



UNIVERSIDAD DE MATANZAS

**Facultad:** Ciencias Técnicas

**Carrera:** Ingeniería Informática

## TESIS EN OPCIÓN AL GRADO DE INGENIERO INFORMÁTICO

**Título:** Aplicación web para la venta y administración de productos de la empresa AICROS.

**Autor:** Ernesto David Escariz Ramos.

**Tutor:** Ms.C. Mayli Estopiñán Lantigua.

**-Matanzas, 2022-**

*“La innovación distingue a los líderes de los seguidores”*

*Steve Jobs*

***A mi familia, que siempre ha sido lo más importante.***

## **Agradecimientos**

- A mi madre por haberme dado la oportunidad de existir y apoyarme en todo momento.
- A mi primo, que ha sido un gran punto de apoyo y le estoy muy agradecido.
- A toda mi familia que siempre ha estado presente cuando lo he necesitado.
- A mi tutora Ms.C. Mayli Estopiñán Lantigua, por brindarme su extraordinaria, valiosa e incondicional ayuda, todo lo cual le agradezco infinitamente.
- A todo el claustro de profesores, por haber ocupado en mí, parte de ese espacio tan infinito que es el saber y que les aseguro, que continuaré enriqueciendo durante toda mi vida.
- A mis amigos, por brindarme su ayuda y especial apoyo en todo momento de mi vida y en especial en la realización de esta investigación.
- Y a todos aquellos que de una forma u otra hicieron posible la culminación de dicha investigación.

Si falta alguien por mencionar, ruego que me disculpen, porque es difícil mencionar a todos aquellos que han tenido que ver conmigo y de una forma u otra han contribuido en mi formación tanto personal como profesional.

A todos muchas gracias.

## **Declaración de Autoridad**

Yo, Ernesto David Escariz Ramos, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, especialmente a la Facultad de Ciencias Técnicas a que hagan el uso que estimen pertinente de él. Y para que así conste, firmo la presente a los 2 días del mes de diciembre del 2022.

---

**Firma de Autor**

---

**Firma de Tutor**

## Opinión del Usuario



### Empresa de Informática y Automatización para la Construcción

Calle 45, No.1107, e/ ave. 26 y Ulloa, Nuevo Vedado,  
Plaza de la Revolución, La Habana, Cuba.

Teléfonos: 7 881-2409, 7 881-1845 y 7 881-7142

Correo electrónico: aicos@aicos.cu



NC 00 9001:2015  
Registro No. 204-2018



## Breve Valoración de la Empresa de Informática para la Construcción (aicos) Grupo Centro sobre la Tienda Virtual <https://tvirtual.aicos.cu>

- Mejora la imagen de la Empresa a través del comercio online prestando un servicio expedito, de mayor calidad, confiabilidad y sencillez para el cliente.
- Facilita ampliar el alcance de los negocios logrando cubrir todo el territorio nacional, empleando la menor cantidad de recursos y portadores energéticos y reduciendo la presencia de los clientes en los procesos de contratación y compra.
- Aumenta el volumen de transacciones y consiguientemente los ingresos para nuestra Empresa.
- Logra un eficiente control de las ventas facilitando el trabajo de los órganos comercial y económico.
- Logra la contratación online Empresa-Cliente, con un nivel de validación satisfactorio de toda la documentación requerida al efecto facilitando el trabajo del órgano jurídico.

Firma la presente valoración

**Ingeniero en Sistemas Automatizados de Dirección  
Avelino Riveiro Martínez**

Fecha: 2 de diciembre de 2022

## Opinión del Tutor

## **Resumen**

Con el desarrollo de la informatización en nuestro país los hábitos de comunicación han cambiado, lo que ha propiciado una evolución en distintos sectores de la economía, principalmente en el sector comercial, donde surgen las tiendas virtuales, las cuales han tomado gran auge a partir de la pandemia del COVID-19 y brindan la posibilidad a las empresas dedicadas a la compra y venta de productos de lograr mayores nichos de mercado y eliminar complicadas barreras como la distancia. El presente trabajo se enfoca en el proceso de desarrollo de una aplicación web como plataforma para el comercio electrónico de la empresa AICROS, que ofrece a los clientes, además de productos, independencia, confort y seguridad, y una manera más eficiente de desempeñar la actividad de comercialización; a los que dirigen el proceso, tener más control sobre el estado de las ventas y los cobros, brindándole información oportuna para la toma de decisiones. Para lograr una solución acorde a los requerimientos de la empresa se utilizaron técnicas de Ingeniería de Software y se emplearon tecnologías de punta tales como Nest.js, Angular y PostgreSQL, que demostraron ser competentes para el logro del objetivo de la presente investigación.

## **Abstract**

With the development of computerization in our country, communication habits have changed, which has led to an evolution in different sectors of the economy, mainly in the commercial sector, where virtual stores arise, which have taken a great boom since the COVID-19 pandemic and provide the possibility for companies dedicated to the purchase and sale of products to achieve greater market niches and eliminate complicated barriers such as distance. The present work focuses on the process of developing a web application as a platform for electronic commerce of the company AICROS, which offers customers, in addition to products, independence, comfort and security, and a more efficient way to carry out the marketing activity; to those who direct the process, have more control over the status of sales and collections, providing timely information for decision-making. To achieve a solution according to the company's requirements, Software Engineering techniques were used and cutting-edge technologies such as Nest were used.js, Angular and PostgreSQL, which proved to be competent for the achievement of the objective of the present research.

## Tabla de contenido

Introducción.....	1
Capítulo 1. Marco Teórico Referencial .....	6
1.1 Descripción de la organización Objeto de Estudio.....	6
1.1.1 Descripción de los procesos que serán automatizados.....	6
1.1.2 Análisis crítico de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias. ....	7
1.2 Análisis comparativo de propuestas existentes con la propuesta del trabajo. ....	7
1.3 Metodología de la investigación.....	8
1.3.1 Métodos Teóricos Utilizados.....	8
1.3.2 Métodos Empíricos Utilizados .....	9
1.4 Fundamentación de la metodología utilizada.....	9
1.4.1 Metodologías ágiles de desarrollo.....	10
1.4.2 Análisis comparativo entre metodologías de desarrollo ágiles .....	18
1.4.3 Metodología de Programación Extrema (XP)....Error! Bookmark not defined.	
1.5 Herramientas y tecnologías utilizadas .....	20
1.5.1 Tendencias tecnológicas a considerar.....	20
1.5.2 Lenguajes de programación.....	22
1.6 Conclusiones parciales del capítulo .....	26
Capítulo 2. Solución Teórica al Problema Científico .....	28
2.1 Descripción de la solución .....	28
2.2 Etapa de planificación.....	28
2.2.1 Equipo de trabajo y roles .....	29
2.2.2 Historias de Usuario Iniciales .....	29
2.2.3 Plan de Iteraciones .....	31
2.2.4 Análisis del costo.....	32
2.2.5 Análisis de los beneficios .....	34
2.2.6 Plan de Entregas .....	34
2.3 Etapa de diseño .....	36
2.3.1 Prototipo de la interfaz de usuario .....	36

2.3.2	Tareas de Ingeniería .....	44
2.3.3	Modelo físico de la Base de datos .....	48
2.4	Conclusiones parciales del capítulo .....	49
<b>Capítulo 3. Validación de la solución propuesta .....</b>		<b>50</b>
3.1	Pruebas al software .....	50
3.2.1	Plan de pruebas .....	50
3.2.2	Pruebas de aceptación .....	52
3.2	Análisis de los resultados obtenidos .....	62
3.3	Conclusiones parciales del capítulo .....	62
<b>Conclusiones.....</b>		<b>64</b>
<b>Recomendaciones .....</b>		<b>65</b>
<b>Bibliografía.....</b>		<b>66</b>
<b>Anexos .....</b>		<b>70</b>

## Introducción

La consolidación de la última década de Internet en general y de la World Wide Web en particular, como tecnología del día a día, ha permitido la aparición de un nuevo entorno competitivo en el que las empresas pueden desarrollar sus procesos de negocios (Kowtha & Choon, 2001) (Laudon & Laudon, 2005), expandiéndose estos así a un nivel global como nunca antes había sido posible imaginar (Turban et al., 2008). Las ventajas que ofrece Internet han propiciado que no solo las grandes compañías, ya existentes antes de la aparición de estos nuevos avances, hayan extendido sus procesos de negocios con la ayuda de este entorno, sino que además han permitido la aparición de nuevas compañías cuyos modelos están basados única y exclusivamente en ellos. (Sabariego, 2013)

La alta competitividad en este ambiente ha llevado a las compañías que habitan en él a desarrollar un alto nivel de sofisticación e integración de técnicas computarizadas, lo que ha provocado la aparición de una nueva infraestructura tecnológica con el objetivo de dar soporte a este nuevo paradigma de negocio. Este recibe el nombre de Comercio Electrónico (e-Commerce), el cual puede ser definido como “*cualquier proceso de negocio cuyas transacciones son llevadas a cabo de forma electrónica*” (Cameron, 1997). Los procesos y tecnologías desarrolladas bajo el campo del comercio electrónico han introducido nuevas formas de llevar a cabo los procesos tradicionales de negocio.

En la última década se ha generado un importante crecimiento del comercio electrónico sustentado principalmente por el continuo aumento del número de usuarios de Internet y el crecimiento constante de usuarios que realizan compras en línea (Machín Díaz, 2019). En los últimos años el segmento de ventas online sufrió un alza debido a la pandemia mundial del COVID-19 provocando un aumento de 397 millones de dólares desde 2.038 millones en diciembre del 2019 hasta 2.495 en 2020 y la tendencia sigue en alza: se estima que en algunos años poco a poco el canal online se convertirá en el principal canal de ventas de las compañías, desplazando al tradicional canal offline (UNCTAD, 2021).

El comercio en América Latina y el Caribe ha sido tradicionalmente impulsado por un pequeño número de empresas. Según las Encuestas de Empresas del Banco Mundial, solo alrededor del 13 por ciento de las empresas de la región exportan, y las que lo hacen, exportan a solo 2 o 3 mercados. Otras estimaciones sugieren que el porcentaje de los exportadores de todas las empresas de América Latina y el Caribe es apenas el 1% (Suominen, 2019). Entre 2017 y 2021 se vivió la mayor expansión en la región, con un

incremento promedio anual de 22%, según el informe, el cual, seguirá en crecimiento con un aumento promedio anual de 19% hasta 2025, de acuerdo con un informe de la plataforma de pagos digitales Worldpay (Cimoli et al., 2022).

Cuba marcha cautelosa en la implementación del comercio electrónico que lo ha desarrollado empíricamente basado sobre todo en la calidad y cantidad de profesionales con que cuenta, siendo el capital intelectual el recurso más valioso para la incursión en este negocio (Rendón Matienzo, 2021).

La infraestructura tecnológica en el ámbito empresarial cubano tuvo un crecimiento positivo en los últimos años, siempre en los marcos de la soberanía tecnológica. Se ha logrado una accesibilidad a la red de redes considerable, aunque no ha sido así en el sector del cliente minorista, el cual aún no cuenta con una conectividad cómoda, privada y eficiente a internet, de ahí que los primeros pasos deben ir dirigidos a la modalidad de comercio electrónico mayorista. Es preciso empezar cuanto antes, para hacerse de un lugar en el mercado y esto aportará una experiencia valiosa cuando se sume al proceso el cliente minorista, último eslabón de la cadena de distribución y razón de ser de todo comercio.(Díaz-Canel Bermúdez, 2015)

Seguir el desarrollo del comercio electrónico en el país, más que una estrategia, responde a una necesidad de estos tiempos, que favorece, principalmente, la dinámica de vida de los ciudadanos y, en esta ruta, empresas estatales pertenecientes al sector de las tecnologías, junto con formas de gestión no estatal, trabajan en conjunto para impulsar este reto (Antón Rodríguez, 2022a).

Existe un número considerable de empresas en el país que no han percibido aún los beneficios de muchas de estas tecnologías y desean hacerlo. Conocen por experiencias ajenas las oportunidades que le brindan para mejorar su eficiencia y obtener sustanciosos beneficios económicos. La Empresa de Informática y Automatización para la Construcción (AICROS) perteneciente al Grupo Empresarial de Diseño e Ingeniería de la Construcción del Ministerio de la Construcción es la encargada de elaborar y comercializar sistemas y aplicaciones informáticas y soluciones integrales de informatización y automatización industrial, además de comercializar medios técnicos de computación y periféricos especializados, sus partes, piezas y accesorios, ofrece servicios telemáticos, utilizando la infraestructura de los suministradores públicos autorizados.

La entidad, dispone de espacios muy limitados para exponer, promover y comercializar su cartera de productos, y al mismo tiempo tiene una fuerza laboral muy reducida para atender el servicio de comercialización, lo cual propicia demora en la atención a los clientes. Los usuarios que acuden a las unidades comercializadoras, se encuentran afectados por esa realidad y también por no contar de manera oportuna e inmediata con toda la información acerca de los productos que se proponen adquirir. Tal situación trae consigo que en no pocas ocasiones desistan de la compra que iban a realizar. Todo ello repercute desfavorablemente en la empresa ante la pérdida considerable de ventas e ingresos. Además, se hace engorroso la gestión de la comercialización al no tener informatizado los procesos de facturación y cuentas por cobrar, lo que dificulta el control de las facturas que no han sido pagadas generando cuentas por cobrar y el monto de estas, impidiendo la toma de decisiones oportunas sobre esta temática.

A partir de esta **situación problemática** se define como **problema de la investigación**: ¿Cómo contribuir a la eficiencia del proceso de ventas y administración de los productos de la empresa AICROS?, teniendo como **objeto de estudio** el proceso de ventas y administración de los productos de la empresa AICROS y siendo el **campo de acción** la informatización del proceso de ventas y administración de productos en la empresa AICROS.

Del problema anterior se deduce la siguiente **hipótesis**: Si se informatiza el proceso de ventas y administración de productos en la empresa AICROS se contribuirá a la eficiencia en las ventas e ingresos de dicha empresa.

Con el propósito de validar la hipótesis anterior se define como **objetivo general**: desarrollar una aplicación web como plataforma para el comercio electrónico, que contribuya a la eficiencia del proceso de ventas y administración de productos de la empresa AICROS.

Para lograr el cumplimiento del objetivo general se declaran los siguientes **objetivos específicos**:

1. Revisar los referentes teóricos y metodológicos del proceso objeto de estudio.
2. Seleccionar las herramientas y metodologías acordes para el diseño y desarrollo del software.

3. Definir los requisitos del sistema, así como diseñar el modelo de datos e interfaz de usuario que sirvan como punto de partida en la adecuada determinación de las herramientas y metodología de desarrollo a utilizar.
4. Desarrollar la aplicación web que contribuya a los procesos de venta y administración de los productos de la empresa Aicros.
5. Validar el funcionamiento correcto de la aplicación web desarrollada.

Para el desarrollo de la investigación se emplean los siguientes **métodos y técnicas**:

- 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.

En correspondencia con lo tratado anteriormente, el documento queda estructurado como sigue a continuación:

**Capítulo I: Marco teórico referencial**, se reflejan los conceptos, definiciones y fundamentos asociados al problema de estudio tratado. Es descrito y analizado el flujo actual de trabajo y los antecedentes. Son abordadas las principales características de las herramientas y tecnologías empleadas para el desarrollo del software, entre los que se encuentran el sistema de gestión de base de datos y lenguajes de programación empleados con ese fin.

**Capítulo II: Solución Teórica al Problema Científico**, se expone la propuesta de solución al problema de investigación, la organización y disposición del proyecto mediante la presentación de una planificación inicial haciendo uso del marco de trabajo de desarrollo de software XP. Es analizado y estudiado los beneficios obtenidos mediante la implementación del proyecto de software.

**Capítulo III: Validación de la solución propuesta**, se realiza el análisis de los resultados obtenidos. Son aplicadas pruebas al software para determinar la funcionalidad y calidad

del mismo con el objetivo de entregar al cliente un producto terminado que satisfaga sus requerimientos.

Finalmente, son presentadas las **Conclusiones y Recomendaciones** de la investigación para dejar el camino abierto a futuros estudios relacionados con la temática abordada.

De igual forma, quedan recogidas las **Bibliografías** utilizadas y **Anexos** que fueron necesarios para el desarrollo del trabajo de investigación y mejor entendimiento del mismo.

## **Capítulo 1. Marco Teórico Referencial**

En el presente capítulo se aborda los fundamentos teóricos, antecedentes y métodos que posee la investigación. Son reflejados los principales conceptos que soportan el objeto de estudio para brindar una mejor comprensión del mismo. Es analizada la metodología de desarrollo escogida para garantizar la adecuada realización de la aplicación en correspondencia con las características del proyecto a implementar. También, se hace referencia a las distintas tecnologías y herramientas seleccionadas para el desarrollo del software.

### **1.1 Descripción de la organización Objeto de Estudio**

Aicros es una empresa estatal cubana, dedicada por más de 20 años al desarrollo y comercialización de aplicaciones y soluciones integrales de informatización y automatización industrial, que incluye servicios telemáticos y de formación, especializada en el sector de la construcción. (Antón Rodríguez, 2022b)

La cartera de servicios contiene:

- Desarrollo y mantenimiento de aplicaciones informáticas y sus servicios de postventas asociados.
- Automatización y sostenibilidad a sistemas de control de procesos industriales.
- Soluciones integrales para centros de datos y servicios de redes telemáticas.
- Consultoría y capacitación en las materias de informática, automática y telemática.

