



**ESPE**  
**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**“Desarrollo de una aplicación web, mediante el uso de software libre, utilizando la metodología xp, basado en la plantilla AdminLTE para el control de inventario y facturación de productos de la Agrícola Moreplant”**

Taipe Rocha, Carla Estefanía

Departamento de Ciencias de la Computación

Carrera de Tecnología en Computación.

Monografía, previo a la obtención del título de Tecnólogo en Computación.

Ing. Casa Guayta, Carlos Wellington

03 de marzo del 2021

## Certificación



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE TECNOLOGÍA EN COMPUTACIÓN.  
CERTIFICACION

Certifico que la monografía, "Desarrollo de una aplicación web, mediante el uso de software libre, utilizando la metodología xp, basado en la plantilla AdminLTE para el control de inventario y facturación de productos de la Agrícola Moreplant", fue realizado por la señorita Taipe Rocha Carla Estefania, la cual ha sido revisada y analizada en su totalidad por la herramienta de verificación de similitud de contenidos; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas Espe, razón por la cual me permito acreditar y autorizar para que los sustente públicamente.

Latacunga 03, de Marzo del 2021

Firma

Ing. Carlos Wellington Casa Guayta

C.C.: 0502352180

## Reporte de Verificación



### Document Information

Analyzed document	Taipe Rocha Carla Estefania.docx (D97120143)
Submitted	3/3/2021 6:12:00 PM
Submitted by	
Submitter email	cetaipe@espe.edu.ec
Similarity	7%
Analysis address	cwcasa.espe@analysis.arkund.com

### Sources included in the report

<b>W</b>	URL: <a href="https://repositorio.umsa.bo/bitstream/handle/123456789/7939/T.2829.pdf?sequence=1">https://repositorio.umsa.bo/bitstream/handle/123456789/7939/T.2829.pdf?sequence=1</a> Fetched: 8/8/2020 12:12:29 AM	11
<b>W</b>	URL: <a href="https://fr.slideshare.net/EvelingGiselleCruzVs/metodologia-monografia">https://fr.slideshare.net/EvelingGiselleCruzVs/metodologia-monografia</a> Fetched: 7/26/2020 7:59:17 PM	2
<b>W</b>	URL: <a href="https://docplayer.es/80214920-Departamento-de-ciencias-de-la-computacion.html">https://docplayer.es/80214920-Departamento-de-ciencias-de-la-computacion.html</a> Fetched: 1/21/2021 2:09:40 AM	6
<b>W</b>	URL: <a href="https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc...">https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc...</a> Fetched: 3/3/2021 6:12:00 PM	2
<b>SA</b>	<b>Universidad de las Fuerzas Armadas ESPE / MOROCHO ROLDAN JUAN DANIEL.pdf</b> Document MOROCHO ROLDAN JUAN DANIEL.pdf (D78835361) Submitted by: jdmorocho@espe.edu.ec Receiver: dkchicaiza.espe@analysis.arkund.com	12
<b>SA</b>	<b>Universidad de las Fuerzas Armadas ESPE / MOROCHO ROLDAN JUAN DANIEL.docx</b> Document MOROCHO ROLDAN JUAN DANIEL.docx (D78835362) Submitted by: jdmorocho@espe.edu.ec Receiver: dkchicaiza.espe@analysis.arkund.com	4
<b>SA</b>	<b>MAURICIO YÉPEZ.pdf</b> Document MAURICIO YÉPEZ.pdf (D62175851)	1

## Responsabilidad de Autoría



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN.

RESPONSABILIDAD DE AUTORÍA

Yo, Taipe Rocha Carla Estefanía, cedula de ciudadanía N°1723114425 declaro que el contenido, ideas y criterios de la monografía: "Desarrollo de una aplicación web, mediante el uso de software libre, utilizando la metodología ~~xx~~, basado en la plantilla AdminLTE para el control de inventario y facturación de productos de la Agrícola Moreplant", es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas Espe, razón por la cual me permito acreditar y autorizar para que los sustente públicamente.

Latacunga 03, de marzo del 2021

Firma

-|

-----  
Taipe Rocha, Carla Estefanía

C.C.:1723114425

## Autorización de Publicación



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍA EN COMPUTACIÓN.

### AUTORIZACION DE PUBLICACIÓN

Yo Taipei Rocha, Carla Estefania con cedula de ciudadanía N° 1723114425 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar la monografía "Desarrollo de una aplicación web, mediante el uso de software libre, utilizando la metodología xp, basado en la plantilla AdminLTE para el control de inventario y facturación de productos de la Agrícola Moreplant", en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga 25, de febrero del 2020

Firma

Taipe Rocha, Carla Estefania

C.C.:1723114425

### **Dedicatoria**

No es fácil mencionar a todas las personas que han sido parte de este trayecto en mi vida estudiantil ya que de alguna manera u otra han sido participes han colaborado e influido en el desarrollo del presente trabajo, mi principal dedicatoria es a mí madre (Elsa Beatriz Rocha Toaquiza), por su entrega total en el desarrollo de este nuevo desafío, por la paciencia que ha tenido antes, durante y después de toda la investigación y por todo el tiempo designado permitiéndome estudiar en un ambiente satisfactorio y tranquilo, sin su apoyo creo que esto no hubiera sido posible. Agradezco a Dios por que mi madre ha sido el pilar fundamental y apoyo en mi formación académica. Por ser una madre maravillosa, ejemplar del mundo que Dios me ha dado la dedico este trabajo.

A mis hermanos

Víctor, Hernán, Daniela, Carlos quienes han sido participes en este trayecto de lucha y cumplir unas de mis metas planteadas, agradezco por todo el apoyo moral que me han brindado en mi formación académica.

A mis sobrinos

Micaela, Josué, Romina son ustedes mi mayor motivación verlos crecer y saber que estamos por el mismo camino de aprendizaje, los agradezco por cada sonrisa y abrazos de ustedes era la motivación de seguir formándome como profesional.

A mi cuñada

Rake quien has sido y has cumplido con un rol muy importante en mi vida, tu apoyo moral, la motivación que me has brindado me ha servido mucho para dar oído a lo verdadero importante, agradezco por todo lo brindado.

**TAIPE ROCHA, CARLA ESTEFANIA**

## **Agradecimiento**

A Dios, por darme vida, salud y sobre todo sabiduría en este transcurso de mi vida estudiantil, ahora he cumplido un objetivo planteado, no me queda más que agradecer a Dios.

A mi familia en especial a mi madre, hermanos y sobrinos por ser el impulso en mi vida ya que al verlos día a día era mi motivación y continuar en la lucha diaria, agradezco todo su apoyo y entrega que me brindaron en estos años de estudio. ¡Gracias por todo su apoyo!

A mi novio Luis por formar parte de mi vida, por haberme brindado su tiempo, espacio en mi vida estudiantil, agradezco que siempre hayas estado cuando más te necesitaba, los consejos, tu apoyo moral me sirvió para continuar y guiarme en ti como profesional, te agradezco infinitamente que aun permanezcas en mi vida y decirte que te amo.

