

Universidad de Matanzas

Facultad de Ciencias Técnicas

Departamento de Informática



TÍTULO: Aplicación Web para la implementación del Comercio Electrónico en la Empresa DIVEP Matanzas

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO INFORMÁTICO

AUTOR: Ing. José Renier Martínez Serpa

TUTOR: MsC. Jorge Ilich Noda

**Ciudad de Matanzas
2020**

Frase

“Todo ser humano, si se lo propone, puede ser escultor de su propio cerebro”

Santiago Ramón y Cajal



Dedicatoria

A mi hija por ser la mayor motivación e inspiración para alcanzar las metas del presente y futuro.

A mis padres por haberme formado y conducido por el camino correcto, por la entrega sin límites de apoyo y amor.

A mis hermanos por la comprensión y ayuda brindada en cada momento que necesitaba de ellos para cumplir con esta y otras labores.

Agradecimientos

Agradezco a toda la familia por la comprensión, apoyo y amor brindado durante esta etapa de estudios. A mis tutores, amigos y compañeros de trabajo que fueron soporte y guía en este viaje a través del conocimiento.

Gracias a todos los profesores que incidieron en mi formación y a mis compañeros de grupo por compartir juntos cada jornada de aprendizaje, momentos tensos y de alegría, por la empatía y solidaridad existente entre nosotros.

Yo, José Renier Martínez Serpa, por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas y a la Empresa UEB DIVEP Matanzas, a que hagan el uso que estimen pertinente de él.

Y para que así conste, firmo la presente a los 14 días del mes de julio del año 2020.

Firma del Autor

José Renier Martínez Serpa

Firma del Tutor



Resumen

El progreso de las tecnologías de las informaciones y las comunicaciones en el país han aumentado el espectro de posibilidades para realizar, de forma más eficiente, labores de gran incidencia en la economía de la nación, haciendo uso de las modernas herramientas que estas proveen. La actual investigación trata el uso de algunas de estas ventajas, al desarrollar una aplicación web para la implementación del comercio electrónico en la empresa Unidad Empresarial de Base Dirección Integral de Ventas y Piezas Matanzas, y con ello lograr una mayor eficiencia en la comercialización del inventario que posee la entidad. El software es responsable de divulgar la cartera de productos, proporcionar datos acerca de los mismos y permitir a aquellos clientes contractuales realizar la reserva de estos. Para alcanzar tales objetivos se emplean métodos científicos de obtención de información teóricos y empíricos. El cumplimiento del propósito y calidad de la herramienta son probados frecuentemente como la metodología específica, obteniéndose resultados satisfactorios. La empresa dispone de un sistema que ofrece a los clientes, además de productos, independencia, confort y seguridad, y una manera más eficiente de desempeñar la actividad comercial.

Abstract

The progress of information and communication technologies in the country have increased the spectrum of possibilities to carry out, in a more efficient way, tasks of great impact on the nation's economy, making use of the modern tools that they provide. The current research deals with the use of some of these advantages, when developing a web application for the implementation of electronic commerce in the company Unidad Empresarial de Base Dirección Integral de Ventas y Piezas Matanzas, and thereby achieve greater efficiency in the marketing of inventory that owns the entity. The software is responsible for disseminating the product portfolio, providing data about them and allowing those contractual customers to book them. To achieve these objectives, scientific methods of obtaining theoretical and empirical information are used. The fulfillment of the purpose and quality of the tool are frequently tested as the specific methodology, obtaining satisfactory results. The company has a system that offers customers, in addition to products, independence, comfort and security, and a more efficient way of carrying out the commercial activity.

Contenido

Introducción 1

Capítulo I Marco teórico referencial..... 6

1.1 Introducción..... 6

1.2 Flujo actual de trabajo. 6

1.3 Análisis crítico de cómo se ejecutan los procesos. 7

1.4 Antecedentes del trabajo. 7

1.5 Métodos de la investigación..... 8

1.6 Tendencias tecnológicas. 9

1.7 Herramientas, tecnologías y metodologías de desarrollo. 9

1.7.1 Software libre. 10

1.7.2 Arquitectura Cliente – Servidor..... 10

1.7.3 Arquitectura basada en N-capas. 11

1.7.4 Patrones de diseño..... 12

1.7.5 Metodología de desarrollo de software. 13

1.7.6 Aplicaciones Web. 18

1.7.7 Herramientas y tecnologías. 19

1.8 Conclusiones parciales del capítulo..... 31

Capítulo II Análisis, diseño y construcción de la solución propuesta..... 32

2.1 Introducción..... 32

2.2 Modelo de Proceso. 32

2.2.1 Diagrama de Procesos de Negocio (BPD). 32

2.2.2 Descripción del Negocio..... 32

2.2.3 Modelo Conceptual de la Base de Datos..... 34

2.3 Descripción de la solución. 35

2.4 Etapa de Planificación. 36

2.4.1 Equipo de Trabajo..... 36

2.4.2 Pila del Producto (Product Backlog). 37

2.4.3 Requisitos No Funcionales. 38

2.4.4 Historias de usuarios..... 40

2.4.5 Plan de Sprint..... 41

2.4.6 Plan de Entrega. 42

2.4.7 Resumen de Tareas generadas por Historias de Usuario. 42

2.5 Análisis de Costo. 43

2.6 Conclusiones parciales del capítulo.....	47
Capítulo III Resultados obtenidos y pruebas realizadas al software.....	48
3.1 Introducción.....	48
3.2 Pruebas realizadas y resultados.....	48
3.2.1 Pruebas de aceptación.....	48
3.3 Conclusiones parciales del capítulo.....	53
Conclusiones Generales.....	55
Recomendaciones.....	56
Bibliografía.....	57
Anexos.....	59
Anexo 1: Diagrama de funcionamiento de la metodología SCRUM (Mariño & Alfonso, 2014).	59
Anexo 2: Historias de usuarios	60
Anexo 3: Tabla de ponderaciones para EI, EQ, EO, ILF, EIF	64
Anexo 4: Tabla para obtener el Factor de Ajuste de Valor (FAC).	65
Anexo 5: Tabla para obtener Horas por Puntos de Función (HPF).....	66

Índice de Figuras

Ilustración 1 Diagrama BPD "Oferta y Reserva de Productos - Seguimiento de Reserva de Productos" (elaboración propia).....34

Ilustración 2 Diagrama BPD "Seguimiento de Lista de Deseos" (elaboración propia).....34

Ilustración 3 Modelo Conceptual de la Base de Datos (elaboración propia).....35



Índice de Tablas

Tabla 1 Cuadro comparativo metodologías ágiles (ALVAREZ CASADIEGO, 2012). 15

Tabla 2 Soporte a CSS de navegadores actuales (Eguíluz Pérez, 2008). 21

Tabla 3 Definición de los roles del equipo SCRUM (elaboración propia). 37

Tabla 4 Pila del Producto o Product Backlog (elaboración propia). 38

Tabla 5 Historia de usuario 01 (elaboración propia). 41

Tabla 6 Historia de usuario 02 (elaboración propia). 41

Tabla 7 Historia de usuario 05 (elaboración propia). 41

Tabla 8 Plan de Sprint (elaboración propia). 42

Tabla 9 Tareas de Ingeniería (elaboración propia). 43

Tabla 10 Cálculos de puntos de función (Pressman, 2010). 45

Tabla 11 Cálculo del Factor de Ajuste de Valor (Pressman, 2010). 46

Tabla 12 Cálculo de Tiempo Laborable (elaboración propia). 46

Tabla 13 PA1 Seguridad y autenticación de usuario (elaboración propia). 50

Tabla 14 PA2 Gestión de contrato (elaboración propia). 51

Tabla 15 PA3 Gestión de nomencladores (elaboración propia). 52

Tabla 16 PA4 Gestión de Compra (elaboración propia). 53

Introducción

En la actualidad el empleo de las Tecnologías de la Información y las Comunicaciones (TICs) en las organizaciones es imprescindible para alcanzar objetivos y metas imprescindibles para el desarrollo. A través del empleo de las redes locales, nacionales y globales, han surgido nuevas formas de gestionar las distintas organizaciones de los diferentes sectores de la economía, la salud, la educación, la seguridad, entre otros, que hacen uso de la información y las comunicaciones en su desempeño diario. Las naciones necesitan cada vez más de estos adelantos para realizar una eficiente gestión de sus recursos y con ello obtener mejores resultados económico y social.

Cuba no se encuentra ajeno a esta realidad y ha decidido emplear con determinación la explotación de dichas tecnologías en aras de mejorar su economía, ampliar su desarrollo y beneficiar al pueblo mediante el disfrute de todas sus bondades. El país está llamado a revolucionar desde las diferentes formas de gestión todos los procesos que sean necesarios para contribuir a la informatización de la sociedad. Ejemplo de ello ha sido la reciente creación de softwares que hacen uso de diversas plataformas para la realización de pagos de servicios como la electricidad, el teléfono o el suministro de gas licuado. El desarrollo de software va transformando positivamente la vida de los ciudadanos y las organizaciones. Muchas de las aplicaciones que mayor acogida ha tenido son las basadas en el comercio electrónico, que permiten a privados y estatales comercializar productos y servicios de una manera más eficiente.

Esta forma novedosa de comercializar proporciona diversas ventajas, entre las cuales se pueden citar el contar con un espacio ilimitado para la exposición de productos, la presentación de la mercancía de manera atractiva y organizada, y el conocimiento oportuno del comportamiento del inventario, la demanda y los ingresos obtenidos.

Existe un número considerable de empresas en el país que no han percibido aun 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 entidad Unidad Empresarial de Base Dirección Integral de Ventas y Piezas Matanzas (UEB DIVEP Matanzas), surge en el año 2012. La misma forma parte del Grupo Empresarial de la Industria Sidero Mecánica (GESIME) del Ministerio de Industria, la cual es considerada como la organización comercializadora por excelencia de GESIME en la provincia. Su objeto empresarial se enmarca en la comercialización de partes, piezas, componentes, agregados y accesorios de equipos asociados al sector automotor, industrial y bienes de consumo, así como las producciones de las empresas de las Industrias Sidero Mecánica y las generadas por la industria nacional que demanden la capacidad de distribución de su red de ventas en pesos cubanos y pesos convertibles.

La entidad en el presente dispone de espacios muy limitados para exponer 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 al 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.

Producto a lo anterior, la organización espera ser más eficiente en el desempeño de su principal función, la comercialización. Explotando las bondades que proporciona el comercio electrónico, pretende proporcionar a los clientes un servicio de ventas con mejores prestaciones y a su vez, obtener beneficios en recursos que pasan de ser solo el financiero, como son un mejor aprovechamiento del tiempo y la fuerza de trabajo. A partir de esta **situación problemática** se define como **problema de la investigación**: ¿Cómo contribuir desde el desarrollo de una aplicación web para el comercio electrónico a la mejora de la eficiencia de la oferta y la reserva de la cartera de productos de la Empresa UEB DIVEP Matanzas?, teniendo como **objeto de estudio** la implementación del comercio electrónico en la gestión comercial en empresas comercializadoras y siendo el **campo de acción** la implementación del comercio electrónico para mejorar la eficiencia de la oferta y la reserva de la cartera de productos que posee la Empresa UEB DIVEP Matanzas.

A partir de lo anteriormente tratado se determina el siguiente planteamiento como hipótesis:

De desarrollarse una aplicación web que implementase el comercio electrónico en la UEB DIVEP Matanzas se logra mejorar la eficiencia de la oferta y la reserva de la cartera de productos que sustenta la actividad comercial de la empresa.

En la hipótesis planteada se identifican las siguientes variables:

Variable independiente: la aplicación web para la implementación del comercio electrónico en la UEB DIVEP Matanzas.

Variable dependiente: la oferta y reserva de la cartera de productos que comercializa la UEB DIVEP Matanzas.

Se plantea como **objetivo general** desarrollar una aplicación web como plataforma para el comercio electrónico, que contribuya a la mejora de la eficiencia de la oferta y la reserva de la cartera de productos que posee la Empresa UEB DIVEP Matanzas. Para lograr el cumplimiento del objetivo general se declaran los siguientes **objetivos específicos**:

- Seleccionar los métodos y técnicas que permitan obtener los fundamentos teóricos y la comprensión necesaria de los distintos elementos del tema que aborda la investigación.
- 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.
- Utilizar la metodología y herramientas concebidas para el desarrollo de la aplicación y obtener un software con la calidad esperada.
- Probar el funcionamiento del software desarrollado para así conocer la calidad y el grado de cumplimiento de los objetivos del proyecto.

Para la elaboración del marco teórico se emplearon distintos métodos y técnicas en los que figura:

- ✓ 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.

Para el desarrollo de la aplicación se utilizó el marco de trabajo SCRUM:

Se define a SCRUM, como una colección de procesos para la gestión de proyectos, que permite centrarse en la entrega de valor para el cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua (Mariño & Alfonzo, 2014).

Entre sus características fundamentales se encuentran la inspección y adaptación como práctica sistemática, la vinculación de la gente del negocio con los equipos de desarrollo y la autoorganización que rige en tales equipos. Dentro de sus principales beneficios se encuentran la obtención temprana y continua de software funcional.

En correspondencia con lo anterior tratado, 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, los antecedentes y las tecnologías utilizadas. Son abordadas las principales características 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: Análisis, diseño y construcción de la solución propuesta, 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 SCRUM. Es analizado y estudiado los beneficios obtenidos mediante la implementación del proyecto de software. Se realiza la construcción de la aplicación web siguiendo las pautas de desarrollo como lo concibe la metodología, a través de iteraciones planificadas.

Capítulo III: Resultados obtenidos y pruebas realizadas al software, 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 requisitos.

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 I Marco teórico referencial

1.1 Introducción.

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.

