la metodología ágil XP, mientras que para la implementación se utiliza el lenguaje de programación C# y se emplea MySQL como gestor de base de datos. La herramienta se evalúa continuamente para asegurar que cumpla con los estándares de calidad establecidos, obteniendo resultados satisfactorios. Esta solución proporciona a la empresa y a sus clientes independencia, comodidad y seguridad, mejorando significativamente la eficiencia en las actividades comerciales.

Abstract

The advancement of information and communication technologies in our country has significantly expanded the possibilities for efficiently carrying out tasks that have a substantial impact on the national economy, leveraging the modern tools these technologies offer. This study focuses on harnessing these advantages by developing a web application to facilitate wholesale e-commerce at the CIMEX Matanzas branch, aiming to enhance inventory management efficiency within the entity. The application enables showcasing the range of available products, providing detailed information about them, and allowing contracted clients to make reservations, while also offering administrators a section for inventory management. To achieve these goals, scientific methods, both theoretical and empirical, are employed to gather information. The development follows the agile XP methodology, utilizing the C# programming language for implementation and MySQL as the database manager. The tool undergoes continuous evaluation to ensure it meets established quality standards, yielding satisfactory results. This solution grants the company and its clients independence, convenience, and security, significantly enhancing efficiency in commercial activities

Contenido

Introducción	1
Capítulo I Marco teórico referencial	4
Introducción.....	4
Proceso actual de trabajo.....	4
Análisis crítico de los procesos	4
Antecedentes	5
Métodos de investigación	5
Tendencias Tecnológicas.....	6
Herramientas, Tecnologías y Metodologías de Desarrollo	6
Software gratuito y de código abierto	6
Arquitectura Cliente-Servidor.....	7
Arquitectura basada en N-capas	7
Patrones de diseño.....	8
Metodología de desarrollo de software.....	8
Aplicaciones Web	10
Herramientas y tecnologías	11
Conclusiones del primer capítulo.....	20
Capítulo II Análisis y diseño del sistema.....	21
Introducción.....	21
Modelo de Proceso	21
Diagrama de Procesos de Negocio (BPD)	21
Descripción del Negocio	21
Modelo de la base de datos.....	22
Explicación de la solución propuesta	24
Patrones de diseño usados	24
Patrón Modelo – Vista – Controlador (MVC).	24
Patrón repositorio	24
Inyección de dependencia	25
Fase de Planificación	25
Equipo de Trabajo	25

Requisitos funcionales.....	26
Requisitos no funcionales.....	27
Historias de usuario.....	29
Estudio de factibilidad.....	33
Pruebas de aceptación.....	33
Conclusiones del segundo capítulo.....	37
Capítulo III Análisis de los resultados.....	38
Vistas del software.....	38
Conclusiones del tercer capítulo.....	42
Conclusiones.....	43
Bibliografía.....	44
Anexos.....	47
Anexo 1: Valores de la metodología XP.....	47

Introducción

Con el rápido avance de las Tecnologías de la Información y las Comunicaciones (TIC) en las últimas décadas, el uso de la informática se ha convertido en un requisito fundamental para alcanzar metas y objetivos cruciales en el desarrollo. A través de la aplicación de redes locales, nacionales y globales, han surgido nuevas maneras de gestionar diversos sectores, como la economía, la salud, la educación y la seguridad, entre otros. Estos ámbitos se apoyan en el empleo de información y conocimiento en sus actividades cotidianas. Las naciones dependen cada vez más de estos avances para gestionar eficientemente sus recursos y lograr mejores resultados en aspectos económicos y sociales.

Nuestro país, Cuba, no es ajeno a esta realidad. Ha decidido comprometerse decididamente en la adopción de estas tecnologías con el propósito de mejorar su economía, impulsar su desarrollo y beneficiar a su población a través del aprovechamiento de estas herramientas. En la actualidad, el país está llamado a innovar en diferentes formas de gestión laboral para contribuir a la digitalización de la sociedad. Un ejemplo reciente de esto es la creación de software que facilita el pago de servicios como electricidad, teléfono o suministro de gas licuado a través de plataformas como transfermóvil. Estas aplicaciones, entre otras, están teniendo un impacto positivo en la vida de los ciudadanos y las entidades, particularmente aquellas centradas en el comercio electrónico, que permiten a empresas públicas y privadas comercializar sus productos y servicios de manera más eficiente.

A pesar de estas oportunidades, muchas empresas en el país aún no han aprovechado completamente los beneficios de estas tecnologías y desean hacerlo. Con base en experiencias de otros, comprenden las oportunidades que estas tecnologías ofrecen para mejorar su eficiencia y obtener beneficios económicos sustanciales. Una de estas entidades es la sucursal CIMEX Matanzas, Grupo empresarial privado, de capital estatal cubano, que se ha caracterizado desde su creación hace más de 20 años, por el crecimiento constante y la estabilidad financiera, tanto dentro como fuera del país. Su organización la integran más de 80 empresas, con 10 sucursales territoriales en toda la Isla, y divisiones especializadas que comprenden amplios e importantes sectores de la economía cubana.

Consciente de las ventajas que supondría la implementación del comercio electrónico, la sucursal CIMEX Matanzas ha decidido adentrarse en este campo y unirse al mundo de las tiendas virtuales dedicadas al sector mayorista. El objetivo es ofrecer a los clientes un espacio donde puedan visualizar toda la gama de productos disponibles en la red de tiendas y almacenes de la empresa de manera ágil, organizada y con la información más detallada sobre las características de cada producto.

Esta iniciativa busca mejorar la eficiencia de la Sucursal CIMEX Matanzas en su función de comercializadora. Aprovechando las ventajas del comercio electrónico, la empresa aspira a brindar un servicio de ventas mayoristas con mejores

Introducción

prestaciones a sus clientes y, al mismo tiempo, obtener beneficios que van más allá del aspecto financiero, como la optimización del tiempo y la gestión del inventario. De esta manera, el problema de investigación se centra en: "¿Cómo mejorar la eficiencia en la oferta y reserva de productos de la Sucursal CIMEX Matanzas a sus clientes mayoristas?" El objeto de estudio es la aplicación del comercio electrónico mayorista en la gestión comercial empresarial, con el enfoque específico en la implementación del comercio electrónico para mejorar la eficiencia de la oferta y la reserva de productos en posesión de la Sucursal CIMEX Matanzas.

A partir de lo discutido anteriormente, se formula la siguiente hipótesis:

La implementación de una herramienta web para el comercio electrónico mayorista en la Sucursal CIMEX Matanzas potenciará la eficiencia en la oferta y reserva de la cartera de productos, fundamentales para las operaciones comerciales de la empresa.

Esta hipótesis identifica las siguientes variables:

- Variable independiente: la herramienta web para el comercio electrónico mayorista en la Sucursal CIMEX Matanzas.
- Variable dependiente: potenciará la eficiencia en la oferta y reserva de la cartera de productos de la Sucursal CIMEX Matanzas.

El objetivo general es implementar una herramienta web para el comercio electrónico mayorista que contribuya a mejorar la eficiencia en la oferta y reserva de productos de la Sucursal CIMEX Matanzas. Para lograr este objetivo, se plantean los siguientes objetivos específicos:

- Seleccionar y aplicar métodos y técnicas para obtener los fundamentos teóricos necesarios sobre el tema de investigación.
- Establecer los requisitos del sistema, diseñar el modelo de datos y la interfaz de usuario para luego aplicar las metodologías y herramientas correspondientes en el desarrollo, asegurando así la calidad del software final.
- Validar el funcionamiento de la herramienta web implementada a través de pruebas para evaluar el cumplimiento de los objetivos de la investigación.

Para desarrollar el marco teórico, se emplearon varios métodos y técnicas, incluyendo:

Métodos teóricos:

- Método inductivo – deductivo.
- Método de análisis histórico – lógico.
- Método de análisis y síntesis.

Métodos empíricos abordados a través de las técnicas siguientes:

- La entrevista.
- La observación científica.
- Análisis de documentos.

Introducción

En el desarrollo de la aplicación, se utilizó la metodología ágil XP, que se define como un enfoque ágil de desarrollo de software que se centra en prácticas para mejorar la calidad del código, la velocidad de entrega y la capacidad de respuesta a los cambios en los requisitos del cliente (Beck, 1999). XP promueve la colaboración cercana entre los miembros del equipo, la retroalimentación continua y la entrega frecuente de versiones funcionales del producto (Jeffries et al., 2000). Sus prácticas clave incluyen programación en parejas, pruebas unitarias continuas, integración continua, refactorización y simplicidad en el diseño. Este enfoque busca adaptarse rápidamente a los cambios y satisfacer las necesidades cambiantes del cliente a lo largo del desarrollo del proyecto.

Siguiendo el orden lógico, la estructura del documento es la siguiente:

Capítulo I: Marco teórico referencial, donde se exponen conceptos, definiciones y fundamentos asociados al problema de estudio. Se analiza el flujo actual de trabajo, antecedentes y tecnologías utilizadas, así como herramientas y tecnologías empleadas para el desarrollo del software.

Capítulo II: Análisis, diseño y construcción de la solución propuesta. Se presenta la propuesta de solución al problema de investigación, se organiza y planifica el proyecto usando la metodología ágil XP. Se analizan los beneficios obtenidos con la implementación del proyecto de software y se construye la herramienta web siguiendo las pautas de desarrollo de dicha metodología.

Capítulo III: Análisis de los resultados. Se analizan los resultados obtenidos y se exponen varias capturas de la aplicación.

Finalmente, se presentan las Conclusiones de la investigación. Además, se incluyen las Referencias Bibliográficas y Anexos necesarios para el desarrollo y mejor comprensión del trabajo de investigación.

Capítulo I Marco teórico referencial

Introducción

En este capítulo se exploran los fundamentos teóricos, antecedentes y métodos relacionados con la investigación. Se presentan los conceptos clave que sustentan el objeto de estudio para facilitar su comprensión. Se examina la metodología de desarrollo elegida para asegurar la correcta ejecución de la aplicación, en consonancia con las particularidades del proyecto a llevar a cabo. Además, se mencionan las diversas tecnologías y herramientas seleccionadas para el desarrollo del software.

Proceso actual de trabajo

En la actualidad, la empresa lleva a cabo la comercialización de su gama de productos a través de una red reducida de tiendas ubicadas en la provincia. Estos establecimientos disponen de espacios de exhibición de alrededor de 50 metros cuadrados.

Cuando un cliente llega a una de estas unidades comerciales, es recibido por el Administrador o el Dependiente de Piso de Venta. Estos empleados inician la comunicación con el visitante para comprender sus necesidades de compra y brindar asistencia para lograr sus objetivos o proporcionar alternativas que puedan satisfacer sus requerimientos. Dado que no toda la mercancía está disponible para su exhibición, es común que los clientes busquen productos que no estén presentes o que simplemente no existan en el lugar. Esta situación a menudo provoca tiempos de espera prolongados mientras los empleados atienden a clientes que llegaron previamente a la tienda.

El proceso de atención al cliente implica desde el diálogo inicial hasta la elaboración de presupuestos, reservas y facturación, si es necesario. Estos procedimientos generan tiempos de espera considerables para aquellos que aún no están siendo atendidos.

En muchas ocasiones, se requiere la intervención del Dependiente de Almacén para mostrar productos que no están en exhibición. En casos específicos que involucran productos técnicamente complejos o tecnológicos, es necesario mostrar el embalaje y/o documentación que detalla las características y funcionalidades de los productos. Todo esto se hace con el propósito de asegurar la elección adecuada de la mercancía que el cliente necesita adquirir.

Análisis crítico de los procesos

El enfoque tradicional utilizado para la exposición, comunicación, oferta y reservación de productos disponibles para la comercialización conlleva a largos tiempos de espera para los clientes, lo que les genera molestias y afecta su jornada

laboral. Esta metodología dificulta la obtención inmediata y necesaria de información para tomar decisiones de compra acertadas. Los empleados que interactúan directamente con los clientes se ven sobrecargados, lo que resulta en un servicio subóptimo y de calidad cuestionable.