A mi amigo Cristian Chuquitarco quien has sido ese único amigo que no me ha dejado en ningún momento me has ayudado en conocimiento y como persona, agradezco que hayas estado en todo este proceso de vida estudiantil.

Así como también a todos quienes conforman la Carrera de Tecnología en Computación y a la Universidad de las Fuerzas Armadas ESPE que de una u otra manera hicieron posible llegar a culminar el presente estudio.

Al Ing.: Carlos Wellington Casa Guayta (Tutor de tesis), Ing. Jorge Pardo (Director de Carrera) por compartir y permitir ser partícipe de sus valiosos conocimientos, sugerencias y comentarios que me permitieron desenvolverme con éxito en el transcurso de la presente investigación.

**TAIPE ROCHA, CARLA ESTEFANIA**

## Tabla de Contenido

Certificación .....	2
Reporte de verificación .....	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Agradecimiento.....	7
Índice de contenidos .....	8
Índice de tablas .....	8
Índice figuras.....	15
Resumen .....	18
Abstract.....	19
Problema de investigación .....	20
Tema ....	20
Antecedente.....	20
Planteamiento del problema.....	21
Justificación .....	23
Objetivos .....	23
<i>Objetivo general</i> .....	24
<i>Objetivos específicos</i> .....	24
Alcance .....	24
Marco teórico .....	26
Definiciones Básicas .....	26



<i>Qué es un software.</i> .....	26
<i>¿Clasificación de software de sistema?</i> .....	26
<i>¿ Qué es Software de Programación?</i> .....	26
<i>¿ Que es Software de Aplicación?</i> .....	26
<i>¿Qué es ingeniería en software?</i> .....	27
<i>¿Qué es un proceso de software?</i> .....	27
<i>¿Qué es un ciclo de software?</i> .....	27
<i>Modelo de un Software</i> .....	28
<i>Modelado de sistemas.</i> .....	28
<i>Contruccion de Modelos.</i> .....	28
Historia de una Aplicación Web.....	29
<i>Concepto Aplicación Web</i> .....	29
<i>Beneficios de la aplicación web</i> .....	30
Estándar IEEE 830-1998.....	30
Norma Iso 27001 .....	31
Requerimientos de software. ....	32
<i>Los requerimientos del usuario (Alto Nivel).</i> .....	32
<i>Los requerimientos del sistema (Software).</i> .....	32
Requerimientos funcionales y no funcionales. ....	32
<i>Requerimientos funcionales.</i> .....	33
<i>Requerimientos no funcionales</i> .....	33
<i>Requerimientos del dominio.</i> .....	34
Metodologías Ágiles.....	36
<i>Manifestó ágil</i> .....	36
<i>Comparación de la metodología ágil y tradicional</i> .....	37

Metodología XP .....	39
<i>Conceptos Básicos</i> .....	39
<i>Metodología XP toma cuatro variables</i> .....	40
<i>Ventajas de metodología XP</i> .....	40
Roles de la Metodología Xp .....	41
<i>Programador</i> .....	41
<i>Jefe de Proyecto</i> .....	41
<i>Cliente</i> .....	42
<i>Encargado De Pruebas</i> .....	42
<i>Rastreador</i> .....	42
<i>Entrenador</i> .....	42
Artefacto de la Metodología XP .....	42
<i>Historia de Usuario</i> .....	42
<i>Iteración</i> .....	44
<i>Tarjetas Crc (Tarjetas, Responsabilidades Y Colaboración)</i> .....	45
Fases de la metodología XP .....	46
<i>Planificación</i> .....	46
<i>Diseño</i> ... ..	49
<i>Desarrollo</i> .....	52
<i>Pruebas</i> .....	54
Programación Laravel 5 .....	56
<i>Beneficios de Laravel</i> .....	57
Plantilla admilte .....	58
Case Powerdesigner .....	58
<i>Power Designer dataarchitect:</i> .....	59

<i>Power Designer physicalarchitect:</i> .....	59
<i>Power Designer Developer:</i> .....	59
<i>Power Designer objectarchitect:</i> .....	59
Desarrollo .....	60
Introducción .....	60
Planificación .....	60
<i>Propósito</i> .....	60
<i>Alcance</i> .....	60
<i>Personal involucrado</i> .....	61
<i>Perspectiva del producto</i> .....	61
Requisitos funcionales .....	61
<i>Modulo Inicilizacion</i> .....	61
<i>Modulo Administrador</i> .....	62
<i>Modulo Vendedor</i> .....	62
Requisitos no funcionales.....	69
Planificación de los lanzamientos .....	73
Iteraciones .....	73
Diseño .....	74
<i>Modelación de la Base De Datos</i> .....	74
<i>Modelo Entidad Relación</i> .....	75
<i>Modelo Relacional</i> .....	75
Codificación.....	77
<i>Implementación en el Sistema Gestor de Base De Datos</i> .....	77
<i>Visual Studio Code</i> .....	77
<i>Modelo, Vista, Controlador</i> .....	78

<i>Modelo</i> .....	78
<i>Vista</i> .....	80
<i>Controlador</i> .....	82
Pruebas.....	85
Creación e Inicialización del Sistema.....	94
Acceso - Usuarios.....	95
Validación y verificación.....	111
Tabla de costos primarios y secundarios. ....	113
Conclusiones y Recomendaciones .....	113
Conclusiones. ....	114
Recomendaciones .....	115
Bibliografía .....	116
Anexos .....	121

## Índice de Tablas

<b>Tabla 1</b> Aspectos.....	38
<b>Tabla 2</b> Personal involucrado 1.....	61
<b>Tabla 3</b> Requisito funcional 1.....	63
<b>Tabla 4</b> Requisito Funcional 2.....	63
<b>Tabla 5</b> Requisito Funcional 3.....	64
<b>Tabla 6</b> Requisito Funcional 4.....	64
<b>Tabla 7</b> Requisito Funcional 5.....	65
<b>Tabla 8</b> Requisito Funcional 6.....	65
<b>Tabla 9</b> Requisito Funcional 7.....	66
<b>Tabla 10</b> Requisito Funcional 8.....	66
<b>Tabla 11</b> Requisito Funcional 9.....	67
<b>Tabla 12</b> Requisito Funcional 10.....	67
<b>Tabla 13</b> Requisito Funcional 11.....	68
<b>Tabla 14</b> Requisito Funcional 12.....	68
<b>Tabla 15</b> Requisito Funcional 13.....	69
<b>Tabla 16</b> Requisito no Funcional 1 .....	70
<b>Tabla 17</b> Requisito no Funcional 2 .....	70
<b>Tabla 18</b> Requisito no Funcional 3 .....	71
<b>Tabla 19</b> Requisito no Funcional 4 .....	71
<b>Tabla 20</b> Requisito no Funcional 5 .....	72
<b>Tabla 21</b> Requisito no Funcional 6 .....	72
<b>Tabla 22</b> Plan de Lanzamiento .....	73
<b>Tabla 23</b> Pruebas de funcionalidad Administrador.....	111

<b>Tabla 24</b> Pruebas de funcionalidad Vendedor.....	112
<b>Tabla 25</b> Costos primarios y secundarios.....	113