1.2 Flujo actual de trabajo.

La comercialización de la cartera de productos con que cuenta la empresa se realiza actualmente a través de la pequeña red de tiendas que posee la entidad en la provincia, específicamente tres tiendas, dos de ellas ubicadas en el municipio Cárdenas y otra en el municipio Matanzas. Los espacios de exposición de la mercancía para mostrar a los clientes que tienen estas tiendas son de alrededor de los 50 metros cuadrados. En las unidades se encuentran desempeñando la actividad comercial un total de 4 trabajadores los cuales cumplen con los roles siguientes: Administrador, Económica, Dependiente Piso Venta y Dependiente Almacén.

El Cliente una vez que arriba a una Unidad Comercial, es recibido por el Administrador o Dependiente Piso Venta, los cuales establecen comunicación con el visitante con el objetivo de conocer sus intenciones de compras y poder ayudarlo a cumplir con su propósito de ser posible u ofrecerle otras alternativas que contribuyan a la satisfacción del mismo. Debido a que no toda la mercancía puede encontrarse representada en el área de exhibición, es normal que se pregunte por productos que no se encuentran representados o son inexistentes. Es frecuente que se realice tiempos de esperas prolongados para ser atendido mientras los trabajadores antes mencionados se encuentran ocupados con clientes que se presentaron en la instalación con anticipación respecto a los que lo hicieron posteriormente a ellos.

La atención a un cliente abarca desde sostener comunicación con este hasta la realización de cotizaciones, reservaciones y facturas si es requerido. Estos procedimientos suman considerables tiempos de espera para quienes no están siendo atendidos.

En no pocas ocasiones es solicitado la presencia del Dependiente Almacén ante el Cliente para que muestre algunos de los productos sin representación y en casos específicos de productos con cierta complejidad técnica o tecnológica, debe ser presentado su envase y/o documentación que aborda las características y funcionalidades de estos, todo ello con el objetivo de tener seguridad de realizar la selección correcta de la mercancía que se necesita adquirir.

1.3 Análisis crítico de cómo se ejecutan los procesos.

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.

1.4 Antecedentes del trabajo.

Anterior a la elaboración de este trabajo se realizó la investigación necesaria para conocer la existencia de softwares similares, al propuesto desarrollar, que puedan satisfacer las necesidades identificadas en la actividad comercial que desempeña la entidad objeto de estudio. Aunque en la búsqueda se halló 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 UEB DIVEP Matanzas 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.

1.5 Métodos de la investigación.

Para lograr los objetivos del presente trabajo, resulta imprescindible la realización de la investigación adoptando diferentes métodos empíricos y teóricos que posibilitan la obtención de la información necesaria para el desarrollo de dicha labor.

Los métodos teóricos fundamentales empleados fueron:

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

Los métodos empíricos utilizados se manifiestan mediante el uso de las siguientes técnicas:

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

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

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

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

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

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

Análisis de documentos fue empleado como la forma más robusta y precisa de obtención de información y concepción de ideas durante la investigación.

1.6 Tendencias tecnológicas.

Hoy en día existe una tendencia creciente en el empleo de metodologías ágiles para el desarrollo de softwares, así como la utilización de lenguajes informáticos y herramientas pertenecientes a las clasificadas como de softwares libres, las mismas son utilizadas para la implementación de todo tipo de proyectos, sobresaliendo de ellos los destinados a la realización de aplicaciones web. Tal tendencia está basada en la libertad de explotación de esos softwares, la constitución robusta que presentan, seguridad que ofrecen y los beneficios económicos que derivan de su uso con absoluta independencia.

1.7 Herramientas, tecnologías y metodologías de desarrollo.

En este epígrafe se describen de manera general las diferentes herramientas, tecnologías y metodologías seleccionadas para el desarrollo de la aplicación web propuesta para dar solución a la situación problemática tratada. Se ha optado por el uso de softwares actuales, seguros y con alto índice de usabilidad en el presente. También se ha decidido emplear metodologías que son concebidas para asumir proyectos con las características del que se aborda en este trabajo. El uso de los medios antes mencionado, tiene como objetivo garantizar el aseguramiento del ambiente productivo para la construcción de un software eficiente que satisfaga los requerimientos del producto informático demandado.

1.7.1 Software libre.

El software libre es aquel que no oculta su código, permanece libre y permite la libre distribución del mismo. Su concepción refleja que el conocimiento contenido en él pertenece a toda la humanidad. La aplicación de ello posibilita que países e individuos, sobre todo los más pobres, tengan acceso sin limitaciones a tecnologías desarrolladas muchas veces por otros con mayores recursos.

Según (Stallman, 2004) a veces es malinterpretado el termino de software libre, pensando que guarda relación con el precio, y no es así, lo que interesa es la libertad. Él refleja que un programa es software libre para el usuario siempre que, como usuario particular, tenga:

1. La libertad de ejecutar el programa sea cual sea el propósito.
2. La libertad para modificar el programa para ajustarlo a tus necesidades. (Para que se trate de una libertad efectiva en la práctica, deberás tener acceso al código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil.)
3. La libertad de redistribuir copias, ya sea de forma gratuita, ya sea a cambio del pago de un precio.
4. La libertad de distribuir versiones modificadas del programa, de tal forma que la comunidad pueda aprovechar las mejora introducidas.

Para poder ejercer las cuatro libertades es preciso recibir el código fuente (algo así como los planos del programa), que permite hacer modificaciones de forma cómoda. De esta característica del software libre viene su otro nombre en inglés, “open source software”. En español, es común verlo traducido como “software de fuentes abiertas” o “código abierto” (González-Barahona, 2011).

1.7.2 Arquitectura Cliente – Servidor.

El modelo Cliente/Servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Las aplicaciones Clientes realizan peticiones a una o varias

aplicaciones Servidores, que deben encontrarse en ejecución para atender dichas demandas (Marini, 2012).

Otra manera de interpretar esta arquitectura es verla como una forma de dividir y especializar programas y equipos para que la función que cada uno de ellos realiza sea con la mayor eficiencia posible y posibilite la simplificación de las actualizaciones y mantenimiento del sistema que la implementa.

(Lizama, Kindley, & Jeria Morales, 2016) refleja que las ventajas de este modelo respecto a de otras posibles arquitecturas de red serían:

- Centralizado.
- Es de fácil mantención.
- Escalabilidad: se puede aumentar la capacidad de clientes y servidores por separado.

1.7.3 Arquitectura basada en N-capas.

La arquitectura basada en capas se enfoca principalmente en el agrupamiento de funcionalidad relacionada dentro de una aplicación en distintas capas que son colocadas verticalmente una encima de otra, la funcionalidad dentro de cada capa se relaciona con un rol o responsabilidad específica (Muñoz Serafín, 2018).

En este estilo de programación se destaca el utilizado para el desarrollo de software a tres capas. Esta técnica es frecuentemente empleada en la implementación de sistemas que contengan conexiones a base de datos. Esas capas son conocidas de manera individual como capa de presentación, capa de lógica y capa de acceso a datos. En la parte superior se encuentra la capa de presentación, la misma tiene la responsabilidad de interactuar con el usuario, gestionando la información que este solicita o requiere. En el nivel intermedio está la capa de lógica cuya responsabilidad es hacer cumplir los requerimientos del sistema o reglas del negocio. Y en la posición más baja se haya la capa de accesos a datos que realiza la función de recuperar, modificar datos del origen de datos.

Los beneficios de implementar este tipo de arquitectura no son pocos, entre ellos destacan:

- Reutilización de componentes.
- Reducción de costos de mantenimiento.
- Interoperabilidad entre componentes de distintos fabricantes.
- Fácil escalabilidad.

1.7.4 Patrones de diseño.

Los patrones de diseño describen un problema que se presenta continuamente en un contexto y comunica una solución para ello. Ellos son fruto del conocimiento recolectado a través de las prácticas de expertos. En el caso de los softwares, estos suministran un esquema o esqueleto para la refinación de subsistemas o componentes, o las relaciones existentes entre ellos. Brindan una estructura común y recurrente de comunicación entre componentes que da solución a un problema general de diseño de un entorno específico.

Refleja (Cáceres Tello) que dentro de los objetivos de los patrones de diseño está ofrecer:

- Reducción de tiempos.
- Disminución del esfuerzo de mantenimiento.
- Aumentar la eficiencia.
- Asegurar la consistencia.
- Aumenta la fiabilidad

1.7.4.1 Patrón de diseño: Modelo – Vista – Controlador (MVC).

Es un patrón de arquitectura de software que asume a una aplicación dividida en tres aspectos: procesamiento, salida y entrada. Para cumplir esto emplea las siguientes abstracciones:

- **Modelo:** es el encargado de encapsular los datos y las funcionalidades que le competen. Se encuentra exento de representar alguna salida o comportamiento de entrada de información.
- **Vista:** tiene la responsabilidad de mostrar la información al usuario, y para ello tiene asociado un controlador el cual hace uso del modelo para obtener los datos requeridos.

- Controlador: se encarga de recibir las peticiones y se encarga de traducirlas a solicitudes de servicio para el modelo o la vista.

(Camarena Sagredo, Trueba Espinosa, Martínez Reyes, López, & García, 2012) plantea que se perciben las siguientes ventajas al hacer uso del patrón MVC:

- a) Permitir la sustitución de las interfaces de usuario.
- b) Generar componentes de las interfaces.
- c) Diseñar vistas simultáneas del mismo modelo.
- d) Aplicar fácilmente cambios de las interfaces.

1.7.5 Metodología de desarrollo de software.

(Espinoza-Meza, 2013) describe a las metodologías de desarrollo como actividades, procedimientos, técnicas, herramientas y documentos, en su conjunto, normados y comprendidos en un marco de trabajo. Funcionan de soporte en la estructuración, planificación y control requeridos para lograr la conversión de una necesidad o un grupo de necesidades a un sistema de información de manera eficiente.

Las metodologías nos indican un plan adecuado de gestión y control del proyecto de software: definición de etapas, ingresos y salidas, restricciones, comunicaciones, tareas ordenadas y distribución de recursos.

En el presente las metodologías de desarrollo se dividen en dos grandes grupos, las metodologías tradicionales y las metodologías ágiles; las primeras se enfocan en el uso íntegro de la documentación, el cual abarca todo el ciclo de vida del proyecto, mientras que las segundas confieren mayor enfoque a la colaboración constante entre los individuos y respuestas a los cambios. Las características principales que poseen tales metodologías son tratadas a continuación:

Metodologías tradicionales se caracterizan por la división en etapas ordenadas y secuenciales, donde cada una posee objetivos y reglas prefijadas. En la misma, la planificación del proyecto se realiza antes de iniciar las actividades, haciendo uso de documentación

específica y detallada. La interacción con el cliente, para la obtención de requisitos y sugerencias, se efectúan al inicio del proyecto, el resto del tiempo las comunicaciones del equipo de desarrollo con los interesados ocurren mayormente para anunciar avances. Al concluir cada etapa definida del proyecto se materializa un entregable y se acude a la retroalimentación. Las estimaciones de alcance, tiempo y costo se asumen que no son variables y deben respetarse. La estructura del equipo se encuentra definida por roles que cumplen funciones y tareas específicas.

Metodologías ágiles se basan en la pronta entrega del software de manera incremental, originado del desarrollo iterativo durante el ciclo de vida del software. Procura flexibilizar restricciones, la burocratización del trabajo y las comunicaciones. Predomina la interacción entre los miembros del equipo y la colaboración con el cliente, la adaptación a entornos y parámetros cambiantes, y la baja demanda de tiempo de capacitación para aprender el uso de herramientas complejas.

PROGRAMACIÓN EXTREMA XP

Es una metodología de desarrollo ligero (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.

KAMBAN

Es una palabra japonesa que significa “tarjetas visuales” (kan significa visual, y ban tarjeta). Esta técnica se creó en Toyota, y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción. Actualmente está siendo aplicado en la gestión de proyectos software.

MOBILE - D

Esta metodología está basada en diversas tecnologías como Rational Unified Process, Extreme Programming y Crystal Mehodologies, y su finalidad es intentar obtener pequeños ciclos de desarrollo de forma rápida en dispositivos pequeños.

SCRUM

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

CARACTERÍSTICAS

- Desarrollo iterativo e incremental
- **Pruebas unitarias.**
- Programación en parejas.
- Corrección de todos los errores antes de añadir nueva funcionalidad.
- Refactorización del código
- Simplicidad en el código.
- Propiedad del código compartida.

- Metodología visual y sencilla
- **Calidad garantizada**
- **Mejora continua**
- Reducción del desperdicio
- **Flexibilidad**

- Un costo bajo al realizar un cambio en el proyecto.
- Entrega de resultados de manera rápida.
- Asegura el software adecuado en el momento adecuado.

- Metodología ágil y flexible
- **Resultados anticipados.**
- Flexibilidad y adaptación.
- **Retorno de inversión.**
- **Mitigación de riesgos.**
- **Productividad y calidad.**
- Alineamiento entre cliente y equipo.

FASES

- Fase planificación del proyecto	- Petición de tareas	- Fase de Exploración	- Reunión de planificación de Sprint
- Fase de diseño	- Selección de tareas	- Fase de inicialización	- Scrum diario
- Fase de codificación	- Desarrollo	- Fase de producción	- Trabajo de desarrollo durante el sprint
- Fase de pruebas	- Prueba	- Fase de estabilización	- Revisión del sprint
	- Terminado	- Fase de pruebas	- Retrospectiva del sprint

Tabla 1 Cuadro comparativo metodologías ágiles (ALVAREZ CASADIEGO, 2012).

Para este trabajo se ha optado por el empleo de las metodologías ágiles, por las ventajas y características que estas presentan, que se ajustan al proyecto concebido a desarrollar para dar solución a la situación problemática planteada con anterioridad. Específicamente, se eligió dentro de tales metodologías, a SCRUM, un marco de trabajo de desarrollo que surge en los años 80 concebido inicialmente para la producción de productos tecnológicos y que fue adaptado en 1993 para el desarrollo de software. El mismo, admite un desarrollo de software adaptable frente a cambios en los requisitos, refuerza el trabajo en equipo de los desarrolladores e involucra en el proyecto a los clientes propiciando un ambiente de compromiso y motivación.