Antecedentes

Previo a este trabajo, se realizó una investigación para identificar softwares similares al propuesto que pudieran satisfacer las necesidades comerciales de la entidad. Aunque se encontraron varias aplicaciones web de comercio electrónico, ninguna se ajustaba a las particularidades de la Sucursal CIMEX Matanzas. Por lo tanto, se decidió desarrollar un software personalizado que se adapte a las condiciones de trabajo específicas de la entidad para el comercio mayorista.

Métodos de investigación

Se emplearon métodos teóricos y empíricos para obtener la información necesaria:

Métodos teóricos fundamentales:

- Método inductivo-deductivo
- Método histórico-lógico
- Método de análisis y síntesis

Métodos empíricos a través de las siguientes técnicas:

- Entrevista
- Observación científica
- Análisis de documentos

Método inductivo – deductivo, sirvió para determinar la dirección correcta a seguir en la construcción de la aplicación, para el análisis de los resultados y la obtención de conclusiones inferidas a partir de propiedades y relaciones existentes entre los elementos que integran el fenómeno estudiado.

Método de análisis y síntesis, permitió el razonamiento y comprensión de la bibliografía consultada, precisar las bases teóricas que soportan el tema tratado y la interpretación adecuada de los resultados obtenidos para la elaboración certera de las conclusiones.

Método histórico – lógico, permitió el estudio minucioso del comportamiento del negocio desde sus inicios hasta el presente, así como las características principales que reflejan su esencia.

La entrevista resultó efectiva en la extracción de información que permitiera una exploración preliminar del fenómeno estudiado.

La observación se ha puesto de manifiesto durante toda la investigación y ha sido fundamental en la percepción de los aspectos esenciales del objeto de estudio.

Análisis de documentos fue empleado como la forma más robusta y precisa de obtención de información y concepción de ideas durante la investigación. Estos métodos permitieron el análisis, comprensión y obtención de conclusiones para el desarrollo del trabajo.

Tendencias Tecnológicas

En el panorama actual, se observa una marcada tendencia hacia el empleo de metodologías ágiles en el desarrollo de software, así como la preferencia por lenguajes de programación y herramientas asociadas al software de código abierto. Estos recursos se aplican ampliamente en una variedad de proyectos, especialmente en el desarrollo de aplicaciones web. Esta tendencia se fundamenta en la libertad de uso inherente a estos softwares, su robusta constitución, la seguridad que ofrecen y los beneficios económicos derivados de su utilización sin restricciones.

Herramientas, Tecnologías y Metodologías de Desarrollo

En esta sección, se esbozan de forma general las diversas herramientas, tecnologías y metodologías seleccionadas para el desarrollo de la herramienta web propuesta, dirigida a resolver la problemática planteada. Se ha optado por el empleo de software actual, seguro y altamente usable en el contexto actual. Asimismo, se ha decidido utilizar metodologías adaptadas a proyectos con las características específicas abordadas en este trabajo. El objetivo primordial de utilizar estos recursos es garantizar un entorno productivo que permita la construcción de un software eficiente, cumpliendo con los requisitos del producto informático demandado.

Software gratuito y de código abierto

El de código abierto se define como aquel que no oculta su código, permaneciendo accesible y permitiendo su distribución libre. Su concepción refleja que el conocimiento contenido en él es patrimonio de toda la humanidad. Esta aplicación posibilita que países e individuos, especialmente aquellos con menos recursos, accedan sin limitaciones a tecnologías desarrolladas por otros con mayores capacidades.

Según Stallman (2004), a menudo se malinterpreta el término "software libre", pensando que se refiere al precio, pero lo verdaderamente relevante es la libertad. Él enfatiza que un programa es considerado software libre para el usuario cuando se cumplen las siguientes condiciones:

- Libertad para ejecutar el programa con cualquier propósito.

- Libertad para modificar el programa para satisfacer las necesidades del usuario (siendo esencial tener acceso al código fuente para lograr cambios significativos).
- Libertad para distribuir copias, ya sea de manera gratuita o a cambio de un precio.
- Libertad para distribuir versiones modificadas del programa, permitiendo que la comunidad se beneficie de las mejoras introducidas.

Para ejercer estas libertades, es fundamental recibir el código fuente, lo que permite realizar modificaciones de manera conveniente. Esta característica del software libre también se conoce en inglés como "open source software" o, en español, como "software de fuentes abiertas" o "código abierto" (González-Barahona, 2011).

Arquitectura Cliente-Servidor

El modelo Cliente/Servidor representa una aplicación distribuida donde las tareas se dividen entre los proveedores de recursos o servicios (servidores) y los solicitantes (clientes). Los clientes realizan peticiones a una o varias aplicaciones servidores, que deben estar en ejecución para atender estas demandas (Marini, 2012).

Esta arquitectura también puede interpretarse como una forma de dividir y especializar programas y equipos para que cada uno realice sus funciones de manera eficiente, facilitando las actualizaciones y el mantenimiento del sistema que la implementa, según lo expresado por Lizama, Kindley y Jeria Morales (2016). Sus ventajas incluyen:

- Centralización.
- Facilidad de mantenimiento.
- Escalabilidad: la capacidad de clientes y servidores puede incrementarse de manera independiente.

Arquitectura basada en N-capas

La arquitectura basada en capas se concentra en agrupar funcionalidades relacionadas en una aplicación en distintas capas, dispuestas verticalmente unas sobre otras, cada una asociada con un rol o responsabilidad específica (Muñoz Serafín, 2018).

Se destaca el enfoque de tres capas para el desarrollo de software, frecuentemente empleado en sistemas con conexiones a bases de datos. Estas capas son: presentación, lógica y acceso a datos. La capa de presentación se sitúa en la parte superior, interactuando con el usuario y gestionando la información requerida. La capa intermedia es la de lógica, encargada de cumplir con los requisitos del sistema

o reglas de negocio. La capa más baja, la de acceso a datos, se encarga de recuperar y modificar datos desde el origen de datos.

Los beneficios de implementar esta arquitectura incluyen:

- Reutilización de componentes.
- Reducción de costos de mantenimiento.
- Interoperabilidad entre componentes de diferentes fabricantes.
- Facilidad de escalabilidad.

Patrones de diseño.

Los patrones de diseño identifican problemas recurrentes en un contexto particular y proponen soluciones comprobadas para abordarlos. Surgidos de la experiencia acumulada de expertos, en el ámbito del software proporcionan estructuras o esquemas para mejorar subsistemas, componentes o las relaciones entre ellos. Estos patrones establecen una forma estandarizada y repetible de interacción entre componentes, resolviendo así problemas generales de diseño en un entorno específico.

Según (Cáceres Tello), los objetivos de los patrones de diseño incluyen:

- Reducción de tiempos.
- Menor esfuerzo de mantenimiento.
- Incremento de la eficiencia.
- Garantía de la consistencia.
- Mejora en la fiabilidad.

Metodología de desarrollo de software.

Existen una gran variedad de metodologías para el desarrollo de software. Entre ellas se encuentran las tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Esta forma de desarrollo no resulta ser la más adecuada para muchos de los proyectos actuales donde el entorno del sistema es muy cambiante y se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. (Panadés & Letelier, 2003)

Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, en este mismo escenario emergen las metodologías ágiles como una posible respuesta para llenar este vacío metodológico. La filosofía de estas es centrarse en otras dimensiones como por ejemplo el factor humano o el producto software dándole mayor valor al individuo, a la colaboración con el cliente

y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. (Caballero Nuñez, 2009)

En su artículo, (Caballero Nuñez, 2009) también expresa las ventajas de las metodologías ágiles sobre las tradicionales:

- Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- Entrega continua y en plazos breves de software funcional.
- Trabajo conjunto entre el cliente y el equipo de desarrollo. Importancia y la simplicidad, eliminando el trabajo innecesario.
- Atención continua a la excelencia técnica y el buen diseño.
- Mejora continua de los procesos y el equipo de desarrollo.

Entre las metodologías ágiles más utilizadas se encuentran SCRUM y XP, de las cuales se realizará breve análisis a continuación.

SCRUM

Esta metodología centra su atención en las actividades de Gerencia y no especifica prácticas de Ingeniería. Fomenta el surgimiento de equipos auto dirigidos cooperativos y aplica inspecciones frecuentes como mecanismo de control. (Peralta, 2003)

SCRUM es un proceso ágil que nos permite centrarnos en ofrecer el más alto valor de negocio en el menor tiempo. (Cohen, 2013)

Es una metodología para el desarrollo de software iterativa e incremental, debe su nombre a la jugada de rugby llamada de la misma manera, se dice que es iterativa ya que se ejecuta en bloques temporales cortos y fijos (de no menos de dos semanas) que reciben el nombre de sprints y es incremental en tanto se obtienen funcionalidades del producto final al terminar cada iteración. Dentro de las características de SCRUM, se relaciona que está definido en base a roles, reuniones y artefactos. (Linares, Tovar, & Salazr, 2018)

Programación Extrema (XP)

La programación Extrema o eXtreme Programming (XP) es una metodología de desarrollo de la ingeniería de software. Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. (Mederos Candelario, 2016)

Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir

todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. (Díaz Labrador & Collazo Garcia, 2013)

Surge como una nueva manera de encarar proyectos de software, proponiendo una metodología basada esencialmente en la simplicidad y agilidad. (Joskowicz, 2008)

Los objetivos de XP son muy simples: la satisfacción del cliente, ya que esta metodología trata de dar al cliente el software que necesita y cuando lo necesita, por lo que debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo, tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

Diferencias entre SCRUM y XP según (Almseidin, Alrfou, & Alnidami, 2015):

- XP contiene prácticas de ingeniería a diferencia de SCRUM.
- XP permite los cambios sobre la marcha sin restricciones, pero en SCRUM si ya comenzó el Sprint no es posible hasta que comience el próximo.
- SCRUM toma más tiempo que XP para terminar un proyecto.
- XP posee un equipo con menos de 10 integrantes mientras que SCRUM posee más de un equipo.

Se ha decidido seleccionar la metodología XP para ponerla en práctica en el desarrollo de esta aplicación por las razones que se exponen a continuación:

- El equipo de trabajo cuenta con tres integrantes.
- Necesidad de realizar cambios sobre la marcha sin restricciones.
- Tiempo de desarrollo corto.

Aplicaciones Web

Las aplicaciones web son aquellas que los usuarios pueden acceder a través de un navegador al conectarse a un servidor. Permiten una interacción directa entre el usuario y la información, independientemente de la versión del sistema operativo instalado en el dispositivo del cliente.

- Entre sus ventajas destacan:
- Ahorro de tiempo: realización de tareas simples sin necesidad de instalar programas adicionales.
- Compatibilidad: funcionan con un navegador web actualizado.
- Ahorro de espacio: no utilizan capacidad de almacenamiento en el disco duro del usuario.
- Actualizaciones inmediatas: cualquier modificación realizada por el desarrollador se refleja de inmediato.
- Consumo eficiente de recursos: la mayor parte del procesamiento y almacenamiento se gestiona en el servidor.
- Multiplataforma: accesibles desde cualquier sistema operativo con un navegador.

- Disponibilidad global: al estar en línea, se pueden utilizar desde cualquier lugar con acceso a internet.

Herramientas y tecnologías

Lenguajes del lado del cliente

HTML (Lenguaje de Marcado de Hipertexto)

HTML es un lenguaje simple que estructura y presenta texto de manera agradable, permitiendo la incorporación de enlaces a documentos, fuentes de información y elementos multimedia. Su función principal es definir la estructura lógica del contenido, como títulos, párrafos, listas, entre otros, y aplicar diferentes efectos de presentación mediante otros lenguajes especializados.