## Índice Figuras

<b>Figura 1</b> Tipos de Modelado .....	29
<b>Figura 2</b> Evolución de la IEEE .....	31
<b>Figura 3</b> Descripción de un sistema en la aplicación de requerimientos.....	33
<b>Figura 4</b> Requerimientos Funcionales y No funcionales.....	35
<b>Figura 5</b> Proceso Ágil del Desarrollo de Software .....	36
<b>Figura 6</b> Modelo de Historia de Usuario .....	44
<b>Figura 7</b> Modelo de ficha para tareas.....	45
<b>Figura 8</b> Modelo nombre de la clase .....	46
<b>Figura 9</b> Fases de la Metodología XP .....	56
<b>Figura 10</b> Plan de Iteraciones.....	74
<b>Figura 11</b> Relación de Tablas.....	76
<b>Figura 12</b> Creación de Tablas .....	77
<b>Figura 13</b> Interfaz Visual Studio Code.....	78
<b>Figura 14</b> Modelo de Empresa .....	79
<b>Figura 15</b> Modelo de Ventas.....	79
<b>Figura 16</b> Modelo de Artículo.....	80
<b>Figura 17</b> Vista Bienvenido.....	81
<b>Figura 18</b> Vista Inicio .....	81
<b>Figura 19</b> Vista Login .....	82
<b>Figura 20</b> Controlador Persona .....	83
<b>Figura 21</b> Controlador Ingreso.....	83
<b>Figura 22</b> Controlador Proveedor.....	84
<b>Figura 23</b> Login de usuarios .....	85
<b>Figura 24</b> Vista página principal .....	86

<b>Figura 25</b> Selección del menú .....	87
<b>Figura 26</b> Submenú de Almacén .....	88
<b>Figura 27</b> Submenú de Compras.....	88
<b>Figura 28</b> Submódulo Ventas .....	89
<b>Figura 29</b> Submódulo Cartera .....	89
<b>Figura 30</b> Submódulo de Caja .....	90
<b>Figura 31</b> Submenú Producción .....	91
<b>Figura 32</b> Submódulo Acceso.....	92
<b>Figura 33</b> Submódulo Reportes.....	93
<b>Figura 34</b> Submódulo Administración.....	94
<b>Figura 35</b> Registro de Usuario.....	95
<b>Figura 36</b> Acceso Administración - Empresa .....	96
<b>Figura 37</b> Acceso Administración - Sucursales.....	96
<b>Figura 38</b> Administración - Caja .....	97
<b>Figura 39</b> Registro de Medida .....	97
<b>Figura 40</b> Almacén - Categoría.....	98
<b>Figura 41</b> Almacén – Artículo.....	98
<b>Figura 42</b> Registro de Proveedor.....	99
<b>Figura 43</b> Compras - Ingresos.....	99
<b>Figura 44</b> Caja - Apertura .....	100
<b>Figura 45</b> Ventas - Ventas .....	100
<b>Figura 46</b> Cartera- Cuentas por Cobrar.....	101
<b>Figura 47</b> Cartera – Cuentas por Pagar .....	101
<b>Figura 48</b> Producción - Plantas .....	102
<b>Figura 49</b> Reportes – Reporte Ingresos .....	102



<b>Figura 50</b> Reportes – Reporte Ventas.....	103
<b>Figura 51</b> Administración – Roles de Pago.....	103
<b>Figura 52</b> Registro de Empresa del almacén principal con sus Sucursales. ....	104
<b>Figura 53</b> Consolidación de Caja. ....	104
<b>Figura 54</b> Guardar documentos de la matriz consolidada .....	105
<b>Figura 55</b> Consolidación de Roles de pagos.....	105
<b>Figura 56</b> Consolidación de listado de Almacén - Categorías.....	106
<b>Figura 57</b> Consolidación de listado de Almacén - Artículos.....	106
<b>Figura 58</b> Consolidación de listado de Almacén – U. Medidas. ....	107
<b>Figura 59</b> Consolidación de listado de Almacén – Existencia. ....	107
<b>Figura 60</b> Consolidación de Compras - Proveedor.....	108
<b>Figura 61</b> Ventas – Guía de Remisión. ....	108
<b>Figura 62</b> Consolidación Cartera – Cuentas por Cobrar.....	109
<b>Figura 63</b> Consolidación Caja - Apertura. ....	109
<b>Figura 64</b> Consolidación Caja – Cierre de Caja.....	110
<b>Figura 65</b> Consolidación Caja – Arqueo.....	110

## Resumen

El tema planteado en este proyecto técnico está enfocada al desarrollo de una aplicación web, para el control de inventarios y facturación de productos de la Agrícola Moreplant, en la cual la investigación ha sido en varios procesos y así buscar alternativas y dar soluciones informáticas. Servirá en la automatización de procesos contables de ventas, compras e inventarios, para esto utilizaremos las siguientes herramientas como laravel, Plantilla AdmiLTE, Metodologías XP, Power Designer que permitirán en el proceso de planificación de ingeniería en software. Planteando el desarrollo de una aplicación web para la empresa Agrícola Moreplant con el objetivo de gestionar procesos contables como son ventas, compras, inventarios, logrando la automatización de documentación. La Ingeniería de Requerimiento tiene como proceso de recopilar, analizar, y verificar las necesidades del cliente, además es la forma de comprender y definir el software. El proyecto de tesis planteada es una investigación Tecnológica, para ello tenemos la función a dos variables (Independiente y Dependiente), la cual servirá para el desarrollo del sistema de información, conjuntamente de los ERS los requerimientos que se establezca tendrán que ser demostrado en el desarrollo, comprendiendo que los requisitos funcionales y no funcionales son las necesidades que tiene el cliente.

Palabras clave:

- **APLICACIONES WEB**
- **AUTOMATIZACION**
- **GESTION**
- **INGENIERIA DE REQUERIMIENTOS**

### **Abstract**

The issue raised in this technical project is focused on the development of a web application for inventory control and invoicing of products of Agrícola Moreplant, in which the research has been in various processes and thus seek alternatives and provide solutions. It will serve in the automation of accounting processes of sales, purchases and inventories, for this we will use the following tools such as laravel, AdmiLTE Template, XP Methodologies, Power Designer that will allow in the planning process of software engineering. Proposing the development of a web application for the company Agrícola Moreplant in order to manage accounting processes such as sales, purchases, inventories, achieving the automation of documentation. The Requirements Engineering is the process of collecting, analyzing and verifying the customer's needs, it is also the way to understand and define the software. The thesis project proposed is a technological research, for this we have the function to two variables (Independent and Dependent), which will serve for the development of the information system, together with the ERS requirements to be established will have to be demonstrated in the development, understanding that the functional and non-functional requirements are the needs of the customer.

Key words:

- **WEB APPLICATIONS**
- **AUTOMATION**
- **MANAGEMENT**
- **REQUIREMENTS ENGINEERING**

## **1. Problema de Investigación**

### **1.1 Tema**

“DESARROLLO DE UNA APLICACIÓN WEB, MEDIANTE EL USO DE SOFTWARE LIBRE, UTILIZANDO LA METODOLOGÍA XP, BASADO EN LA PLANTILLA ADMINLTE PARA EL CONTROL DE INVENTARIO Y FACTURACIÓN DE PRODUCTOS DE LA AGRÍCOLA MOREPLANT”.