SCRUM es un marco de trabajo de buenas prácticas, de mayor flexibilidad y mayor comprensión, pero que requiere de un trabajo constante, que permita una armonía para que no se pierda la secuencia que se logra articular desde el mismo momento en que se emprende un proyecto bajo esta metodología (ALVAREZ CASADIEGO, 2012). ([Ver Anexo 1](#))

Este marco de trabajo está constituido en los **Equipos SCRUM** y sus **roles, eventos, artefactos y reglas asociadas**. Cada uno de estos componentes tienen un fin específico y son imprescindibles para el éxito de SCRUM y para su uso.

SCRUM se sostiene sobre tres principios obtenidos de la teoría del control empírico de procesos:

- **La transparencia** se manifiesta mediante el dominio, por parte de los integrantes de la administración, del conocimiento de los aspectos y resultados que inciden sobre el proceso.
- **La inspección** dada por la capacidad para identificar variaciones que pudiesen afectar a un proyecto, a través de la realización de inspecciones frecuentes.
- **La adaptación** que permite el ajuste del proceso cuando se producen desviaciones del objetivo principal o se hallan aspectos fuera de los límites.

Los **Equipo SCRUM (SCRUM Team)** se encuentran integrados por un **Dueño de Producto (Product Owner)**, el **Equipo de Desarrollo (Development Team)** y un **SCRUM Master** que desempeñan un rol determinado dentro del equipo.

El **Dueño de Producto (Product Owner)** tiene la responsabilidad de determinar la visión del producto, recolectar los requerimientos de este, conocer en detalle las características funcionales del mismo, definir las prioridades de cada una de las características por sobre el resto y maximizar el valor entregado a los clientes. Debe encontrarse cerca de donde ocurre la toma de decisiones para poder conocer con transparencia y frecuencia el estado de los desarrollos, y así poder controlarlos. Es vital que dedique tiempo a la interacción con el Equipo de Desarrollo para velar por que estos vayan en la dirección correcta.

SCRUM Master es la persona que tiene el deber de enseñar SCRUM, servir de guía de la metodología y prestarle seguimiento y control al rendimiento del equipo.

El **Equipo de Desarrollo (Development Team)** se halla formado por un grupo de tres a nueve miembros. Es responsable de acometer las actividades necesarias respecto al desarrollo del proyecto y velar por la mejora continua de este. Para cumplir con ese objetivo, el equipo interacciona con los otros roles obteniendo de ellos información, orientación y guía metodológica.

En SCRUM existe una serie de eventos predefinidos con el objetivo de crear regularidad y a la vez, minimizar la necesidad de reuniones no definidas. Todos esos eventos son bloques de tiempo (time-boxes) y tienen un máximo de duración establecido. Todos los eventos están diseñados para facilitar la transparencia e inspección. Si se faltara alguno de ellos traería como consecuencia una reducción de la transparencia y una pérdida de oportunidad para inspeccionar y adaptarse. Los eventos de SCRUM son los siguientes:

- **El Sprint:** es el corazón de SCRUM y cuenta con un tiempo de duración máximo de un mes o menos durante el cual es creado un incremento de producto “Terminado”, listo para usar y potencialmente desplegable. Los Sprint a su vez están constituidos por otros eventos entre los que se encuentran: la Reunión de Planificación de Sprint (Sprint Planning Meeting), los SCRUMs Diarios (Daily SCRUMs), el trabajo de desarrollo, la Revisión de Sprint (Sprint Review), y la Retrospectiva de Sprint (Sprint Retrospective).
Es válido destacar que durante la ocurrencia del Sprint no se realizan cambios que puedan perjudicar el Objetivo del Sprint (Sprint Goal). Sin embargo, el alcance puede ser renegociado entre el Equipo de Desarrollo y el Dueño del Producto.
- **Reunión de Planificación de Sprint (Sprint Planning Meeting):** es la planificación del trabajo que se realiza durante el Sprint, y es creada con la colaboración del Equipo SCRUM completo. La misma tiene un tiempo máximo de duración de 8 horas.
- **SCRUM Diario (Daily SCRUM):** es una reunión de trabajo en la cual se inspecciona toda la labor realizada desde el último SCRUM Diario y se proyecta que trabajo podría acometerse antes del siguiente. Es desarrollada por el Equipo de Desarrollo y cuenta con un máximo de tiempo de 15 minutos.
- **Revisión de Sprint (Sprint Review):** se efectúa al final del Sprint con el objetivo de inspeccionar el Incremento y realizarle cambios de adaptación a la Lista de Producto si así lo requiriera. Participa el Equipo SCRUM y los interesados en colaborar respecto a lo hecho durante el Sprint. Esta reunión está determinada para un bloque de tiempo de 4 horas.
- **Retrospectiva de Sprint (Sprint Retrospective):** es el evento que proporciona la oportunidad de inspeccionarse el Equipo SCRUM a sí mismo, adoptando un plan de mejoras a acometer en el siguiente Sprint. Es una reunión con un límite superior de tiempo de tres horas, programada para realizarse después de la Revisión de Sprint y anterior a la próxima Reunión de Planificación de Sprint.

El marco de trabajo SCRUM posee una cantidad mínima de artefactos. A continuación, son descritos cada uno de ellos:

- **Lista de Producto (Product Backlog):** se trata básicamente de un listado ordenado de ítems que recoge las características, funcionalidades, requisitos, mejoras y correcciones a ser realizados sobre el producto a desarrollar, el cual es gestionado y priorizado por el Dueño del Producto. Este artefacto evoluciona a medida que el producto y el medio en que se empleará también lo hacen.
- **Lista de Pendientes del Sprint (Sprint Backlog):** es la selección del conjunto de elementos de la Lista de Producto para el Sprint, acompañada de un plan para conseguir el Objetivo del Sprint y la entrega del Incremento. Básicamente, es una predicción realizada por el Equipo de Desarrollo respecto a que funcionalidad integrará el próximo Incremento y del esfuerzo a realizar para lograrlo
- **Incremento:** es el total de todas las funcionalidades logradas durante el Sprint.

Como se ha tratado, SCRUM es un marco de trabajo diseñado para acometer problemas complejos adaptativos y entregar productos con un máximo de valor productivo posible, es ligero y fácil de entender, pero a la vez, difícil de llegar a dominar.

1.7.6 Aplicaciones Web.

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar a través del acceso a un servidor mediante un navegador. La misma posibilita la comunicación activa entre el usuario y la información, su funcionamiento es independiente de la versión del sistema operativo instalado en el lado del cliente.

Se destacan en su empleo las siguientes ventajas:

- **Ahorro de tiempo:** se pueden acometer tareas simples sin la necesidad de instalar programa alguno.
- **Compatibilidad:** suficiente con tener el navegador web actualizado para poder utilizarlas.
- **Espacio:** no hacen uso de la capacidad de nuestro disco duro.

- **Actualizaciones inmediatas:** siempre que el desarrollador realice una actualización del software, se estará percibiendo la misma de forma inmediata.
- **Bajo consumo de recursos:** la mayor demanda de información y procesamiento de la aplicación es gestionada en el servidor que la hospeda.
- **Multiplataforma:** puede ser utilizada desde cualquier sistema operativo, solamente ese necesario contar con la presencia de un navegador.
- **Disponibilidad:** ofrece sus servicios desde múltiples localizaciones, hoy en día con la existencia de internet, se podría asegurar el acceso en el mundo entero.

1.7.7 Herramientas y tecnologías.

1.7.7.1 Lenguajes de informática del lado del cliente.

1.7.7.1.1 HTML (Lenguaje de marcas de hipertexto).

Es un lenguaje muy sencillo que presenta texto de manera estructurada y agradable, con acceso a documentos o fuentes de información relacionada, y con inserciones multimedia. Tal presentación se basa en especificar en el texto la estructura lógica del contenido, pudiendo ser estos: títulos, párrafos de texto, enumeraciones, definiciones u otros. Además, se encarga de proporcionar distintos efectos para esa presentación a través del uso de otros lenguajes especializados para ello.

A continuación, se exponen las distintas versiones que ha poseído este lenguaje de forma resumida:

HTML o HTML1: En 1991 Tim Berners-Lee da a conocer el lenguaje HTML creado a partir del Estandar de Lenguaje de Marcación Generalizado (SGML). En ese mismo contexto, él crea algo más, el World Wide Web (también conocido como la Web, sistema de documentos de hipertextos y/o hipermedias enlazados entre sí y accesibles desde internet).

HTML+ o HTML2: su desarrollo comenzó en 1993 y su diseño corrió a cargo de David Ragget. Se destacó por formalizar un conjunto de extensiones de HTML, realizar un replanteamiento,

unificación y mejora de los diferentes dialectos de HTML usados hasta ese momento. Esta versión incorpora nuevas características como la utilización de imágenes, tablas, formularios y líneas anidadas.

HTML3: fue propuesto en 1995 por el recién formado World Wide Web Consortium (W3C) el que se encontraba dirigido en ese entonces por Tim Berners-Lee. En él se introducen nuevas características como la fluidez de texto alrededor de las figuras mostradas y la posibilidad de mostrar elementos matemáticos complejos. Sin embargo, presentó muchas complejidades e inadaptaciones que lo condujeron a poseer diferentes versiones partiendo del HTML3.0 y transcurriendo por los HTML3.1 y HTML3.2, que lamentablemente tampoco obtuvieron la simpatía de muchos de los programadores de navegadores web de aquel entonces.

HTML4: fue presentado en junio de 1998 y se acentuó por adoptar muchos de los elementos específicos desarrollados inicialmente para un navegador web, pero también por limpiar el lenguaje HTML de sus elementos catalogados como “desaprobados”.

HTML5: es la versión más joven de HTML, fue desarrollada a partir del año 2007 por el Proyecto de Estándares Web (The Web Standards Project). Fue concebido como un lenguaje capaz de ejecutar aplicaciones completas. Recibió el respaldo del consorcio W3C que decidió abandonar otros proyectos que tenía en marcha y centrarse en el desarrollo de HTML5, siendo este el empujón definitivo al nuevo estándar. Se caracteriza por poseer una amplia gama de mejoras en controles para formularios, APIs, multimedia y también en la estructura y la semántica. En la actualidad, casi todos los navegadores soportan la mayoría de las etiquetas HTML5.

1.7.7.1.2 CSS (Cascade Style Sheet).

Hoja de Estilo en Cascada (Cascade Style Sheet), es un lenguaje que permite definir el aspecto de documentos de la familia HTML. Mediante su empleo se puede aplicar un estilo concreto para los elementos de una página web, y combinándolo con lenguajes como JavaScript se pueden aplicar efectos especiales para interactuar con los usuarios.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores (Eguíluz Pérez, 2008).

CSS3 es la última versión disponible y de mayor potencial, le preceden las versiones CSS 1 y CSS 2.1. En la tabla siguiente se muestra el soporte que brindan algunos de los navegadores más utilizados hoy en día a las distintas versiones de CSS.

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

Tabla 2 Soporte a CSS de navegadores actuales (Eguíluz Pérez, 2008).

El navegador Internet Explorer sólo puede considerarse adecuado desde el punto de vista de CSS a partir de su versión 7. De hecho, la versión Internet Explorer 6 que aún utilizan muchos usuarios sufre carencias muy importantes y contiene decenas de errores en su soporte de CSS. Afortunadamente, Internet Explorer 8 asegura que su soporte de CSS 2.1 será tan completo como el del resto de navegadores.

Algunas de las ventajas de usar CSS son:

- **Separación del contenido y la presentación:** cuanto más se separen estos dos elementos, más sencillo será el mantenimiento de las páginas y procesamiento de la información.

- **Consistencia del sitio:** una hoja de estilos global puede ser utilizada para aplicar estilos a todo un sitio, y en el caso que en el futuro algunos de esos estilos deben ser modificados, basta con hacerlo de forma única en esa hoja de estilos global, siendo mucho más fácil.
- **Ahorro en la transferencia:** Una hoja de estilo externa normalmente es alojada en la cache del navegador, pudiendo ser reutilizada por múltiples páginas sin ser cargada nuevamente, reduciendo de forma significativa la transferencia de datos en la red.
- **Formateo de página:** mediante un cambio simple de una sola línea, puede modificarse la hoja de estilos para la misma página.
- **Compatibilidad:** la mayoría de los usuarios poseen navegadores que soportan características avanzadas de las CSS, pero en el caso de que no sea así, estos serán capaces de leer el código de la página y mostrarlo sin ningún posicionamiento.

1.7.7.1.3 JavaScript.

JavaScript es un sencillo lenguaje de programación, que presenta una característica especial: sus programas, llamados comúnmente scripts, se incrustan en las páginas HTML y se ejecutan en el navegador (Mozilla Firefox, Microsoft Internet Explorer y otros). Estos scripts normalmente consisten en unas funciones que son llamadas desde el propio HTML cuando algún evento sucede. De ese modo, podemos añadir efectos como que un botón cambie de forma al pasar el ratón por encima, o abrir una ventana nueva al pulsar en un enlace.

JavaScript fue desarrollado por Netscape, a partir del lenguaje Java, el cual sigue una filosofía similar, aunque va más allá. Java es un lenguaje de programación por sí mismo, como lo pueden ser C, Pascal o Visual Basic. Esto quiere decir, que se puede ejecutar un programa Java fuera de un navegador. La diferencia fundamental es que Java es un lenguaje completo, que puede ser utilizado para crear aplicaciones de todo tipo, mientras que JavaScript sólo “funciona” dentro de una página HTML.