A continuación, se presenta un resumen de las diferentes versiones:

- HTML o HTML1: presentado en 1991 por Tim Berners-Lee, surgió del Estándar de Lenguaje de Marcado Generalizado (SGML) junto con la creación de la World Wide Web.
- HTML+ o HTML2: iniciado en 1993 por David Ragget, formalizó extensiones de HTML, introdujo imágenes, tablas, formularios y líneas anidadas.
- HTML3: propuesto en 1995 por el World Wide Web Consortium (W3C), presentaba nuevas características como el flujo de texto alrededor de imágenes y la representación de elementos matemáticos, aunque tuvo complicaciones y diferentes versiones (HTML3.0, HTML3.1, HTML3.2) no bien recibidas por los desarrolladores de navegadores web
- HTML4: lanzado en junio de 1998, adoptó elementos específicos de los navegadores web y limpió el lenguaje HTML de elementos "desaprobados".
- HTML5: desarrollado a partir de 2007 por el Proyecto de Estándares Web, respaldado por el consorcio W3C y destinado a ejecutar aplicaciones completas. Introdujo mejoras significativas en controles para formularios, APIs, multimedia y estructura semántica, y en la actualidad es compatible con la mayoría de los navegadores.

CSS (Hojas de Estilo en Cascada)

CSS es un lenguaje que define el aspecto visual de los documentos HTML, permitiendo aplicar estilos específicos a elementos de una página web. Combinado con lenguajes como JavaScript, se pueden implementar efectos interactivos para los usuarios.

CSS3 es la versión más reciente y potente, precedida por CSS1 y CSS2.1. A continuación se muestra el soporte ofrecido por algunos de los navegadores más utilizados a las diferentes versiones de CSS.

Navegador	Motor	CSS 1	CSS 2.1 Casi	CSS 3
Internet Explorer	Trident	Completo desde la versión 6.0	completo desde la versión 7.0	Prácticamente nulo
Firefox	Gecko	Completo	Casi completo	Selectores, pseudo-clases y algunas propiedades
Safari	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Google Chrome	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades

Imagen 1 Soporte a CSS de navegadores actuales (Eguíluz Pérez, 2008)

El navegador Internet Explorer es considerado adecuado en términos de compatibilidad con CSS a partir de su versión 7. Sin embargo, la versión 6, aún utilizada por muchos usuarios, presenta deficiencias significativas y numerosos errores en el soporte de CSS. Por su parte, Internet Explorer 8 asegura un soporte completo de CSS 2.1 equiparable al de otros navegadores.

Entre las ventajas de usar CSS se incluyen:

- Separación contenido-presentación: al separar estos dos elementos, el mantenimiento de las páginas y el procesamiento de la información se vuelven más sencillos.
- Consistencia del sitio: una hoja de estilos global puede aplicarse a todo el sitio, facilitando futuras modificaciones al realizar cambios únicos en esa hoja.
- Ahorro en la transferencia: al ser alojada en la caché del navegador, una hoja de estilo externa puede ser reutilizada en múltiples páginas, reduciendo significativamente la transferencia de datos en la red.
- Formateo de página: pequeños cambios en la hoja de estilos pueden modificar la apariencia de toda una página.
- Compatibilidad: la mayoría de los navegadores soportan características avanzadas de CSS, permitiendo que incluso aquellos que no lo hacen puedan leer el código de la página y mostrarlo sin afectar su posicionamiento.

JavaScript

JavaScript es un lenguaje de programación simple que se inserta en las páginas HTML y se ejecuta en el navegador, como Mozilla Firefox o Microsoft Internet Explorer. Estos scripts, compuestos principalmente por funciones, se activan cuando ocurre algún evento en la página. Por ejemplo, pueden cambiar la apariencia de un botón al pasar el ratón por encima o abrir una nueva ventana al hacer clic en un enlace.

Desarrollado por Netscape, JavaScript se originó a partir del lenguaje Java, aunque tiene diferencias importantes. Java es un lenguaje de programación completo, ejecutable fuera de un navegador, mientras que JavaScript está diseñado para funcionar dentro de una página HTML.

Si bien JavaScript fue estandarizado por ECMA e ISO en 1997, los diferentes modelos de objetos implementados por navegadores como Netscape y Explorer no cumplían con el estándar, lo que causaba problemas de compatibilidad. Para abordar esto, el W3C introdujo el Document Object Model (DOM), adoptado por la mayoría de navegadores actuales, como Explorer o Firefox.

Lenguajes de programación del lado del servidor

C# (Asp .NET CORE)

ASP.NET Core es un marco de desarrollo de software de código abierto y multiplataforma diseñado por Microsoft para construir aplicaciones web modernas y escalables. Esta tecnología proporciona un entorno flexible y potente para crear aplicaciones web y servicios RESTful. Algunas de sus características principales incluyen:

- **Multiplataforma:** Compatible con sistemas operativos como Windows, macOS y Linux, lo que permite desarrollar aplicaciones que pueden ser desplegadas en diferentes entornos (Documentación de Microsoft).
- **Alto rendimiento:** Ofrece un alto rendimiento debido a su arquitectura modular y ligera, lo que permite que las aplicaciones se ejecuten más rápido y consuman menos recursos (Freeman, A., Pro ASP.NET Core MVC 2).
- **Soporte para contenedores:** Integra características que facilitan la creación y el despliegue de aplicaciones en contenedores como Docker, lo que proporciona portabilidad y escalabilidad (Documentación de Microsoft).
- **API flexible y moderna:** Proporciona una arquitectura que permite el desarrollo de servicios RESTful y APIs de forma sencilla y eficiente (Galloway, J., & Hanselman, S., ASP.NET Core in Action).
- **Sistema de enrutamiento potente:** Ofrece un enrutamiento flexible y potente para manejar solicitudes HTTP entrantes y dirigir las a los controladores correspondientes (Documentación de Microsoft).

Ventajas:

- **Rendimiento mejorado:** Su arquitectura modular y ligera permite un rendimiento superior en comparación con versiones anteriores de ASP.NET (Freeman, A., Pro ASP.NET Core MVC 2).
- **Multiplataforma:** La capacidad de ejecutarse en diferentes sistemas operativos amplía su alcance y su uso en distintos entornos (Documentación de Microsoft).

Capítulo I

- Flexible y modular: Permite a los desarrolladores elegir e implementar solo los componentes necesarios para sus aplicaciones, lo que resulta en una mayor flexibilidad (Galloway, J., & Hanselman, S., ASP.NET Core in Action).

Desventajas:

- Curva de aprendizaje: Puede tener una curva de aprendizaje más empinada para aquellos nuevos en el desarrollo web o en el ecosistema de Microsoft (Freeman, A., Pro ASP.NET Core MVC 2).
- Madurez de la comunidad y librerías: Aunque ha crecido significativamente, algunas librerías y recursos de la comunidad pueden ser más limitados en comparación con otros marcos más establecidos (Galloway, J., & Hanselman, S., ASP.NET Core in Action).

Gestor de bases de datos. MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto muy popular y ampliamente utilizado en el desarrollo de aplicaciones web y empresariales. Algunas de sus características principales incluyen:

- Escalabilidad y rendimiento: MySQL está diseñado para manejar grandes volúmenes de datos y proporcionar un rendimiento eficiente en entornos de alta demanda (DuBois, P., MySQL).
- Amplia compatibilidad: Es compatible con múltiples plataformas y sistemas operativos, lo que permite su implementación en una variedad de entornos (Manual de referencia de MySQL).
- Seguridad: Ofrece funciones de seguridad robustas, como la gestión de usuarios y permisos, así como el cifrado de datos para proteger la información sensible (Forta, B., MySQL Crash Course).
- Replicación y alta disponibilidad: MySQL admite técnicas de replicación que permiten crear copias de bases de datos para mejorar la disponibilidad y la redundancia en caso de fallos (DuBois, P., MySQL).
- Soporte transaccional: Proporciona soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), garantizando la integridad de los datos en entornos de transacciones complejas (Forta, B., MySQL Crash Course).

Ventajas:

- Rendimiento sólido: Ofrece un rendimiento eficiente y escalabilidad para manejar grandes volúmenes de datos (DuBois, P., MySQL).

- **Amplia compatibilidad:** Es compatible con una amplia gama de sistemas operativos y lenguajes de programación, lo que facilita su integración en diversos entornos (Manual de referencia de MySQL).
- **Comunidad y soporte extenso:** Tiene una gran comunidad de usuarios y una abundancia de recursos disponibles en línea para ayudar en el desarrollo y la resolución de problemas (Manual de referencia de MySQL).

Desventajas:

- **Complejidad en configuración avanzada:** Configurar y optimizar MySQL para escenarios avanzados puede requerir un conocimiento profundo de sus características y ajustes (Manual de referencia de MySQL).
- **Limitaciones en características avanzadas:** Algunas características más avanzadas pueden no ser tan completas o avanzadas como en otros sistemas de gestión de bases de datos (Manual de referencia de MySQL).

Frameworks y librerías

Un framework, también conocido como entorno o marco de trabajo, representa un conjunto estandarizado de conceptos, prácticas y criterios diseñados para abordar problemáticas específicas. Sirve como una referencia fundamental para resolver nuevos desafíos que presenten similitudes con situaciones previamente establecidas.

En el ámbito de las aplicaciones web, un framework se concibe para respaldar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Estos frameworks están ideados para aligerar la carga asociada a las tareas habituales en el desarrollo web.

Un framework web proporciona una infraestructura de programación para tus aplicaciones, permitiéndote enfocarte en la creación de código claro y de fácil mantenimiento, sin la necesidad de reinventar soluciones ya existentes.

React JS

React JS es una biblioteca de JavaScript de código abierto y mantenida por Facebook (Freeman & Robins, 2017). Es ampliamente utilizada para construir interfaces de usuario interactivas y dinámicas en aplicaciones web de una sola página (Single Page Applications - SPAs) (Documentación de React). Algunas de sus características clave son:

- **Componentización:** React JS se basa en el concepto de componentes reutilizables. Permite crear interfaces de usuario mediante la composición de componentes más pequeños, lo que facilita el mantenimiento y la escalabilidad del código (Freeman & Robins, 2017).

Capítulo I

- Virtual DOM (Document Object Model): Utiliza un Virtual DOM para mejorar el rendimiento de las aplicaciones. Al realizar actualizaciones, React compara el Virtual DOM con el DOM real y aplica solo los cambios necesarios, lo que reduce la manipulación directa del DOM y mejora la velocidad (Documentación de React).
- JSX (JavaScript XML): React utiliza JSX, una extensión de JavaScript que permite escribir código HTML dentro de archivos JavaScript. Esto facilita la creación de componentes y mejora la legibilidad del código (Freeman & Robins, 2017).
- Unidireccionalidad de datos (One-way data binding): Emplea un flujo de datos unidireccional, lo que significa que los datos fluyen en una sola dirección, desde el componente principal hacia los componentes secundarios, facilitando el seguimiento de los cambios y la depuración (Documentación de React).

Ventajas:

- Reutilización de componentes: La capacidad de crear componentes reutilizables facilita el desarrollo y el mantenimiento del código (Freeman & Robins, 2017).
- Rendimiento mejorado: El uso del Virtual DOM y su enfoque en la actualización eficiente del DOM contribuyen a un rendimiento óptimo de las aplicaciones (Documentación de React).
- Comunidad activa y soporte: React cuenta con una gran comunidad de desarrolladores y abundante documentación que facilita el aprendizaje y la resolución de problemas (Freeman & Robins, 2017).

Desventajas:

- Curva de aprendizaje inicial: Para los desarrolladores nuevos en React, puede haber una curva de aprendizaje al comprender los conceptos fundamentales, como JSX y el flujo de datos unidireccional (Documentación de React).
- Configuración inicial compleja: Configurar un entorno de desarrollo puede ser complejo debido a la necesidad de herramientas adicionales como Babel y Webpack para aprovechar al máximo las características de React (Freeman & Robins, 2017).