#### **1.1.1 Descripción de los procesos que serán automatizados**

Actualmente para poder comprar un producto a la empresa Aicros, el comprador debe asistir de forma presencial a uno de los establecimientos de la empresa, en el cual el comercial le informa los productos disponibles, de los cuales selecciona los que desea y se le emite una prefactura. Cualquier modificación, cancelación o nueva copia que el cliente desee hacer a la prefactura debe realizarse en la empresa con el comercial, por lo que se debe concertar una nueva reunión. Para poder realizar la compra el cliente debe tener un contrato vigente con la empresa, el cual se gestiona de forma manual y presencial.

Si el cliente desea saber cuáles son los productos disponibles, o de estos conocer alguna de sus especificaciones, debe realizar una llamada a la sede de la empresa o presentarse en la misma para obtener la información. En el caso que la empresa incorpore un software

nuevo a su lista de productos no tiene forma efectiva para hacérselo saber a todos los clientes, tendría que informárselo de manera individual (Ver diagrama de flujo del proceso en el anexo 1).

Por otra parte, los directivos no poseen una herramienta que les permita tomar decisiones y llegar a conclusiones de manera rápida y eficaz, pues la información se almacena en softwares de terceros que no satisfacen sus necesidades, ya que la información no llega de primera mano y no cuentan con funcionalidades necesarias para el modelo de negocio de la empresa AICROS, imposibilitando la obtención de datos para la toma de decisiones oportunas. Haciendo que tanto el proceso de venta como de toma de decisiones sea poco eficaz y engorroso.

### **1.1.2 Análisis de cómo se ejecutan actualmente esos procesos, las causas que originan la situación problemática y las consecuencias.**

Al realizarse la exposición, comunicación, oferta y reservación de los productos que dispone la entidad para comercializar, de manera tradicional, se somete a los clientes a largas esperas para ser atendidos provocándoles incomodidades por tal motivo y por el detrimento de valiosa parte de su jornada laboral, se les dificulta la obtención de información inmediata y necesaria para obtener seguridad en la selección correcta de la mercancía a adquirir, los trabajadores que interactúan directamente con los usuarios se encuentran saturados en esa labor conduciéndolos a ofrecer un servicio no óptimo y de cuestionable calidad.

### **1.2 Antecedentes del trabajo.**

Como antecedentes se han tenido en cuenta soluciones informáticas existentes vinculadas con el objeto de estudio de la presente, que satisfacen las necesidades identificadas en la actividad de comercialización que desempeña la entidad objeto de estudio.

Las plataformas existentes que se analizaron fueron (ver anexo 2):

- **Tienda virtual GEDEME:** Esta plataforma de comercio electrónico es utilizada por la Empresa Industrial para la Informática, las Comunicaciones y la Electrónica (Gedeme) para la promoción y venta de sus productos. En entrevista realizada a trabajadores de la empresa, estos explicaron que este sistema no cuenta con presencia en otras provincias del país y que los términos de garantía estarán descritos en los detalles de

cada mercancía, si cuenta con ella. Entre los artículos promocionados por Gedeme se encuentran teléfonos en moneda nacional, luminarias, así como productos de informática y telecomunicaciones.

- **MallCubano:** es una plataforma de comercialización online perteneciente a la Agencia Exportadora SoyCubano de ARTex.S.A., que garantiza todas las etapas del comercio electrónico, desde la gestión del producto o servicio con los proveedores hasta la entrega puerta a puerta del producto y servicio posventa. Esta tienda virtual online está segmentada por departamentos de productos y servicios culturales genuinamente cubanos tales como: literatura, música, arte en casa, artesanía, confecciones y otros que se ofertan bajo las monedas: USD (dólar americano) y CAD (dólar canadiense).
- **Superfácil:** es un centro comercial digital a través del cual el cliente, mediante Comercio Electrónico Nacional, podrá adquirir desde cualquier punto de conexión todas las ofertas que posee esta empresa cubana. Para eso solo necesitará una tarjeta del Banco Metropolitano, Banco Popular de Ahorro, Banco de crédito y comercio más la conocida de Telebanca.

Aunque en la búsqueda se hallaron diferentes aplicaciones web que tratan el tema del comercio electrónico, se desecharon como alternativa de solución al problema presentado ya que ninguna se ajusta a las características y funcionamiento que posee la Empresa AICROS en el desempeño del negocio. Por ese motivo, se decide desarrollar un software personalizado que se ajuste a las condiciones de trabajo presentes en la manera particular que comercializa la entidad su cartera de productos.

### **1.3 Metodología de la investigación**

#### **1.3.1 Métodos Teóricos Utilizados**

**Análisis – Síntesis:** Este método se evidenció en el análisis de los elementos de la situación problemática, posibilitando descomponer mentalmente un todo complejo en sus partes y cualidades, para luego sobre la base de los resultados alcanzados previamente por el análisis establecer mentalmente la unión entre las partes, permitiendo descubrir relaciones y características generales entre los elementos de la realidad.

**Histórico – Lógico:** Permitió estudiar la trayectoria, evolución y desarrollo de los sistemas de gestión de compras, directamente los dirigidos al comercio electrónico.

**Inductivo – Deductivo:** Su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

### **1.3.2 Métodos Empíricos Utilizados**

**Entrevista:** Este método fue esencial para el desarrollo de la investigación ya que permitió conocer las necesidades reales del cliente y los beneficios que podría aportar la aplicación.

**Análisis documental:** El análisis de documentos incluye la revisión y estudio del procedimiento de compras del sistema de gestión de la calidad, el manual de contabilidad para saber las formas de pago aprobadas por la organización y los tiempos de cobro y pago, se revisaron el expediente jurídico para determinar el personal aprobado para firmar facturas y ofertas comerciales.

**Observación:** La observación fue necesaria desde el comienzo para comprender como se realizaban los procesos y detectar sus insuficiencias para posteriormente darles solución.

### **1.4 Fundamentación de la metodología utilizada**

En la década de los noventa surgieron metodologías de desarrollo de software ligeras, más conocidas como metodologías ágiles, que buscaban reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Estas metodologías nacieron como reacción a las metodologías existentes con el propósito de disminuir la burocracia que implica la aplicación de las metodologías tradicionales en los proyectos de pequeña y mediana escala. Las metodologías tradicionales buscan imponer disciplina al proceso de desarrollo de software y de esa forma volverlo predecible y eficiente. Para conseguirlo se soportan en un proceso detallado con énfasis en planeación propio de otras ingenierías. El principal problema de este enfoque es que hay muchas actividades que hacer para seguir la metodología y esto retrasa la etapa de desarrollo (Navarro Cadavid et al., 2013).

Diferentes investigadores comparan enfoques tradicionales y ágiles en sus diferentes perspectivas, en el Anexo 3 se describen las diferencias entre las metodologías ágiles de desarrollo y las metodologías tradicionales (Letelier Torres et al., 2003; Matharu et al., 2015; Moniruzzaman & Hossain, 2013; Overhage & Schlauderer, 2012; Stoica et al., 2013).

### 1.4.1 Metodologías ágiles de desarrollo

El enfoque de las metodologías ágiles está teniendo una amplia efectividad en proyectos donde los requisitos son muy cambiantes, ya que en este tipo de proyectos la comunicación con el cliente debe ser fundamental y precisamente ese es uno de los principios básicos de las metodologías ágiles.

En marzo de 2001, 17 críticos de los modelos de producción basados en procesos resumieron en cuatro postulados lo que hoy en día se conoce como el Manifiesto Ágil (Menzinsky et al., 2019). Los cuatro postulados mencionados en aquella reunión mencionaban los siguientes criterios: Individuos e interacciones sobre procesos y herramientas, Software que funciona sobre documentación exhaustiva, Colaboración de clientes sobre la negociación del contrato y Respuestas a cambios sobre seguir un plan. Estos cuatro postulados son considerados como la definición canónica del desarrollo ágil.

En la actualidad existen un sin número de metodologías ágiles, unas más populares que otras, cada una aportando al desarrollo ágil distintos métodos que ayudan a mejorar de una manera eficaz la calidad del software. Entre las metodologías ágiles más populares se encuentran las siguientes:

#### 1.4.1.1 Programación Extrema (XP)

La Programación Extrema o XP (Extreme Programming) es un enfoque de desarrollo de sistemas que acepta lo que se conoce como buenas prácticas en esta área y las lleva al extremo (Rosado Gómez et al., 2012; Suardiyana Putra et al., 2012). En esta metodología cabe resaltar la importancia del cliente, las pruebas, la refactorización, la simplicidad, la propiedad colectiva del código que se ven reflejadas en las cuatro prácticas esenciales de XP (Pressman, 2010; Yong & Zhou, 2009):

- Entregas limitadas o pequeñas: Consiste en realizar entregas parciales de módulos del sistema.
- Semana de trabajo de 40 horas: Los equipos de desarrollo de XP trabajan de manera intensa durante una semana típica de 40 horas.
- Cliente en el sitio: Esta práctica insiste en que el cliente debe hacer parte fundamental y activa del grupo de trabajo y debe estar presente durante todo el proceso de desarrollo.

- Programación en Pareja: Con esto se busca aumentar la calidad del código, ahorrar tiempo, estimula la creatividad y la reducción de código fuente.

La programación extrema engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. En la Figura 1.2 se muestra cada una de las actividades de XP, y resalta las tareas claves de cada una (Pressman, 2010; Rosado Gómez et al., 2012; Sommerville, 2011; Xu, 2009; Yong & Zhou, 2009):



**Figura 1.2** El proceso de la programación extrema. **Fuente:** Pressman (2010)

**Planeación:** Esta actividad comienza escuchando a los clientes, para entender el contexto del negocio y definir las características principales y funcionalidad que se requiere, estas características se transforman en requerimientos del negocio que se especifican mediante Historias de Usuario; las cuales recogen la interacción hablada entre desarrolladores y usuarios. Una vez hechas las Historias de Usuario, el equipo de desarrollo las divide en tareas, estima el esfuerzo, recursos requeridos para su implementación, se genera el plan de entregas, las iteraciones, la rotación de parejas y las reuniones diarias.

**Diseño:** Es la etapa en donde son evaluadas las historias de usuario por el equipo del proyecto para dividir las en tareas, cada tarea representa una característica distinta del sistema y se puede diseñar una prueba de unidad que verifique cada tarea, estas tareas se representan por medio de las tarjetas CRC (Clase-Responsabilidad-Colaborador). Las tarjetas CRC identifican y organizan las clases bajo el paradigma orientado a objetos (lo que incluye asignación de responsabilidades), cada tarjeta contiene el nombre de la clase (que representa una o más historias de usuario), una descripción de las responsabilidades o métodos asociados con la clase, así como la lista de las clases con que se relaciona o

que colaboran con ella. Las tarjetas CRC son el único trabajo de diseño que se genera como parte del proceso de XP.

**Codificación:** Se lleva a cabo la programación en pareja, la unidad de pruebas y la integración del código. Durante esta etapa se espera la disponibilidad del cliente para que éste pueda resolver cualquier duda que se presente durante una jornada de trabajo.

**Prueba:** Cada tarea que se identificó con las historias de usuario, representa una característica distinta del sistema y se realiza una prueba de unidad por cada una de ellas, existen pruebas unitarias las cuales son diseñadas para probar cada uno de los métodos y clases, dichas pruebas son realizadas por los programadores.

#### 1.4.1.2 Etapas de la Metodología XP

Esta metodología consta de cuatro fases las cuales se muestran a continuación (Beck, 1999; Beck & Andres, 2004) (ver anexo 5):

##### **Fase I - Planificación del proyecto.**

En esta fase se obtienen como resultados las Historias de usuario, el plan de entrega y la las iteraciones.

##### **Fase II - Diseño**

En esta fase se tienen en cuenta el diseño simple y los riesgos, además se desarrolla el Glosarios de Términos.

##### **Fase III – Codificación**

El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

##### **Fase IV - Pruebas**

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. El uso de los test en XP es el siguiente:

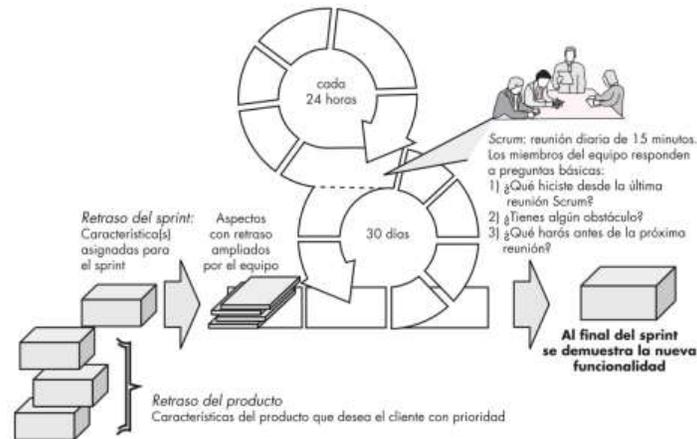
- Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.
- Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.
- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican. Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos.

#### 1.4.1.3 Scrum

Su nombre no corresponde a una sigla, sino a un concepto deportivo, propio del rugby, relacionado con la formación requerida para la recuperación rápida del juego ante una infracción menor (Faniran et al., 2017; Navarro Cadavid et al., 2013). Scrum es un modelo de desarrollo ágil caracterizado por (Kaur et al., 2015; Menzinsky et al., 2019; Romano & da Silva, 2015):

- Adoptar una estrategia de desarrollo incremental, en lugar de la planificación y ejecución completa del producto.
- Basar la calidad del resultado más en el conocimiento tácito de las personas en equipos autoorganizados, que en la calidad de los procesos empleados.
- Solapamiento de las diferentes fases del desarrollo, en lugar de realizarlas una tras otra.

Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega (Pressman, 2010; Srivastava et al., 2017). El flujo general del proceso Scrum se ilustra en la Figura 1.3:



**Figura 1.3** Flujo del proceso Scrum. **Fuente:** Pressman (2010)

**Retraso:** lista de prioridades de los requerimientos o características del proyecto que dan al cliente un valor del negocio.

**Sprints:** consiste en unidades de trabajo que se necesitan para alcanzar un requerimiento definido en el retraso que debe ajustarse en una caja de tiempo predefinida (lo común son 30 días).

**Reuniones Scrum:** son reuniones breves (de 15 minutos, por lo general) que el equipo Scrum efectúa a diario.

**Demostraciones preliminares:** entregar el incremento de software al cliente de modo que la funcionalidad que se haya implementado pueda demostrarse al cliente y éste pueda evaluarla.

Las características del desarrollo basado en Scrum son (Ahmed et al., 2010; Matharu et al., 2015):

**Colaboración:** El desarrollo basado en Scrum promueve la colaboración ya que está impulsado por equipos multifuncionales donde cada persona con sus habilidades y experiencia contribuye a la mejor solución de diseño. Un equipo multifuncional incluye una combinación de programadores, arquitectos de software, analistas de software y expertos en control de calidad.

**Reuniones diarias:** La metodología Scrum está marcada por reuniones diarias de corta duración en las que el equipo de desarrollo de productos se comunica y evalúa el estado de progreso del desarrollo de software, lo que aumenta la productividad de los miembros del equipo.

**Product Backlog:** El product backlog captura los requisitos para que un producto de software se entregue con éxito. Mantiene una lista ordenada de características, correcciones de errores, requisitos no funcionales.

**Sprint Backlog:** El sprint backlog registra la lista de tareas que realizará el equipo de desarrollo durante el próximo sprint. Esta lista se elabora recogiendo las tareas desde la parte superior de la cartera de productos hasta que se realice el trabajo suficiente para el próximo sprint, teniendo en cuenta la capacidad de trabajo y los resultados pasados del equipo de desarrollo.

**Roles:** El desarrollo basado en Scrum se rige por 3 funciones principales (Faniran et al., 2017; Mahalakshmi & Sundararajan, 2015):

- Product Owner: responsable de definir, priorizar y comunicar los requisitos del producto y guía el proceso de desarrollo del producto.
- Equipo de desarrollo: responsable de ejecutar las tareas asignadas por el propietario del producto dentro del plazo del sprint. Por lo general, un equipo multifuncional de 3 a 9 individuos implementa las tareas de desarrollo del producto previstas por el propietario del producto.
- Scrum Master: responsable de hacer cumplir las reglas y los principios del desarrollo basado en Scrum. El Scrum Master elimina los impedimentos para el desarrollo y ayuda a mejorar el proceso, el equipo de desarrollo y el producto de software que se está desarrollando.

#### 1.4.1.4 Desarrollo adaptativo de software (ASD)

El desarrollo adaptativo de software (ASD) es una técnica para elaborar software y sistemas complejos. Los fundamentos filosóficos del ASD se centran en la colaboración humana y en la organización propia del equipo (Abdelaziz et al., 2015; García Rodríguez, 2015; Pressman, 2010). La Figura 1.4 ilustra el ciclo de vida del ASD:



**Figura 1.4** Desarrollo adaptativo de software. **Fuente:** Pressman (2010)

El “ciclo de vida” del ASD incorpora tres fases: especulación, colaboración y aprendizaje (Chowdhury & Huda, 2011; García Rodríguez, 2015; Pressman, 2010).