### **1.2 Antecedente**

La mayoría de los microempresarios por lo general no cuentan con software que ayude a optimizar el tiempo, orden y sobre todo el control, generalmente se trabaja con registros manuales que son carpetas folder para el registro de sus compras, un libro contable para el registro de ventas, un directorio manual de clientes y proveedores, es la manera en la que se registra datos. Esto se ha convertido en un problema ya que impide optimizar tiempo, provocando el bajo desempeño laboral.

En el trabajo de tesis Desarrollo e Implementación de Aplicación Web para el Control de Inventarios del Local Comercial Máquinas Hidalgo, desarrollado por Kerly Brigitte Lucas Vega (2017), en relación a los sistemas de información, manifiesta lo siguiente

En las organizaciones es frecuente el uso de la tecnología como una herramienta para la ejecución y control de procesos e información. La finalidad de este proyecto es otorgar una herramienta propia a la empresa, la cual estará diseñada acorde a las necesidades actuales de la misma, proporcionando un mejor control de la información y la optimización de los procesos que realizan, tanto en recurso humano como en el uso de materiales de oficina.

El desarrollo e implementación del Sistema de Control de Inventario será de gran beneficio a los colaboradores de cada sucursal de “Máquinas Hidalgo” de las diferentes provincias del país, ya que les facilitará suministrar la mercadería a cada local de manera oportuna y eficiente. (pág. 4)

En el trabajo de tesis Aplicación web para el Control de Inventario y Facturación de la Empresa BINACOM SYS S.A. (2018) en relación a los sistemas de información, manifiesta lo siguiente:

Se propone implementar un sistema piloto para mejorar la gestión de facturación, que proporcione a los usuarios, una herramienta que facilite la búsqueda, adquisición y facturación de productos y servicios tecnológicos, mediante la utilización de tecnología de información y comunicación

Propuesta que justifica la implementación del proyecto, que es optimizar y mejorar el área de facturación de la empresa, a través del uso de una herramienta de plataformas LAMP (Linux, Apache, MySQL y PHP), lo que propiciará agilidad en atención a los consumidores en el proceso de facturar los productos adquiridos, proporcionando un entorno de negociación electrónico para minimizar el tiempo de comercialización y mejorar el proceso de ventas en forma integral.

En conclusión, se elaboró la propuesta en base a los juicios y fundamentos obtenidos de la observación del proceso al momento de facturar en la empresa y la opinión de algunos usuarios los que serán quienes utilicen el sistema, se recogió sus opiniones a través de una entrevista aplicada para realizar el proyecto de mejor manera identificando aspectos que sirvan para esquematizar, diseñar e implementar el sistema propuesto y que garantice la usabilidad y facilidad del mismo a largo plazo. (YURY, 2018)

### 1.3 Planteamiento del problema

Las organizaciones empresariales hoy en día enfrentan un sin número de problemas, provocados por una serie de factores que influyen directamente en la organización, retrasando y haciéndola incompetente ante las demás, por lo cual con la ayuda de la tecnología se desarrolla nuevos enfoques empresariales los cuales están siendo aceptados y adoptados por numerosas empresas debido a las ventajas profesionales que ofrecen, las cuales afrontarán estas dificultades; logrando así una organización enérgica ante amenazas de diversas índole.

Los Almacenes AGRÍCOLA MOREPLANT, fue inaugurada el 17 de marzo del 2016 por el sr Christian Orlando Mangui, Gerente y Administrador de los almacenes, la idea que fue creada con el propósito de innovar la idea estudiantil desarrollando la carrera estudiantil, explotación de viveros y venta al por mayor de productos químicos de uso agrícola.

Los Almacenes AGRÍCOLA MOREPLANT se dedica a la producción y venta de distintas variedades hortícolas y productos para el control de enfermedades causadas en los cultivos, afirmando que desde el año 2016 el personal se encuentra llevando su información de forma manual o en los últimos de los casos no se tiene el control.

Los controles que se ejecutan en el área de producción y venta son realizados en hojas manuales con formatos que no se lleva un control de inicialización ni finalización de producción, datos que retrasa considerablemente el proceso y la actualización de productos existente en sus almacenes, además los insumos agrícola que se encuentran disponibles para los clientes, no se lleva el control de la venta, no controla inventarios, ni cuadro de caja los cuales resultan los problemas primordiales dentro de los almacenes AGRÍCOLA MOREPLANT.

De no dar una solución efectiva a lo mencionado seguirá causando pérdida de información desorden del mismo incumplimiento con clientes productores, mal manejo de inventario dentro de los almacenes, por la falta de un sistema ágil que ayude al procesamiento de la actualización de datos y la creación de facturación.

Por el cual existe la necesidad de desarrollar e implementar un aplicativo web POS que ayude a mejorar el proceso del control de inventarios, producción de variedades hortícolas, venta de insumos agrícolas.

#### **1.4 Justificación**

Esta aplicación web POS será un implemento en el desarrollo del trabajo de los almacenes AGRÍCOLA MOREPLANT, porque se llevará el control de inventarios, facturación, registro de producción aproximadas a la venta, control de clientes, control de pagos lo que facilitará información actualizada que será de gran utilidad para el control y el desempeño laboral dentro de los almacenes.

Para los almacenes AGRÍCOLA MOREPLANT, el desarrollo e implementación de una página web permitirá el control de inventarios y facturación, esto resulta ser muy útil para el desempeño laboral, automatizando los procesos que se manejan diariamente, dándole la posibilidad de tener toda la información dentro de un software y posibilitando que las actividades puedan ser manejadas por el usuario que lo requiera.

El propósito de este trabajo es contribuir con los almacenes proporcionando mejoras significativas en el desarrollo de las actividades laborales, permitiendo el manejo de la aplicación web con la finalidad de mantener información actualizada de inventarios y facturación.

## **1.5 Objetivos**

### **1.5.1 Objetivo general**

Desarrollar una aplicación web, mediante el uso de software libre, utilizando la metodología XP, basado en la plantilla AdminLTE, para el control de inventario y facturación de productos de la Agrícola Moreplant.

### **1.5.2 Objetivos específicos**

- Establecer información sobre las herramientas que se incluiría en el desarrollo de la aplicación web.
- Implementar un sistema para el control de inventarios y facturación de la empresa Agrícola Moreplant.
- Realizar pruebas de unidad, pruebas de integración, y test de aceptación para la verificación de los requisitos funcionales.

## **1.6 Alcance**

El presente proyecto tiene como meta el desarrollo e implementación de una aplicación web POS para el control de inventarios y facturación de los almacenes AGRÍCOLA MOREPLANT.

### **Inventarios Productos Agrícolas**

Mediante los inventarios se controlará nombre de producto, cantidades, valor, bodega, descripción de ingrediente activo, fecha de caducidad, disponibilidad de los productos, presentación, distribuidor, informe de los productos más vendidos.