Material UI

Material UI es una popular biblioteca de componentes de interfaz de usuario para React JS que implementa los principios del diseño Material de Google (Darmawan, 2020). Esta biblioteca proporciona un conjunto de componentes predefinidos, como

botones, barras de progreso, tarjetas, entre otros, diseñados siguiendo las pautas de diseño de Material Design (Material-UI Documentation).

Características clave de Material UI:

- Componentes consistentes: Ofrece una amplia gama de componentes predefinidos que siguen las directrices de diseño de Material Design, lo que garantiza la consistencia visual en las aplicaciones (Material-UI Documentation).
- Personalización sencilla: Permite una fácil personalización de estilos y temas para adaptarse a las necesidades específicas del proyecto (Darmawan, 2020).
- Fácil integración con React: Está diseñado específicamente para trabajar con React JS, lo que facilita su integración y uso dentro de aplicaciones React (Material-UI Documentation).

Ventajas de Material UI:

- Consistencia visual: Al seguir las directrices de Material Design, proporciona una interfaz de usuario consistente y atractiva (Darmawan, 2020).
- Facilidad de uso: Ofrece una amplia gama de componentes listos para usar que simplifican el desarrollo de interfaces de usuario complejas (Material-UI Documentation).
- Personalización: Permite personalizar fácilmente estilos y temas para adaptarse al diseño específico del proyecto (Darmawan, 2020).

Desventajas:

- Curva de aprendizaje inicial: Para los desarrolladores nuevos en Material UI, puede llevar tiempo familiarizarse con su amplia gama de componentes y opciones de personalización (Material-UI Documentation).
- Posible sobrecarga de estilos: En proyectos complejos, la gestión de estilos personalizados podría ser complicada y resultar en una sobrecarga de estilos (Darmawan, 2020).

Axios

Axios es una popular biblioteca de JavaScript utilizada para realizar solicitudes HTTP desde navegadores web o entornos de Node.js (Hussain, 2020). Esta biblioteca proporciona una interfaz simple y fácil de usar para realizar peticiones HTTP, lo que incluye soporte para solicitudes GET, POST, PUT, DELETE y otras operaciones HTTP (Axios GitHub Repository).

Características clave de Axios:

- Sencillez de uso: Proporciona una sintaxis simple y clara para realizar solicitudes HTTP, lo que facilita su implementación en aplicaciones web (Axios GitHub Repository).
- Soporte amplio para promesas: Axios utiliza promesas nativas de JavaScript para manejar solicitudes asíncronas, lo que permite un código más legible y mantenible (Hussain, 2020).
- Funcionalidades adicionales: Ofrece características avanzadas como la posibilidad de interceptar solicitudes y respuestas, establecer configuraciones globales y cancelar solicitudes (Axios GitHub Repository).

Entity Framework Core

Entity Framework Core es un ORM (Object-Relational Mapping) de código abierto desarrollado por Microsoft y es una evolución de Entity Framework 6, diseñado para interactuar con bases de datos relacionales en aplicaciones .NET Core y .NET 5 (Smith, 2020). Proporciona un conjunto de herramientas y funciones para mapear objetos de dominio a esquemas de bases de datos relacionales y viceversa, permitiendo a los desarrolladores trabajar con bases de datos utilizando objetos y consultas LINQ (Language Integrated Query) (Smith, 2020).

Características clave de Entity Framework Core:

- Mapeo objeto-relacional (ORM): Permite a los desarrolladores trabajar con objetos de dominio en lugar de escribir consultas SQL directamente, facilitando el desarrollo y mantenimiento del código (Smith, 2020).
- Abstracción de la base de datos: Proporciona una capa de abstracción sobre la base de datos, lo que significa que los desarrolladores pueden utilizar modelos de datos independientes del proveedor de la base de datos (Smith, 2020).
- Migraciones de base de datos: Permite realizar cambios en el esquema de la base de datos mediante migraciones de código, lo que simplifica la administración de versiones y actualizaciones de la base de datos (Smith, 2020).
- Soporte para múltiples proveedores de bases de datos: Ofrece la capacidad de trabajar con varios motores de bases de datos como SQL Server, MySQL, PostgreSQL, SQLite, entre otros (Smith, 2020).

IDE - Rider

JetBrains Rider es un entorno de desarrollo integrado (IDE) multiplataforma creado por JetBrains, diseñado específicamente para desarrolladores que trabajan con

tecnologías .NET, .NET Core, ASP.NET, Xamarin y Unity (JetBrains, s.f.). Se destaca por su soporte integral para múltiples lenguajes de programación y su capacidad para ofrecer herramientas avanzadas de productividad y depuración.

Características clave de JetBrains Rider:

- Soporte multiplataforma: Está disponible para Windows, macOS y Linux, permitiendo a los desarrolladores trabajar en sus sistemas operativos preferidos (JetBrains, s.f.).
- Compatibilidad con lenguajes de programación: Ofrece un amplio soporte para varios lenguajes, incluyendo C#, VB.NET, F#, JavaScript, TypeScript, HTML, CSS, SQL, entre otros (JetBrains, s.f.).
- Herramientas de productividad avanzadas: Incluye características como refactorización de código, finalización de código inteligente, navegación rápida, y generación de código automática para mejorar la eficiencia del desarrollo (JetBrains, s.f.).
- Depuración y perfilado: Proporciona herramientas avanzadas de depuración y perfilado para ayudar a identificar y solucionar problemas de manera eficiente durante el desarrollo de aplicaciones (JetBrains, s.f.).
- Control de versiones e integración: Ofrece integración con sistemas de control de versiones como Git, SVN, Mercurial, y TFS, facilitando la colaboración en equipo (JetBrains, s.f.).

Herramientas de Modelado

Bizagi Process Modeler.

(OTAVALO MOROCHO, 2019) plantea que Bizagi es una herramienta de uso ágil y sencillo que permite el diseño, documentación y evolución de un BPM (Gestión de procesos de negocios). Con él se puede elaborar un diagrama de flujo de un proceso en función de sus variables u otros elementos que intervienen en el mismo. Este software de libre uso es una herramienta gerencial importante que permite mejorar la gestión en todo tipo de empresa, pues la aplicación en el campo administrativo es universal y entendible.

(VILCA OQUENDO, 2017) refleja que entre las características que presenta este software están:

- Se pueden crear documentos en alta calidad en Word, PDF, SharePoint o Wiki, así mismo pueden ser importados y exportados usando Visio, XML, entre otras herramientas.
- Permite documentar y mapear rápido y fácil los procesos de manera eficiente, usando fomentar la colaboración en la organización.
- La aplicación es gratuita de Internet.

Conclusiones del primer capítulo

En este capítulo se establecen los fundamentos teóricos que respaldan la investigación, explorando diversos aspectos y conceptos relacionados con el ámbito del problema. El análisis de los antecedentes del proceso bajo estudio, junto con las características generales de la entidad, así como el examen de herramientas, tecnologías y metodologías de desarrollo, conduce a las siguientes conclusiones:

- Se reconoce la necesidad de implementar un software adaptado a las especificaciones de comercialización mayorista de la Sucursal CIMEX Matanzas, con el propósito de mejorar la eficiencia en esa área.
- Se comprende que la implementación de metodologías ágiles para el desarrollo de software, como XP, junto con la combinación de las tecnologías y herramientas seleccionadas, resultan las más idóneas para la creación de la herramienta web propuesta, buscando resolver la problemática planteada.

Capítulo II Análisis y diseño del sistema

Introducción

En este capítulo se exploran los componentes esenciales para describir la solución propuesta. Su implementación se apoya en el uso de tecnologías y herramientas previamente definidas en la sección anterior, y se gestiona siguiendo los principios y reglas de la metodología XP. Se detalla el desempeño de los roles, eventos y artefactos involucrados en la construcción del software, haciendo uso de esta metodología. Además, se incorporan otros artefactos que no forman parte de la metodología mencionada, con el fin de mejorar la comprensión del sistema a implementar. Se examinan los resultados obtenidos a través del análisis de viabilidad realizado.

Modelo de Proceso

Diagrama de Procesos de Negocio (BPD)

El Diagrama de Procesos de Negocio (BPD) es una representación gráfica diseñada para mostrar la secuencia de todas las actividades que tienen lugar durante un proceso. Basado en la técnica de "Flow Chart", este diagrama representa visualmente una secuencia lógica de operaciones de trabajo. Utiliza diferentes símbolos para representar operaciones, datos, direcciones de flujo y recursos, con el propósito de definir, analizar o resolver un problema (Gómez Mondragón, 2016).

Descripción del Negocio

El proceso de oferta y reserva de productos comienza cuando el usuario interesado en percibir el servicio accede a la herramienta web destinada para ello. El usuario debe estar registrado anteriormente. El usuario puede cargar los productos que desee en su carrito de compra. Un usuario puede cargar productos en el carrito, pero necesitará estar asociado a un cliente para efectuar la reserva. El carrito almacena la lista de productos, pero no representa una reserva de los productos. Una vez que el usuario cargue todos los productos que desea comprar procederá a confirmar el carrito y un pedido será generado.

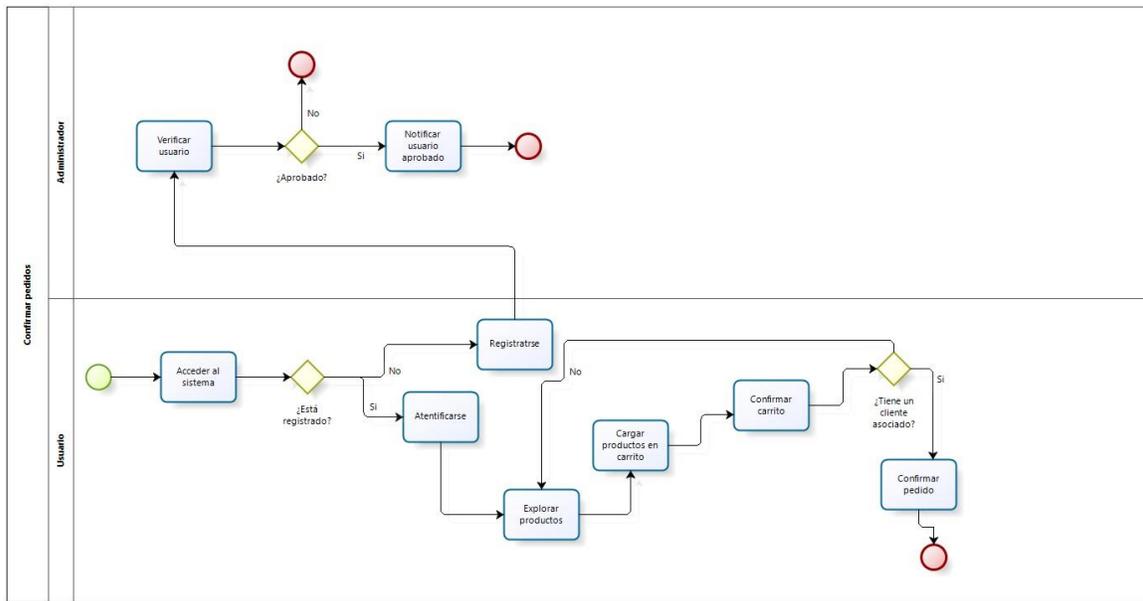


Imagen 2 Modelado del negocio (Elaboración propia)

Modelo de la base de datos

Un Modelo Conceptual es un componente, ya sea lógico o físico, que contribuye a la comprensión de un problema. Basándose en lo mencionado y considerando el análisis de la información recopilada durante las entrevistas con los clientes sobre los datos, sus características y el flujo de información a abordar, se establece el siguiente Modelo Conceptual de la Base de Datos:

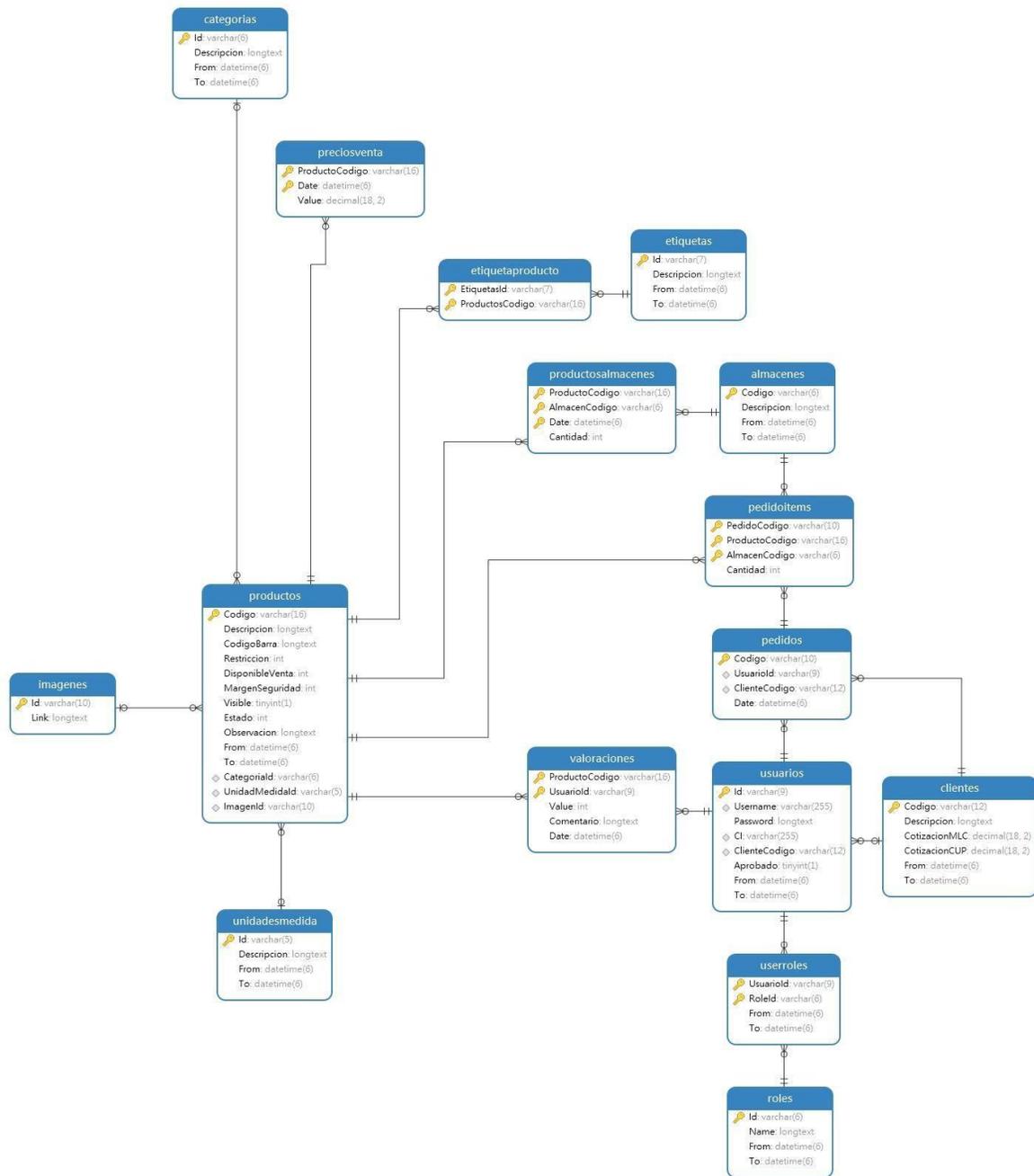


Imagen 3. Modelo conceptual de la base de datos (Elaboración propia)

Explicación de la solución propuesta

Se propone el desarrollo de una herramienta web que permita a la Sucursal CIMEX Matanzas implementar el comercio electrónico mayorista, aprovechando sus beneficios para ofrecer una manera eficiente de presentar y reservar los productos que comercializa. Esto busca mejorar la eficacia de las operaciones comerciales de la entidad y brindar a los clientes un servicio de mayor calidad, comodidad y seguridad.

Los usuarios que interactuarán con la aplicación desempeñarán los siguientes roles:

Administrador del sistema: encargado de gestionar usuarios internos, asignar permisos, configurar y auditar la aplicación.

Usuario: usuario registrado que interactúa principalmente con el sistema, explorando, cargando productos en el carrito y generando pedidos.

Para mostrar la oferta de productos disponibles y aquellos con saldo reciente, se utiliza una plantilla que los presenta de manera ordenada y clasificada, además de ofrecer opciones de búsqueda mediante filtros y la agrupación por categorías.

Patrones de diseño usados

Patrón Modelo – Vista – Controlador (MVC).

El MVC es un patrón de arquitectura de software que divide una aplicación en tres aspectos: procesamiento, salida y entrada. Para lograrlo, emplea las siguientes abstracciones:

- **Modelo:** encapsula los datos y las funcionalidades pertinentes, sin representar salidas o entradas.
- **Vista:** muestra la información al usuario, usando un controlador asociado que utiliza el modelo para obtener los datos necesarios.
- **Controlador:** recibe las peticiones, traduciéndolas en solicitudes al modelo o vista.

Patrón repositorio

El patrón repositorio está destinado a crear una capa de abstracción entre la capa de acceso a datos (DAL) y la capa empresarial para que pueda ayudar a aislar la aplicación de los cambios en el manejo del almacén de datos. En su libro, (Evans, 2003) define el patrón Repositorio como un mecanismo para encapsular el comportamiento de almacenamiento, obtención y búsqueda, de una forma similar a una colección de objetos.

(Camarena Sagredo, Trueba Espinosa, Martínez Reyes, López & García, 2012) señala las ventajas del uso del patrón MVC:

1. Facilita la sustitución de interfaces de usuario.
2. Genera componentes para interfaces.
3. Diseña vistas simultáneas para un mismo modelo.
4. Permite cambios sencillos en las interfaces.

Inyección de dependencia

La Inyección de Dependencia es un patrón que ayuda a desacoplar componentes en un programa. En lugar de que un objeto cree sus propias dependencias, estas se proporcionan desde fuera, permitiendo así mayor flexibilidad, reutilización de código y facilitando las pruebas. Un contenedor gestiona estas dependencias y las pasa a los objetos que las necesitan, mejorando el modularidad del software.

Fase de Planificación

La etapa inicial de planificación es fundamental en todo proyecto. Su objetivo es organizar eficientemente el prototipo inicial del problema y así garantizar un buen inicio para alcanzar una solución efectiva. En este sentido, se desarrollarán las Historias de Usuario en línea con las ideas del cliente sobre el software. Estas historias servirán como punto de partida para el resto de la planificación del proyecto.

Equipo de Trabajo

El equipo de trabajo está conformado por el jefe de proyecto MsC. Orlenys Machin Díaz, el cliente, gerente de ventas mayoristas Jany Rodríguez y el programador y encargado de las pruebas, estudiante David Chávez Armenteros.

Las características fundamentales del equipo de trabajo son:

- El cliente forma parte del equipo, revisa lo planificado en cada encuentro y acuerda los plazos de entrega para la solución de cada historia de usuario.
- Se trabaja cuarenta horas semanales, cumpliendo el principio de que horas extras no aumenta la productividad, sino que desmotiva al equipo, ya que así se evita el cansancio de los programadores y con esto disminuyen las probabilidades de introducir errores.
- Se adopta un diseño que propicie la escalabilidad y flexibilidad de cara los cambios, teniendo en cuenta que el software es un sistema de uso corporativo y las necesidades de la empresa pueden llegar variar con el tiempo. De esta forma se garantiza que el software pueda evolucionar a la par que la unidad a la cual está destinada e incluso pueda adaptarse a las necesidades de otras empresas.

- Se trabaja desarrollando pequeñas versiones funcionales que se incrementan poco a poco, ante los fallos o cambios se produce la reprogramación del código, sin variar su funcionalidad.
- El desarrollo es guiado a través de pruebas, las que definen el alcance del proyecto trabajando para aumentar el nivel de conformidad del cliente, contribuyendo a que se detecten errores que los programadores pudieran pasar por alto.

Tabla 1. Equipo de trabajo y roles

Miembros	Roles XP
MsC. Orlenys Machin Díaz	Jefe del Proyecto
Jany Rodríguez	Cliente
David Chávez Armenteros	Programador y Encargado de las Pruebas

Requisitos funcionales

Los requisitos funcionales son aquellos que describen qué debe hacer el sistema, desde el punto de vista de las necesidades del usuario, son capacidades o condiciones que debe cumplir el sistema y que están fuertemente ligados a las opciones del programa (Cardozzo y col., 2016). A continuación, se muestran los requisitos funcionales del sistema

1. Integración con Sentai
 - 1.1. Leer productos de Sentai
 - 1.2. Leer almacenes de Sentai
 - 1.3. Exportar datos a Sentai
2. Autenticar usuario
 - 2.1. Iniciar sesión
 - 2.2. Cerrar sesión
3. Registro usuario
 - 3.1. Registrar usuario
4. Gestión de usuarios
 - 4.1. Listar usuarios
 - 4.2. Aprobar usuarios

- 4.3. Actualizar usuarios
- 4.4. Eliminar usuarios
5. Gestión de productos
 - 5.1. Actualizar productos
6. Gestionar carrito de compra
 - 6.1. Añadir productos al carrito
 - 6.2. Cambiar cantidades de producto en el carrito
 - 6.3. Bajar productos del carrito
7. Gestionar clientes
 - 7.1. Listar clientes
 - 7.2. Adicionar clientes
 - 7.3. Actualizar clientes
 - 7.4. Eliminar clientes
8. Gestión de pedidos
 - 8.1. Listar pedidos
 - 8.2. Confirmar pedidos
 - 8.3. Eliminar pedidos
9. Gestión de categorías
 - 9.1. Listar categorías
 - 9.2. Adicionar categorías
 - 9.3. Actualizar categorías
 - 9.4. Eliminar categorías
10. Gestión de unidades de medidas
 - 10.1. Listar unidades de medidas
 - 10.2. Adicionar unidades de medidas
 - 10.3. Actualizar unidades de medidas
 - 10.4. Eliminar unidades de medidas
11. Gestionar Almacenes
 - 11.1. Listar almacenes
 - 11.2. Actualizar almacenes

Requisitos no funcionales

Los requisitos no funcionales son aquellas características que definen las cualidades del producto más allá de su funcionalidad directa. Estos requisitos buscan asegurar aspectos como la usabilidad, rendimiento, seguridad, entre otros, y suelen ser fundamentales para el éxito del producto.

En el contexto del software abordado en este proyecto, se han identificado los siguientes requisitos no funcionales:

- Interfaz:
 - La interfaz debe ser intuitiva, fácil de leer y amigable para los usuarios. Debe ser atractiva visualmente y garantizar una experiencia de uso sencilla.

- Usabilidad:
 - El sistema debe ofrecer asistencia clara para guiar a los usuarios, especialmente aquellos con conocimientos básicos en informática, asegurando que puedan utilizar todas las funcionalidades de manera efectiva.
- Rendimiento:
 - Se requiere que la aplicación utilice recursos mínimos y pueda admitir simultáneamente un total aproximado de 100 a 150 usuarios. Además, los tiempos de respuesta para actualizaciones no deben exceder los 3 segundos, y las recuperaciones de información no deben tardar más de 10 segundos.
- Soporte:
 - Se necesita un servidor de bases de datos capaz de manejar grandes volúmenes de datos. Se recomienda emplear el servicio de hosting ofrecido por ETECSA a través de su Centro de Datos.
- Portabilidad:
 - El sistema debe ser compatible con múltiples plataformas, principalmente Linux, Mac y Windows, para garantizar su accesibilidad a través de diferentes sistemas operativos.
- Seguridad:
 - Se requiere autenticación de usuarios antes de acceder al sistema, así como limitación de acceso a interfaces e información basada en roles de usuarios.
 - Para prevenir la captura y robo de información, se debe implementar un Token en los métodos POST para el envío de información.
- Integridad:
 - Tanto la herramienta Web como la Base de Datos se hospedarán en un Servidor encontrándose protegidos de accesos no autorizados, divulgación y comprometimiento de la fuente de datos.
- Confiabilidad:
 - El software debe contar con el respaldo necesario para garantizar su recuperación ante fallas eléctricas o de comunicación.
- Requisitos de diseño:
 - Para el diseño e implementación del sistema se debe utilizar el Marco de Trabajo Django que presenta grandes ventajas para la elaboración de aplicaciones web por su robustez, flexibilidad y portabilidad.
- Requerimiento de Ayuda y Documentación:
 - Se propone que el sistema cuente con un manual de ayuda general, y además ofrecer guía u orientación a través de las diferentes interfaces que presente.