En la especulación, se inicia el proyecto y se lleva a cabo la planeación adaptativa del ciclo. La especulación emplea la información de inicio del proyecto para definir el conjunto de ciclos de entrega (incrementos de software) que se requerirán para el proyecto.

Las personas motivadas usan la colaboración de manera que multiplica su talento y producción creativa más allá de sus números absolutos. Este enfoque es un tema recurrente en todos los métodos ágiles. Sin embargo, la colaboración no es fácil. Incluye la comunicación y el trabajo en equipo, pero también resalta el individualismo porque la creatividad individual desempeña un papel importante en el pensamiento colaborativo.

Conforme los miembros de un equipo ASD comienzan a desarrollar los componentes que forman parte de un ciclo adaptativo, el énfasis se traslada al aprendizaje de todo lo que hay en el avance hacia la terminación del ciclo.

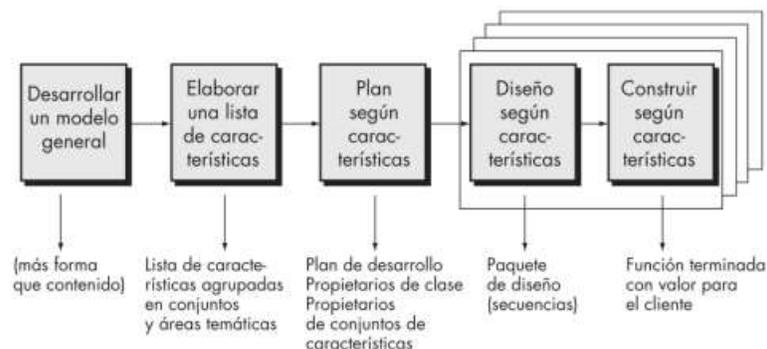
#### 1.4.1.5 Feature Drive Development (FDD)

El desarrollo impulsado por las características es una metodología ágil para el desarrollo de sistemas, basado en la calidad del software, que incluye un monitoreo constante del proyecto. Al igual que otros proyectos ágiles, adopta una filosofía que (Chowdhury & Huda, 2011; Doshi & Patil, 2016; Pressman, 2010):

- pone el énfasis en la colaboración entre los integrantes del equipo;

- administra la complejidad de los problemas y del proyecto con el uso de la descomposición basada en las características, seguida de la integración de incrementos de software, y
- comunica los detalles técnicos en forma verbal, gráfica y con medios basados en texto.

El desarrollo impulsado por las características se divide en 5 fases y es de tipo incremental, constando cada incremento (iteración) en 2 fases: diseño y construcción de una característica. En la Figura 1.5 se pueden distinguir estas fases:



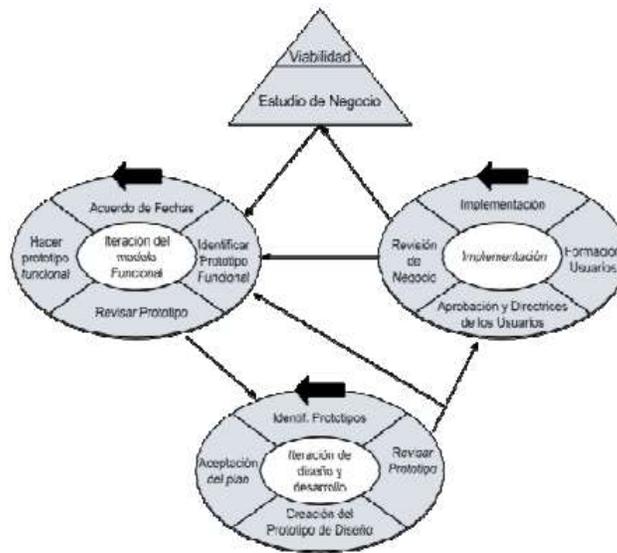
**Figura 1.4** Desarrollo impulsado por las características (Feature Drive Development (FDD)).

**Fuente:** Pressman (2010), García Rodríguez (2015)

#### 1.4.1.6 Dynamic Systems Development Method (DSDM)

El método de desarrollo de sistemas dinámicos (DSDM) es un enfoque de desarrollo ágil de software que “proporciona una estructura para construir y dar mantenimiento a sistemas que cumplan restricciones apretadas de tiempo mediante la realización de prototipos incrementales en un ambiente controlado de proyectos” (Menzinsky et al., 2019; Pressman, 2010). El DSDM es un proceso iterativo de software en el que cada iteración sigue la regla de 80 por ciento. Es decir, se requiere sólo suficiente trabajo para cada incremento con objeto de facilitar el paso al siguiente. Los detalles restantes se determinan más tarde, cuando se conocen los requerimientos del negocio y se han pedido y efectuado cambios (Mead et al., 2008; Moniruzzaman & Hossain, 2013; Pressman, 2010).

El ciclo de vida DSDM, define tres ciclos iterativos distintos, precedidos de dos actividades adicionales al ciclo de vida (figura 1.5).



**Figura 1.5** Ciclo de vida DSDM. **Fuente:** Jabeen et al. (2014)

### 1.4.2 Análisis comparativo entre metodologías de desarrollo ágiles

Aunque existen varias metodologías de desarrollo ágiles disponibles, cada una se aplica a un conjunto específico de proyectos. Un proyecto de desarrollo de software tiene varios factores asociados, como el tamaño del proyecto, la complejidad, el tiempo asignado, el presupuesto, etc. La selección de la metodología adecuada para el desarrollo de software depende de tales factores. Por lo tanto, un análisis comparativo de metodologías ágiles ayudará a decidir cuál se puede utilizar en una situación dada (Kirmani, 2017; Priyanka & Kantha, 2016).

#### 1.4.2.1 Documentación

Uno de los principios del desarrollo ágil es reducir la cantidad de tiempo y esfuerzo dedicado a la documentación. Pero la documentación que es importante no puede ser eliminada por completo (Almseidin et al., 2015; Pawar, 2015):

- En metodologías como Scrum, XP y ASD la documentación es de menor importancia.
- En proyectos con FDD se requiere más documentación.
- DSDM requiere un nivel moderado de documentación que aún es menor que FDD.

#### 1.4.2.2 Interacción con el cliente

Las metodologías ágiles otorgan una importancia primordial a la comunicación frecuente con los usuarios finales. Aun así, el grado de implicación es diferente en cada metodología (Carvalho et al., 2011):

- XP y Scrum tienen una alta participación del cliente en el proceso de desarrollo.
- En ASD y DSDM, la participación del cliente o usuario final se puede ver durante el inicio y el final de la iteración.
- Mientras que FDD utiliza informes para comunicarse con los clientes.

#### 1.4.2.3 Reuniones

La comunicación es uno de los principios básicos establecidos en el manifiesto ágil. El éxito de las metodologías ágiles depende de la comunicación efectiva entre los miembros del equipo (Moniruzzaman & Hossain, 2013):

- Las reuniones son de carácter informal y no se conserva documentación.
- Debido al uso de la técnica de programación en pares, el éxito de XP depende en gran medida de la comunicación.
- FDD y DSDM se basan en informes y documentación para la comunicación.
- Las reuniones cara a cara se utilizan para la comunicación en ASD.

#### 1.4.2.4 Tamaño y complejidad del proyecto

- Cada metodología es adecuada para un tipo particular de proyecto:
- XP y ASD son usualmente preferidos para proyectos pequeños y menos complejos.
- XP es adecuado para proyectos donde hay un cambio constante en la especificación del producto.
- Scrum, FDD y DSDM se puede aplicar a cualquier tamaño de proyecto.

En el anexo 4 se comparan las metodologías ágiles revisadas anteriormente con respecto a diferentes parámetros (Moniruzzaman & Hossain, 2013).

Para el desarrollo de la aplicación objeto de estudio se utiliza la metodología XP debido a que se utiliza en proyectos pequeños o medianos, y permite actualizarse permanentemente; además, es muy eficiente durante el proceso de pruebas y planificación, su tasa de error es muy pequeña, facilita los cambios, origina una programación muy organizada y la satisfacción del programador, además de fomentar la comunicación entre los desarrolladores y los clientes; también, se puede aplicar a cualquier lenguaje de

programación, el cliente tiene el control sobre las prioridades; durante todo el proyecto se pueden realizar diversas pruebas; y, sobre todo, permite ahorrar mucho tiempo y dinero.

## **1.5 Herramientas y tecnologías utilizadas**

### **1.5.1 Tendencias tecnológicas a considerar**

#### **1.5.1.1 Arquitectura Cliente Servidor:**

Es una arquitectura de procesamiento cooperativo de información donde uno de los componentes pide servicios a otro. Es una relación entre procesos corriendo en máquinas separadas. El cliente es un consumidor de servicios y el servidor es un proveedor de servicios e interactúan por un mecanismo de pasaje de datos: pedido de servicios – respuesta.

Cliente: Una estación de trabajo o microcomputador (PC: Computadora Personal) conectado a una red que le permite acceder y gestionar una serie de recursos. En este caso se refiere a un microcomputador conectado al sistema de información y en el que se realiza una parte mayoritaria de los procesos.

Servidor o Back-end: Está dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANs o WANs y suministran servicios tales como acceso a base de datos, archivos, comunicaciones, impresión, procesamiento de imágenes, correo entre otros. Este mecanismo favorece flexibilidad y dinamismo en las organizaciones (Varela García, 2018).

La principal característica de esta arquitectura es que permite conectar a varios clientes a los servicios que provee un servidor, permitiendo que la aplicación sea consumida por distintos usuarios de forma simultánea, requisito indispensable para las aplicaciones web, principalmente las tiendas virtuales.

#### **1.5.1.2 Patrón de diseño Modelo Vista Controlador**

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, entre otras). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean

independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación (Guerra Velázquez & de la Vega Valenzuela, 2013):

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El controlador se encarga de procesar las interacciones del usuario, manejo de las peticiones, de la seguridad, cargar la configuración de la aplicación y realiza los cambios apropiados en el modelo o en la vista.

Este patrón de diseño al separar los distintos componentes de la aplicación permite un más fácil mantenimiento y una mejor escalabilidad de la aplicación, aspectos que se quieren lograr en el presente proyecto.

#### 1.5.1.3 Sistema Gestor de Base de Datos (SGBD)

Un Sistema Gestor de Base de Datos es un conjunto de programas no visibles que administran y gestionan la información que contiene una base de datos. A través de él se maneja todo acceso a la base de datos con el objetivo de servir de interfaz entre ésta, el usuario y las aplicaciones (Lopez Campeny & Ponce, 2022).

#### 1.5.1.4 PostgreSQL

PostgreSQL es un sistema gratuito y de código abierto de administración de bases de datos del tipo relacional, aunque también es posible ejecutar consultas que sean no relaciones. Su desarrollo es llevado adelante por una gran comunidad de colaboradores de todo el mundo que día a día ponen su granito de arena para hacer de este sistema una de las opciones más sólidas a nivel de bases de datos.

Dos detalles a destacar de PostgreSQL es que posee data types (tipos de datos) avanzados y permite ejecutar optimizaciones de rendimiento avanzadas, que son características que por lo general solo se ven en sistemas de bases de datos comerciales, como por ejemplo SQL Server de Microsoft u Oracle de la compañía homónima (Borges, 2019).

Lo anteriormente descrito junto a la petición del cliente, de que se considerase este gestor de base de datos para el proyecto, hace que sea seleccionada esta herramienta.

### 1.5.1.5 Entorno de desarrollo integrado (IDE): Visual Studio Code

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.

Para tener una idea de la popularidad de Visual Studio Code y la aceptación que ha tenido en el mundo de desarrollo, según una encuesta realizada por Stack Overflow a más de 80,000 desarrolladores en mayo del 2021, Visual Studio Code es el entorno de desarrollo más usado y con mucha diferencia, un 71.06% (Flores, 2022).

La gran customización y la amplia gama de funcionalidades que se le pueden añadir, permiten al desarrollador crear un entorno de trabajo agradable y útil, mejorando la velocidad de desarrollo y la comodidad del programador.

### 1.5.2 Lenguajes de programación

Entre los distintos lenguajes de programación para la Web que existen en la actualidad, se destacan dos grupos, que se diferencian entre sí por el lugar que ocupan en la arquitectura Cliente/Servidor, característica de los sistemas Web.

1. El primer grupo lo integran los lenguajes que se ejecutan en el cliente, ejemplos de ellos son: HTML, CSS, JavaScript, estos se encargan de aportar dinamismo a la aplicación en los navegadores.
2. El segundo grupo está formado por los lenguajes que se ejecutan en el servidor, dentro de los que se encuentra ASP, PHP y Java, estos lenguajes se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a bases de datos y el tratamiento de la información.

Un lenguaje del lado cliente es totalmente independiente del servidor, lo cual permite que la página pueda ser albergada en cualquier sitio. El navegador es una especie de aplicación capaz de interpretar las órdenes recibidas en forma de código HTML fundamentalmente y convertirlas en las páginas que son el resultado de dicha orden.

Los lenguajes de programación utilizados para el desarrollo de la aplicación objeto de estudio fueron dados como requisitos por el cliente.

### 1.5.2.1 Frontend

#### - HyperText Markup Language (HTML)

Es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces (hyperlinks) que conducen a otros documentos o fuentes de información relacionadas, y con inserciones multimedia (gráficos, sonido). Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. La utilización de HTML como uno de los lenguajes para el desarrollo, responde a las necesidades de manipulación y maquetación de los elementos visuales de la aplicación. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, entre otras), así como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (Gauchat, 2012).

Este lenguaje es parte fundamental de las normas de la web, ya que es la base para construir páginas en internet, y el uso de sus etiquetas semánticas ayudan a definir la estructura del documento y permiten que las páginas web sean mejor indexadas por los navegadores.

#### - Cascade Style Sheets (CSS)

Hojas de estilo en cascada (o CSS, siglas en inglés de Cascading Stylesheets) no es como tal un lenguaje de programación, sino un lenguaje de estilos, con el cual se define la apariencia de una página web definiendo como debe mostrarse cada uno de sus elementos HTML. CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento (Gauchat, 2012).

Del mismo modo que HTML se utiliza para estructurar el contenido de un sitio, CSS se usa para estructurar la presentación del mismo, convirtiéndolo en uno de los lenguajes más importantes que se utilizan para ordenar las instrucciones referentes a la apariencia de un sitio y presentar los contenidos de una página de forma atractiva.

## - TypeScript

TypeScript es un lenguaje de programación de distribución y código abierto desarrollado por Microsoft. Su propio fabricante lo define como un superset (superconjunto) de JavaScript que, en esencia, mejora JavaScript añadiéndole funciones de tipado y de programación orientada a objetos. El propósito de TypeScript es cubrir las carencias o deficiencias del lenguaje JavaScript, haciéndolo un lenguaje más completo, escalable, seguro e incluso más fácil de utilizar. Una de sus principales características frente al JavaScript es la programación orientada a objetos basado en clases, lo cual, aunque de alguna manera era posible con JavaScript, resultaba complicado ya que el lenguaje no está pensado para ello. Últimamente este lenguaje está ganando popularidad ya que el potente framework Angular está totalmente desarrollado con este lenguaje y ellos mismos recomiendan utilizar TypeScript para desarrollar en Angular (Fernández Guerra et al., 2017).

JavaScript es el lenguaje de programación que agrega elementos de interactividad, animación y movimiento de datos a las páginas web. TypeScript ofrece todas las ventajas de JavaScript incorporándole tipado estático lo cual evita errores ocultos, permite refactorizar el código sin grandes efectos secundarios y permite una mejor escalabilidad y lectura del código.

### 1.5.2.2 Backend

## - Node.js

Node.JS es un entorno en tiempo de ejecución multiplataforma que suele usarse en la capa del servidor (pero no limitada a esta) y que está basado en ECMAScript. Fue creado con el objetivo principal de poder construir plataformas web con una alta escalabilidad ya que los entornos tradicionales sufrían cuando se veían sometidos a una alta concurrencia de usuarios.

Al estar basado en ECMAScript el lenguaje en el que se codifica es casi idéntico a JavaScript ya que utiliza el motor de código V8, al igual que el navegador Google Chrome. Las principales ventajas de este entorno son las siguientes (Fernández Sáenz de Navarrete, 2019):

- Se trata de un entorno asíncrono, por lo que permite procesar una gran cantidad de solicitudes sin que su rendimiento se vea afectado. La forma en la que atiende las peticiones no se asemeja a los entornos tradicionales en los que para cada petición se asigna un hilo del SO, si no que cada conexión ejecuta un evento dentro del entorno que se traduce en un hilo del SO que se ejecuta en bucle, esto permite ahorrar mucho tiempo ya que, por la propia naturaleza del motor, este no permite bloqueos y porque no se bloquea directamente cuando realizas una llamada de E/S.
- Orientado a eventos, la arquitectura está dirigida por eventos ya que por un lado es más útil que utilizar la orientada a objetos en este tipo de entornos. Esto es debido a que el funcionamiento del servidor se basa en solicitudes que requieren una respuesta, es decir, se realiza una conexión, se reciben ciertos datos y se devuelve la respuesta, estas acciones son mucho más cercanas a eventos que a un sistema puramente interactivo. Además, esta orientación permite que las operaciones no bloqueantes informen de que han finalizado mediante señales.
- Mayor tasa de transferencia, este entorno permite unas tasas de transferencia o throughput mucho mayores a los entornos tradicionales, la desventaja en este caso es que la latencia puede verse incrementada, por lo que depende de la orientación que tenga la plataforma web. Sin embargo, cuando la concurrencia crece, esta latencia suele tener una tendencia lineal.
- Modular, al igual que la mayoría de librerías y frameworks tiene una estructura modular que permite la carga y descarga de diferentes módulos que amplían la funcionalidad base y permiten una gestión y actualización mucho más eficiente en comparación con otros entornos en los que la funcionalidad viene incluida en el core.