### **Inventarios Producción Hortícola**

Mediante los inventarios se controlará nombre de variedad, cantidad de producción, fecha de inicio, fecha final, porcentaje de mortalidad (cuantas plantas han quedado a la fecha), stock de productos, control de pedidos, informe de los productos más vendidos.



**Facturación**

La facturación controlará la gestión de ventas con diversos medios de pagos, registro de las compras, clientes, proveedores, reporte diario, reporte mensual, impresión de factura, reporte de venta en Excel, informe de clientes, informe de vendedores.

## 2 Marco Teórico

### 2.1 Definiciones Básicas

#### 2.1.1 *Qué es un software.*

Un proceso de desarrollo de software es la descripción de una secuencia de actividades que deben ser seguida por un equipo de trabajadores para generar un conjunto coherente de productos, uno de los cuales es el programa del sistema deseado. (Drake, 2008, pág. 2)

#### 2.1.2 *¿Clasificación de software de sistema?*

Su objetivo es desvincular adecuadamente al usuario y al programador de los detalles del sistema informático en particular que se use, aislándolo especialmente del procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc. El software de sistema le procura al usuario y programador adecuadas interfaces de alto nivel, controladores, herramientas y utilidades de apoyo que permiten el mantenimiento del sistema global. (Lopez, 2013, pág. 3)

#### 2.1.3 *¿Qué es Software de Programación?*

Es el conjunto de herramientas que permiten al programador desarrollar programas informáticos, usando diferentes alternativas y lenguajes de programación, de una manera práctica. (Lopez, 2013, pág. 3)

#### 2.1.4 *¿Que es Software de Aplicación?*

Es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. (Lopez, 2013, pág. 3)

### **2.1.5 ¿Qué es ingeniería en software?**

La Ingeniería del Software requiere la comprensión y aplicación de principios de ingeniería, habilidades de diseño, buenas prácticas de gestión, fundamentos de la Ciencia de la Computación y formalismos matemáticos. Es tarea de la Ingeniería del Software juntar estas áreas de trabajo tan dispares y utilizarlas en las fases de obtención de los requisitos, especificación, diseño, verificación, implementación, prueba, documentación y mantenimiento de sistemas software complejos y de gran tamaño. (Peñalvo, 2018, pág. 7)

### **2.1.6 ¿Qué es un proceso de software?**

Un proceso del software es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades son llevadas a cabo por los ingenieros de software. (Sommerville I. , 2005, pág. 7)

1. Especificación del software. Donde los clientes e ingenieros definen el software a producir y las restricciones para su operación.
2. Desarrollo del software donde el software se diseña y programa.
3. Validación del software donde el software se valida para asegurar que es lo que el cliente requiere.
4. Evolución del software donde el software se modifica para modificar para adaptarlo a los cambios requeridos por el cliente y el mercado.

(Sommerville I. , 2005, pág. 7)

### **2.1.7 ¿Qué es un ciclo de software?**

El periodo de tiempo que comienza cuando se concibe un software y concluye cuando el producto ya no está disponible para su uso.

El ciclo de vida del software típicamente incluye una fase de requisitos, una fase de diseño, una fase de pruebas, una fase de instalación y aceptación, una fase de

operación y mantenimiento, y, en ocasiones, una fase de retirada. (Simon Pickin, 2012, pág. 16)

Un modelo de ciclo de vida es una abstracción particular que representa un ciclo de vida de software. Un modelo de ciclo de vida se denomina con frecuencia un ciclo de vida de desarrollo software (SDLC, siglas inglesas). (Simon Pickin, 2012)

### **2.1.8 Modelo de un Software**

Perspectiva escéptica sobre modelos de software: “burbujas y flechas, al contrario que los programas, nunca cascan”, el uso de modelos es tan antiguo como la ingeniería, antes de construir el ente real, los ingenieros construyen modelos y aprenden de ellos.

Algunas características deseables de un modelo:

- abstracto
- comprensible
- preciso
- predictivo
- no muy caro de construir. (Simon Pickin, 2012, pág. 17)

### **2.1.9 Modelado de sistemas.**

Los modelos ofrecen un medio de abstracción que ayuda a la comunicación. El idioma mismo es un proceso de abstracción, y las matemáticas son una clase particular de idioma simbólico. (Farias, pág. 2)

La creación de modelos es la clave de la conceptualización; los modelos son creados para describir, explicar o predecir los fenómenos pertinentes en el mundo real. (Farias, pág. 2)

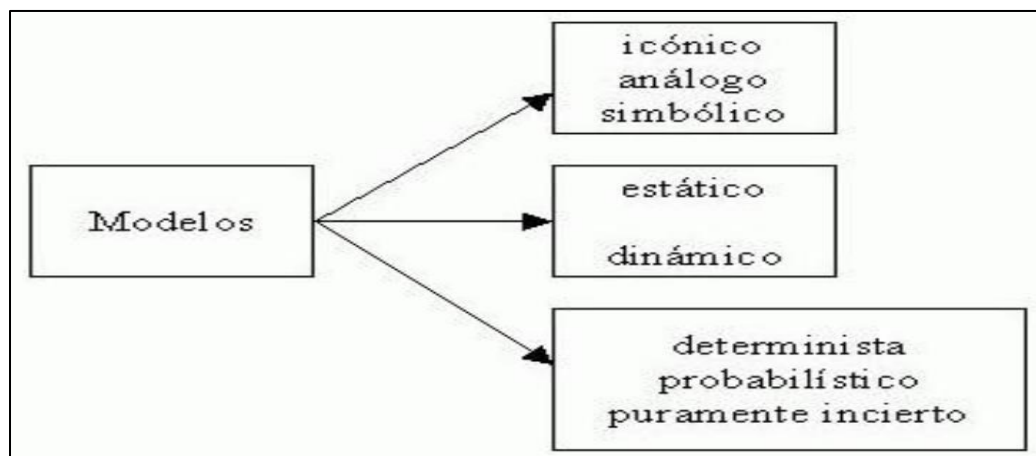
### **2.1.10 Contrucción de Modelos.**

Significa que el modelo se inicia con una descripción verbal del problema y los objetivos a lograr para darle solución. En esta etapa, el investigador debe proponer,

según su criterio, un modelo que se acerque lo más posible a la situación real del sistema. (Farias, pág. 6)

### Figura 1

*Tipos de Modelado*



*Nota:* Marco de referencia para la administración. (Farias, pág. 2)

## 2.2 Historia de una Aplicación Web

La inclusión de un método para confeccionar páginas dinámicas que permitiesen que lo mostrado fuese dinámico (generado o calculado a partir de los datos de la petición). Dicho método fue conocido como CGI (common gateway interface) y definía un mecanismo mediante el cual podíamos pasar información entre el servidor HTTP y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores web los soportan debido a su sencillez. Además, nos proporcionan total libertad a la hora de escoger el lenguaje de programación para desarrollarlos. (Jiménez, Escuela Especializada en Ingeniería, 2016, pág. 20)