Este lenguaje fue declarado como estándar del European Computer Manufacturers Association (ECMA) en 1997, y poco después también fue estandarizado por ISO. Sin embargo, la estructura de objetos que implementaban los diferentes navegadores (Netscape y Explorer en aquellos momentos) no se ajustaba al estándar, lo que provocaba numerosos

problemas de compatibilidad. Para solventarlos el W3C publicó un nuevo modelo de objetos, DOM (Document Object Model), que incorporan la mayoría de navegadores actuales como Explorer o Firefox.

1.7.7.2 Lenguajes de programación del lado del servidor.

1.7.7.2.1 Python.

Python es un lenguaje de programación divertido y extremadamente fácil de usar que en los últimos años se ha hecho muy popular. Desarrollado hace diez años por Guido van Rossum, su sintaxis simple y la sensación general se deriva en gran parte del ABC, un lenguaje desarrollado en los 1980s para la enseñanza. Sin embargo, Python también se creó para resolver problemas reales y presenta una variedad amplia de características de lenguajes de programación como C++, Java, Modula-3 y Scheme. Debido a esto, una de las características notables de Python es su atractivo para los desarrolladores profesionales de programación, científicos, investigadores, artistas, y educadores (Downey, Elkner, & Meyers, 2002).

Según (Challenger Pérez, Díaz Ricardo, & Becerra García, 2014) existe una lista de principios de diseño que se deberían seguir cuando se escribe código en Python:

- Hermoso es mejor que feo.
- Explícito es mejor que implícito.
- Simple es mejor que complejo.
- Plano es mejor que anidado.
- Disperso es mejor que denso.
- El código legible cuenta.
- Casos especiales no son lo suficientemente especiales para romper las reglas.
- Casi siempre lo práctico vence a lo formal.
- Los errores no deben pasar nunca desapercibidos, a menos que se especifique este comportamiento.
- Ante una ambigüedad, descarte la tentación a adivinar.

- Debe haber una, y preferentemente una sola, manera obvia de lograr algo, aunque esta generalmente no está clara a primera vista a menos que seas un genio.
- Ahora es mejor que nunca, aunque en muchas ocasiones nunca es mejor que ahora mismo.
- Si la implementación es difícil de explicar, entonces es una mala idea.
- Si la implementación es fácil de explicar, entonces pudiera ser una buena idea.
- Los espacios de nombre son una buena idea, hagamos más de eso.

Ventajas y desventajas.

(J. & G., 2019) presenta algunas ventajas y desventajas del lenguaje Python:

Ventajas

Las ventajas del lenguaje Python son las siguientes:

- **Simplificado y rápido:** Este lenguaje simplifica mucho la programación “hace que te adaptes a un modo de lenguaje de programación, Python te propone un patrón”. Es un gran lenguaje para scripting, si usted requiere algo rápido (en el sentido de la ejecución del lenguaje), con unas cuantas líneas ya está resuelto.
- **Elegante y flexible:** El lenguaje le da muchas herramientas, si usted quiere listas de varios tipos de datos, no hace falta que declares cada tipo de datos. Es un lenguaje tan flexible usted no se preocupa tanto por los detalles.
- **Programación sana y productiva:** Programar en Python se convierte en un estilo muy sano de programar: es sencillo de aprender, direccionado a las reglas perfectas, le hace como dependiente de mejorar, cumplir las reglas, el uso de las líneas, de variables”. Además, es un lenguaje que fue hecho con productividad en mente, es decir, Python le hace ser más productivo, le permite entregar en los tiempos que me requieren.
- **Ordenado y limpio:** El orden que mantiene Python, es de lo que más le gusta a sus usuarios, es muy legible, cualquier otro programador lo puede leer y trabajar sobre el programa escrito en Python. Los módulos están bien organizados, a diferencia de otros lenguajes.
- **Portable:** Es un lenguaje muy portable (ya sea en Mac, Linux o Windows) en comparación con otros lenguajes. La filosofía de baterías incluidas, son las librerías que más usted

necesita al día a día de programación, ya están dentro del interprete, no tiene la necesidad de instalarlas adicionalmente con en otros lenguajes.

- **Comunidad:** Algo muy importante para el desarrollo de un lenguaje es la comunidad, la misma comunidad de Python cuida el lenguaje y casi todas las actualizaciones se hacen de manera democrática.

Desventajas

Las desventajas del lenguaje Python son las siguientes:

- **Curva de aprendizaje:** La “curva de aprendizaje cuando ya estás en la parte web no es tan sencilla”.
- **Hosting:** La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es un poco difícil.
- **Librerías incluidas:** Algunas librerías que trae por defecto no son del gusto de amplio de la comunidad, y optan a usar librerías de terceros.

1.7.7.3 Sistema de Gestión de Base de Datos. PostgreSQL.

En la actualidad, las bases de datos cumplen una función muy importante en los sistemas de información, la mayoría de las empresas sean públicas o privadas tiene sus procesos automatizados y esto hace que los sistemas manuales queden obsoletos a la hora de realizar búsquedas, modificaciones y cualquier obtención de información de dicha empresa, es por ello que la utilización de una base de datos se hace indispensable al momento de almacenar grandes volúmenes de información con la que cuenta la organización. Una base de datos es un banco de información, el cual contiene datos relacionados entre sí y se encuentran agrupados o estructurados; además son manipulados por programas conocidos actualmente como Sistema de Gestión de Base de Datos (SGBD). En este caso se ha utilizado PostgreSQL como SGBD para la realización y ejecución del proyecto (Zea Ordóñez, Molina Ríos, & Redrován Castillo, 2017).

Entre los sistemas de bases de datos existentes hoy en día, PostgreSQL juega un papel muy importante ya que es un sistema que tiene muchas cualidades que lo hacen ser una muy buena

alternativa para instalar sistemas en empresas, universidades y una gran cantidad de otras aplicaciones (Denzer, 2002).

PostgreSQL es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD. Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Al encontrarse PostgreSQL liberado bajo la licencia BSD, su empleo incluye las siguientes ventajas:

- Ahorros importantes al liberarse del pago de licencias y especialmente por la replicación casi gratuita de aplicaciones comunes a toda la administración. El muy bajo costo del software permitirá la ejecución de programas y proyectos cuyos costos actuales los hacen prohibitivos.
- La empresa deja de depender de terceros (a menudo transnacionales) para el diseño, desarrollo y mantenimiento de sus sistemas de información, retomando el control total de sus procesos, en particular de los procesos críticos.
- El acceso al código fuente, la libertad de inspeccionar el funcionamiento del software, la libertad de decidir la manera en que almacenan los datos y la posibilidad de modificar cualquiera de estos aspectos queda en manos de la empresa, lo cual le permite el control total de la información.
- El software libre realizado por comunidades está sometido a la inspección de un importante número de personas, este número de verificadores es mucho mayor que el del software propietario. Estas personas identifican los problemas, los resuelven, y comparten las soluciones con los demás. Por tal razón los programas libres de las comunidades gozan de gran confiabilidad y estabilidad.
- La información que la empresa maneja generalmente es importante y/o confidencial, puede ser muy peligroso que esta información caiga en manos incorrectas. Por esta razón es

imprescindible que la empresa pueda verificar que su software no tenga puertas de entrada traseras, voluntarias o accidentales, y que pueda cerrarlas en caso de encontrarlas; tal control sólo es posible con el software libre.

1.7.7.4 Frameworks y Librerías.

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

Un **framework para aplicaciones web** es un *framework* diseñado para apoyar el desarrollo de sitios web dinámicos, aplicaciones web y servicios web. Este tipo de frameworks intenta aliviar el exceso de carga asociado con actividades comunes usadas en desarrollos web.

Un framework Web provee una infraestructura de programación para tus aplicaciones, para que puedas concentrarte en escribir código limpio y de fácil mantenimiento sin tener que reinventar la rueda.

1.7.7.4.1 Django.

(García M., 2015) expresa que Django es un framework, desarrollado totalmente con Python, destinado para la construcción de aplicaciones web. El mismo, permite construir en profundidad, de forma dinámica, sitios interesantes en un tiempo extremadamente corto. Se encuentra diseñado para hacer foco en la diversión, en las partes interesantes del trabajo, al mismo tiempo que alivia el dolor de las partes repetitivas. Al hacerlo, proporciona abstracciones de alto nivel a patrones comunes del desarrollo Web, agrega atajos para tareas frecuentes de programación y claras convenciones sobre cómo resolver problemas. Al mismo tiempo, intenta mantenerse fuera de tu camino, dejando que se trabaje fuera del alcance del framework cuando sea necesario. Está diseñado para poner rápidamente en movimiento un proyecto.

1.7.7.4.2 Bootstrap.

Bootstrap es un framework que posee una estructura responsive fácilmente de montar, ya que muchas de sus clases y funciones están desarrolladas, en su accionar se vale de muchas librerías como normalice, jQuery, Less, por lo que proporciona muchos efectos y funciones sin tener que programar de cero, posibilita un diseño que pueda ser visualizado de forma correcta en distintos dispositivos y a distintas escalas y resoluciones.

Características principales de Bootstrap:

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
- Incorpora un sistema de grid basado en 12 columnas fluidas para poder acoplarse en función del dispositivo desde el que se acceda.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo, JQuery.
- Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTMLShim para que reconozca los tags HTML5.

1.7.7.4.3 JQuery.

JQuery es una biblioteca de JavaScript rápida y concisa que simplifica el documento HTML, manejo de eventos, animación y las interacciones AJAX para el desarrollo web.

Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio (Villarreal Fuentes, 2013).

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Otras de las características que posee son las siguientes:

- Selección de elementos.
- Eventos.
- Manipulación de la hoja de estilos CSS.
- Efectos y animaciones.

- Animaciones personalizadas.
- AJAX.
- Soporta extensiones.
- Utilidades varias como obtener información del navegador, operar con objetos y vectores.

1.7.7.5 Entorno de Desarrollo Integrado (IDE). PyCharm.

PyCharm es un Entorno de Desarrollo Integrado que ofrece una colección completa de herramientas para los desarrolladores profesionales de Python. Está constituido por un editor que comprende el código profundamente, y un depurador que ofrece una vista clara del código en acción. PyCharm proporciona la integración con herramientas de colaboración como son los sistemas de control de versiones y los rastreadores. La Edición Profesional abarca los elementos esenciales para la integración perfecta con marcos de trabajos web, herramientas de JavaScript, virtualización y el soporte de contenedores.

1.7.7.6 Herramientas de modelado.

El modelado de softwares es una técnica para tratar con la complejidad inherente a estos sistemas. El uso de modelos ayuda al ingeniero de software a "visualizar" el sistema a construir. Además, los modelos de un nivel de abstracción mayor pueden utilizarse para la comunicación con el cliente. Por último, las herramientas de modelado y las de Ingeniería de Software, pueden ayudar a verificar la corrección del modelo.

1.7.7.6.1 Visual Paradigm for UML8.

El UML es el lenguaje unificado para facilitar el desarrollo de software, donde se incluyen estandartes de calidad y se hacen uso de metodologías de desarrollo. Casi en todos los desarrollos de software se hace uso alguna vez de este lenguaje, ya sea para diseñar alguna interfaz o algún diagrama del software, aunque también se puede hacer uso de un pizarrón o una hoja de papel y un lápiz o marcador, es cierto que algunos sistemas son más grandes que una interfaz, o incluso esa sola interfaz o puede ser más compleja de lo que pensamos.

En el mercado existen varias aplicaciones que nos ayudan a diseñar modelos UML, dentro de ellas encontramos:

Visual Paradigm para UML (VP-UML), una poderosa herramienta visual del tipo UML CASE. La misma está diseñada para una gran variedad de usuarios, incluyendo Ingenieros de Software, Sistema Analysts, Analistas de Empresas, Sistema Architects por igual, cualquiera que se encuentre interesado en construir soportes lógicos confiables a través del uso de la Programación Orientada a Objetos.

El entorno de trabajo de VP-UML provee unos recursos intuitivos para llevar a cabo diseño y análisis de sistemas Orientado a Objetos, donde usted puede crear diagramas UML a través de operaciones simples de funcionalidad de arrastre de imágenes usando el ratón en una computadora. Es una colección de menús, barras de herramientas y ventanas que hacen el área de trabajo de desarrollo, que le deja crear tipos diferentes de diagramas en un entorno completamente visual.

1.7.7.6.2 Bizagi Process Modeler.

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

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

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

- Proporciona un módulo de simulación de gran alcance que le permite tomar mejores decisiones visualizando el impacto de ideas y cambios propuestos antes de su implementación en el mundo real.
- Con el Modelador de procesos BPMN Bizagi, se podrá compartir ideas de mejoramiento con los otros miembros del equipo, así como también presentar los procesos en un formato de aceptación mundial.

1.8 Conclusiones parciales del capítulo.

Durante este capítulo quedaron sentadas las bases teóricas que sustentan la investigación a través del estudio de los distintos aspectos y conceptos que se encuentran asociados al dominio del problema. El análisis de los antecedentes del proceso objeto de estudio y las características generales de la entidad, así como las herramientas, tecnologías y metodologías de desarrollo, permiten arribar a las siguientes conclusiones:

- Se identifica la necesidad de desarrollar un software que se ajuste a las características de comercialización que posee la UEB DIVEP Matanzas para contribuir a mejorar la eficiencia con que se realiza esa actividad.
- Se entiende que el uso de las metodologías ágiles de desarrollo de softwares, dentro de ellas el marco de trabajo SCRUM, así como la combinación de las tecnologías y herramientas de desarrollo seleccionadas, son las más apropiadas para la construcción de la aplicación web propuesta para dar solución a la situación problemática planteada.

Capítulo II Análisis, diseño y construcción de la solución propuesta

2.1 Introducción.

En el presente capítulo se abordan los elementos necesarios para realizar una descripción de la solución propuesta. La implementación de la misma se soporta en el empleo de las tecnologías y herramientas definidas en el capítulo anterior, gestionada mediante los principios y reglas que rigen el marco de trabajo SCRUM. A partir del uso de este marco de trabajo, se explica el desempeño de los roles, eventos y artefactos que intervienen durante la construcción del software. Además, son incorporados otros artefactos no comprendidos dentro de las metodologías ágiles que son usados con el objetivo de brindar una mayor comprensión del sistema a implementar. Y son analizados los resultados obtenidos a través del estudio de factibilidad realizado.