### 1.5.2.3 Framework de desarrollo web

#### - Nest.js

NestJS es un framework progresivo de Node.js para la creación de aplicaciones eficientes, confiables y escalables del lado del servidor, el cual está construido y es completamente compatible con TypeScript (no obstante aún nos permite la codificación en JS puro), combinando elementos de la programación orientada a objetos (POO en español; OOP, según sus siglas en inglés, Object-oriented programming), programación funcional (FP,

según sus siglas en inglés, functional programming) y programación reactiva funcional (FRP, según sus siglas en inglés, functional reactive programming).

Algunas de las propiedades que potencian NestJS y lo separan del grupo de los demás frameworks de Node.js (Murillo, 2022):

- Extensibilidad: gracias a su arquitectura modular, es flexible y nos permite utilizar las otras bibliotecas existentes en nuestro proyecto.
- Arquitectura: tiene una arquitectura de proyecto que proporciona capacidad de prueba, escalabilidad y mantenimiento sin mucho esfuerzo.
- Versatilidad: proporciona un ecosistema adaptable, el cual está desarrollado para crear todo tipo de aplicaciones del lado del servidor.
- Progresividad: hace uso de las últimas funciones de JavaScript e implementa soluciones maduras y patrones de diseño en el desarrollo de software.

En definitiva, NestJS agiliza increíblemente el trabajo que supone desarrollar aplicaciones backend, principalmente APIs. En ese sentido es un framework que facilita mucho el trabajo que supone trabajar con NodeJS puesto que soluciona sus principales carencias.

#### - Angular

Angular es una plataforma de desarrollo, construida sobre TypeScript. Es un framework basado en componentes para crear aplicaciones web escalables. Una colección de bibliotecas bien integradas que cubren una amplia variedad de características, que incluyen enrutamiento, administración de formularios, comunicación cliente-servidor y más. Un conjunto de herramientas para desarrolladores que permiten desarrollar, compilar, probar y actualizar el código fuente de la aplicación (Gonçalves, 2021).

Posee una estructura basada en componentes lo que hace que los componentes sean altamente reutilizables y simplifica el proceso de desarrollo. Esto unido a que el cliente pidió que se utilizara este framework, ya que otros proyectos de la empresa están desarrollados en el mismo, hace tenerlo en cuenta al seleccionar las herramientas de desarrollo.

### **1.6 Conclusiones parciales del capítulo**

En el desarrollo del primer capítulo se dio a conocer las bases teóricas sobre las cuales se sustenta la propuesta de trabajo. Se analiza cómo se efectúa el proceso y se corroboró la necesidad de diseñar esta aplicación para solucionar los problemas existentes. Se realiza

un estudio detallado de las tendencias existentes para el desarrollo de software y se definen las tecnologías y técnicas que se aplicarán al diseño e implementación de la aplicación informática. Se justifica la utilización de una metodología ágil para el desarrollo de la aplicación, específicamente la metodología de programación extrema XP. Se fundamenta la utilización de los lenguajes HTML, CSS y TypeScript en el lado del cliente, siendo Angular escogido como framework y el lenguaje Node.js con Nest.js como framework en el lado del servidor, además del uso de Visual Studio Code como entorno de desarrollo y el gestor de bases de datos PostgreSQL, ya que se ajustan a los requerimientos del cliente y de desarrollo del software propuesto.

## **Capítulo 2. Solución Teórica al Problema Científico**

En el capítulo se exponen los elementos necesarios para la descripción de la solución propuesta. A través de las Historias de Usuarios (HU) que acumulan la necesidad existente definida por el cliente, es llevado a cabo el análisis de los requerimientos. Se aplica la Metodología XP, Extreme Programming (Programación Extrema), metodología ágil de desarrollo, con el objetivo de garantizar el diseño de un programa lo más ajustado posible y se logra como ventaja la incorporación del cliente como un miembro del equipo de desarrollo.

### **2.1 Descripción de la solución**

Se propone desarrollar una aplicación web que permita informatizar el proceso de comercialización de la Empresa de Informática y Automatización para la Construcción, AICROS, logrando una mayor eficiencia de este proceso de ventas y permita un mayor control sobre la facturación de sus productos y servicios, proporcionando a sus directivos más información para la toma de decisiones efectiva.

Los usuarios que interactúan con la aplicación tendrán los siguientes roles:

- Administrador del sistema: es el usuario que tiene la responsabilidad de usuarios internos, asignarles permisos, configurar y auditar la aplicación.
- Cliente o Representante: es el usuario registrado que interactúa mayormente con el sistema, crea listas de deseos y reserva mercancía.
- Visitante: es aquel usuario no registrado que tiene solo acceso a la percibir la oferta de la mercancía.

Para realizar la oferta de productos disponibles y otros que han tenido saldo recientemente, se hace uso de una plantilla que los muestra con cierto orden y clasificación, y ofrece facilidades de búsqueda a partir de la aplicación de filtros.

### **2.2 Etapa de planificación**

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando "Historias de usuarios", las que sustituyen a los tradicionales "casos de uso". Una vez obtenidas las "Historias de Usuarios", los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios” (Beck & Andres, 2004).

### 2.2.1 Equipo de trabajo y roles

La metodología XP define roles de trabajo asociando a cada uno con diversas actividades. A continuación, se definen los roles, quedando designado el programador que sería el encargado de producir el código del sistema, el jefe de proyecto y el cliente que no es más que el que escribe las historias de usuario, les asigna la prioridad y diseña las pruebas funcionales para validar su implementación. A continuación, se muestra la asignación de estos roles a las personas responsables.

**Tabla 2.1** Equipo de trabajo y roles **Fuente:** Elaboración Propia.

Miembro	Roles
Mayli Estopiñán Lantigua	Jefe de proyecto, Tester
Ernesto David Escariz Ramos	Programador
Avelino Martínez	Cliente, Tester

### 2.2.2 Historias de Usuario Iniciales

Las “Historias de usuarios” sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Son utilizadas para estimar tiempos de desarrollo de la parte de la aplicación que describen.

También se utilizan en la fase de pruebas, para verificar si la aplicación cumple con lo que especifica la historia de usuario (Laínez Fuentes, 2015).

#### **Escalas equivalentes a la prioridad en el negocio:**

- Alta: Asignada a las Historias de Usuario que corresponden a funcionalidades esenciales en el desarrollo del proyecto, a las que el cliente define como primordiales.
- Media: Dada a las Historias de Usuario que resultan para el cliente como funcionalidades a tener en cuenta, sin que estas tengan una afectación directa sobre el proyecto que se esté desarrollando.
- Baja: Se le otorga a las Historias de Usuario que constituyen funcionalidades que sirven de ayuda al control de elementos asociados al equipo de desarrollo, a la estructura y no tienen nada que ver con el proyecto en desarrollo.

#### **Escala Nominal de Riesgo en Desarrollo:**

- Alta: Cuando para la implementación de la Historia de Usuario se considera la posible existencia de errores que lleven a inoperatividad del código.
- Media: Cuando pueden aparecer errores en la implementación de la Historia de Usuario que puedan retrasar la entrega de la versión.
- Baja: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

A continuación, en la tabla 2.2 se colocan los requisitos funcionales definidos.

**Tabla 2.2** Planificación de las Historias de Usuario.

**Fuente:** Elaboración Propia.

No	Nombre	Prioridad	Riesgo	Esfuerzo	Iteración	Entrega
1	Diseño y creación de la base de datos	Media	Alto	1	1	1
2	Diseño de la interfaz de usuario	Media	Medio	1		
3	Autenticación de usuario	Alta	Medio	1		
4	Vincular usuario a empresa	Alta	Alto	1		
5	Gestionar productos	Alta	Alto	2	2	2

6	Gestionar carrito	Alta	Medio	2		
7	Generar prefactura	Alta	Alto	3		
8	Gestionar perfil	Media	Medio	2	3	3
9	Gestionar usuario	Alta	Alto	2		
10	Gestionar descuento	Alta	Alto	2		
11	Gestionar centro de costo	Alta	Alto	2		
12	Gestionar favoritos	Media	Medio	1	4	4
13	Gestionar prefactura	Media	Medio	2		
14	Generar reportes	Media	Medio	1		

Una vez definidos los requisitos funcionales pues es importante elaborar el plan de entrega y de iteraciones de manera tal que se cumpla en cada entrega con los requerimientos hechos por el cliente.

### 2.2.3 Plan de Iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación.

El proyecto fue dividido en cuatro iteraciones, por lo que se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales. Para la determinación de cada una de las iteraciones se tuvo en cuenta la opinión del cliente a través de las entrevistas que se le realizaron continuamente antes de comenzar a desarrollar cada iteración donde se tomaron todos los acuerdos necesarios. Una vez concluida la iteración que va a estar conformada por un conjunto de historias de usuarios, teniendo en cuenta los requisitos exigidos por el cliente, la aplicación poseerá mayor número de funcionalidades.

En la figura 2.1 se muestran las cuatro iteraciones y las semanas en las que se desarrollan.



**Figura 2.1** Plan de Iteraciones. **Fuente:** Elaboración Propia.

Posterior al plan de iteraciones se realiza la estimación del proyecto. En este caso se utilizó puntos de función para tener una estimación temprana.

#### 2.2.4 Análisis del costo

La estimación es el proceso de medición anticipada de la duración, esfuerzos y costes necesarios para realizar todas las actividades y obtener todos los productos asociados a un proyecto. Es necesario tener en cuenta numerosos aspectos que afectan a la estimación como la complejidad del proyecto, su estructuración, el tamaño, los recursos involucrados y los riesgos asociados (Pressman, 2010).

La estimación del costo de un software es el proceso de predecir la cantidad de esfuerzo requerido para el desarrollo del sistema y el tiempo para ello. Existen diversos modelos para realizar la estimación del costo de un software como, por ejemplo: COCOMO I, COCOMO II, Puntos de Función, Botton – Up, Top – Down, entre otros.

##### 2.2.4.1 Método Puntos de Función

Es una métrica que permite traducir en un número el tamaño de la funcionalidad que brinda un producto de software desde el punto de vista del usuario, a través de una suma ponderada de las características del producto.

Componentes:

- El: Procesos en los que se introducen datos y que suponen la actualización de cualquier archivo interno.

- EO: Procesos en los que se envía datos al exterior de la aplicación.
- EQ: Procesos consistentes en la combinación de una entrada y una salida, en el que la entrada no produce ningún cambio en ningún archivo y la salida no contiene información derivada.
- ILF: Grupos de datos relacionados entre sí internos al sistema.
- EIF: Grupos de datos que se mantienen externamente.

Una vez obtenidos los diferentes elementos del sistema se utilizan las tablas reflejadas en el Anexo 6 para asignar pesos en función del número de atributos que tengan y el número de archivos a los que afecte (Suardiyana Putra et al., 2012).

**Tabla 2.3** Cantidad de Componentes por su peso.

**Fuente:** Elaboración Propia

Componente	Bajo	Medio	Alto	Total
EI	$5 * 3 = 15$	$14 * 4 = 56$	$6 * 6 = 36$	107
EO	$3 * 4 = 12$	$6 * 5 = 30$	$0 * 7 = 0$	42
EQ	$4 * 3 = 12$	$5 * 4 = 20$	$0 * 6 = 0$	32
ILF	$0 * 7 = 0$	$14 * 10 = 140$	$0 * 15 = 0$	140
EIF	$0 * 5 = 0$	$0 * 7 = 0$	$0 * 10 = 0$	0
				<b>321</b>

### Cálculo de los Puntos de Función sin Ajustar (PFSA)

Los PFSA se calculan como la suma de los productos de cada componente por su peso determinado en la tabla correspondiente.

$$PFSA = 107 + 42 + 32 + 140 + 0 = 321$$

### Ajuste de Complejidad Técnica (ACT)

Para calcular el ACT se le va dando un valor entre 0 y 5 a cada Factor de Ajuste como se muestra en el Anexo 7. Cuando cada Factor tenga un valor, se suman todos y así obtenemos el ACT. A continuación, se muestra dicho procedimiento.

### Cálculo de los Puntos de Función Ajustados (PFA)

$$PFA = PFSA * [0.65 + (0.01 * ACT)]$$

$$PFA = 321 * [0.65 + (0.01 * 31)] = 208.96$$

### **Cálculo del Esfuerzo**

$$\text{Lineas de Código}(LC) = PFA + \text{Lineas} * PF$$

$$LC = 208.96 * 20 = 4180$$

### **Esfuerzo en horas / persona (E)**

$$E = \frac{PFA}{\left(\frac{1 \text{ persona}}{8 \text{ horas}}\right)}$$

$$E = \frac{208}{(20)} = 1671.68 \text{ horas}$$

Tomando 26 días laborables en el mes y 8 horas productivas al día, se obtiene 208 horas laborables al mes.

$$\text{Duración al mes (DM)} = \frac{E}{\left(208 \frac{\text{horas}}{\text{mes}}\right)}$$

$$\text{Duración al mes (DM)} = \frac{1671.6 \text{ horas}}{\left(208 \frac{\text{horas}}{\text{mes}}\right)} = 8 \text{ meses}$$

### **Cálculo del Presupuesto del Proyecto**

Costo Total del proyecto = sueldo de 1 participante \* cantidad de participantes \* DM

$$\text{Costo Total del proyecto} = 2000 * 1 * 8 = 16000$$

#### **2.2.5 Análisis de los beneficios**

Realizando un análisis del costo respecto a los beneficios descritos a lo largo de este trabajo se puede concluir que los aspectos positivos son superiores a los costos. Además, debe señalarse que para esta empresa este software resultó gratuito puesto que es el resultado del trabajo de diploma del autor.

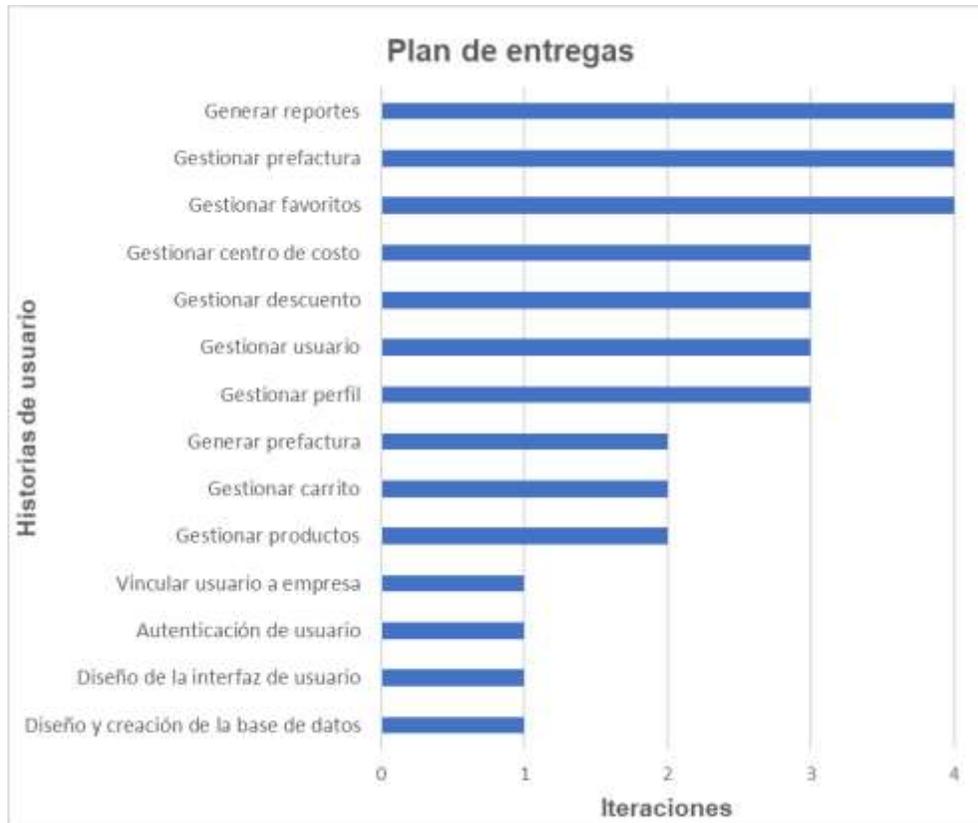
#### **2.2.6 Plan de Entregas**

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de

una reunión entre todos los actores del proyecto. Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Luego de algunas iteraciones es recomendable realizar nuevamente una reunión con los actores del proyecto, para evaluar nuevamente el plan de entregas y ajustarlo si es necesario.