### 2.2.1 Concepto Aplicación Web

Las aplicaciones Web se basan en un modelo cliente/servidor, pero su forma de interactuar es diferenciada puesto que las conexiones cliente/servidor típicas no son

persistentes. Esto implica que no es necesario mantener abierta la comunicación de forma permanente, sino que esta se realiza bajo la demanda del usuario. (YURY, 2018, pág. 9)

### **2.2.2 Beneficios de la aplicación web**

Los beneficios de las aplicaciones web, considerando aspectos de su aplicación como: herramientas de comunicación, colaboración, acceso a la información y trabajo en equipo. Entre los principales beneficios de las aplicaciones web se tiene:

- Son aplicaciones multiplataforma que en la gran mayoría de los casos no requieren de software adicional para su funcionamiento.
- Las aplicaciones web no requieren de conocimientos avanzados sobre comunicaciones.
- Son adaptables a medios de uso a conveniencia y bajo entornos de seguridad controlados a través de intranets, extranet e internet.
- No demandan de grandes cantidades de procesamiento de parte de los equipos de punto final puesto que los procesos se ejecutan del lado del servidor. (YURY, 2018, pág. 9)

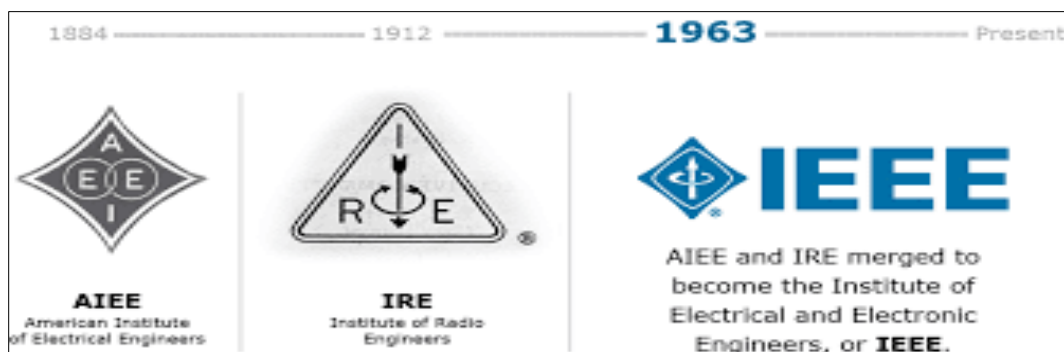
### **2.3 Estándar IEEE 830-1998.**

Según IEEE, un buen documento de Requisitos, pese a no ser obligatorio que siga estrictamente la organización y el formato dados en el estándar 830, sí deberá incluir, de una forma o de otra, toda la información presentada en dicho estándar. El estándar de IEEE 830 no está libre de defectos ni de prejuicios, y por ello ha sido justamente criticado por múltiples autores y desde múltiples puntos de vista, llegándose a cuestionar incluso si es realmente un estándar en el sentido habitual que tiene el termino en otras ingenierías. El presente documento no pretende pronunciarse ni a favor ni en contra de unos u otros: tan sólo reproduce, con propósitos

fundamentalmente docentes, cómo se organizaría un Documento de Requisitos según el estándar IEEE 830. (Mendez, 2008, pág. 1)

## Figura 2

*Evolución de la IEEE*



*Nota.* The American Institute of Electrical Engineers + Institute of Radio Engineers 1912 = Institute of Electrical and Electronics Engineers. (Software., 2010)

### 2.4 Norma Iso 27001

La ISO 27001 es una norma internacional emitida por la Organización Internacional de Normalización (ISO) y describe como gestionar la seguridad de la información en una empresa. La revisión más reciente de esta norma fue publicada en el 2013 y ahora su nombre completo es ISO/IEC 27001:2013. La primera revisión fue público en 2005 y fue desarrollada en base a la norma británica BS 7799-2. (SARA, s.f., pág. 6)

ISO 27001 puede ser implementada en cualquier tipo de organización con o sin fines de lucro, privada o pública, pequeña o grande. Está redacta por los mejores especialistas del mundo en el tema y proporciona una metodología para implementar la gestión de la seguridad de la información en una organización. También permite que una empresa se certificada esto significa que una entidad de certificación independiente confirma que la seguridad de la información ha sido implementada en esa organización

en cumplimiento con la norma ISO 27001 se ha convertido en la principal norma a nivel mundial para la seguridad de la información y muchas empresas han certificado su cumplimiento. (SARA, s.f., págs. 6,7)

## **2.5 Requerimientos de software.**

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos (RE). Este apartado se centra en dichos requerimientos y cómo describirlos. (Sommerville I. , 2005)

### **2.5.1 Los requerimientos del usuario (Alto Nivel).**

Declaraciones en un lenguaje natural más diagramas de los servicios que el sistema proporciona y sus restricciones operacionales. (Sommerville, 2004, pág. 7)

### **2.5.2 Los requerimientos del sistema (Software).**

Un documento estructurado que establece descripciones detalladas de las funciones servicios y restricciones operacionales del sistema. Define lo que debería ponerse en práctica así que se debe ser parte de un contrato entre cliente y el contratista. (Sommerville, 2004, pág. 7)

## **2.6 Requerimientos funcionales y no funcionales.**



**Figura 3**

*Descripción de un sistema en la aplicación de requerimientos*



*Nota:* Interacción de los requerimientos funcionales y no funcionales (Atom, 2016)

### **2.6.1 Requerimientos funcionales.**

Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer. (Atom, 2016)