Historias de usuario

Una historia de usuario es una técnica utilizada en el desarrollo de software para capturar los requisitos funcionales desde la perspectiva del usuario. Una historia de usuario describe una funcionalidad o característica específica del sistema en un formato simple y comprensible, generalmente narrado desde el punto de vista del usuario final (Cohn,2004b)

En la Tabla siguiente se muestran todas las HU planificadas inicialmente, en las que queda definido el nivel de prioridad con el que deben darle solución a las HU (P), el riesgo en desarrollo (R). Se define también la estimación del esfuerzo (E) requerido, que no es más que el tiempo en el que se concibió inicialmente el desarrollo de cada HU.

Tabla 2. Resumen de historias de usuarios (Elaboración propia)

No.	Nombre	P	R	E	Iteración	Entrega
1	Diseño de la base de datos	A	A	1	1	1
2	Creación de arquitectura base	A	A	2	1	1
3	Lectura Sentai	A	M	1	1	1
4	Autenticación	A	A	0.5	1	1
5	Registro	A	A	0.5	1	1
6	Gestionar Usuarios	A	A	1	2	2
7	Gestionar Unidades de Medida	M	M	0.5	2	2
8	Gestionar Almacenes	A	A	1	2	2
9	Gestionar Categorías	M	M	0.5	2	2

10	Gestionar Productos	A	A	1	3	3
11	Gestionar Carrito de Compra	A	A	1	3	3
12	Gestionar Clientes	A	A	1	4	4
13	Gestionar Pedidos	A	A	1	4	4
14	Confirmar pedidos	A	A	1	5	5
15	Escritura Sentai	A	M	0.5	5	5
Totales				13.5	5	5

A continuación, se muestran algunas de las historias de usuarios más importantes para la construcción del sistema.

Tabla 3. Historia de Usuario 4. Fuente (Elaboración propia)

Historia de Usuario	
Número: 4	
Nombre historia: Autenticación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1

<p>Descripción:</p> <p>El sistema debe brindar la funcionalidad de autenticación por medio de correo electrónico y contraseña. El sistema no debe permitir al usuario acceder al sistema si no está autenticado y debe dar accesos a los diferentes recursos de acuerdo a los roles del usuario.</p>
<p>Observaciones: El sistema no debe permitir autenticarse ni cerrar la sesión más de una vez.</p> <p>Un Rol de Usuario = Rol de usuario que describe un conjunto de permisos sobre las funcionalidades de la aplicación.</p>

Tabla 4. Historia de Usuario 6. Fuente (Elaboración propia)

Historia de Usuario	
Número: 6	
Nombre historia: Gestionar usuarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
<p>Descripción:</p> <p>El sistema debe brindar las funcionalidades de listar, actualizar y eliminar usuarios a través del panel de administración. El sistema brindara controles de búsqueda y paginación de usuarios.</p>	
<p>Observaciones: El sistema debe tener en cuenta las reglas de validaciones de usuarios como mismo están en el registro.</p>	

Tabla 5. Historia de Usuario 11. Fuente (Elaboración propia)

Historia de Usuario	
Número: 11	
Nombre historia: Gestionar Carrito de Compra	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 3
<p>Descripción:</p> <p>El sistema debe brindar las funcionalidades de montar, bajar y modificar las cantidades de productos en el carrito de compra</p> <p>El carrito de compra solo podrá ser confirmado si tiene productos en el</p>	
<p>Observaciones: El sistema debe tener en cuenta las reglas de validaciones de stock disponible de cada producto.</p> <p>El carrito de compra se almacenará en el almacenamiento local del navegador.</p>	

Tabla 6. Historia de Usuario 14. Fuente (Elaboración propia)

Historia de Usuario	
Número: 14	
Nombre historia: Confirmar pedidos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 5
<p>Descripción:</p> <p>El sistema debe brindar la funcionalidad de confirmación de pedidos.</p> <p>Un pedido se puede confirmar una vez tenga al menos un producto en el carrito de compra</p> <p>El sistema debe mostrar las cotizaciones para cada moneda del cliente asociado al usuario</p>	

Observaciones: El usuario debe estar asociado a un cliente para confirmar un pedido.

Estudio de factibilidad

Uno de los factores importantes que se deben tener en cuenta en el desarrollo de un software, es si las ventajas de la aplicación propuesta justifican o no su costo. Además de realizar una estimación del esfuerzo y el tiempo de desarrollo es necesario llevar a cabo un análisis de los beneficios tangibles e intangibles que reportará el proyecto. Se estiman un total de 13.5 semanas de trabajo de implementación y si a esto se suma el tiempo de realización de pruebas de aceptación, 5 semanas, una para cada iteración del plan, resulta un acumulado total de 18.5 semanas, que sería un total de 5 meses aproximadamente.

Utilizando la fórmula de Boehm: $\text{Costo} = \text{CH} * \text{SM} * \text{TD}$

Donde:

CH: Cantidad de hombres

SM: Salario mensual por persona

TD: Tiempo de desarrollo total estimado para el proyecto

Costo estimado del proyecto:

CH: 1, SM: \$3000.00, TD: 5

$\text{Costo} = \text{CH} * \text{SM} * \text{TD}$

$\text{Costo} = 1 * 3000 * 5$

$\text{Costo} = \$15000.00$

Considerando como valor para SM de \$3000.00 (salario mínimo) donde la cantidad de hombres es 1 y la duración estimada es de 5 meses, entonces al sustituir en la fórmula y calcular se obtiene un costo de \$15000.00. Teniendo en cuenta que no es un proyecto costoso, resulta provechoso el empleo de tiempo y esfuerzo en la realización de este.

Pruebas de aceptación

Las pruebas de aceptación son un recurso fundamental durante el desarrollo de un sistema basado en la metodología ágil. Con base en los principios propuestos en ésta, es posible aplicar los “tests” durante las diferentes iteraciones. Permiten evaluar el grado de satisfacción del cliente con el producto que se está desarrollando. Se elaboran en paralelo con el desarrollo del sistema, y adaptándose a sus cambios. De esta manera, las pruebas de aceptación se ejecutan ágilmente para corregir los errores oportunamente.

A continuación, se muestran algunas de las pruebas realizadas

Tabla 7. Caso de Prueba 3 (Elaboración propia)

Caso de Prueba	
Número Caso de Prueba: 3	Número Historia de Usuario: 4
Nombre Caso de Prueba: Acceso a la aplicación mediante usuarios registrados.	
Descripción: Se inserta el nombre de usuario y la contraseña para acceder a la aplicación. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación, se tratará de acceder a la aplicación poniendo una ruta destino para comprobar que no acceda a menos que se autentique correctamente. Se intentará acceder desde un usuario marcado como noaprobado para constatar que se deniegue el acceso. Luego se insertarán los datos de manera correcta con un usuario activo para comprobar el acceso.	
Condiciones de ejecución: El usuario está registrado y la contraseña es correcta.	
Entradas: <ol style="list-style-type: none"> 1. Presionar el botón Acceder con los campos en blanco. 2. Fijar una ruta para acceder a ella sin haberse autenticado. 3. Intentar acceder a la aplicación con un usuario inactivo. 4. Insertar los datos correctamente y presionar el botón Acceder. 	
Resultado esperado: devuelve un mensaje de error si no existe, está en estado no-aprobado o la contraseña es incorrecta. No se puede acceder suministrando una ruta de acceso sin antes estar autenticado. El sistema solo permite acceder si el usuario está registrado, la contraseña es correcta y está marcado como aprobado.	
Evaluación: Prueba Satisfactoria	

Tabla 8. Caso de Prueba 4 (Elaboración propia)

Caso de Prueba

Número Caso de Prueba: 4	Número Historia de Usuario: 5
Nombre Caso de Prueba: Registro de usuario	
Descripción: Se probará si el sistema permite crear usuarios nuevos. Se verifica que el usuario no exista aún en la base de datos. Retornando un mensaje de error si existe o se suministraron valores erróneos. Si todo es correcto se almacenará el usuario en la base de datos en estado no-aprobado.	
Condiciones de ejecución: El usuario no puede estar autenticado.	
Entradas: <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar un registro con un nombre de usuario ya existente en la base de datos. 4. Modificar los datos del proceso dejando campos en blanco. 5. Verificar que se muestren los datos insertados. 	
Resultado esperado: la aplicación debe alertar al usuario cuando se inserten datos incorrectos o ya exista un usuario con el mismo nombre. Cuando se inserten los datos correctamente, el sistema debe almacenar en la base de datos el nuevo registro con estado no-aprobado.	
Evaluación: Prueba Satisfactoria	

Tabla 9. Caso de Prueba 14 (Elaboración propia)

Caso de Prueba	
Número Caso de Prueba: 14	Número Historia de Usuario: 10
Nombre Caso de Prueba: Gestionar producto	

<p>Descripción: Se probará si el sistema permite modificar los datos de los productos correctamente. Se insertarán descripciones vacías o muy largas y margen de seguridad negativo.</p>
<p>Condiciones de ejecución: El usuario autenticado debe tener Rol Administrador</p>
<p>Entradas:</p> <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar campos negativos 3. Insertar los datos correctamente.
<p>Resultado esperado: la aplicación debe alertar al usuario cuando se inserten datos incorrectos. Cuando se inserten los datos correctamente, el sistema debe modificar la entidad en la base de datos de forma correcta.</p>
<p>Evaluación: Prueba Satisfactoria</p>

Tabla 10. Caso de Prueba 15 (Elaboración propia)

<p>Caso de Prueba</p>	
<p>Número Caso de Prueba: 15</p>	<p>Número Historia de Usuario: 11</p>
<p>Nombre Caso de Prueba: Gestionar Carrito de Compra</p>	
<p>Descripción: Se probará si el sistema permite agregar y bajar productos del carrito y modificar las cantidades correctamente</p>	
<p>Condiciones de ejecución: El usuario debe estar autenticado</p>	

Entradas: <ol style="list-style-type: none">1. Insertar cantidades negativas
<ol style="list-style-type: none">2. Insertar cantidades mayores al stock del producto3. Insertar cantidades correctas de acuerdo al rango de stock disponible
<p>Resultado esperado: la aplicación debe agregar correctamente los productos al carrito, en caso de agregar el mismo producto se debe añadir 1 a la cantidad en lugar de insertar una nueva entrada.</p> <p>La aplicación debe quitar correctamente productos del carrito.</p> <p>La aplicación debe verificar que la cantidad esté en los rangos especificados.</p>
Evaluación: Prueba Satisfactoria

Conclusiones del segundo capítulo

Durante el desarrollo de este capítulo se elaboró el plan principal de iteraciones, siempre teniendo en cuenta que las mismas pueden sufrir modificaciones durante el proceso de implementación. También se plasmó una selección de algunas de las historias de usuario más importantes para el desarrollo del software.

Se describieron algunos de los casos de pruebas de aceptación efectuadas.

A partir de la descripción de la solución propuesta en este capítulo se llega a un mejor entendimiento de los requerimientos funcionales para que el software cumpla de manera satisfactoria con las necesidades del cliente, siendo la comunicación con el mismo el factor más importante para lograrlo.