**Tabla 2.4** Plan de entrega. **Fuente:** Elaboración Propia.

Entrega	Descripción
1	Esta entrega comprende: - Diseño y creación de la base de datos. - Diseño de la interfaz de usuario. - Autenticación de usuario. - Vincular usuario a empresa.
2	Esta entrega comprende: - Gestionar productos. - Gestionar carrito. - Generar prefactura.
3	Esta entrega comprende: - Gestionar perfil. - Gestionar usuario. - Gestionar descuento. - Gestionar centro de costo.
4	Esta entrega comprende: - Gestionar favoritos. - Gestionar prefactura. - Generar reportes.



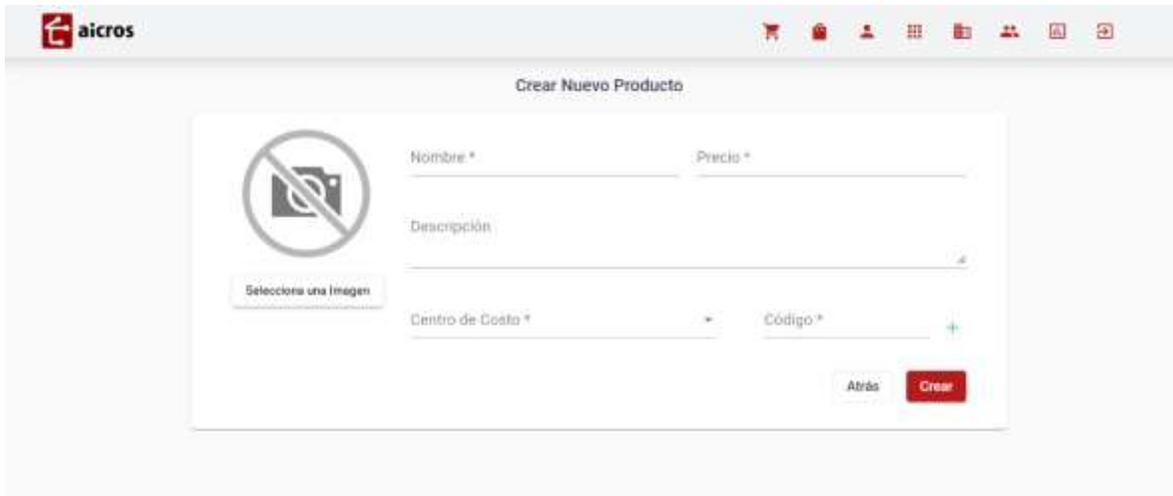
**Figura 2.2** Plan de entregas. **Fuente:** Elaboración Propia.

## 2.3 Etapa de diseño

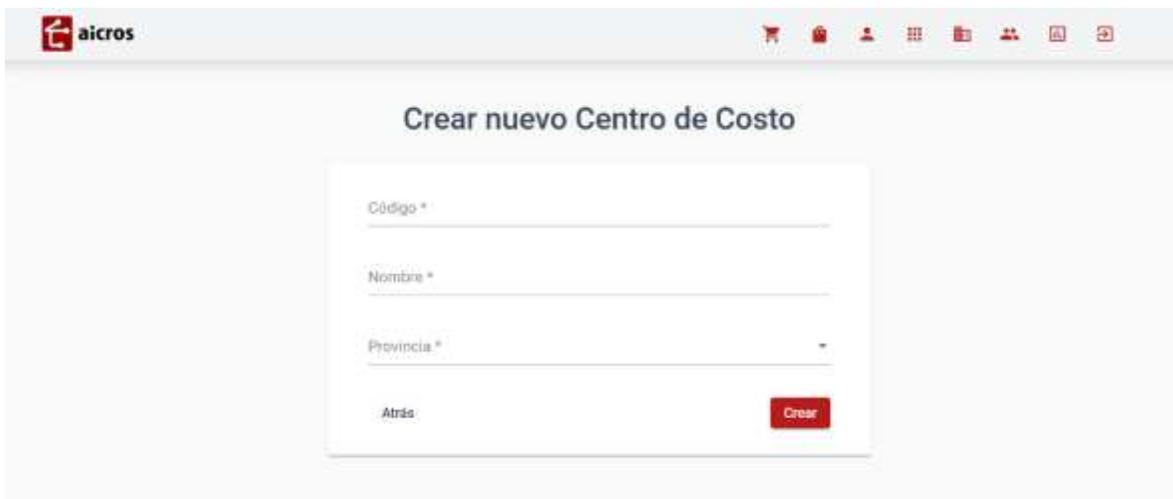
La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible que a la larga costará menos tiempo y esfuerzo desarrollar. Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases ayudará a comprender el diseño y facilitará sus posteriores ampliaciones y reutilización del código. Si surgen problemas potenciales durante el diseño, XP sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el riesgo que supone ese problema (Pressman, 2010).

### 2.3.1 Prototipo de la interfaz de usuario

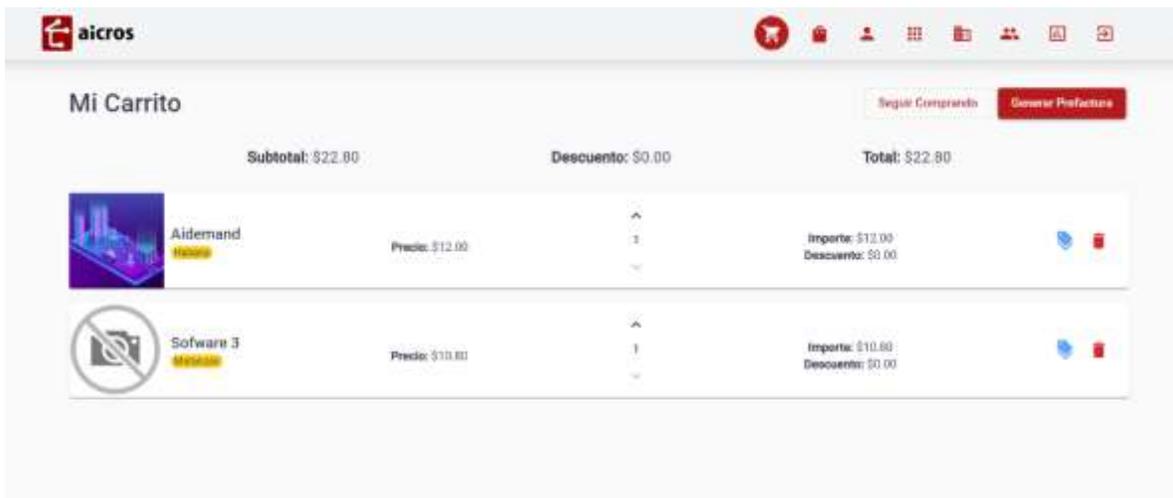
Se muestran ilustraciones de la interfaz de usuario en la figura 2.3 a la figura 2.5.



**Figura 2.3** Interfaz de insertar un nuevo producto. **Fuente:** Elaboración Propia.



**Figura 2.4** Interfaz de crear un nuevo centro de costo. **Fuente:** Elaboración Propia.



**Figura 2.5** Interfaz de carrito. **Fuente:** Elaboración Propia.

A continuación, se muestran en detalle para que se pueda comprender el proceso, las historias de usuarios que se consideran fundamentales para lograr el objetivo esperado.

**Tabla 2.5** *Diseño y creación de la base de datos.* **Fuente:** *Elaboración Propia.*

Historia de usuario	
<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre:</b> Diseño y creación de la base de datos	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Se diseña e implementa la base de datos en el gestor PostgreSQL.	
<b>Observaciones:</b> Crear los roles del sistema en el gestor de base de datos para una mayor seguridad.	

**Tabla 2.6** *Diseño de la interfaz de usuario.* **Fuente:** *Elaboración Propia.*

Historia de usuario	
<b>Número:</b> 2	<b>Usuario:</b> Administrador
<b>Nombre:</b> Diseño de la interfaz de usuario	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Se diseñan las interfaces de la aplicación.	
<b>Observaciones:</b> Resaltar los colores de la empresa y potenciar la experiencia de usuario.	

**Tabla 2.7** *Autenticación de usuario.* **Fuente:** *Elaboración Propia.*

Historia de usuario	
<b>Número:</b> 3	<b>Usuario:</b> Administrador, Cliente
<b>Nombre:</b> Autenticación de usuario	

<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> El usuario introduce sus credenciales, de ser correctas, se le muestra el entorno correspondiente a su rol, de lo contrario se mostrará un error.	
<b>Observaciones:</b> Dar la posibilidad de registrarse en el sistema a los nuevos clientes.	

**Tabla 2.8** Vincular usuario a empresa.

**Fuente:** Elaboración Propia.

<b>Historia de usuario</b>	
<b>Número:</b> 4	<b>Usuario:</b> Cliente
<b>Nombre:</b> Vincular usuario a empresa	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> El usuario busca su empresa y de ser una empresa nueva se le da la opción de crearla.	
<b>Observaciones:</b> La búsqueda de la empresa debe realizarse por los códigos NIT y Reep.	

**Tabla 2.9** Gestionar productos.

**Fuente:** Elaboración Propia.

<b>Historia de usuario</b>	
<b>Número:</b> 5	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gestionar productos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Ernesto David Escariz Ramos	

<b>Descripción:</b> Inicia cuando el usuario procede a gestionar los productos de la tienda. Podrá buscar, insertar, editar y modificar los productos existentes.
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.

**Tabla 2.10** Gestionar carrito.

**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 6	<b>Usuario:</b> Cliente
<b>Nombre:</b> Gestionar carrito	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario procede a gestionar los productos que desea comprar. Podrá añadir, quitar y editar la cantidad que desea en el carrito.	
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.	

**Tabla 2.11** Generar prefactura.

**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 7	<b>Usuario:</b> Cliente
<b>Nombre:</b> Generar prefactura	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario está conforme con los productos que dispuso en el carrito y desea proceder con la compra. Se exportará en formato PDF.	
<b>Observaciones:</b> La aplicación bloquea esta funcionalidad si el carrito está vacío.	

**Tabla 2.12** Gestionar perfil.**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 8	<b>Usuario:</b> Cliente, Administrador
<b>Nombre:</b> Gestionar perfil	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario accede a editar su perfil para cambiar algunas de sus credenciales. Si el usuario está asociado a una empresa se le habilita la posibilidad de editarla también.	
<b>Observaciones:</b> La aplicación solo mostrará la información del usuario que está actualmente autenticado en el sistema.	

**Tabla 2.13** Gestionar usuario.**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 9	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gestionar usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario accede a gestionar los usuarios de la aplicación. Podrá buscar, eliminar, actualizar y crear usuarios en la aplicación.	
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.	

**Tabla 2.14** Gestionar descuento.**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 10	<b>Usuario:</b> Administrador

<b>Nombre:</b> Gestionar descuento	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario accede a gestionar los descuentos de un producto de la tienda. Podrá editar, crear y eliminar los descuentos de cada producto.	
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.	

**Tabla 2.15** Gestionar centro de costo.

**Fuente:** Elaboración Propia.

<b>Historia de usuario</b>	
<b>Número:</b> 11	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gestionar centro de costo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alto
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario accede a gestionar los centros de costo de la tienda. Podrá editar, buscar, eliminar y crear centros de costos.	
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.	

**Tabla 2.16** Gestionar favoritos.

**Fuente:** Elaboración Propia.

<b>Historia de usuario</b>	
<b>Número:</b> 12	<b>Usuario:</b> Cliente
<b>Nombre:</b> Gestionar favoritos	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Ernesto David Escariz Ramos	

<b>Descripción:</b> Inicia cuando el usuario accede a gestionar los productos marcados como favoritos. Permite al usuario añadir, quitar y buscar favoritos.
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.

**Tabla 2.17** Gestionar prefectura.

**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 13	<b>Usuario:</b> Cliente
<b>Nombre:</b> Gestionar prefectura	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 2	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario accede a gestionar las prefecturas que ha generado. Se muestra el estado de cada prefectura.	
<b>Observaciones:</b> Las prefecturas se muestran ordenadas por fecha.	

**Tabla 2.18** Gestionar reportes.

**Fuente:** Elaboración Propia.

Historia de usuario	
<b>Número:</b> 14	<b>Usuario:</b> Administrador
<b>Nombre:</b> Generar reportes	
<b>Prioridad en negocio:</b> Media	<b>Riesgo en desarrollo:</b> Medio
<b>Puntos estimados:</b> 1	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Inicia cuando el usuario accede a ver reportes. Los reportes se mostrarán en gráficas.	
<b>Observaciones:</b> La aplicación solo debe dar acceso a esta funcionalidad si el usuario está en correspondencia con el rol que cumple en la aplicación.	

### 2.3.2 Tareas de Ingeniería

Es importante descomponer cada una de las historias de usuario en tareas de ingeniería.

**Tabla 2.18** Tareas de Ingeniería.

**Fuente:** Elaboración Propia.

No	Nombre HU	No	Tarea de Ingeniería
1	Diseño y creación de la base de datos	1	Diseñar la base de datos
		2	Crear la base de datos
2	Diseño de la interfaz de usuario	3	Diseñar la interfaz de usuario
3	Autenticación de usuario	4	Autenticar usuario existente
		5	Registrar nuevo usuario
		6	Recuperar contraseña
4	Vincular usuario a empresa	7	Listar empresa
		8	Filtrar empresa
		9	Crear empresa
		10	Asignar usuario a empresa
5	Gestionar productos	11	Listar producto
		12	Filtrar producto
		13	Crear producto
		14	Editar producto
		15	Eliminar producto
6	Gestionar carrito	16	Añadir al carrito
		17	Eliminar del carrito
		18	Editar cantidad
7	Generar prefactura	19	Generar prefactura
8	Gestionar perfil	20	Editar usuario
		21	Editar empresa
9	Gestionar usuario	22	Listar usuario
		23	Filtrar usuario
		24	Crear usuario
		25	Editar usuario
		26	Eliminar usuario
10	Gestionar descuento	27	Listar descuento

		28	Crear descuento
		29	Editar descuento
		30	Eliminar descuento
11	Gestionar centro de costo	31	Listar centro de costo
		32	Filtrar centro de costo
		33	Crear centro de costo
		34	Editar centro de costo
		35	Eliminar centro de costo
12	Gestionar favoritos	36	Añadir favorito
		37	Eliminar favorito
		38	Filtrar favorito
13	Gestionar prefactura	39	Listar prefactura
		40	Filtrar prefactura
		41	Cancelar prefactura
14	Generar reportes	42	Reporte de prefacturas por estado
		43	Reporte de ingresos por mes
		44	Reporte de productos más vendidos

En las tablas que se muestran a continuación se relacionan algunas tareas de ingeniería que tenían mayor peso en el desarrollo de esta investigación.

**Tabla 2.19** *Diseño de la base de datos.* **Fuente:** *Elaboración Propia.*

Tarea de Ingeniería	
<b>Número de tarea:</b> 1	<b>Número de historia de usuario:</b> 1
<b>Nombre:</b> Diseño de la base de datos	
<b>Tipo de tarea:</b> Diseño	<b>Puntos estimados:</b> 0.5
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Analizar profundamente el negocio para crear la base de datos que permita almacenar dicha información.	

**Tabla 2.20** *Creación de la base de datos.* **Fuente:** *Elaboración Propia.*

Tarea de Ingeniería
---------------------

<b>Número de tarea:</b> 2	<b>Número de historia de usuario:</b> 1
<b>Nombre:</b> Creación de la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Crear la base de datos con sus respectivas relaciones y la integridad correspondiente entre las tablas.	

**Tabla 2.21** Insertar producto. *Fuente: Elaboración Propia.*

Tarea de Ingeniería	
<b>Número de tarea:</b> 13	<b>Número de historia de usuario:</b> 5
<b>Nombre:</b> Insertar producto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Se mostrará un formulario donde el usuario con los permisos necesarios podrá insertar un nuevo producto, que será mostrado en la tienda.	

**Tabla 2.22** Añadir al carrito. *Fuente: Elaboración Propia.*

Tarea de Ingeniería	
<b>Número de tarea:</b> 16	<b>Número de historia de usuario:</b> 6
<b>Nombre:</b> Añadir al carrito	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Las tarjetas de producto tendrán un botón que permitirá añadir dicho producto al carrito de compras. El mismo será accesible desde el listado de productos, los detalles del producto y si el producto se encuentra marcado como favorito, en la pantalla de favoritos.	

**Tabla 2.23** Generar prefactura. **Fuente:** Elaboración Propia.

Tarea de Ingeniería	
<b>Número de tarea:</b> 19	<b>Número de historia de usuario:</b> 7
<b>Nombre:</b> Generar prefactura	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Una vez que el usuario esté conforme con la cantidad y los productos que va a comprar, procede a generar la prefactura, la cual se le muestra con su estructura final y se da la opción de descargarla en formato PDF.	