### **2.6.2 Requerimientos no funcionales.**

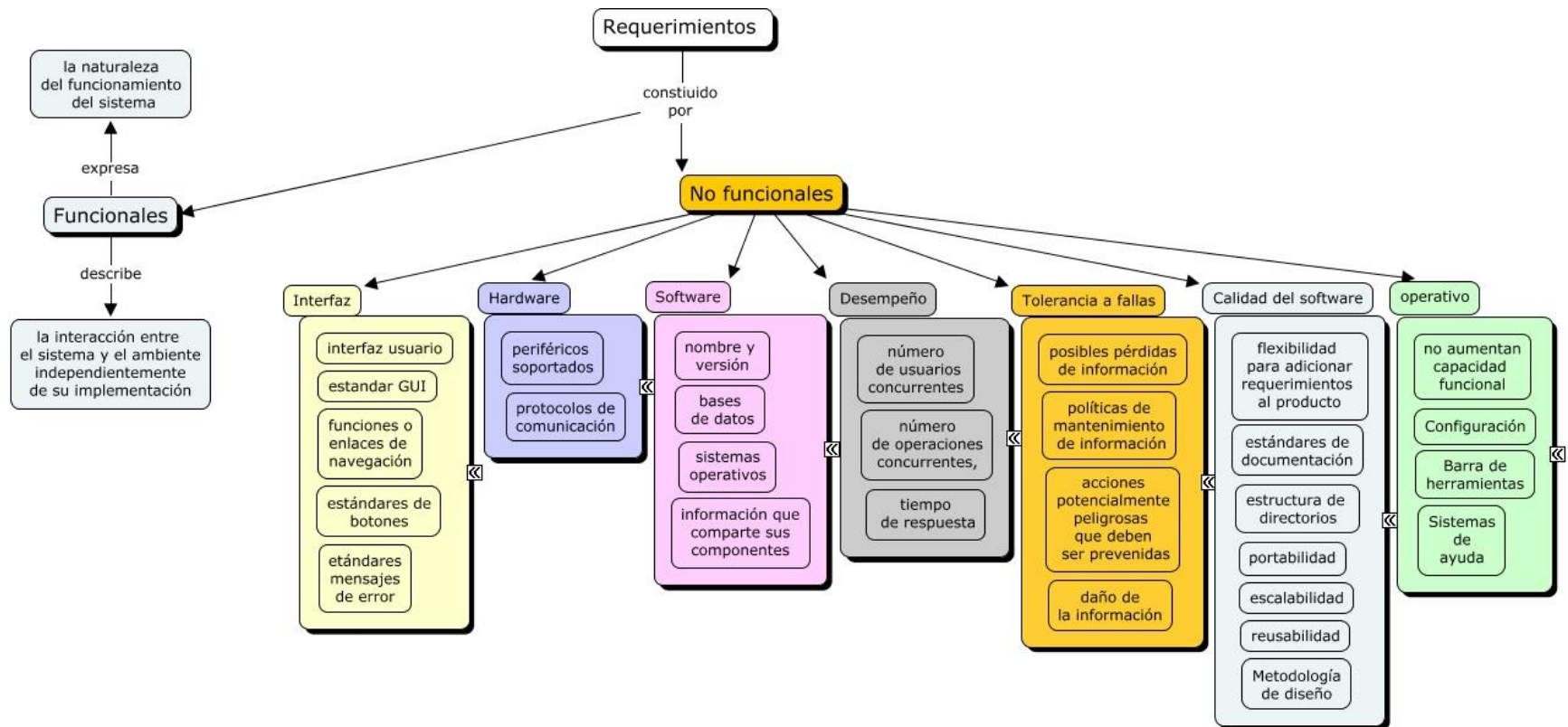
Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema. (Atom, 2016)

### **2.6.3 *Requerimientos del dominio.***

Son requerimientos que provienen del dominio de aplicación del sistema y que reflejan las características y restricciones de ese dominio. Pueden ser funcionales o no funcionales. En realidad, la distinción entre diferentes tipos de requerimientos no es tan clara como sugieren estas definiciones. Por ejemplo, un requerimiento del usuario sobre seguridad podría parecer un requerimiento no funcional. Sin embargo, cuando se desarrolla en detalle, puede generar otros requerimientos que son claramente funcionales, como la necesidad de incluir en el sistema recursos para la autenticación del usuario. (Atom, 2016)

Figura 4

Requerimientos Funcionales y No funcionales



Nota. Especificación de requerimientos funcionales y no funcionales. (Atom, 2016, p. 2)

## 2.7 Metodologías Ágiles.

El término ágil surge como iniciativa de un conjunto de expertos en el área de desarrollo de software con el fin de optimizar el proceso de creación del mismo, el cual era caracterizado por ser rígido y con mucha documentación. El punto de partida fue el manifiesto ágil, el cual es un documento donde se detalla todo lo que involucra la filosofía ágil. (Jiménez, 2016, pág. 7)

**Figura 5**

*Proceso Ágil del Desarrollo de Software*



*Nota.* Engloba principios y valores que hacen diferente un proyecto de desarrollo de software ágil de uno en su forma tradicional. (Jiménez, 2015, pág. 7)

### 2.7.1 *Manifestó ágil*

Según el manifiesto ágil se valora a:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funcione más que la documentación del mismo.
- La colaboración con el cliente más que la negociación de su contrato.
- Responde a los cambios más que seguir con el plan establecido.

Esta metodología ágil está regida además por doce principios que ayudan a que el proceso de desarrollo se vuelva menos complejo y responda de manera oportuna a los cambios que surgen a lo largo del mismo, siempre contando con el punto de vista del cliente. (Jiménez, 2015, pág. 7)

### **2.7.2 Comparación de la metodología ágil y tradicional**

La incorporación de nuevas tecnologías y formas de llevar a cabo el proceso de desarrollo de software ha venido revolucionando de manera tal que se han dejado de lado los largos procesos y documentación exhaustiva, por métodos más enfocados en el cliente y en el equipo de desarrollo. Basándose en este enfoque se realiza una comparación de la forma antigua de realizar el proceso versus el actual. A continuación, se muestra una tabla resumen con aspectos importantes de ambas metodologías. (Jiménez, 2016, pág. 11)

**Tabla 1***Aspectos*

<b>ASPECTOS</b>	<b>AGIL</b>	<b>DIRIGIDO POR MODELOS</b>
<i>PERSONAS</i>	<i>Alta prioridad: se facilita relación cliente desarrollador</i>	<i>No prioritario; el modelo del espacio del problema es la base de la discusión entre el cliente y el desarrollador.</i>
<i>PROCESOS</i>	<i>Prioridad media; incremental y evolutivo.</i>	<i>Tiende el proceso en cascada, poco incremental</i>
<i>TECNOLOGIA</i>	<i>Baja Prioridad; solo cobra importancia al final</i>	<i>Es relevante; se usa para la generación del software (usando un PSM)</i>
<i>MODELOS</i>	<i>Artefactos secundarios; se produce cuando es absolutamente necesario</i>	<i>Artefacto prioritario; fuente de la implementación</i>
<i>SOFTWARE</i>	<i>Artefacto Prioritario; es la única medida de progreso</i>	<i>Artefacto secundario depende del espacio de la solución</i>

*Nota:* Aspectos diferenciadores entre procesos ágiles y dirigidos por modelos. (Jiménez, 2015, pág. 11)

## 2.8 Metodología XP

La metodología XP fue creada para desarrollar sistemas de corto plazo por Kent Beck en 1999; el éxito de esta metodología es el uso de una excelente comunicación, su sencillez y el permitir interactuar con el cliente. “Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar. (Fernández, 2019, pág. 47)

Las características más importantes de la metodología XP son:

- Actualmente es utilizada en Ingeniería de Software para desarrollar sistemas a corto plazo.
- Permite desarrollar proyectos sin dar mucha importancia a la documentación.
- Realiza la codificación y pruebas de forma paralela y no secuencial, así como permite la colaboración y comunicación directa de los clientes, otorgando respuestas inmediatas ante la eventualidad de presentarse algún cambio.
- Dispone de herramientas informáticas y no informáticas para desarrollar por completo un proyecto de desarrollo Web.
- Verifica que el desarrollo del proyecto funcione por medio de prototipos. Por regla general estos se deben producir cuando sean realmente necesarios.

(Fernández, 2019, pág. 47)

### 2.8.1 *Conceptos Básicos*

La metodología XP está fundamentada en los siguientes valores que permiten facilitar el trabajo para realizar la implementación de la aplicación.

- **COMUNICACIÓN** Mantener un intercambio de información entre el cliente y los

programadores durante todo el desarrollo del sistema, para evitar dejar de lado puntos importantes que al inicio fueron presentados.