2.2 Modelo de Proceso.

2.2.1 Diagrama de Procesos de Negocio (BPD).

BPD es un diagrama diseñado para representar gráficamente la secuencia de todas las actividades que ocurren durante un proceso, basado en la técnica de “Flow Chart”, el cual se define como representación gráfica de una secuencia lógica de procesos de trabajo, mediante la utilización de diferentes simbologías, el cual representa operaciones, datos, direcciones de flujo y recursos, para la definición, análisis o solución de un problema (Gómez Mondragón, 2016).

2.2.2 Descripción del Negocio.

El proceso de oferta y reserva de productos comienza cuando el usuario interesado en percibir el servicio accede a la aplicación web destinada para ello. Este usuario tiene la posibilidad de acceder como un usuario Visitante, el cual solo dispondrá de los accesos para ver la oferta de la mercancía, o como un usuario Cliente que dispone además de los accesos del usuario anterior, los de realizar la reserva de la mercancía a nombre de la empresa que representa y/o

agregar productos a la Lista de Deseos. Para que el usuario Cliente pueda realizar la reserva de cualquier producto, la entidad a la cual representa debe poseer un contrato de venta vigente y el producto debe tener disponibilidad. Sin embargo, para agregar productos a la Lista de Deseos basta con que se cumpla el primer requisito. El tiempo de vida de la reserva estará determinado por el tiempo de vigencia de reserva especificado para la entidad que la realiza, o por la realización de una factura que la consuma. Si a una reserva se le termina su tiempo de vigencia, esta es cancelada y la cantidad de cada producto contenido en ella es reincorporada a la disponibilidad de los mismos. La Lista de Deseos funciona como un subproceso capaz de notificar al usuario Cliente, mediante de un email o una alerta mostrada por la aplicación, cuando un producto contenido en ella adquiere disponibilidad. Esta notificación es reiterada al usuario Cliente pasado 7 días si este no realiza interacción alguna con tal producto.

Este proceso se puede apreciar en la figura 1 y 2.

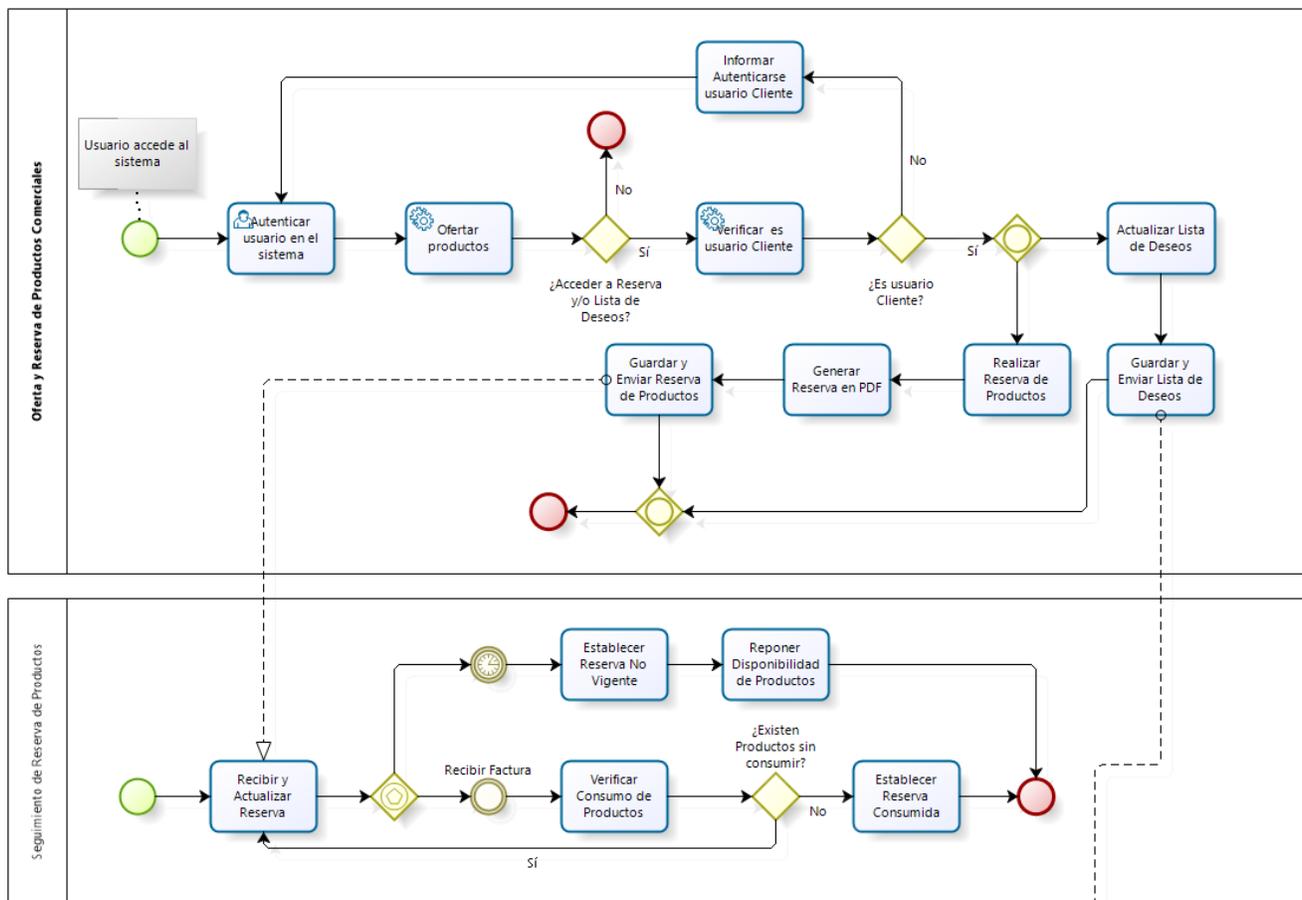


Ilustración 1 Diagrama BPD "Oferta y Reserva de Productos - Seguimiento de Reserva de Productos" (elaboración propia).

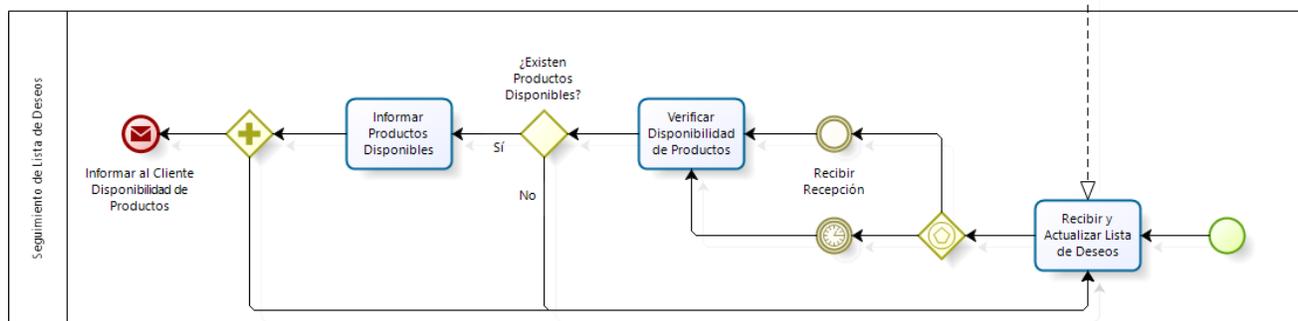


Ilustración 2 Diagrama BPD "Seguimiento de Lista de Deseos" (elaboración propia).

2.2.3 Modelo Conceptual de la Base de Datos.

Un Modelo Conceptual es aquel elemento lógico o físico que facilita la comprensión de un problema. Aplicando lo anterior y teniendo en consideración el análisis de la información

obtenida a través de las entrevistas con los clientes acerca de los datos, las características de estos y el flujo informativo acometer, se determina el siguiente Modelo Conceptual de la Base de Datos:

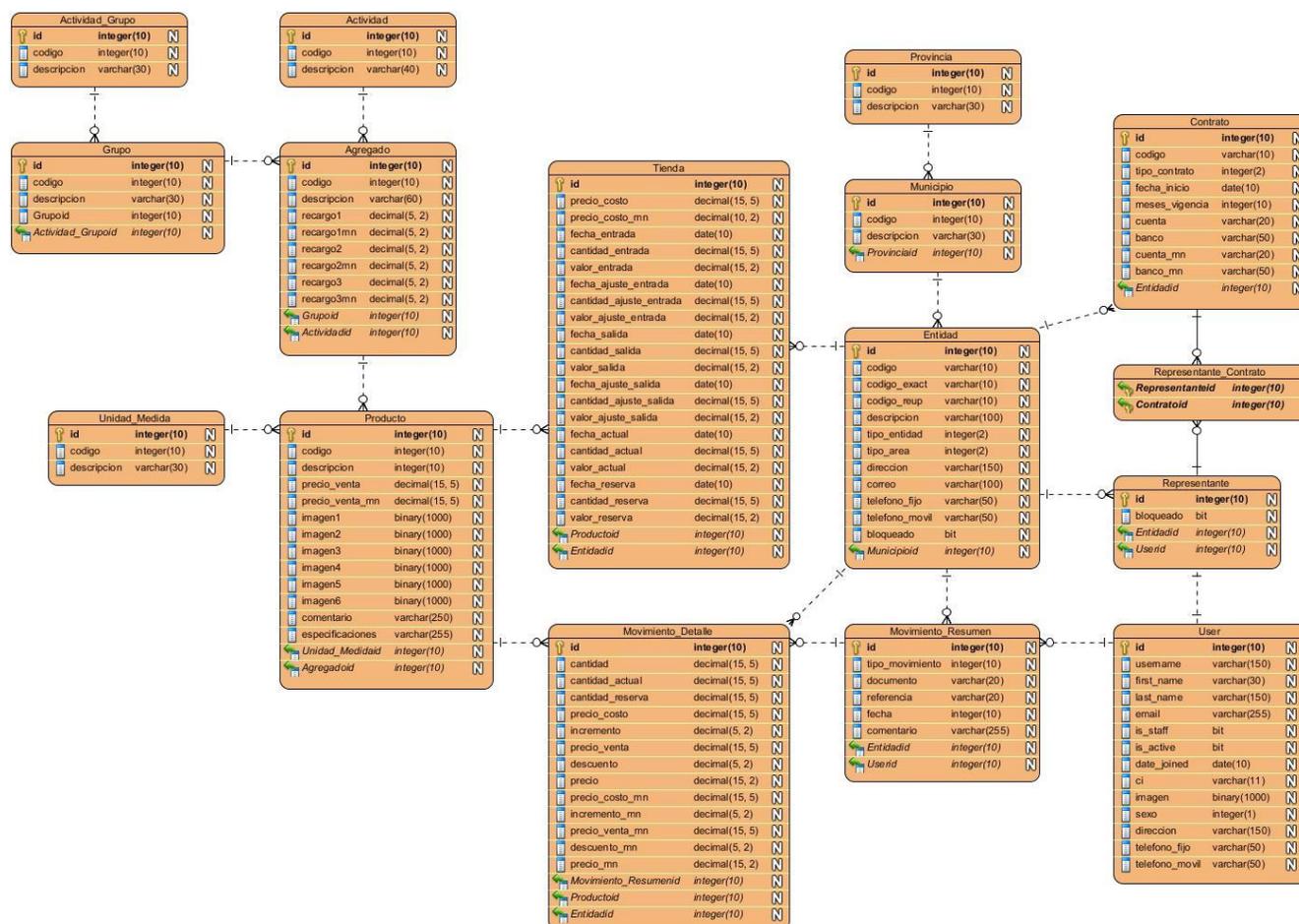


Ilustración 3 Modelo Conceptual de la Base de Datos (elaboración propia).

2.3 Descripción de la solución.

Se propone el desarrollo de una aplicación web que permita a la UEB DIVEP Matanzas implementar el comercio electrónico aprovechando las ventajas que este ofrece para realizar la oferta y reserva de los productos que comercializa de una manera eficiente. De esa forma, se pretende mejorar la eficiencia en la actividad comercial que desempeña la entidad y ofrecer a los clientes un servicio de mayor calidad, confort y seguridad.

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.

Jurídico: es el usuario responsable de incorporar y actualizar las entidades, representantes de estas y sus respectivos contratos.

Comercial: es el usuario encargado de incorporar nuevos productos en la aplicación y monitorear el comportamiento de los mismos.

Contador: es el usuario que vela por la correcta y frecuente retroalimentación de la aplicación a partir de los movimientos producidos en el sistema de gestión de inventario.

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 y el agrupamiento por categorías.

2.4 Etapa de Planificación.

La etapa de planificación es la primera de todo el proyecto. La misma es realizada con el objetivo de obtener una eficiente organización del prototipo inicial del problema y propiciar de esa manera un buen comienzo en aras de lograr una solución eficaz. Con ese fin y en correspondencia con las ideas del cliente acerca del software se desarrollarán las Historias de Usuario, a partir de las cuales se obtendrá un punto de partida para el resto de la planificación del proyecto. De igual forma, se realiza un estimado de cada una de las entregas del proyecto y del tiempo que puede tomar, teniendo presente que la planificación inicial podría estar sujeta a cambios a partir de modificaciones que pudiesen sufrir estos aspectos durante el desarrollo del proyecto.

2.4.1 Equipo de Trabajo.

La metodología SCRUM concibe que los equipos de desarrollo sean auto organizados, que escojan como es mejor llevar a cabo su tarea, sin ser dirigidos por otros fuera del equipo. Estos equipos multifuncionales deben poseer todas las capacidades necesarias para completar su trabajo sin depender de otros que no pertenecen al equipo.