Siguiendo esta planificación inicial se procedió a construir el sistema que constituye la propuesta de solución. Se concretaron los patrones de diseño y arquitectura aplicados a la propuesta con el objetivo de satisfacer la necesidad por parte del cliente de un software extensible y flexible a los cambios.

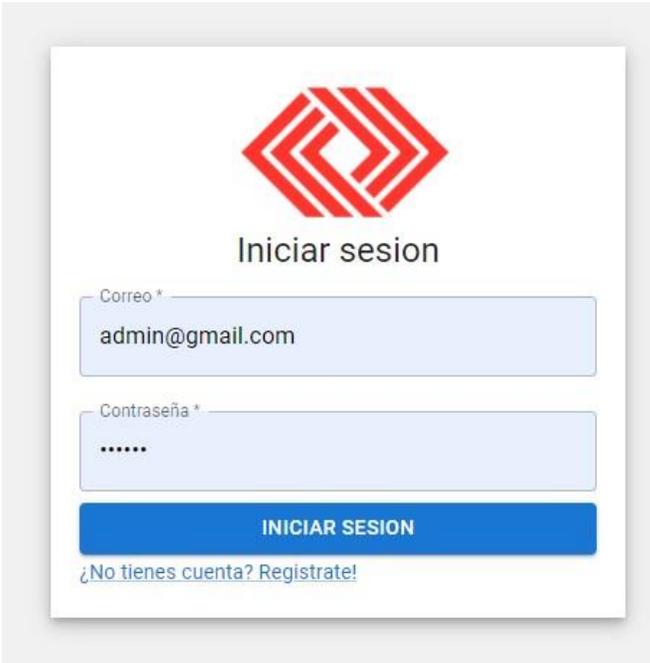
Capítulo III Análisis de los resultados

Mediante la implementación de tecnologías, herramientas y una metodología de desarrollo específica, se creó una herramienta web que cautivó a los clientes por su atractivo, comodidad y seguridad. Esta aplicación no solo fomentó la fidelidad de los usuarios, sino que los convenció de utilizar este software debido a los múltiples beneficios que les ofrece. Además, contribuye significativamente a la comercialización de productos, permitiendo a la entidad objeto de estudio realizar esta tarea de manera más eficiente y ventajosa.

Entre las ventajas y beneficios que esta solución aportó a la empresa se incluyen:

- Disponer de un espacio ilimitado para exhibir la cartera completa de productos.
- Presentar los productos de forma atractiva, organizada, confiable y de fácil acceso para los clientes.
- Liberar a los empleados involucrados en el proceso para que puedan desempeñar otras funciones.
- Optimizar la gestión y entrega de productos reservados mediante un proceso más ordenado.
- Obtener información temprana y precisa sobre el comportamiento del inventario y los ingresos gestionados a través de esta plataforma.

Vistas del software





Iniciar sesion

Correo *

admin@gmail.com

Contraseña *

INICIAR SESION

[¿No tienes cuenta? Regístrate!](#)

Imagen 4. Pantalla de autenticación

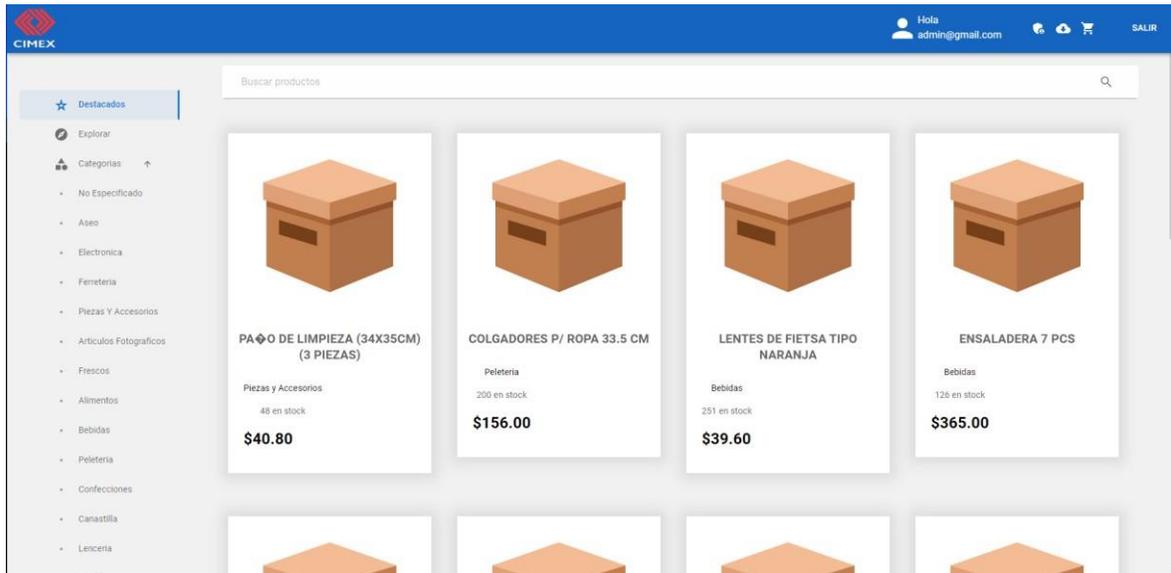


Imagen 5. Explorador de productos

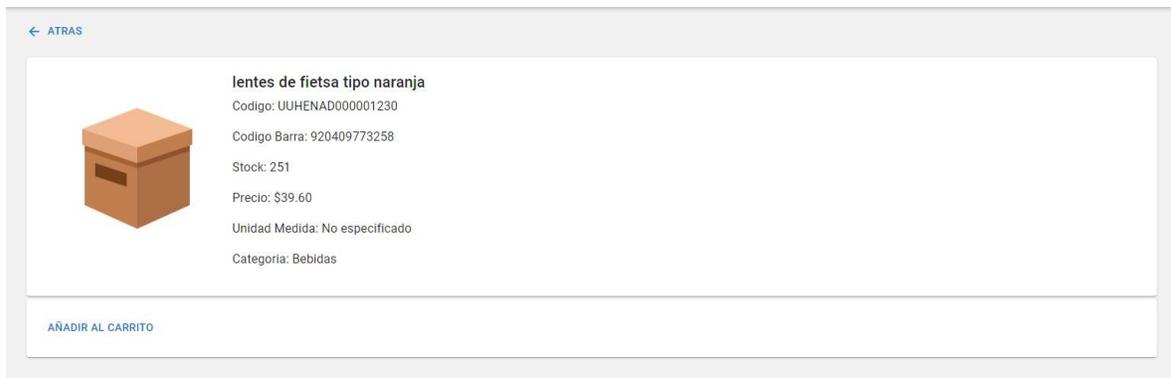


Imagen 6. Detalles de producto (Usuario)

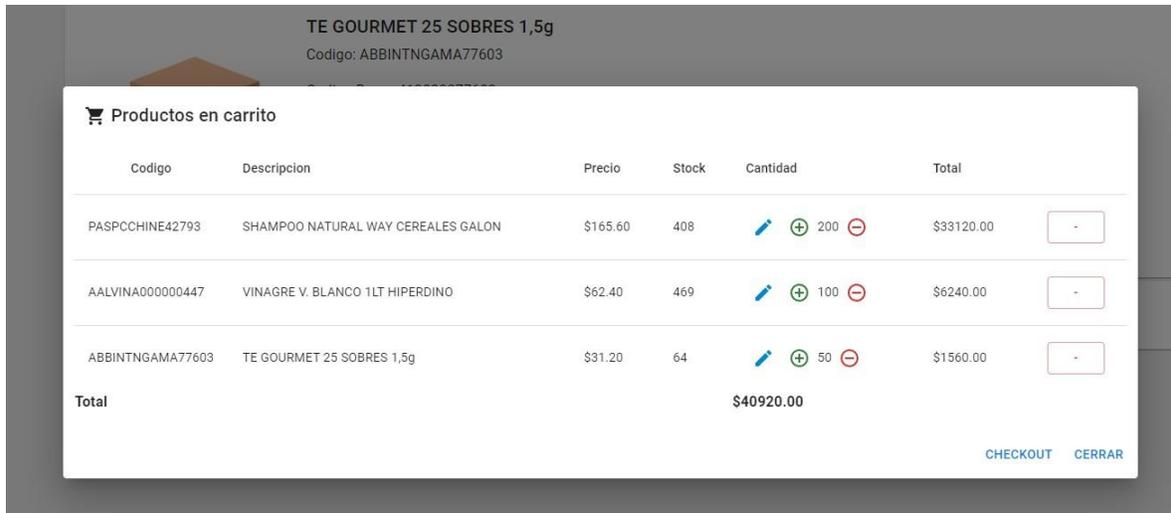


Imagen 7. Carrito de compra

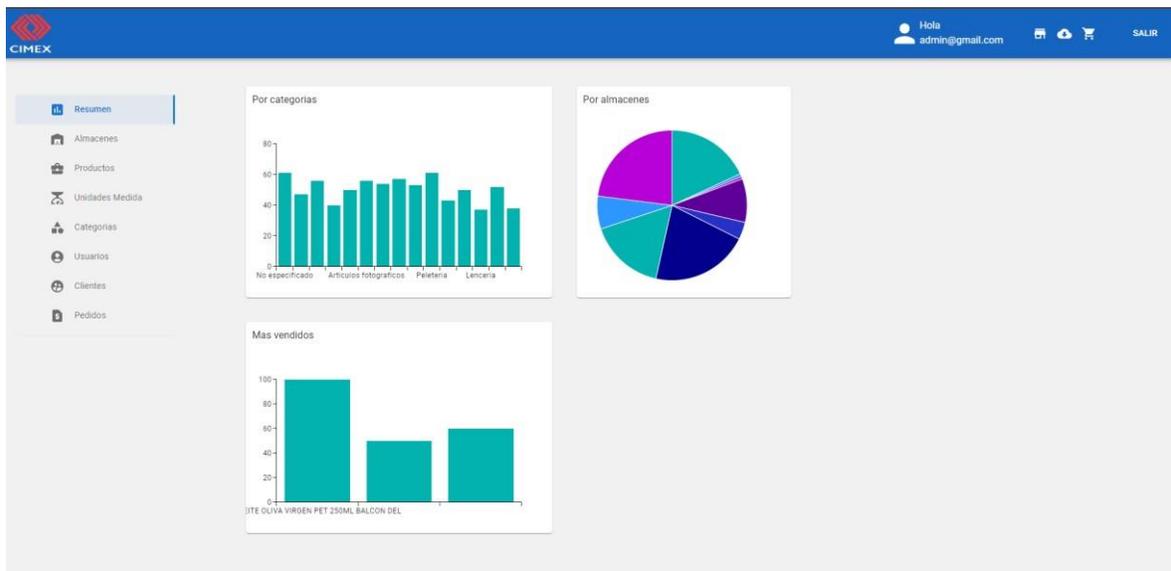


Imagen 8. Panel de administración

Buscar...