**Tabla 2.23** Crear usuario. **Fuente:** Elaboración Propia.

Tarea de Ingeniería	
<b>Número de tarea:</b> 24	<b>Número de historia de usuario:</b> 9
<b>Nombre:</b> Crear usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Programador responsable:</b> Ernesto David Escariz Ramos	
<b>Descripción:</b> Se mostrará un formulario donde el usuario con los permisos necesarios podrá insertar un usuario con su rol correspondiente.	

**Tabla 2.23** Cancelar prefactura. **Fuente:** Elaboración Propia.

Tarea de Ingeniería	
<b>Número de tarea:</b> 41	<b>Número de historia de usuario:</b> 13
<b>Nombre:</b> Cancelar prefactura	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Programador responsable:</b> Ernesto David Escariz Ramos	

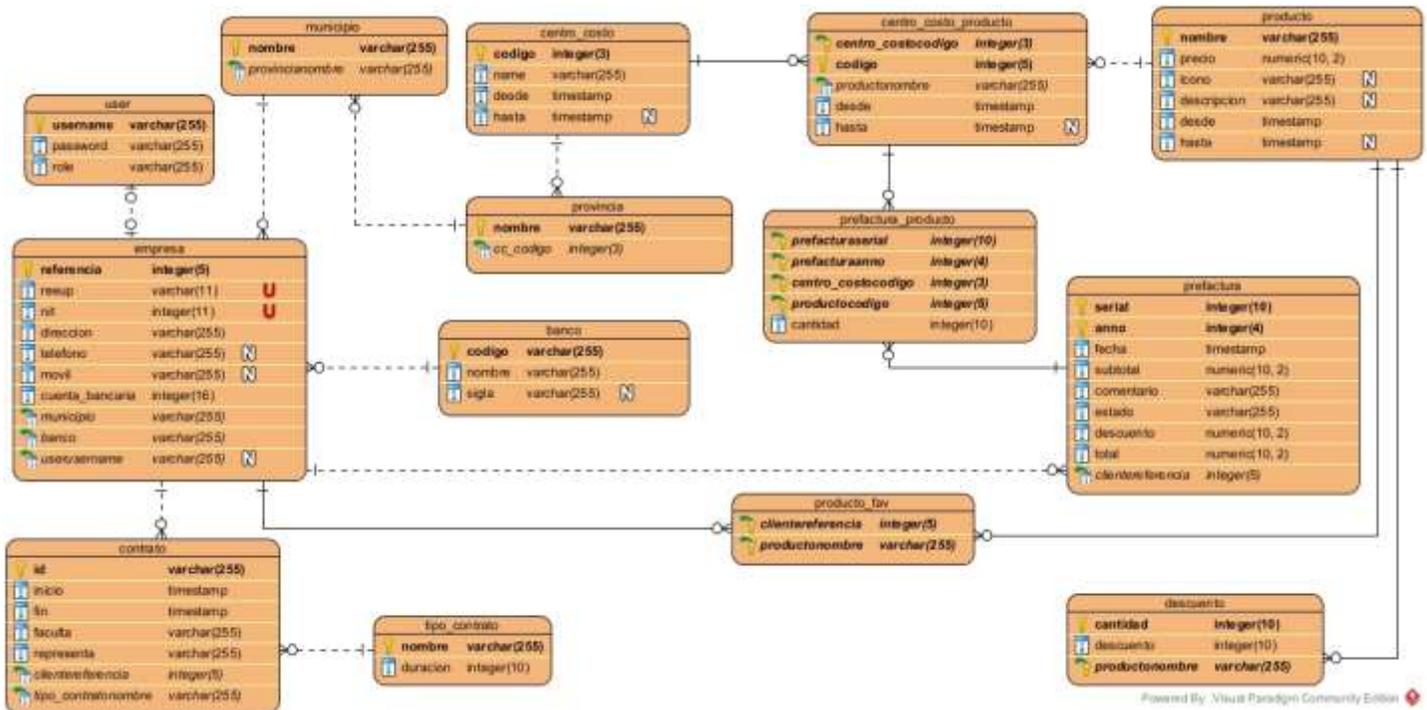
**Descripción:** Por cada prefectura que se muestre en el listado de prefecturas del usuario, habrá la opción de cancelar, siempre que el estado de la prefectura no sea facturado. Se pasará el estado de la prefectura hacia cancelado.

**Tabla 2.23** Reporte de productos más vendidos. **Fuente:** Elaboración Propia.

Tarea de Ingeniería	
Número de tarea: 44	Número de historia de usuario: 14
Nombre: Reporte de productos más vendidos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Programador responsable: Ernesto David Escariz Ramos	
Descripción: Se mostrará en un gráfico los productos que más vendidos de la tienda junto a las respectivas cantidades.	

### 2.3.3 Modelo físico de la Base de datos

En la ilustración se observa el modelo físico de la base de datos de la aplicación web.



**Figura 2.3** Modelo de la base de datos.

**Fuente:** Elaboración Propia.

#### **2.4 Conclusiones parciales del capítulo**

Entre las principales conclusiones a las que se pudieron llegar en este capítulo está que mediante la metodología XP, se reflejaron las etapas imprescindibles para desarrollar el software, exceptuando las pruebas del mismo. Se planteó la composición del equipo de desarrollo, las diferentes historias de usuarios y las respectivas interacciones necesarios para acometer la labor de construcción de la aplicación. Se realizó la planificación del proyecto y análisis de costo de este, aplicando el modelo de Puntos de Función, siendo de \$ 16000. Los aspectos abordados anteriormente permitieron reflejar el diseño, estructura y funcionalidad del software a desarrollar, proporcionando una visión clara del proyecto en su conjunto para que las distintas partes involucradas en el mismo llegasen a un acuerdo.

## **Capítulo 3. Validación de la solución propuesta**

En este capítulo se realizan las pruebas al software que permiten comprobar la calidad de este producto, lo que constituye uno de los pasos más importantes en el desarrollo de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de mostrar un error no descubierto hasta entonces y con el fin de verificar la fiabilidad y calidad de la aplicación como un todo.

### **3.1 Pruebas al software**

El proceso de pruebas es el instrumento más adecuado para determinar el status de la calidad de un producto. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos o si es el software que se quería desarrollar. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante Técnicas de Prueba.

Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces. Los niveles de trabajo en los cuales se pueden realizar las pruebas son:

- Prueba de Unidad.
- Prueba de Integración.
- Prueba de Sistema.
- Prueba de Aceptación.
- Prueba de Seguridad.

#### **3.2.1 Plan de pruebas**

El plan de pruebas de software se elabora con el fin de especificar qué elementos o componentes se van a probar para que el grupo de trabajo pueda realizar el proceso de Validación y Verificación de los requerimientos funcionales y no funcionales. Además, a través del plan de pruebas se puede continuar con la trazabilidad de los requerimientos, con lo cual el grupo de trabajo, identifica el porcentaje de avance que se ha logrado hasta cierto momento (Reyes Chirino & Delgado Fernández, 2021).

Al desarrollar el plan de pruebas, se puede obtener información sobre los errores, defectos o fallas que tiene el prototipo, así se realizan las correcciones pertinentes, según el caso y

se asegura la calidad del producto que se está entregando al cliente. A continuación, se muestra en la Tabla 3.1 el plan de prueba.

**Tabla 3.1** Plan de pruebas.

**Fuente:** Elaboración Propia.

No	Nombre HU	Pruebas a realizar
1	Diseño y creación de la base de datos	Test base de datos
2	Diseño de la interfaz de usuario	Test diseño de la interfaz de usuario
3	Autenticación de usuario	Test autenticación de usuario Test registro nuevo usuario Test recuperar contraseña
4	Vincular usuario a empresa	Test listar empresa Test filtrar empresa Test crear empresa Test asignar usuario a empresa
5	Gestionar productos	Test listar producto Test filtrar producto Test crear producto Test editar producto Test eliminar producto
6	Gestionar carrito	Test añadir al carrito Test eliminar del carrito Test editar cantidad
7	Generar prefactura	Test generar prefactura
8	Gestionar perfil	Test editar usuario Test editar empresa
9	Gestionar usuario	Test listar usuario Test filtrar usuario

		Test crear usuario Test editar usuario Test eliminar usuario
10	Gestionar descuento	Test listar descuento Test crear descuento Test editar descuento Test eliminar descuento
11	Gestionar centro de costo	Test listar centro de costo Test filtrar centro de costo Test crear centro de costo Test editar centro de costo Test eliminar centro de costo
12	Gestionar favoritos	Test añadir favorito Test eliminar favorito Test filtrar favorito
13	Gestionar prefactura	Test listar prefactura Test filtrar prefactura Test cancelar prefactura
14	Generar reportes	Test reportes

### 3.2.2 Pruebas de aceptación

Cuando se construye software a la medida para un cliente, se realiza una serie de pruebas de aceptación a fin de permitir al cliente validar todos los requerimientos. Realizada por el usuario final en lugar de por los ingenieros de software, una prueba de aceptación puede variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente. De hecho, la prueba de aceptación puede realizarse durante un periodo de semanas o meses, y mediante ella descubrir errores acumulados que con el tiempo puedan degradar el sistema (Pressman, 2010).

Uno de los pilares de la metodología XP es el uso de pruebas para comprobar el funcionamiento de los códigos que se vayan implementando. Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información.

**Tabla 3.2** Test base de Datos. **Fuente:** Elaboración Propia.

<b>Caso de prueba</b>	
<b>Número:</b> 1	<b>Historia de usuario:</b> 1
<b>Nombre de caso de pruebas:</b> Test base de Datos	
<b>Descripción:</b> Verifica el funcionamiento de la base de datos.	
<b>Condición de ejecución:</b> Estar conectado a la base de datos.	
<b>Entradas:</b> Valores para leer o escribir en la base de datos, por ejemplo: usuario, contraseña.	
<b>Resultado esperado:</b> Se muestran o guardan los datos correctamente.	
<b>Evaluación:</b> Prueba satisfactoria.	

**Tabla 3.3** Test autenticación de usuario. **Fuente:** Elaboración Propia.

<b>Caso de prueba</b>	
<b>Número:</b> 3	<b>Historia de usuario:</b> 3
<b>Nombre de caso de pruebas:</b> Test autenticación de usuario	
<b>Descripción:</b> Se inserta el nombre de usuario y contraseña para entrar al sistema. Se insertarán de forma incorrecta, dejando campos en blanco para verificar la validación. Luego se insertarán los datos de manera correcta para comprobar esta funcionalidad.	
<b>Condición de ejecución:</b> La aplicación web debe estar en ejecución.	

<b>Entradas:</b> Nombre de usuario y contraseña
<b>Resultado esperado:</b> El sistema notifica al usuario cuando se inserten datos erróneos o cuando se dejen campos en blanco. Cuando se inserten los datos correctamente, según el rol que tenga el usuario, se le otorgan los permisos correspondientes.
<b>Evaluación:</b> Prueba satisfactoria.

**Tabla 3.4** Test gestionar producto.

**Fuente:** Elaboración Propia.

Caso de prueba	
<b>Número:</b> 5	<b>Historia de usuario:</b> 5
<b>Nombre de caso de pruebas:</b> Test gestionar producto	
<b>Descripción:</b> Se inserta los datos para crear un nuevo producto, se inserta de forma incorrecta dejando el campo en blanco, luego se inserta de forma correcta para verificar que los datos sean almacenados. Se modifican los datos de producto, se modifican incorrectamente dejando campos en blanco para verificar la validación, se modifican correctamente para que los sean almacenados. Se elimina el producto aceptando el mensaje de confirmación.	
<b>Condición de ejecución:</b> La aplicación web debe estar en ejecución.	
<b>Entradas:</b> Datos del producto	
<b>Resultado esperado:</b> El sistema debe alertar cuando se insertan datos erróneos. Cuando se insertan los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos.	
<b>Evaluación:</b> Prueba satisfactoria	

En la Tabla 3.5 se muestran las clases de equivalencia de las pruebas de aceptación al sistema en la HU.

**Tabla 3.5** Clases de equivalencia

**Fuente:** Elaboración Propia.

Condición de entrada	Clases válidas	Representante	Clases inválidas	Representante
nombre	1. Cualquier cadena de caracteres alfabéticos. Permite espacio.	Aibalan	2. Cadena vacía	NULL

precio	3. Cualquier número positivo con los decimales separados por un punto.	1725.25	4. Número negativo	-1750.5
			5. Cadena que contenga letras	100CUP
			6. Separación de los decimales por coma	3500,25
			7. Cadena vacía	NULL
centro de costo	8. Seleccionar alguna de las opciones	902 Centro Oficial Matanzas	9. No seleccionar ninguna opción	NULL
código	10. Cadena de 5 dígitos	24910	11. Cadena vacía	NULL
			12. Cadena que contiene caracteres no numéricos	23t.1
			13. Cadena con menos de 5 caracteres	2491
			14. Cadena con más de 5 caracteres	249100

En la Tabla 3.6 se muestra un resumen de algunos casos de pruebas necesarios para realizar las pruebas de aceptación.

**Tabla 3.6** Resumen de Casos de prueba.

**Fuente:** Elaboración Propia.

No.	Clases equiv.	Nombre	Precio	Centro de costo	Código	Resultado
1	1, 3, 8, 10	Aibalan	1725.25	902 Centro Oficial Matanzas	24901	Producto insertado
2	1, 3, 8, 11	Aibalan	1725.25	902 Centro Oficial Matanzas	NULL	El código es requerido.

3	1, 3, 8, 12	Aibalan	1725.25	902 Centro Oficial Matanzas	24.01	El código debe ser un número de 5 dígitos.
4	1, 3, 8, 13	Aibalan	1725.25	902 Centro Oficial Matanzas	9012	El código debe ser un número de 5 dígitos.
5	1, 3, 8, 14	Aibalan	1725.25	902 Centro Oficial Matanzas	901289	El código debe ser un número de 5 dígitos.
6	1, 3, 9, 10	Aibalan	1725.25	NULL	90128	El centro de costo es requerido.
7	1, 4, 8, 10	Aibalan	- 1725.25	902 Centro Oficial Matanzas	90128	El precio debe tener el formato 00.00
8	1, 5, 8, 10	Aibalan	18u	902 Centro Oficial Matanzas	90128	El precio debe tener el formato 00.00
9	1, 6, 8, 10	Aibalan	1900,25	902 Centro Oficial Matanzas	90128	El precio debe tener el formato 00.00
10	1, 7, 8, 10	Aibalan	NULL	902 Centro Oficial Matanzas	90128	El precio es requerido.
11	2, 3, 8, 10	NULL	1200	902 Centro Oficial Matanzas	90128	El nombre es requerido.
12	2, 3, 8, 11	NULL	1200	902 Centro Oficial Matanzas	NULL	El nombre es requerido. El código es requerido.
13	2, 3, 8, 12	NULL	1200	902 Centro Oficial Matanzas	9o123	El nombre es requerido. El código debe ser un número de 5 dígitos.
14	2, 3, 8, 13	NULL	1200	902 Centro Oficial Matanzas	9123	El nombre es requerido. El código debe ser un número de 5 dígitos.
15	2, 3, 8, 14	NULL	1200	902 Centro Oficial Matanzas	912300	El nombre es requerido. El código debe ser un número de 5 dígitos.

16	2, 3, 9, 10	NULL	1200	NULL	91230	El nombre es requerido. El centro de costo es requerido.
17	2, 3, 9, 11	NULL	1200	NULL	NULL	El nombre es requerido. El centro de costo es requerido. El código es requerido.
18	2, 7, 9, 11	NULL	NULL	NULL	NULL	El nombre es requerido. El centro de costo es requerido. El código es requerido.