Los equipos de SCRUM realizan la entrega de las partes del producto de forma iterativa e incremental, maximizando las oportunidades de retroalimentación. Tales entregas incrementales de producto “hecho” garantiza que siempre exista una versión potencialmente utilizable del producto, es decir, que se encuentre funcional.

A continuación, se relacionan los roles y las funciones del equipo SCRUM en el desarrollo de la solución:

Miembros	Roles
Directivos de UEB DIVEP Matanzas	Stakeholders
Ing. Jesús Alberto Suárez	Product Owner (Dueño del Producto)
Ing. Renier Martínez Serpa	SCRUM Master
Ing. Renier Martínez Serpa	SCRUM Team (Equipo de Trabajo)

Tabla 3 Definición de los roles del equipo SCRUM (elaboración propia).

Para la definición de los roles en el actual trabajo se adecuaron las definiciones expuestas en la metodología de desarrollo abordada, debido a que ha sido desarrollado por una única persona que desempeñará los roles de Facilitador y Equipo. Esta adaptación trae como consecuencia la pérdida de la relación entre tales roles, que tienen un papel relevante para el desarrollo del software; pero a diferencia de otras metodologías, SCRUM permite transformar el flujo de trabajo y adaptar las condiciones del entorno a su metodología.

2.4.2 Pila del Producto (Product Backlog).

Id	Prioridad	Descripción	Importancia	Estimación (semanas)
----	-----------	-------------	-------------	----------------------

01	1	Diseño y creación de la Base de Datos	10	1
02	2	Diseño y creación de la Interfaz de Usuario	10	3
03	3	Gestionar nomencladores: <ul style="list-style-type: none"> • Actividad • Actividad-Grupo • Grupo • Agregado • Área • Producto 	8	3
04	4	Gestionar Entidades - Contratos	8	2
05	5	Gestionar Tienda (entradas, salidas)	10	4
06	6	Gestionar Carro de Compra	8	2
07	7	Gestionar Lista de Deseos (Seguimiento a Productos)	6	2
08	8	Gestionar Seguridad del Sistema	8	2
09	9	Gestionar Auditoria del Sistema	7	2
10	10	Gestionar Informes y Estadísticas	5	3

Tabla 4 Pila del Producto o Product Backlog (elaboración propia).

2.4.3 Requisitos No Funcionales.

Estos requisitos reflejan las propiedades o cualidades que el producto debe poseer. Tratan de asegurar que el producto resulte atractivo, usable, rápido o confiable. Por lo general se encuentran vinculados a los Requisitos Funcionales. En no pocas ocasiones, estos resultan fundamentales para el éxito del producto.

Para el Software tratado en este trabajo se determina los Requisitos No Funcionales siguientes:

Interfaz:

- La interfaz debe ser simple, legible y amigable para los usuarios, que les resulte atractiva y a la vez fácil de usar.

Usabilidad:

- El sistema debe proporcionar la ayuda necesaria para que el usuario se pueda orientar al hacer uso de cualquier funcionalidad, debe estar diseñada pensando en aquellos que tengan niveles básicos de computación.

Rendimiento:

- La aplicación debe estar concebida para hacer un uso mínimo de recursos. Permitir la conexión de un total aproximado de 100-150 usuarios de forma simultánea.
- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 3 segundos para las actualizaciones y 10 para las recuperaciones.

Soporte:

- Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos. Se recomienda el empleo del servicio de hosting que ofrece ETECSA a través de su Centro de Datos.

Portabilidad:

- El sistema será multiplataforma (Linux, Mac y Windows fundamentalmente).

Seguridad:

- El usuario debe autenticarse antes de entrar al sistema. Debe existir limitación de accesos a las interfaces e información por roles de usuarios.
- El envío de información a través de los métodos POST debe poseer un Token para impedir la captura y robo de información.

Integridad:

- Tanto la Aplicación Web como la Base de Datos se hospedarán en un Servidor encontrándose protegidos de accesos no autorizado, divulgación y comprometimiento de la fuente de datos.

Confiabilidad:

- El software debe contar con el respaldo necesario para garantizar su recuperación ante fallas eléctricas o de comunicación.

Requisitos de diseño:

- Para el diseño e implementación del sistema se debe utilizar el Marco de Trabajo Django que presenta grandes ventajas para la elaboración de aplicaciones web por su robustez, flexibilidad y portabilidad.

Requerimiento de Ayuda y Documentación:

- Se propone que el sistema cuente con un manual de ayuda general, y además ofrecer guía u orientación a través de las diferentes interfaces que presente.

Políticos Culturales:

- El producto en su interfaz no debe contener palabras en otro idioma que no sea el español.

2.4.4 Historias de usuarios.

A continuación, se muestra las Historias de Usuarios (HU) de mayor importancia en el desarrollo de la investigación. Para ver todas las HU comprendidas en el proceso de planificación: [Ver Anexo 2](#)

Historia de usuario 01.

HISTORIA DE USUARIO	
ID: HU01	Usuario: Administrador del Sistema
Nombre Historia: Diseño y creación de la Base de Datos	
Prioridad en el Negocio: 1	Importancia del Desarrollo: 10

Tiempo Estimado: 1 semana	Sprint: 1
Descripción: Diseñar e implementar la base de datos que almacenará la información.	
Observaciones: El diseño debe garantizar la integridad de la información.	

Tabla 5 Historia de usuario 01 (elaboración propia).

Historia de usuario 02.

HISTORIA DE USUARIO	
ID: HU02	Usuario: Administrador del Sistema
Nombre Historia: Diseño y creación de la Interfaz de Usuario	
Prioridad en el Negocio: 2	Importancia del Desarrollo: 10
Tiempo Estimado: 2 semanas	Sprint: 2
Descripción: Diseñar y elaborar la interfaz de usuario.	
Observaciones: La interfaz debe mostrarse atractiva y amigable.	

Tabla 6 Historia de usuario 02 (elaboración propia).

Historia de usuario 05.

HISTORIA DE USUARIO	
ID: HU05	Usuario: Contador
Nombre Historia: Gestionar Tienda (entradas, salidas)	
Prioridad en el Negocio: 5	Importancia del Desarrollo: 10
Tiempo Estimado: 3 semanas	Sprint: 5
Descripción: Gestionar los movimientos de los productos referido a su inventario.	
Observaciones: Se debe garantizar la realización correcta de entradas, salidas y movimientos que afectan la cantidad y disponibilidad de los productos.	

Tabla 7 Historia de usuario 05 (elaboración propia).

2.4.5 Plan de Sprint.

Sprint	Pila del Producto	Fecha inicial	Fecha final
1	Diseño y creación de la Base de Datos	02/12/2019	09/12/2019
2	Diseño y creación de la Interfaz de Usuario	10/12/2019	31/12/2019

3	Gestionar nomencladores	02/01/2020	23/01/2020
4	Gestionar Entidades - Contratos	24/01/2020	07/02/2020
5	Gestionar Tienda (entradas, salidas)	10/02/2020	09/03/2020
6	Gestionar Carro de Compra	10/03/2020	24/03/2020
7	Gestionar Lista de Deseos	25/03/2020	08/04/2020
8	Gestionar Seguridad del Sistema	09/04/2020	23/04/2020
9	Gestionar Auditoria del Sistema	24/04/2020	08/05/2020
10	Gestionar Informes y Estadísticas	11/05/2020	01/06/2020

Tabla 8 Plan de Sprint (elaboración propia).

2.4.6 Plan de Entrega.

Las entregas se realizarán al concluir cada Sprint según las fechas especificadas para cada uno en la Tabla 8 Plan del Sprint. El cliente tendrá la oportunidad de especificar las pruebas de aceptación mediante las cuales comprobará que cada una de las Historias de Usuarios han sido desarrollada exitosamente.

2.4.7 Resumen de Tareas generadas por Historias de Usuario.

No.	Historia de Usuario	Tareas Asignadas	Sprint
1	Diseño y creación de la Base de Datos	Diseño de la base de datos. Creación de la base de datos.	1
2	Diseño y creación de la Interfaz de Usuario	Diseño y creación de la interfaz de usuario.	2
3	Gestionar nomencladores	Agregar nomencladores. Editar nomencladores. Eliminar nomencladores. Listar nomencladores.	3
4	Gestionar Entidades - Contratos	Agregar entidades, representantes y contratos. Editar entidades, representantes y contratos. Eliminar entidades, representantes y contratos. Listar entidades, representantes y contratos. Realizar validaciones de fecha y vigencia de los contratos.	4

5	Gestionar Tienda (entradas, salidas)	Agregar entradas y salidas del inventario. Listar entradas y salidas del inventario. Realizar validaciones afectación de disponibilidad y cuadro de cantidades y valores.	5
6	Gestionar Carro de Compra	Buscar, ordenar y filtrar productos. Agregar reserva de productos. Editar reserva de productos. Eliminar reserva de productos. Listar reserva de productos. Realizar validaciones afectación de disponibilidad.	6
7	Gestionar Lista de Deseos	Buscar, ordenar y filtrar productos. Agregar selección de productos. Editar selección de productos. Eliminar selección de productos. Listar selección de productos. Realizar validaciones afectación de disponibilidad.	7
8	Gestionar Seguridad del Sistema	Agregar grupos de seguridad. Agregar usuarios anclados a grupos de seguridad. Establecer permisos y accesos a funcionalidades.	8
9	Gestionar Auditoria del Sistema	Registrar eventos a partir de los accesos a funcionalidades. Listar los eventos registrados según el formato pertinente.	9
10	Gestionar Informes y Estadísticas	Crear informes y estadísticas para el estudio y control de la información almacenada.	10

Tabla 9 Tareas de Ingeniería (elaboración propia).

2.5 Análisis de Costo.

Según (Pressman, 2010), la estimación es el proceso de medición anticipada de la duración, esfuerzos y costes necesarios para desarrollar todas las actividades y obtener todos los productos comprendidos en un proyecto. Para ello, es sumamente necesario tener en cuenta numerosos aspectos que afectan el proyecto como son: la complejidad del mismo, su estructuración y tamaño, los recursos que intervienen y los riesgos asociados.

Para realizar este análisis existen distintos modelos entre los que se encuentran COCOMO I, COCOMO II, Puntos de Función, Diseño Temprano, Composición de Aplicación, Reutilización, entre otros. En la investigación se selecciona como modelo para estimar los costos el de Puntos de Función.

(Pressman, 2010) plantea que la métrica de punto de función (PF) puede emplearse de manera efectiva como medio para medir la funcionalidad que entra a un sistema. Y al usar datos históricos, la métrica PF puede entonces utilizarse para estimar el costo o esfuerzo requerido para diseñar, codificar y probar el software.

Componentes:

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

- (Pantallas donde el usuario ingresa datos)

EO: Procesos en los que se envía datos al exterior de la aplicación.

- (Informes, gráficos, Listado de datos)

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.

- (Recuperar y mostrar datos al usuario (Buscar))

ILF: Grupos de datos relacionados entre sí internos al sistema.

- Archivo del punto de vista lógico, no como en un sistema operativo.
- Pueden ser tablas en la base de datos

EIF: Grupos de datos que se mantienen externamente.

- Datos referenciados a otros sistemas.
- Datos mantenidos por otros sistemas, pero usados por el sistema actual.

Una vez obtenidos los diferentes elementos del sistema se utilizan las tablas reflejadas en el [Anexo 3](#) para asignar la importancia en función del número de atributos que tengan y el número de archivos a los que afecte.

Cálculos de puntos de función.

Valor de dominio de información	Conteo		Factor ponderado				
			Simple	Promedio	Complejo		
Entradas externas (EI)	3	x	3	4	6	=	12
Salidas externas (EO)	4	x	4	5	7	=	20
Consultas externas (EQ)	6	x	3	4	6	=	24
Archivos lógicos internos (ILF)	7	x	7	10	15	=	70
Archivos de interfaz externos (EIF)	0	x	5	7	10	=	0
Conteo total							126

Tabla 10 Cálculos de puntos de función (Pressman, 2010).

Cálculo de los Puntos de Función Sin Ajustar (PFSA):

Los PFSA se calculan como la suma de los productos de cada valor de dominio por su factor ponderado determinado en la tabla anterior.

$$PFSA = 12 + 20 + 24 + 70 + 0$$

$$PFSA = 126$$

Cálculo del Factor de Ajuste (FA)

Para calcular el FA se le va dando un valor entre 0 y 5 a cada Factor de Ajuste como se expone en la tabla perteneciente al [Anexo 4](#). Cuando cada Factor de Ajuste tenga un valor, se suman todos y así obtenemos el FA. A continuación, se muestra en la Tabla 11 como se refleja dicho procedimiento.

Cálculo del Factor de Ajuste de Valor

No.	Factor de Ajuste	Puntos
1	Comunicación de Datos	4
2	Proceso Distribuido	5
3	Objetivos de Rendimiento	2
4	Configuración de Explotación Compartida	1
5	Tasa de transacciones	3
6	Entrada de Datos en Línea	5

7	Eficiencia con el Usuario Final	4
8	Actualizaciones en Línea	1
9	Lógica de Proceso Interno Compleja	1
10	Reusabilidad del Código	4
11	Conversión e Instalación contempladas	1
12	Facilidad de Operación	4
13	Instalaciones Múltiples	1
14	Facilidad de Cambios	4
	Factor de Ajuste	40

Tabla 11 Cálculo del Factor de Ajuste de Valor (Pressman, 2010).