Usuarios

Usuario	CI	Cliente	Roles	Estado	
admin@gmail.com	94062906975	AGENCIA SEG. PROTECCION AVIACION (ESPACSA) CAMAGUEY	Admin User	Aprobado	 
user@gmail.com	94062906984	No asignado	User	Aprobado	 
test@gmail.com	94062906989	UEB Aeropuerto Camaguey ECASA.	Admin User	Aprobado	 
tesis@gmail.com	94062906988	Academia de Ciencias CENSA	Admin User	Aprobado	 

Imagen 9. Gestor de usuarios

[← ATRAS](#)



Descripcion	ACEITE DE OLIVA EXTRA VIRGEN(CRISTAL) RI	
Codigo	AAAACAORISTO3045	
Codigo Barra	056515243045	
Existencia total	888	
Existencia por almacen	888	
Precio Venta	330.00	
Unidad medida	Kg	
Categoria	Piezas y Accesorios	
Margen seguridad	10	
Visible	No	

ACTUALIZAR

Imagen 10. Detalles del producto (Administrador)

← ATRAS

Pedido

Codigo: PED0

Usuario: admin@gmail.com

Cliente: AGENCIA SEG. PROTECCION AVIACION (ESPACSA) CAMAGUEY

Fecha: Dec 7th 2023

Cotizacion Total: \$16244.00

Cotizacion CUP: \$2923.92 (18%)

Cotizacion MLC: \$555.00 (82%)

Productos

Codigo	Descripcion	Precio Venta	Cantidad	Importe
AAAACA0A0BE250ML	ACEITE OLIVA VIRGEN PET 250ML BALCON DEL	\$65.00	100	\$6500.00
AAAACA0VGEXTR500	ACEITE OLIVA VIRGEN EXTRA 500ML PET BALC	\$120.00	50	\$6000.00
AALVINA00000447	VINAGRE V. BLANCO 1LT HIPERDINO	\$62.40	60	\$3744.00

Imagen 11. Detalles del pedido

Conclusiones del tercer capítulo

La implementación exitosa de la herramienta web con tecnología avanzada y metodología específica ha atraído a los clientes por su atractivo, permitiendo una exhibición organizada e ilimitada de productos, liberando recursos laborales, optimizando la gestión de productos reservados y proporcionando información clave sobre inventario y rendimiento financiero. Sus ventajas notables incluyen una presentación atractiva, liberación de empleados para otras tareas, mejora en la gestión de productos y análisis detallado de datos financieros. Las vistas del sistema respaldan su diseño intuitivo, reforzando visualmente estos beneficios. En conjunto, esta herramienta se ha convertido en una inversión estratégica que impulsa la eficiencia operativa y la satisfacción del cliente de manera significativa.

Conclusiones

Como resultado de la presente investigación se obtuvieron las siguientes conclusiones:

- El análisis de los referentes teóricos sobre las tecnologías de comercio electrónico, permitió adquirir la información necesaria para dar solución a la problemática planteada.
- A partir de la puesta en práctica de la metodología de desarrollo de software XP para el desarrollo de la aplicación se facilitó una comunicación constante y directa con el cliente, lo que permitió una respuesta oportuna ante los cambios, sin importar la etapa del ciclo de vida del proyecto.
- Se diseñó e implementó una herramienta web que potencia la efectividad de las actividades económicas mayoristas en la Sucursal CIMEX Matanzas.
- Con el desarrollo de las pruebas funcionales y los resultados obtenidos se logró verificar y demostrar el correcto funcionamiento del software, así como el debido cumplimiento de los requisitos especificados por el cliente.

Bibliografía

- ALVAREZ CASADIEGO, C. A. (2012). ESTUDIO DEL USO DE LA METODOLOGIA AGIL SCRUM EN EL DESARROLLO DE UNA APLICACION NATIVA PARA LA PLATAFORMA ANDROID EN UN ENTORNO CON REQUISITOS CAMBIANTES CASO DE APLICACIÓN APPCOMPOSITORES. UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA, OCAÑA, COLOMBIA.
- Cáceres Tello, J.). Patrones de diseño: ejemplo de aplicación en los Generative Learning Object, 13.
- Camarena Sagredo, J. G., Trueba Espinosa, A., Martínez Reyes, M., López, & García, M. d. L. (2012). Automatización de la codificación del patrón modelo vista controlador (MVC) en proyectos orientados a la Web. Ciencia Ergo Sum, 13.
- Caballero Nuñez, J. L. (2009). *Metodologías Ágiles para el desarrollo de Software*
- Challenger Pérez, I., Díaz Ricardo, Y., & Becerra García, R. A. (2014). El lenguaje de programación Python. Ciencias Holguín, vol. XX, 14.
- Denzer, P. (2002). [PostgreSQL].
- Downey, A., Elkner, J., & Meyers, C. (2002). Aprenda a Pensar Como un Programador con Python. In (pp. 312).
- Eguíluz Pérez, J. (2008). Introducción a CSS. In C. Commons (Ed.). Retrieved from <http://www.librosweb.es/css>
- Espinoza-Meza, A. (2013). MANUAL PARA ELEGIR UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE DENTRO DE UN PROYECTO INFORMÁTICO. Universidad de Piura, Universidad de Piura.
- García M., S. (2015). La guía definitiva de Django: Desarrolla aplicaciones web de forma rápida y sencilla. In. Retrieved from <http://github.com/saulgm/djangobook.com>
- Gómez Mondragón, A. (2016). Propuesta para el mejoramiento del proceso de fabricación del producto Zanjadora para Alce Hidráulico mediante el uso de la gestión de procesos de negocio (BPM). (Maestría en Ingeniería), Pontificia Universidad Javeriana. Cali., Santiago de Cali.
- González-Barahona, J. M. (2011, 2011/12). El concepto de software libre. Revista Tradumática: tecnologías de la traducción, 7.
- J., L., & G., C. (2019). Materiales del entrenamiento de programación en Python - Nivel básico. In C. R.L. (Ed.), (pp. 287).
- Lizama, O., Kindley, G., & Jeria Morales, J. I. (2016). Redes de computadores. Arquitectura Cliente - Servidor. 8.
- Marini, E. (2012). [El Modelo Cliente/Servidor].
- Cohen, M. (2013). *Una Introducción a SCRUM*.

Bibliografía

- Díaz Labrador, M., & Collazo Garcia, A. (2013). *La Programación Extrema*.
- Linares, J. C., Tovar, Á., & Salazr, J. C. (2018). *Scrum vs XP: Similarities and Differences*.
- Almseidin, M., Alrfou, K., & Alnidami, N. (2015). *A Comparative Study of Agile Methods: XP versus SCRUM*.
- Mariño, S. I., & Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. Vol. 19, 7.
<http://www.redalyc.org/articulo.oa?id=84933912009>
- Muñoz Serafín, M. (2018). Introducción al desarrollo de aplicaciones NCapas con tecnologías Microsoft. In (pp. 18).
- OTAVALO MOROCHO, M. D. (2019). IMPLEMENTACIÓN DE FLUJOGRAMAS EN ACTIVIDADES REALIZADAS EN LA FARMACIA MISHHELL MEDIANTE LA SIMBOLOGÍA BIZAGI PARA MEJORAR LOS TIEMPOS DE EJECUCIÓN. In (pp. 25). FACULTAD DE CIENCIAS EMPRESARIALES. Univertsidad Técnica de Machala.: Univertsidad Técnica de Machala.
- Stallman, R. M. (2004). Software libre para una sociedad libre. In M. Vidal (Ed.), *Free Software, Free Society* (pp. 232).
- VILCA OQUENDO, D. H. (2017). OPTIMIZACIÓN DEL PROCESO DE DESARROLLO DE PROYECTOS, BAJO EL ENFOQUE DE BUSINESS PROCESS MANAGEMENT (BPM) EN EL ÁREA DE INTEGRACION DE APLICACIONES EMPRESARIALES(EAI) EN UNA EMPRESA DE TELECOMUNICACIONES. UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR, Villa El Salvador.
- Villarreal Fuentes, C. A. (2013). ¿Qué es jQuery? *Northware Software Development*, 6.
- Zea Ordóñez, M. P., Molina Ríos, J. R., & Redrován Castillo, F. F. (2017). ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL. In S. L. Editorial Área de Innovación y Desarrollo (Ed.), (pp. 82).
- Documentacion de Microsoft. (s.f.). ASP.NET Core Overview. Recuperado de <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0>
- Freeman, A. (2017). *Pro ASP.NET Core MVC 2*. Apress.
- Galloway, J., & Hanselman, S. (2018). *ASP.NET Core in Action*. Manning Publications.
- Manual de referencia de MySQL. Recuperado de <https://dev.mysql.com/doc/>
 - DuBois, P. (2003). *MySQL*. Addison-Wesley Professional.
- Forta, B. (2012). *MySQL Crash Course*. Sams Publishing.
- Freeman, A., & Robins, A. (2017). *Pro React*. Apress.
- Documentación de React. Recuperado de <https://react.dev/learn>
- Darmawan, E. (2020). *Material-UI for ReactJS - Design Beautiful Websites*. Packt Publishing.
- Material-UI Documentation. Recuperado de <https://material-ui.com/>

Bibliografía

- Hussain, M. (2020). React Projects. Packt Publishing.
- Axios GitHub Repository. Recuperado de <https://github.com/axios/axios>

- Smith, A. (2020). Pro Entity Framework Core 3. Apress.
- JetBrains. (s.f.). Rider: The Cross-Platform .NET IDE. Recuperado de <https://www.jetbrains.com/rider/>
- CARDOZZO, D. y ACADEMY, I., 2016. Desarrollo de Software: Requisitos, Estimaciones y Análisis. 2 Edición. CreateSpace Independent Publishing Platform. ISBN 9781530088614
- COHN, M., 2004b. User stories applied: For agile software development. Addison-Wesley Professional.

- Evans, E. (2003). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Westford, Massachusetts: Addison Wesley.

- Beck, K. (1999). Extreme Programming Explained: Embrace Change. Addison-Wesley.
- Jeffries, R., Anderson, A., Hendrickson, C., y Mejia, C. (2000). Extreme Programming Installed. Addison-Wesley.

- Panadés, C., & Letelier, P. O. (2003). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.

- Peralta, A. (2003). *Metodología SCRUM*.

- Mederos Candelario, M. (2016). *Aplicación web de apoyo a la gestión de la información del Consejo Científico de la Universidad de Matanzas*.

Anexos

Anexo 1: Valores de la metodología XP

Los valores de XP

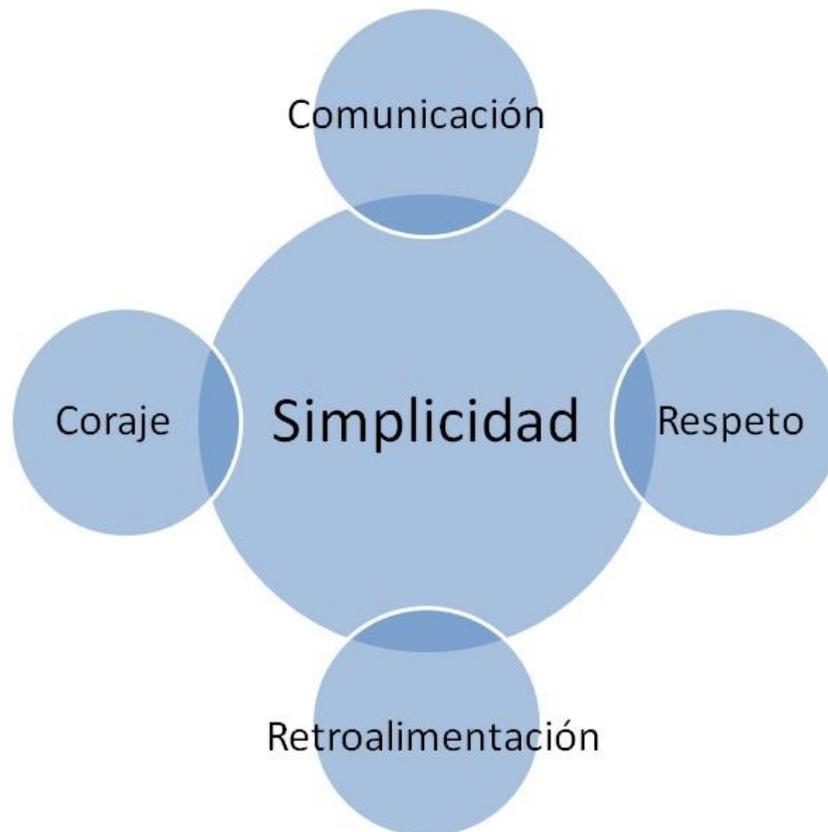


Imagen elaborada por oficinaproyectosinformatica.blogspot.com