A continuación, se muestran algunas Tablas de Prueba por Caso de Prueba

**Tabla 3.7** Test insertar producto caso prueba 1. **Fuente:** Elaboración Propia.

Tabla de prueba	
<b>No. Caso de Prueba</b>	1
<b>Requerimiento</b>	La aplicación web debe estar en ejecución
<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases válidas 1, 3, 8, 10)
<b>Tipo de Prueba</b>	Funcional
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB
<b>Software</b>	Navegador de internet Google Chrome v103.0
<b>Personal</b>	Encargados de pruebas
<b>Datos de entrada</b>	Nombre: Aibalan, precio: 1725.25, centro de costo: 902 Centro Oficial Matanzas, código: 24901
<b>Resultados esperados</b>	Producto insertado correctamente
<b>Resultados obtenidos</b>	SI(X) NO()

<b>Casos de excepción</b>		
<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos

**Tabla 3.8** Test insertar producto, caso prueba 2. **Fuente:** Elaboración Propia.

<b>Tabla de prueba</b>		
<b>No. Caso de Prueba</b>	2	
<b>Requerimiento</b>	La aplicación web debe estar en ejecución	
<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases 1, 3, 8, 11)	
<b>Tipo de Prueba</b>	Funcional	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB	
<b>Software</b>	Navegador de internet Google Chrome v103.0	
<b>Personal</b>	Encargados de pruebas	
<b>Datos de entrada</b>	Nombre: Aibalan, precio: 1725.25, centro de costo: 902 Centro Oficial Matanzas, código: NULL	
<b>Resultados esperados</b>	El código es requerido	
<b>Resultados obtenidos</b>	SI(X) NO()	
<b>Casos de excepción</b>		
<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos

**Tabla 3.8** Test insertar producto, caso prueba 5. **Fuente:** Elaboración Propia.

Tabla de prueba		
<b>No. Caso de Prueba</b>	5	
<b>Requerimiento</b>	La aplicación web debe estar en ejecución	
<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases 1, 3, 8, 14)	
<b>Tipo de Prueba</b>	Funcional	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB	
<b>Software</b>	Navegador de internet Google Chrome v103.0	
<b>Personal</b>	Encargados de pruebas	
<b>Datos de entrada</b>	Nombre: Aibalan, precio: 1725.25, centro de costo: 902 Centro Oficial Matanzas, código: 784501	
<b>Resultados esperados</b>	El código debe ser un número de 5 dígitos.	
<b>Resultados obtenidos</b>	SI(X) NO()	
<b>Casos de excepción</b>		
<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos

**Tabla 3.9** Test insertar producto, caso prueba 6. **Fuente:** Elaboración Propia.

Tabla de prueba		
<b>No. Caso de Prueba</b>	6	
<b>Requerimiento</b>	La aplicación web debe estar en ejecución	
<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases 1, 3, 9, 10)	
<b>Tipo de Prueba</b>	Funcional	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB	

<b>Software</b>	Navegador de internet Google Chrome v103.0	
<b>Personal</b>	Encargados de pruebas	
<b>Datos de entrada</b>	Nombre: Aibalan, precio: 1725.25, centro de costo: NULL, código: 78450	
<b>Resultados esperados</b>	El centro de costo es requerido.	
<b>Resultados obtenidos</b>	SI(X) NO()	
<b>Casos de excepción</b>		
<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos

**Tabla 3.10** Test insertar producto, caso prueba 8. **Fuente:** Elaboración Propia.

<b>Tabla de prueba</b>	
<b>No. Caso de Prueba</b>	8
<b>Requerimiento</b>	La aplicación web debe estar en ejecución
<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases 1, 5, 8, 10)
<b>Tipo de Prueba</b>	Funcional
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB
<b>Software</b>	Navegador de internet Google Chrome v103.0
<b>Personal</b>	Encargados de pruebas
<b>Datos de entrada</b>	Nombre: Aibalan, precio: 1725&25, centro de costo: 902 Centro Oficial Matanzas, código: 78450
<b>Resultados esperados</b>	El precio debe tener el formato 00.00
<b>Resultados obtenidos</b>	SI(X) NO()
<b>Casos de excepción</b>	

<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos
---	---	---

**Tabla 3.11** Test insertar producto, caso prueba 10. **Fuente:** Elaboración Propia.

<b>Tabla de prueba</b>		
<b>No. Caso de Prueba</b>	10	
<b>Requerimiento</b>	La aplicación web debe estar en ejecución	
<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases 1, 7, 8, 10)	
<b>Tipo de Prueba</b>	Funcional	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB	
<b>Software</b>	Navegador de internet Google Chrome v103.0	
<b>Personal</b>	Encargados de pruebas	
<b>Datos de entrada</b>	Nombre: Aibalan, precio: NULL, centro de costo: 902 Centro Oficial Matanzas, código: 78450	
<b>Resultados esperados</b>	El precio es requerido	
<b>Resultados obtenidos</b>	SI(X) NO()	
<b>Casos de excepción</b>		
<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos

**Tabla 3.12** Test insertar producto, caso prueba 11. **Fuente:** Elaboración Propia.

<b>Tabla de prueba</b>	
<b>No. Caso de Prueba</b>	11
<b>Requerimiento</b>	La aplicación web debe estar en ejecución

<b>Objetivo</b>	Probar la acción de insertar un producto (Para cubrir las clases 2, 3, 8, 10)	
<b>Tipo de Prueba</b>	Funcional	
<b>Hardware</b>	Sistema de cómputo con un Procesador Intel Core i3 - Disco sólido de 240GB - Memoria RAM de 8GB	
<b>Software</b>	Navegador de internet Google Chrome v103.0	
<b>Personal</b>	Encargados de pruebas	
<b>Datos de entrada</b>	Nombre: NULL, precio: 450, centro de costo: 902 Centro Oficial Matanzas, código: 78450	
<b>Resultados esperados</b>	El nombre es requerido	
<b>Resultados obtenidos</b>	SI(X) NO()	
<b>Casos de excepción</b>		
<b>Aprobado por:</b> Mayli Estopiñán Lantigua	<b>Cargo:</b> Jefe de Proyecto, Tester	<b>Líder:</b> Ernesto David Escariz Ramos

### 3.2 Análisis de los resultados obtenidos

Después de desarrollar todo un proceso de pruebas con un nivel medio de sencillez se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las pruebas realizadas, se solucionaron varios de ellos que impedían el cumplimiento de los requisitos fundamentales del sistema en cuestión.

Las primeras pruebas fueron planeadas y ejecutadas en módulos individuales del programa y a medida que fueron avanzando se desplazaron a módulos integrados, hasta que finalmente llegaron al sistema completo y se logró obtener un software cuyas funciones se encuentra en correspondencia con las especificaciones acordadas y que además cumple con los requerimientos de rendimiento. El desarrollo del sistema cumple las expectativas trazadas al inicio del proyecto y satisface al cliente en su totalidad.

### 3.3 Conclusiones parciales del capítulo

Al validar la propuesta de solución al problema científico de la investigación a partir del empleo del marco de trabajo XP, y realizando un análisis de los resultados de la misma mediante el empleo principalmente de pruebas funcionales, se concluye que:

1. Fue acertada la planificación concebida, ya que permitió cumplir con los objetivos a los que aspiraba el cliente.
2. El empleo de pruebas funcionales posibilitó la identificación y rectificación de errores en la aplicación que no fueron detectados durante la etapa de desarrollo.
3. A partir del diseño y ejecución de la aplicación propuesta haciendo uso de las tecnologías escogidas para ello, se obtuvo un software que satisface los términos de tiempo y calidad acordados previamente por las partes

## Conclusiones

Luego de haber realizado un estudio exhaustivo de la problemática abordada en la actual investigación y lográndose dar respuesta a ella de manera satisfactoria, se pudo arribar a las siguientes conclusiones:

1. El estudio realizado sobre los antecedentes, la bibliografía y documentos relacionados con el objeto de estudio, permitió confeccionar un marco teórico referencial que, de conjunto al estudio de las herramientas y tecnologías a utilizar, proporcionó los elementos necesarios para dar solución a la problemática abordada.
2. Los sistemas informáticos hallados, que se relacionan al tema no le proporcionan solución al problema planteado por lo que no es factible su utilización.
3. La metodología de desarrollo y herramientas más idóneas para el desarrollo de la propuesta de solución son la metodología de programación extrema XP y los lenguajes HTML, CSS y TypeScript en el lado del cliente, Angular como framework y el lenguaje Node.js con Nest.js como framework en el lado del servidor, además Visual Studio Code como entorno de desarrollo y el gestor de bases de datos PostgreSQL.
4. Se desarrolló la aplicación web para la venta y administración de productos de la empresa AICROS, siguiendo la metodología de desarrollo de software XP, ajustado a los requerimientos del usuario y logrando un producto informático que cumple con el objetivo general trazado en la investigación.
5. Mediante la aplicación de pruebas a la aplicación Web, resultó posible obtener resultados favorables, con el consiguiente análisis de errores detectados que fueron subsanados como parte del desarrollo de esta aplicación.

De manera general, se concluye que el software desarrollado posee la funcionalidad y calidad demandada, presentándose como una herramienta atractiva, confortable y de fácil manejo que permite desempeñar la labor de comercialización de AICROS con mayor eficiencia.

## **Recomendaciones**

Desde el punto de vista del alcance del presente trabajo y teniendo en cuenta el momento de desarrollo del mismo, se proponen las siguientes recomendaciones:

1. Continuar la ampliación de la aplicación mediante la incorporación de nuevas funcionalidades que brinden un mayor control y análisis de los datos almacenados.
2. Proporcionar un número superior de notificaciones, reportes y gráficos que posibiliten una interpretación superior del flujo de información.

## Bibliografía

- Abdelaziz, A., El-Tahir, Y., & Osman, R. (2015). *Adaptive Software Development for developing safety critical software* 2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE),
- Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sheikh, Z. (2010). *Agile software development: Impact on productivity and quality* Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference,
- Almseidin, M., Alrfou, K., Alnidami, N., Tarawneh, A. S., & Tarawneh, A. S. (2015). A Comparative Study of Agile Methods: XP versus SCRUM. *International Journal of Computer Science and Software Engineering (IJCSSE)*, 4(5), 126-129.  
<http://ijcsse.org/published/volume4/issue5/p3-V4I5.pdf>
- Antón Rodríguez, S. (2022a). ¿Ecosistemas digitales para el comercio electrónico en Cuba? *Periódico Granma*. <https://www.granma.cu/doble-click/2022-10-11/ecosistemas-digitales-para-el-comercio-electronico-en-cuba>
- Antón Rodríguez, S. (2022b). El sector de la construcción en el camino de la Industria 4.0. *Periódico Granma*. <https://www.granma.cu/cuba/2022-06-20/el-sector-de-la-construccion-en-el-camino-de-la-industria-40-20-06-2022-23-06-37>
- Beck, K. (1999). *Extreme Programming Explained* (1st Edition ed.). Addison-Wesley Professional.  
<https://www.scrumcn.com/agile/index.php?wpdmact=process&did=MTguaG90bGluaw==>
- Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change, 2nd Edition (The XP Series)*. Addison-Wesley.
- Borges, S. (2019). *Servidor PostgreSQL*. Retrieved 25/11/2022 from  
<https://blog.infranetworking.com/servidor-postgresql/#Caracteristicas>
- Cameron, D. (1997). *Electronic Commerce: The new business platform for the Internet*. Computer Technology Research.
- Carvalho, W., Rosa, P. F., Soares, M. S., Teixeira da Cunha, M. A., & Buiatte, L. C. (2011). *A comparative analysis of the agile and traditional software development processes productivity* 2011 30th International Conference of the Chilean Computer Science Society,
- Chowdhury, A., & Huda, M. (2011). *Comparison between Adaptive Software Development and Feature Driven Development* Proceedings of 2011 International Conference on Computer Science and Network Technology,
- Cimoli, M., García-Buchaca, R., Titelman, D., & Shaw, S. (2022). *Estudio Económico de América Latina y el Caribe: Dinámica y desafíos de la inversión para impulsar una recuperación sostenible e inclusiva*. Publicación de las Naciones Unidas. <https://doi.org/LC/PUB.2022/9-P>
- Díaz-Canel Bermúdez, M. (2015). *Discurso inaugural del I Taller Nacional de Informatización y Ciberseguridad*
- Doshi, V. P., & Patil, V. A. (2016). *Competitor driven development: Hybrid of extreme programming and feature driven reuse development* 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS),
- Faniran, V., Badru, A., & Ajayi, N. A. (2017). *Adopting Scrum as an Agile Approach in Distributed Software Development: A Review of Literature* 1st International Conference on Next Generation Computing Applications (NextComp),  
[https://www.researchgate.net/publication/309120635\\_Adopting\\_Scrum\\_as\\_an\\_Agile\\_Approach\\_in\\_Distributed\\_Software\\_Development\\_A\\_Review\\_of\\_Literature?enrichId=rgreq-cc384e356398f9caa934d02d22c7a006-](https://www.researchgate.net/publication/309120635_Adopting_Scrum_as_an_Agile_Approach_in_Distributed_Software_Development_A_Review_of_Literature?enrichId=rgreq-cc384e356398f9caa934d02d22c7a006-)

- [XXX&enrichSource=Y292ZXJQYWdlOzMwOTEyMDYzNTtBUzoxMTI1ODY1OTU4MTk1MjAwQDE2NDU0Mzg3NzAyNTk%3D&el=1\\_x\\_2&\\_esc=publicationCoverPdf](#)
- Fernández Guerra, E., Peña Tresancos, J., & Álvarez, M. Á. (2017). *Manual de TypeScript*. desarrolloweb.com. <http://desarrolloweb.com/manuales/manual-typescript.html>
- Fernández Sáenz de Navarrete, J. M. (2019). *Diseño e implementación de un entorno web con NodeJS* Universidad de Valladolid]. España.
- Flores, F. (2022). *Qué es Visual Studio Code y qué ventajas ofrece*. Retrieved 25/11/2022 from <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- García Rodríguez, M. J. (2015). *Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos software* Universidad de Oviedo]. <https://digibuo.uniovi.es/dspace/bitstream/handle/10651/32457/TFMMIJGarciaRodriguezRUO.pdf;jsessionid=9C20F4869A08A647D0279C33CE413ECE?sequence=6>
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript* MARCOMBO, S.A.
- Godoy-Rosero, P. C., Hurtado-Sotalin, D. M., Nazareno-Arteaga, I. M., Rentería-Macías, H. J., Plata-Cabrera, C. S., Barboza-Quintero, G., & JCortez-Altafuya, o. J. (2022). Análisis del E-commerce en el entorno comercial basado en la aplicación de las TICs. *Polo del Conocimiento*, 7(4). <https://doi.org/10.23857/pc.v7i4.3869>
- Gonçalves, M. J. (2021). *¿Qué es Angular y para qué sirve?* hiberus Blog. Retrieved 25/11/2022 from <https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/>
- Guerra Velázquez, D., & de la Vega Valenzuela, J. D. (2013). *Solución informática para mejorar la calidad de los datos en el Sistema de Gestión Universitaria* Universidad de las Ciencias Informáticas]. La Habana.
- Kaur, K., Jajoo, A., & Manisha. (2015). *Applying Agile Methodologies in Industry Projects: Benefits and Challenges* 2015 International Conference on Computing Communication Control and automation (ICCUBEA), [https://www.researchgate.net/publication/301476735\\_Applying\\_Agile\\_Methodologies\\_in\\_Industry\\_Projects\\_Benefits\\_and\\_Challenges?enrichId=rgreq-45f373dcf6702340f8a145099f827c78-XXX&enrichSource=Y292ZXJQYWdlOzMwMTQ3NjczNTtBUzoxMTEzOTEyMzY3OTQzNjgxQDE2NDI1ODg4MTI0MDQ%3D&el=1\\_x\\_2&\\_esc=publicationCoverPdf](https://www.researchgate.net/publication/301476735_Applying_Agile_Methodologies_in_Industry_Projects_Benefits_and_Challenges?enrichId=rgreq-45f373dcf6702340f8a145099f827c78-XXX&enrichSource=Y292ZXJQYWdlOzMwMTQ3NjczNTtBUzoxMTEzOTEyMzY3OTQzNjgxQDE2NDI1ODg4MTI0MDQ%3D&el=1_x_2&_esc=publicationCoverPdf)
- Kirmani, M. M. (2017). *Agile methods for mobile application development: A comparative analysis*. International Journal of Advanced Research in Computer Science,
- Kowtha, R., & Choon, T. W. I. (2001). Determinants of Website Development: A study of electronic commerce in Singapore. *Information & Management*, 3(39), 227-242. [https://doi.org/10.1016/S0378-7206\(01\)00092-1](https://doi.org/10.1016/S0378-7206(01)00092-1)
- Laínez Fuentes, J. R. (2015). *Desarrollo de Software Ágil. Extremme Programming y Scrum*. IT Campus Academy.
- Laudon, K. C., & Laudon, J. P. (2005). *Management Information System: Managing the Digital Firm* (9th ed.). Prentice-Hall.
- Letelier Torres, P., Penadés, M. C., Canós, J. H., & Sánchez López, E. A. (2003). *Metodologías Ágiles en el Desarrollo de Software* Taller de Metodologías Ágiles en el Desarrollo de Software, <https://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
- Lopez Campeny, S. M. L., & Ponce, N. A. (2022). Torteros 2.0. Sistema gestor de base de datos (SGBD) para registro, consulta y análisis de pesos para hilar arqueológicos. *Revista del Museo de Antropología*, 15(2). <https://doi.org/http://doi.org/10.31048/1852.4826.v15.n2.36192>