Cálculo de los Puntos de Función Ajustados (PFA)

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

$$PFA = 126 * [0.65 + (0.01 * 40)]$$

$$PFA = 132.3$$

Cálculo del Esfuerzo

H/H: Horas hombre

$$H/H = PFA * \text{Horas PF promedio}$$

Para obtener las Horas PF promedio, nos apoyamos en la Tabla del [Anexo 5](#).

$$H/H = 132.3 * 8$$

$$H/H = 1058 \text{ Horas hombre}$$

Cálculo de Tiempo Laborable

Equipo Scrum	Jornada Laborar (h)	Horas de trabajo al proyecto por día (h)	Horas de trabajo al proyecto por semana (h)	Semanas de Trabajo por mes (u)	Total de horas (h)	Total de días laborables en el mes (días)
Renier Martínez	8	7	42	4	168	24

Tabla 12 Cálculo de Tiempo Laborable (elaboración propia).

Duración del proyecto en horas = (1058 Horas/hombre) / 1 hombre = 1058 horas por miembro

1 Mes = 168 horas laborables

Duración del proyecto en meses = 1058 horas / (168 horas/mes) = 6 meses

Cálculo del Presupuesto del Proyecto

Sueldo mensual 1 participante = 350.00 CUP

Costo Total del proyecto = sueldo 1 participante * cantidad participantes * tiempo de desarrollo

Costo Total del Proyecto = 350.00 * 1 * 6

Costo Total del Proyecto = 2100.00 CUP

2.6 Conclusiones parciales del capítulo.

Mediante la metodología SCRUM, en este capítulo 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 los respectivos Sprints 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.

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 III Resultados obtenidos y pruebas realizadas al software

3.1 Introducción.

Un software cuenta de varias etapas durante su desarrollo, entre las cuales se encuentra la de pruebas que permiten conocer el grado de calidad del mismo y el cumplimiento de los requisitos funcionales identificados con anterioridad en la etapa de planificación. En este capítulo se describe el proceso de prueba, el cual posibilita hallar la mayor cantidad de errores existentes en la aplicación y de esa manera erradicarlos. A continuación, se aborda las pruebas funcionales, de usabilidad y seguridad realizadas, y se brinda un análisis de los resultados que se han obtenido al concluir las mismas.

3.2 Pruebas realizadas y resultados.

Las pruebas intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para buscar errores, anomalías o información de atributos no funcionales del programa (Sommerville, 2011).

El proceso de prueba tiene dos metas distintas:

1. Demostrar al desarrollador y al cliente que el software cumple con los requerimientos.
2. Encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con su especificación.

3.2.1 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).

Previo a concluir cada Sprint o iteración, se realizaron las pruebas correspondientes a las funcionalidades desarrolladas en ese momento. Las mismas permitieron realizar la evaluación del cumplimiento de los objetivos especificados por el cliente. A continuación, se presentan algunas de estas pruebas y los resultados que arrojaron.

Tales pruebas son elaboradas por el propio equipo de desarrollo y sus resultados son guardados en tablas que cuentan con los campos que se exponen a continuación:

Número del caso de Prueba: Número que tendrá el caso de prueba a realizar.

Número de Historia de Usuario: Número que identifica la Historia de Usuario abordada.

Descripción: Contiene una breve descripción de la prueba realizada.

Condiciones de Ejecución: Las condiciones necesarias para que se pueda realizar la prueba.

Entradas / Pasos de Ejecución: Serie de pasos enumerados para lograr realizar la prueba.

Resultado esperado: Breve descripción del resultado que se espera obtener con la prueba realizada.

Evaluación de la prueba: acorde al resultado de la prueba realizada se emitirá una evaluación sobre la misma. Esta evaluación tendrá uno de los tres resultados que a continuación se describen:

- I. Satisfactoria: cuando el resultado de la prueba es exactamente el esperado por el usuario.
- II. Parcialmente satisfactoria: cuando el resultado no es completamente el esperado por el cliente o usuario de la aplicación y muestra resultados erróneos o fuera de contexto.

A continuación, se especifican algunas pruebas de aceptación elaboradas.

Pruebas de Aceptación	
Número de caso de prueba: 1	Número de Historia de Usuario: 8
Nombre del caso de prueba: Seguridad y autenticación de usuario	
Descripción: Se ingresa el nombre de usuario y contraseña en la aplicación para iniciar sesión. Se realiza esta acción de manera incorrecta, especificando usuarios inexistentes, contraseñas no válidas y dejando campos en blanco. Luego se intenta con datos correctos. Posteriormente se trata de acceder a rutas a las cuales no se posee permisos de acceso. Posteriormente se procede a las permitidas.	
Condiciones de ejecución: El usuario dispondrá de las funcionalidades que les están permitidas según los permisos otorgados.	
Entrada/Pasos de ejecución: Ingresar usuario incorrecto con contraseña correcta al autenticarse. Ingresar usuario correcto con contraseña incorrecta al autenticarse. Ingresar usuario y contraseña correctos al autenticarse. Intentar acceder a rutas sin permisos de acceso. Intentar acceder a rutas con permisos de acceso.	
Resultado esperado: La aplicación debe mostrar mensaje de alerta acerca de la introducción de datos erróneos al intentar autenticarse. Mostrar mensaje de error al acceder a rutas de las cuales no se posee permiso de acceso. El sistema debe permitir el inicio de sesión cuando se ingresan correctamente usuario y contraseña. Acceder a las rutas especificadas cuando se dispone de los permisos de acceso.	
Evaluación de la prueba: Satisfactoria	

Tabla 13 PA1 Seguridad y autenticación de usuario (elaboración propia).

Pruebas de Aceptación	
Número de caso de prueba: 2	Número de Historia de Usuario: 4
Nombre del caso de prueba: Gestión de contrato	
Descripción: Se ingresa el código identificador y demás campos para crear un nuevo contrato. Se deja de especificar campos requeridos y a otros se les introducen datos	

<p>con el formato y tipo incorrectos. Posteriormente se realiza el mismo procedimiento de manera correcta y se verifica el registro de la información en la base de datos. Se recupera el contrato guardado y se lista ofreciendo los datos relevantes acerca del mismo.</p> <p>Se editan los datos del contrato introduciendo información de forma incorrecta. Luego se realiza la misma acción de la manera adecuada.</p> <p>Se intenta eliminar el contrato especificando previamente su decisión mediante un cuadro de dialogo.</p>
<p>Condiciones de ejecución: Los usuarios que cuentan con los permisos para realizar este proceder son los que poseen el rol de Jurídico. Debe haberse introducido anteriormente entidades a las cuales se le realizara el contrato y sus respectivos usuarios que la representan.</p>
<p>Entrada/Pasos de ejecución:</p> <p>Ingresar datos de manera incorrecta.</p> <p>Dejar campos requeridos sin llenar.</p> <p>Ingresar datos correctamente.</p> <p>Editar los datos de manera incorrecta.</p> <p>Editar los datos de manera correcta.</p> <p>Listar la información guardada previamente.</p> <p>Eliminar la información guardada previamente.</p>
<p>Resultado esperado: La aplicación debe alertar al usuario a medida que se conduce a introducir los datos por los campos de forma incorrecta, y de igual manera al realizar el guardado de los mismos. Cuando los datos son introducidos correctamente debe guardar la información en la base de datos y posteriormente listar la misma. El sistema debe obtener la confirmación del usuario al intentar eliminar la información.</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Tabla 14 PA2 Gestión de contrato (elaboración propia).

Pruebas de Aceptación	
Número de caso de prueba: 3	Número de Historia de Usuario: 3
Nombre del caso de prueba: Gestión de nomencladores	

<p>Descripción: Se ingresa el código identificador y demás campos para crear un nuevo nomenclador. Se deja de especificar campos requeridos y a otros se les introducen datos con el formato y tipo incorrectos. Posteriormente se realiza el mismo procedimiento de manera correcta y se verifica el registro de la información en la base de datos. Se recupera el nomenclador guardado y se lista ofreciendo los datos relevantes acerca del mismo.</p>
<p>Condiciones de ejecución: Los usuarios que cuentan con los permisos para realizar este proceder son los que poseen el rol de Comercial. Debe haberse introducido anteriormente nomencladores previos a relacionar según lo requiera el nomenclador tratado.</p>
<p>Entrada/Pasos de ejecución:</p> <p>Ingresar datos de manera incorrecta.</p> <p>Dejar campos requeridos sin llenar.</p> <p>Ingresar datos correctamente.</p> <p>Editar los datos de manera incorrecta.</p> <p>Editar los datos de manera correcta.</p> <p>Listar la información guardada previamente.</p> <p>Eliminar la información guardada previamente.</p>
<p>Resultado esperado: La aplicación debe alertar al usuario a medida que se conduce a introducir los datos por los campos de forma incorrecta, y de igual manera al realizar el guardado de los mismos. Cuando los datos son introducidos correctamente debe guardar la información en la base de datos y posteriormente listar la misma. El sistema debe obtener la confirmación del usuario al intentar eliminar la información y en caso de existir otros registros que dependen del actual, alertar al respecto e impedir su eliminación.</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Tabla 15 PA3 Gestión de nomencladores (elaboración propia).

Pruebas de Aceptación	
Número de caso de prueba: 4	Número de Historia de Usuario: 5
Nombre del caso de prueba: Gestión de Compra	

<p>Descripción: Se selecciona la entidad proveedora y de igual manera otros datos son ingresados para crear una nueva compra. Se deja de especificar campos requeridos y a otros se les introducen datos con el formato y tipo incorrectos. Posteriormente se realiza el mismo procedimiento de manera correcta y se verifica el registro de la información en la base de datos. Se recupera la compra guardada y se lista ofreciendo los datos relevantes acerca de la misma.</p>
<p>Condiciones de ejecución: Los usuarios que cuentan con los permisos para realizar este proceder son los que poseen el rol de Contador. Debe haberse introducido anteriormente entidades, representantes, contratos y áreas que se encuentren asociados a la compra.</p>
<p>Entrada/Pasos de ejecución:</p> <ul style="list-style-type: none"> Ingresar datos de manera incorrecta. Dejar campos requeridos sin llenar. Ingresar datos correctamente. Editar los datos de manera incorrecta. Editar los datos de manera correcta. Listar la información guardada previamente. Eliminar la información guardada previamente.
<p>Resultado esperado: La aplicación debe alertar al usuario a medida que se conduce a introducir los datos por los campos de forma incorrecta, y de igual manera al realizar el guardado de los mismos. No debe permitir la selección de una entidad que no posea contrato de compra y este no se encuentre vigente. Debe alertar cuando existe ausencia de información en campos vitales como son la cantidad y el precio. Cuando los datos son introducidos correctamente debe guardar la información en la base de datos y posteriormente listar la misma.</p>
<p>Evaluación de la prueba: Satisfactoria</p>

Tabla 16 PA4 Gestión de Compra (elaboración propia).

3.3 Conclusiones parciales del capítulo.

Al concluir la propuesta de solución al problema científico de la investigación a partir del empleo del marco de trabajo SCRUM, y realizando un análisis de los resultados de la misma mediante el empleo 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 Generales

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, el estado del arte, 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. Se determinó el empleo como herramientas de software más factibles para la construcción de la solución el lenguaje de programación Python, framework Django, servidor web Apache, PyCharm como IDE y como gestor de base de datos PostgreSQL.
4. Se eligió el uso del marco de trabajo SCRUM para el desarrollo del sistema informático por las características del proyecto a realizar.
5. Se realizó el análisis de costo de desarrollo del software, arrojando como resultado la factibilidad de la realización del mismo.
6. El desarrollo del software y la aplicación de las pruebas de funcionalidad con resultados satisfactorios demostraron que la aplicación web construida satisface las especificaciones del cliente.

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 comercial de la UEB DIVEP Matanzas con mayor eficiencia.

Recomendaciones

Luego de analizar los resultados obtenidos de esta investigación se recomienda:

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.
3. Generalizar la aplicación web al resto de las Unidades Empresariales de Base para facilitar la gestión de la actividad comercial en el país.

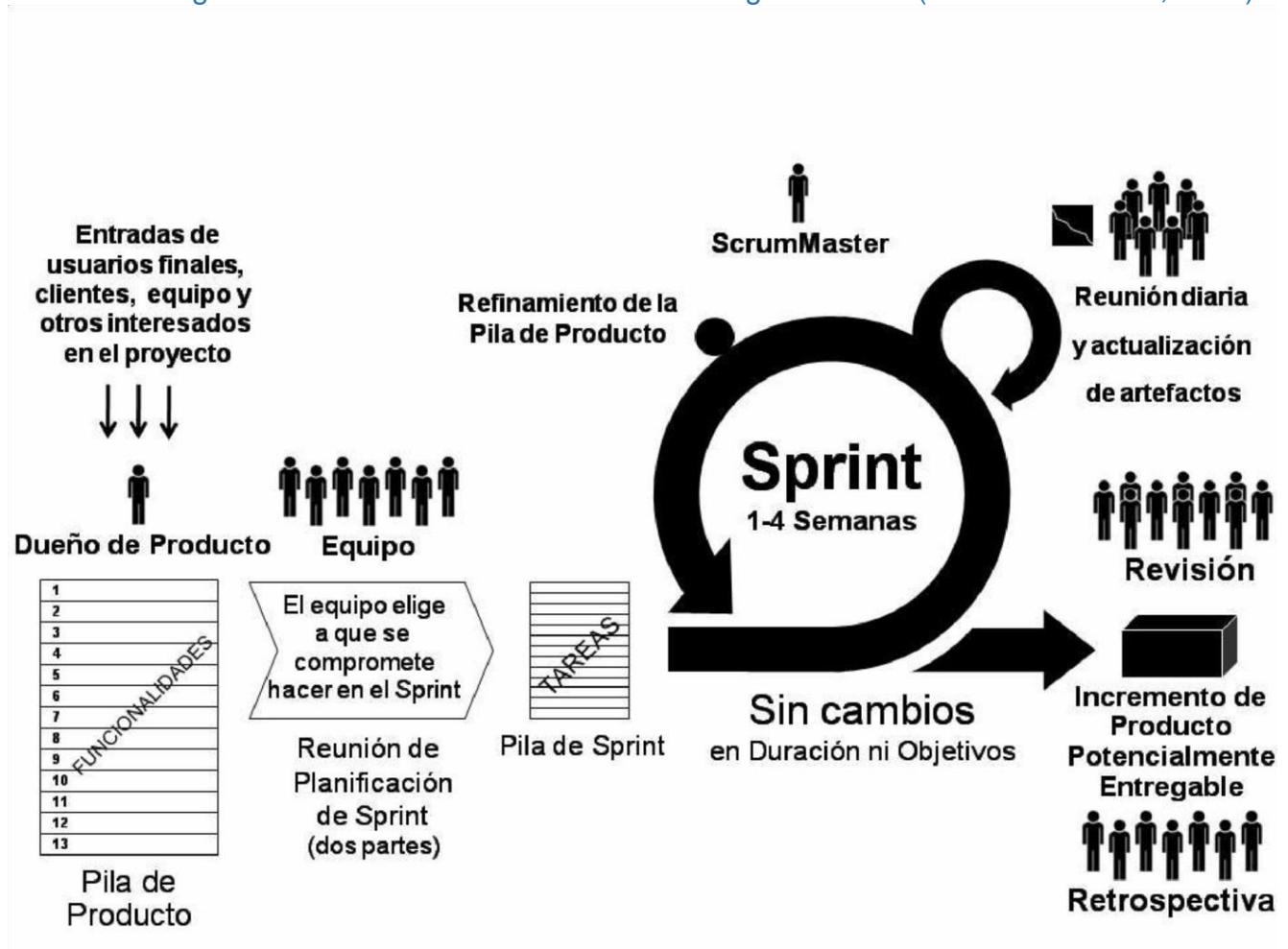
Bibliografía

- ALVAREZ CASADIEGO, C. A. (2012). *ESTUDIO DEL USO DE LA METODOLOGIA AGIL SCRUM EN EL DESARROLLO DE UNA APLICACION NATIVA PARA LA PLATAFORMA ANDROID EN UN ENTORNO CON REQUISITOS CAMBIANTES CASO DE APLICACIÓN APPCOMPOSITORES*. UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA, OCAÑA, COLOMBIA.
- Cáceres Tello, J.). Patrones de diseño: ejemplo de aplicación en los Generative Learning Object, 13.
- Camarena Sagredo, J. G., Trueba Espinosa, A., Martínez Reyes, M., López, & García, M. d. L. (2012). Automatización de la codificación del patrón modelo vista controlador (MVC) en proyectos orientados a la Web. *Ciencia Ergo Sum*, 13.
- Challenger Pérez, I., Díaz Ricardo, Y., & Becerra García, R. A. (2014). El lenguaje de programación Python. *Ciencias Holguín*, vol. XX, 14.
- Denzer, P. (2002). [PostgreSQL].
- Downey, A., Elkner, J., & Meyers, C. (2002). *Aprenda a Pensar Como un Programador con Python*. In (pp. 312).
- Eguíluz Pérez, J. (2008). *Introducción a CSS*. In C. Commons (Ed.). Retrieved from <http://www.librosweb.es/css>
- Espinoza-Meza, A. (2013). *MANUAL PARA ELEGIR UNA METODOLOGÍA DE DESARROLLO DE SOFTWARE DENTRO DE UN PROYECTO INFORMÁTICO*. Universidad de Piura, Universidad de Piura.
- García M., S. (2015). *La guía definitiva de Django: Desarrolla aplicaciones web de forma rápida y sencilla*. In. Retrieved from <http://github.com/saulgm/djangobook.com>
- Gómez Mondragón, A. (2016). *Propuesta para el mejoramiento del proceso de fabricación del producto Zanjadora para Alce Hidráulico mediante el uso de la gestión de procesos de negocio (BPM)*. (Maestría en Ingeniería), Pontificia Universidad Javeriana. Cali., Santiago de Cali.
- González-Barahona, J. M. (2011, 2011/12). El concepto de software libre. *Revista Tradumàtica: tecnologies de la traducció*, 7.
- J., L., & G., C. (2019). *Materiales del entrenamiento de programación en Python - Nivel básico*. In C. R.L. (Ed.), (pp. 287).
- Lizama, O., Kindley, G., & Jeria Morales, J. I. (2016). *Redes de computadores. Arquitectura Cliente - Servidor*. 8.
- Marini, E. (2012). [El Modelo Cliente/Servidor].
- Mariño, S. I., & Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. *Vol. 19, 7*. <http://www.redalyc.org/articulo.oa?id=84933912009>
- Muñoz Serafín, M. (2018). *Introducción al desarrollo de aplicaciones N-Capas con tecnologías Microsoft*. In (pp. 18).
- OTAVALO MOROCHO, M. D. (2019). IMPLEMENTACIÓN DE FLUJOGRAMAS EN ACTIVIDADES REALIZADAS EN LA FARMACIA MISHELL MEDIANTE LA SIMBOLOGÍA BIZAGI PARA MEJORAR LOS TIEMPOS DE EJECUCIÓN. In (pp. 25). FACULTAD DE CIENCIAS EMPRESARIALES. Universidad Técnica de Machala.: Universidad Técnica de Machala.
- Pressman, R. S. (2010). *"INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO"*. In S. A. D. C. V. McGRAW-HILL INTERAMERICANA EDITORES (Ed.).
- Sommerville, I. (2011). *"INGENIERÍA DE SOFTWARE"*. In (pp. 792).

- Stallman, R. M. (2004). *Software libre para una sociedad libre*. In M. Vidal (Ed.), *Free Software, Free Society* (pp. 232).
- VILCA OQUENDO, D. H. (2017). *OPTIMIZACIÓN DEL PROCESO DE DESARROLLO DE PROYECTOS, BAJO EL ENFOQUE DE BUSINESS PROCESS MANAGEMENT (BPM) EN EL ÁREA DE INTEGRACION DE APLICACIONES EMPRESARIALES(EAI) EN UNA EMPRESA DE TELECOMUNICACIONES*. UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR, Villa El Salvador.
- Villarreal Fuentes, C. A. (2013). *¿Qué es jQuery? Northware Software Development*, 6.
- Zea Ordóñez, M. P., Molina Ríos, J. R., & Redrován Castillo, F. F. (2017). *ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL*. In S. L. Editorial Área de Innovación y Desarrollo (Ed.), (pp. 82).

Anexos

Anexo 1: Diagrama de funcionamiento de la metodología SCRUM (Mariño & Alfonzo, 2014).



Anexo 2: Historias de usuarios

Historia de usuario 01.

HISTORIA DE USUARIO	
ID: HU01	Usuario: Administrador del Sistema
Nombre Historia: Diseño y creación de la Base de Datos	
Prioridad en el Negocio: 1	Importancia del Desarrollo: 10
Tiempo Estimado: 1 semana	Sprint: 1
Descripción: Diseñar e implementar la base de datos que almacenará la información.	
Observaciones: El diseño debe garantizar la integridad de la información.	

Fuente: Elaboración Propia.

Historia de usuario 02.

HISTORIA DE USUARIO	
ID: HU02	Usuario: Administrador del Sistema
Nombre Historia: Diseño y creación de la Interfaz de Usuario	
Prioridad en el Negocio: 2	Importancia del Desarrollo: 10
Tiempo Estimado: 2 semanas	Sprint: 2
Descripción: Diseñar y elaborar la interfaz de usuario.	
Observaciones: La interfaz debe mostrarse atractiva y amigable.	

Fuente: Elaboración Propia.

Historia de usuario 03.

HISTORIA DE USUARIO	
ID: HU03	Usuario: Comercial
Nombre Historia: Gestionar nomencladores	
Prioridad en el Negocio: 3	Importancia del Desarrollo: 8
Tiempo Estimado: 3 semanas	Sprint: 3
Descripción: Gestionar la creación y actualización de nomencladores.	
Observaciones: La aplicación debe garantizar la coherencia de la información.	

Fuente: Elaboración Propia.

Historia de usuario 04.

HISTORIA DE USUARIO	
ID: HU04	Usuario: Jurídico
Nombre Historia: Gestionar entidades – contratos	
Prioridad en el Negocio: 4	Importancia del Desarrollo: 8
Tiempo Estimado: 3 semanas	Sprint: 4
Descripción: Gestionar la creación y actualización de entidades, representantes de estas y sus respectivos contratos.	
Observaciones: El sistema debe garantizar la especificación correcta de los términos de contratación.	

Fuente: Elaboración Propia.

Historia de usuario 05.

HISTORIA DE USUARIO	
ID: HU05	Usuario: Contador
Nombre Historia: Gestionar Tienda (entradas, salidas)	
Prioridad en el Negocio: 5	Importancia del Desarrollo: 10
Tiempo Estimado: 3 semanas	Sprint: 5
Descripción: Gestionar los movimientos de los productos referido a su inventario.	
Observaciones: Se debe garantizar la realización de entradas, salidas y de los productos de forma que posibilite el control adecuado sobre los mismos.	

Fuente: Elaboración Propia.

Historia de usuario 06.

HISTORIA DE USUARIO	
ID: HU06	Usuario: Cliente
Nombre Historia: Gestionar Carro de Compra	
Prioridad en el Negocio: 6	Importancia del Desarrollo: 8
Tiempo Estimado: 1 semana	Sprint: 6
Descripción: Gestionar la selección de productos y su posterior reservación.	

Observaciones: Se debe proporcionar los medios de búsqueda, organización y filtrado de los productos para una ágil y fácil selección de los mismos.

Fuente: Elaboración Propia.

Historia de usuario 07.

HISTORIA DE USUARIO	
ID: HU07	Usuario: Cliente
Nombre Historia: Gestionar Lista de Deseos	
Prioridad en el Negocio: 7	Importancia del Desarrollo: 6
Tiempo Estimado: 1 semana	Sprint: 7
Descripción: Gestionar la selección de productos y adecuada notificación.	
Observaciones: Se debe proporcionar los medios de búsqueda, organización y filtrado de los productos para una ágil y fácil selección de los mismos. Así como garantizar la notificación pertinente ante los cambios de disponibilidad.	

Fuente: Elaboración Propia.

Historia de usuario 08.

HISTORIA DE USUARIO	
ID: HU08	Usuario: Administrador del Sistema
Nombre Historia: Gestionar Seguridad del Sistema	
Prioridad en el Negocio: 8	Importancia del Desarrollo: 8
Tiempo Estimado: 1 semana	Sprint: 8
Descripción: Gestionar la seguridad que precisa al sistema.	
Observaciones: Se debe garantizar los permisos y accesos a las distintas funcionalidades que brinda la aplicación según el rol que posean los usuarios.	

Fuente: Elaboración Propia.

Historia de usuario 09.

HISTORIA DE USUARIO	
ID: HU09	Usuario: Administrador del Sistema

Nombre Historia: Gestionar Auditoria del Sistema	
Prioridad en el Negocio: 9	Importancia del Desarrollo: 7
Tiempo Estimado: 1 semana	Sprint: 9
Descripción: Gestionar el registro de accesos al sistema y su posterior auditoria.	
Observaciones: Asegurar que quede registrado los accesos a las distintas opciones del sistema y brinden una robusta información al respecto.	

Fuente: Elaboración Propia.

Historia de usuario 10.

HISTORIA DE USUARIO	
ID: HU10	Usuario: Administrador del Sistema
Nombre Historia: Gestionar Informes y Estadísticas	
Prioridad en el Negocio: 10	Importancia del Desarrollo: 5
Tiempo Estimado: 2 semana	Sprint: 10
Descripción: Gestionar los informes y estadísticas que proporcionan control de la información.	
Observaciones: Proporcionar páginas, informes y ficheros que contengan la información en el formato adecuado para la interpretación correcta de los datos que se almacenan.	

Fuente: Elaboración Propia.

Anexo 3: Tabla de ponderaciones para EI, EQ, EO, ILF, EIF

Clasificación de Entradas Consultas	1-4 Atributos	5-15 Atributos	Más de 15 Atributos
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

Clasificación de Salidas	1-4 Atributos	6-19 Atributos	Más de 19 Atributos
0 o 1 ficheros accedidos	Baja 4	Baja 4	Media 5
2 o 3 ficheros accedidos	Baja 4	Media 5	Alta 7
Más de 3 ficheros accedidos	Media 5	Alta 7	Alta 7

Ficheros Lógicos Internos	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro	Baja 7	Baja 7	Media 10
2-5 Entidades o registros	Baja 7	Media 10	Alta 15
Más de 5 ficheros accedidos	Media 10	Alta 15	Alta 15

Ficheros Lógicos Externos	1-19 Atributos	20-50 Atributos	Más de 50 Atributos
1 Entidad o registro	Baja 5	Baja 5	Media 7
2-5 Entidades o registros	Baja 5	Media 7	Alta 10
Más de 5 ficheros accedidos	Media 7	Alta 10	Alta 10

Anexo 4: Tabla para obtener el Factor de Ajuste de Valor (FAC).

No.	Factor de Ajuste	Puntos	
		Min	Max
1	Comunicación de Datos	0	5
2	Proceso Distribuido	0	5
3	Objetivos de Rendimiento	0	5
4	Configuración de Explotación Compartida	0	5
5	Tasa de transacciones	0	5
6	Entrada de Datos en Línea	0	5
7	Eficiencia con el Usuario Final	0	5
8	Actualizaciones en Línea	0	5
9	Lógica de Proceso Interno Compleja	0	5
10	Reusabilidad del Código	0	5
11	Conversión e Instalación contempladas	0	5
12	Facilidad de Operación	0	5
13	Instalaciones Múltiples	0	5
14	Facilidad de Cambios	0	5

Anexo 5: Tabla para obtener Horas por Puntos de Función (HPF).

Entorno y Lenguaje	Líneas de Código por PF	Horas por PF promedio
Lenguajes de 2GL: Ensamblador, C...	300	25
Lenguajes de 3GL: Cobol	100	15
Lenguajes de 4GL: Python	20	8