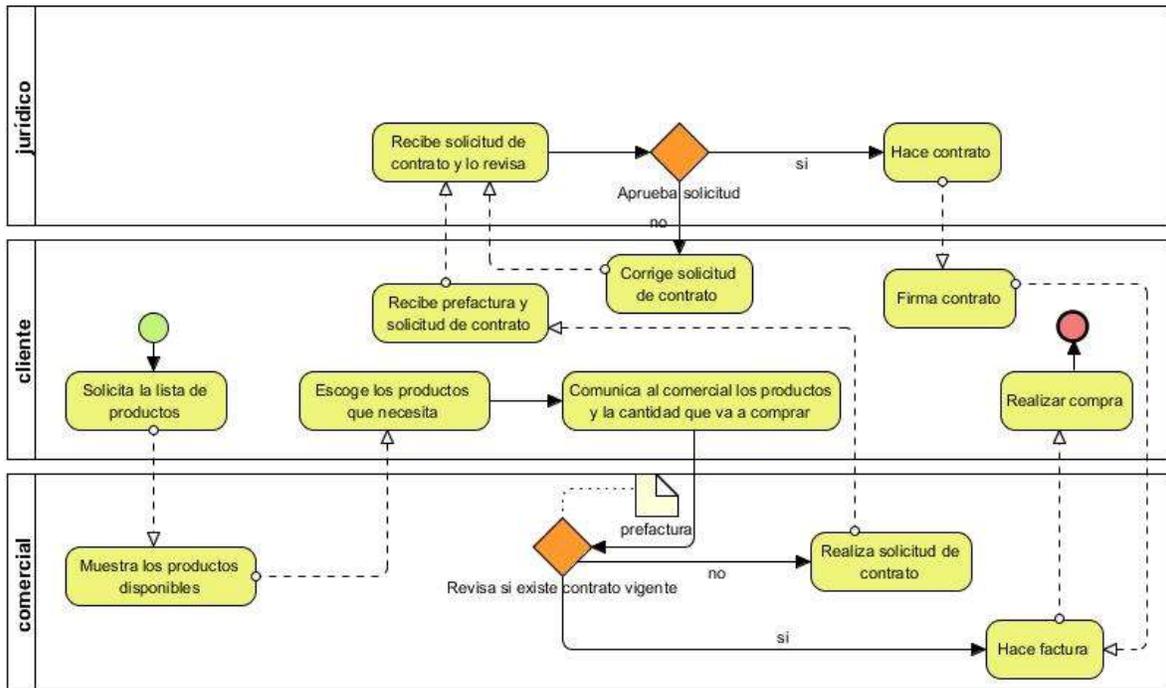
- Machín Díaz, O. (2019). *Procedimiento para la implantación del comercio electrónico en la empresa estatal cubana. Caso Tienda Zona+ Cárdenas* Universidad de Matanzas]. Matanzas.
- Mahalakshmi, M., & Sundararajan, M. (2015). *Tracking the student's performance in Web-based education using Scrum methodology* 015 International Conference on Computing and Communications Technologies (ICCCT),
- Matharu, G. S., Mishra, A., Singh, H., & Upadhyay, P. (2015). Empirical Study of Agile Software Development Methodologies: A Comparative Analysis. *ACM SIGSOFT Software Engineering Notes*, 40(1). <https://doi.org/10.1145/2693208.2693233>
- Mead, N. R., Viswanathan, V., Padmanabhan, D., & Padmanabhan, D. (2008). *Incorporating Security Requirements Engineering into the Dynamic Systems Development Method* Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International,
- Menzinsky, A., López, G., & Palacio, J. (2019). Scrum Manager: Temario Troncal I. In S. Master (Ed.), (Vol. 2.6.1): Iubaris Info 4 Media SL.
- Moniruzzaman, A. B. M., & Hossain, S. A. (2013). Comparative Study on Agile Software Development Methodologies. *Global Journal of Computer Science and Technology*, XIII( VII).  
[https://www.researchgate.net/publication/249011841\\_Comparative\\_Study\\_on\\_Agile\\_software\\_development\\_methodologies?enrichId=rgreq-422ebd559a5b7493daffe6203d904ef3-XXX&enrichSource=Y292ZXJQYWdlOzI0OTAxMTg0MTtBUzo5Nzc1Mjc5Mjk2MTAyNkAxNDAwMzE3NDkwNjI0&el=1\\_x\\_2&esc=publicationCoverPdf](https://www.researchgate.net/publication/249011841_Comparative_Study_on_Agile_software_development_methodologies?enrichId=rgreq-422ebd559a5b7493daffe6203d904ef3-XXX&enrichSource=Y292ZXJQYWdlOzI0OTAxMTg0MTtBUzo5Nzc1Mjc5Mjk2MTAyNkAxNDAwMzE3NDkwNjI0&el=1_x_2&esc=publicationCoverPdf)
- Murillo, N. (2022). *NestJS: qué es y por qué empezar a usarlo*. gfourmis. <https://gfourmis.co/nestjs-que-es-y-por-que-empezar-a-usarlo/>
- Navarro Cadavid, A., Fernández Martínez, J. D., & Morales Vélez, J. (2013). Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA. Universidad Autónoma del Caribe*, 11, 30-39. <http://www.redalyc.org/articulo.oa?id=496250736004>
- Overhage, S., & Schlauderer, S. (2012). *Investigating the Long-Term Acceptance of Agile Methodologies : An Empirical Study of Developer Perceptions in Scrum Projects* Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS). Wailea, HI, USA 2012, Wailea, HI, USA <https://fis.uni-bamberg.de/handle/uniba/39731>
- Pawar, R. P. (2015). *A Comparative study of Agile Software Development Methodology and traditional waterfall model*. <https://www.iosrjournals.org/iosr-ice/papers/NCIEST/Volume%202/01.01-08.pdf>
- Pressman, R. S. (2010). *Ingeniería del software: Un enfoque práctico* (Séptima edición ed.). McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Priyanka, & Kantha, P. (2016). *A Comprehensive Study of Traditional and AGILE Software Development Methodologies* International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE),
- Rendón Matienzo, F. (2021). Destacan potencial del comercio electrónico para desarrollar encadenamientos productivos. *Agencia Cubana de Noticias*.
- Reyes Chirino, R., & Delgado Fernández, T. (2021). Modelo funcional de un laboratorio ciudadano de innovación digital. *UNIVERSIDAD Y SOCIEDAD | Revista Científica de la Universidad de Cienfuegos*, 13(1), 177-188. <http://scielo.sld.cu/pdf/rus/v13n1/2218-3620-rus-13-01-177.pdf>
- Romano, B. L., & da Silva, A. D. (2015). *Project Management Using the Scrum Agile Method: A Case Study within a Small Enterprise* 2015 12th International Conference on Information Technology - New Generations (ITNG),

- [https://www.researchgate.net/publication/301405365\\_Project\\_Management\\_Using\\_the\\_Scrum\\_Agile\\_Method\\_A\\_Case\\_Study\\_within\\_a\\_Small\\_Enterprise](https://www.researchgate.net/publication/301405365_Project_Management_Using_the_Scrum_Agile_Method_A_Case_Study_within_a_Small_Enterprise)
- Rosado Gómez, A., Quintero Duarte, A., & Meneses Guevara, C. D. (2012). Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio*, 5(1), 24-29.  
<https://doi.org/https://doi.org/10.22463/2011642X.2003>
- Sabariego, R. M. (2013). *Diseño e implementación de portales de comercio electrónico. Mejoras basadas en Soft Computing* Universidad de Castilla-La Mancha]. España.
- Sommerville, I. (2011). *Ingeniería de Software*. Pearson Educación de México, S.A. de C.V.  
[https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9\\_compressed.pdf](https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf)
- Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). *SCRUM model for agile methodology* 2017 International Conference on Computing, Communication and Automation (ICCCA),
- Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional *Informatica Economică*, 14, 64-76. <https://doi.org/10.12948/issn14531305/17.4.2013.06>
- Suardiyana Putra, I. P. E., Yuliawati, A., & Mursanto, P. (2012). Industrial extreme programming practice's implementation in rational unified process on agile development theme. Proceedings of the 2012 International Conference on Advanced Computer Science and Information Systems, Piscataway, NJ.
- Suominen, K. (2019). El comercio digital en América Latina: ¿Qué desafíos enfrentan las empresas y cómo superarlos? *serie Comercio Internacional*, N° 145 (LC/TS.2019/76). Retrieved 21/11/2022, from <https://americasmi.com/insights/proyecciones-sobre-el-comercio-electronico-en-latinoamerica-2018-2022/>
- Turban, E., Lee, K. J., King, D., McKay, J., & Marshall, P. (2008). *Electronic Commerce: A managerial Perspective* (5th ed.). Prentice-Hall.
- UNCTAD. (2021). *El comercio electrónico mundial alcanza los 26,7 billones de dólares mientras COVID-19 impulsa las ventas en línea*. <https://unctad.org/es/news/el-comercio-electronico-mundial-alcanza-los-267-billones-de-dolares-mientras-covid-19-impulsa>
- Varela García, L. (2018). *Aplicación web en apoyo a la gestión de la información del Colectivo de Carrera de la Universidad de Matanzas* Universidad de Matanzas].
- Xu, B. (2009). Towards high quality software development with extreme programming methodology: Practices from real software projects. Proceedings - International Conference on Management and Service Science, MASS 2009,
- Yong, Y., & Zhou, B. (2009). *Evaluating Extreme Programming Effect through System Dynamics Modeling* 2009 International Conference on Computational Intelligence and Software Engineering, <https://www.semanticscholar.org/paper/Evaluating-Extreme-Programming-Effect-through-Yong-Zhou/2dd7f5a859d367f5377b247afe1943fa14d24d53>

## Anexos

Anexo 1. Flujo del proceso de comercialización.

Fuente: Aicros.

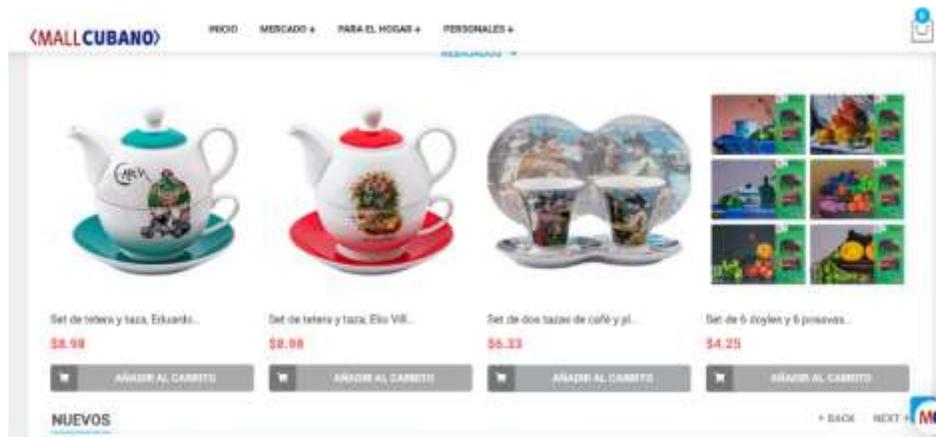


**Anexo 2.** Plataformas de comercio electrónicos existentes. **Fuente:** Elaboración Propia

a) Tienda virtual GEDEME.



b) MallCubano



c) Superfácil



**Anexo 3.** Diferencias entre las metodologías ágiles y las metodologías tradicionales de desarrollo.

**Fuente:** Quiñones-Ku et al. (2019)

<b>Característica</b>	<b>Metodologías tradicionales</b>	<b>Metodologías ágiles</b>
<b>Hipótesis fundamental</b>	Los sistemas son totalmente especificables, predecibles y se desarrollan a través de una planificación detallada y extendida.	El software adaptativo de alta calidad es desarrollado por pequeños equipos que utilizan el principio de mejora continua del diseño y las pruebas basadas en una rápida respuesta y cambio.
<b>Estilo de gestión</b>	Comando y control.	Liderazgo y colaboración.
<b>Conocimiento administrativo</b>	Explícito.	Tácito.
<b>Comunicación</b>	Formal.	Informal.
<b>Modelo de desarrollo</b>	Modelo de ciclo de vida (cascada, espiral o modelos modificados).	Modelo evolutivo de entrega.
<b>Estructura organizacional</b>	Mecánico (burocrático, alta formalización), dirigido a grandes organizaciones.	Orgánico (flexible y participativo, fomenta la cooperación social), dirigido a pequeñas y medianas organizaciones.
<b>Control de calidad</b>	Planificación difícil y control estricto. Pruebas difíciles y tardías.	Control permanente de requisitos, diseño y soluciones. Pruebas permanentes.
<b>Requisitos de usuario</b>	Detallado y definido antes de la codificación / implementación.	Entrada interactiva.
<b>Costo de reinicio</b>	Alto.	Bajo.
<b>Dirección de desarrollo</b>	Fijo.	Fácilmente cambiable.
<b>Pruebas</b>	Después de completar la codificación.	Cada iteración.
<b>Participación del cliente</b>	Bajo.	Alto.
<b>Habilidades adicionales requeridas de los desarrolladores</b>	Nada en particular.	Habilidades interpersonales y conocimientos básicos del negocio.
<b>Escala apropiada del proyecto</b>	Gran escala.	Baja y mediana escala.
<b>Desarrolladores</b>	Planificado, con capacidades adecuadas, acceso a conocimientos externos.	Ágil, con conocimientos avanzados y cooperativos.
<b>Clientes</b>	Con acceso al conocimiento, cooperativo, representativo y empoderado.	Con acceso al conocimiento, cooperativo, representativo y empoderado.
<b>Requerimientos</b>	Muy estable, conocido de antemano.	Emergente, con cambios rápidos.
<b>Arquitectura</b>	Diseño para requerimientos actuales y previsibles.	Diseño para requerimientos actuales.
<b>Remodelación</b>	Costoso.	No es caro.
<b>Tamaño</b>	Grandes equipos y proyectos.	Pequeños equipos y proyectos.
<b>Objetivos principales</b>	Alta seguridad.	Valor rápido.

**Anexo 4.** Comparación de metodologías ágiles. **Fuente:** Moniruzzaman & Hossain (2013)

<b>Características</b>	<b>XP</b>	<b>SCRUM</b>	<b>DSDM</b>	<b>FDD</b>	<b>ASD</b>
<b>Enfoque</b>	Iterativo, incremental	Iterativo, incremental	Iterativo	Iterativo	Iterativo, incremental
<b>Periodo del ciclo de iteración</b>	1 – 6 semanas	2 – 4 semanas	En 20% por ciento del tiempo total 80% de producto.	2 días - 2 semanas	4 – 8 semanas
<b>Tamaño adecuado del proyecto y complejidad</b>	Proyecto pequeño y sencillo	Para proyectos grandes y complejos	Todo tipo de proyectos	Proyectos a gran escala	Proyectos más pequeños y sencillos
<b>Involucramiento del usuario</b>	Activamente involucrado	A través del propietario del producto	A través de lanzamientos frecuentes	A través de informes	A través de lanzamientos frecuentes
<b>Documentación</b>	Documentación básica	Documentación básica	Más que XP y SCRUM	Más alto entre todos	Documentación básica
<b>Principales prácticas</b>	Simplicidad, Programación en pares.	Reuniones de SCRUM	Time boxing, MoSCoW, Prototipado	Modelado de objetos, desarrollo por característica, uso del diagrama UML	Time boxing, Risk Driven, basado en características
<b>Desarrollo de características concurrentes</b>	Posible	Posible	Posible	Posible	Posible

Anexo 5. Fases de la metodología XP.

Fuente: Kent & Andres (2004).



**Anexo 6** Tabla de ponderaciones para EI, EQ, EO, ILF, EIF. **Fuente:** Elaboración Propia.

<b>Clasificación de entradas y consultas</b>	<b>1 – 4 Atributos</b>	<b>5 – 15 Atributos</b>	<b>Mas de 15 Atributos</b>
0 o 1 ficheros accedidos	BAJA 3	BAJA 3	MEDIA 4
2 ficheros accedidos	BAJA 3	MEDIA 4	ALTA 6
Más de 2 ficheros accedidos	MEDIA 4	ALTA 6	ALTA 6

<b>Clasificación de entradas y consultas</b>	<b>1 – 5 Atributos</b>	<b>5 – 19 Atributos</b>	<b>Mas de 19 Atributos</b>
0 o 1 ficheros accedidos	BAJA 4	BAJA 4	MEDIA 5
2 a 3 ficheros accedidos	BAJA 4	MEDIA 5	ALTA 7
Más de 5 Entidades o registros lógicos	MEDIA 5	ALTA 7	ALTA 7

<b>Clasificación de entradas y consultas</b>	<b>1 – 19 Atributos</b>	<b>20 – 50 Atributos</b>	<b>Mas de 50 Atributos</b>
1 entidad o registro lógico	BAJA 7	BAJA 7	MEDIA 10
2 - 5 Entidades o registros lógicos	BAJA 7	MEDIA 10	ALTA 15
Más de 2 ficheros accedidos	MEDIA 10	ALTA 15	ALTA 15

<b>Clasificación de entradas y consultas</b>	<b>1 – 4 Atributos</b>	<b>5 – 15 Atributos</b>	<b>Mas de 15 Atributos</b>
0 o 1 ficheros accedidos	BAJA 5	BAJA 5	MEDIA 7
2 ficheros accedidos	BAJA 5	MEDIA 7	ALTA 10
Más de 2 ficheros accedidos	MEDIA 7	ALTA 10	ALTA 10

**Anexo 7** Ajuste de Complejidad Técnica (ACT).**Fuente:** Elaboración Propia.

<b>Nº</b>	<b>Nombre del Factor</b>	<b>Valor</b>
<b>1</b>	Comunicación de datos	3
<b>2</b>	Procesamiento distribuido	1
<b>3</b>	Rendimiento	4
<b>4</b>	Configuración de Explotación Compartida	1
<b>5</b>	Tasa de transacciones	1
<b>6</b>	Entrada de Datos en Línea	4
<b>7</b>	Eficiencia con el Usuario Final	4
<b>8</b>	Actualizaciones en Línea	1
<b>9</b>	Procesamiento complejo	1
<b>10</b>	Reusabilidad del Código	1
<b>11</b>	Facilidad de implementación	1
<b>12</b>	Facilidad de Operación	4
<b>13</b>	Instalaciones Múltiples	2
<b>14</b>	Facilidad de Cambios	3
	<b>Ajuste de Complejidad Técnica (ACT)</b>	<b>31</b>