- **SIMPLICIDAD** Desarrollar un sistema que sea ágil, unificado y de fácil comprensión para cuando se precise realizar actualizaciones, en caso de ser necesario, no se requiera empezar de cero.
- **RETROALIMENTACIÓN** Visualizar el desarrollo del sistema a tiempo, para ir verificando si todo está correcto o si se debe realizar cambios; esto depende de las necesidades del cliente.
- **TENACIDAD** Es el valor más importante en razón de que a través de éste podemos cumplir los tres valores anteriores, pues se debe poseer: valor para la comunicación, decisión para la simplicidad y enfatizar la retroalimentación.

(Fernández, 2019, pág. 48)

### **2.8.2 Metodología XP toma cuatro variables**

- **COSTE** Los cambios a realizar no afectan el tiempo de entrega.
- **TIEMPO** Debe ser el menor posible pero siempre y cuando cumpla todos los requerimientos especificados durante el desarrollo del sistema.
- **CALIDAD** Poseer una mayor calidad en el menor tiempo posible.
- **ÁMBITO** Especificado por medio de los programadores. (Fernández, 2019, pág. 49)

### **2.8.3 Ventajas de metodología XP**

Las ventajas que se pueden obtener al utilizar la metodología XP son las siguientes:

- Potenciar el trabajo en grupo durante el desarrollo de la aplicación.
- Entregar periódicamente avances del sistema.
- Realizar pruebas continuas hasta obtener la aplicación final.



- Los cambios que se realizan en la aplicación no deben aumentar con el tiempo.
- Realizar un desarrollo de manera eficiente, sencilla y clara; para facilitar el rendimiento y su actualización.
- Da lugar a una programación sumamente organizada.
- Ocasiona eficiencias en el proceso de planificación y pruebas.
- Cuenta con una tasa de errores muy pequeña.
- Propicia la satisfacción del programador.
- Fomenta la comunicación entre los clientes y los desarrolladores.
- Facilita los cambios. o Permite ahorrar mucho tiempo y dinero.
- Puede ser aplicada a cualquier lenguaje de programación.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.
- La XP es mejor utilizada en la implementación de nuevas tecnologías.

(Fernández, 2019, págs. 49,50)

## **2.9 Roles de la Metodología Xp**

Los roles que posee la metodología XP permiten a los desarrolladores organizar y distribuir el trabajo de manera más equilibrada y eficiente. Estos roles se pueden resumir de la siguiente manera: (Fernández, 2019, pág. 53)

### **2.9.1 Programador**

Responsable de construir, analizar, programar, tomar decisiones y realizar pruebas del sistema. (Fernández, 2019, pág. 53)

### **2.9.2 Jefe de Proyecto**

Responsable de coordinar, gestionar y administrar las reuniones para considerar las condiciones de cómo avanza el proyecto. (Fernández, 2019, pág. 54)

### **2.9.3 Cliente**

Persona que debe especificar qué construir, cuándo y dónde realizar las pruebas. (Fernández, 2019, pág. 54)

### **2.9.4 Encargado De Pruebas**

Delegado de ayudar al cliente para que las pruebas sean realizadas y superadas. (Fernández, 2019, pág. 54)

### **2.9.5 Rastreador**

Responsable de obtener datos históricos; encargados de observar sin molestar durante el desarrollo del sistema. (Fernández, 2019, pág. 54)

### **2.9.6 Entrenador**

Facultado de visualizar el proceso y desarrollo del sistema, desde un segundo plano. (Fernández, 2019, pág. 54)

Cuando se utilizan los roles de la metodología XP se trata al cliente como parte del equipo de desarrolladores, lo que evita la redundancia de información, y permite que se realicen pruebas constantes de la aplicación. Otro beneficio de esta inclusión es que se efectúan cambios mientras avanza el desarrollo según los requerimientos solicitados evitando pérdidas de tiempo. (Fernández, 2019, pág. 54)

## **2.10 Artefacto de la Metodología XP**

Los artefactos de la metodología XP son: Historias de usuario, iteración, tarjeta CRC.

### **2.10.1 Historia de Usuario**

Las historias de usuario explican de una manera simple y entendible los requisitos de la empresa ya que son escritas por los clientes involucrados en el desarrollo utilizando un lenguaje común (sin tecnicismos). (Fernández, 2019, pág. 54)

Para su elaboración se utilizan tarjetas de papel o a través de una computadora. En estas se plasman los requisitos del usuario con lo que el desarrollador podrá visualizar el ámbito y fijar prioridades para el desarrollo de la aplicación, así como generar estimaciones, consecuencias y definir procesos a ser realizados. Por medio de las historias de usuario se puede formalizar de manera casi exacta la estimación del tiempo que llevará realizar el desarrollo de la aplicación. Permiten proporcionar la estimación de riesgo de cada requisito y verificar si se cumple correctamente con lo solicitado cuando se realicen las pruebas. (Fernández, 2019, pág. 55)

También ayudan a definir las funciones que realizan los usuarios cuando utilizan el software y determinar quiénes deben poseer acceso ilimitado y limitado para evitar errores. Cada una de las historias debe ser limitada y clara para ser implementada por los desarrolladores máximo en tres semanas; si se requiere más tiempo esta debe ser dividida entre dos personas para obtener un mejor desempeño. (Fernández, 2019, pág. 55)

Al inicio es común que no se especifiquen todas las historias de usuario, pero estas se incrementarán al realizar las iteraciones. Se debe tomar en cuenta que para cada historia de usuario debe existir una prueba de aceptación. (Fernández, 2019, pág. 54)

**Figura 6***Modelo de Historia de Usuario*

<b>Historia de usuario</b>	
<b>Número:</b>	<b>Usuario:</b>
<b>Nombre historia de usuario:</b>	
<b>Prioridad en negocio: (Alta / Media / Baja)</b>	<b>Riesgo en desarrollo: (Alta / Media / Baja)</b>
<b>Puntos estimados:</b>	<b>Iteración asignada:</b>
<b>Descripción:</b>	
<b>Observación:</b>	

*Nota:* Se visualiza el modelo de ficha para realizar las historias de usuario. (Fernández, 2019, pág. 56)

**2.10.2 Iteración**

Las iteraciones permiten realizar un análisis más exhaustivo antes de implementar una historia de usuario; de esta forma el desarrollador en conjunto con el cliente debe realizar las fichas de tareas para obtener la información necesaria para un mejor desarrollo. Las iteraciones son útiles para verificar el avance del proyecto; entre menos errores este tenga se puede observar un gran avance en el mismo, tanto en calidad como en tiempo. Dentro de cualquier desarrollo se establece como mínimo tres interacciones. (Fernández, 2019, pág. 56)

**Figura 7***Modelo de ficha para tareas*

<b>Tarea</b>	
<b>Número tarea:</b>	<b>Nombre historia:</b>
<b>Programador Responsable:</b>	
<b>Tipo de tarea:</b> Desarrollo/Corrección/ Mejora/Otra	<b>Puntos estimados:</b>
<b>Descripción</b>	

*Nota:* Se visualiza el modelo de ficha para realizar las tareas. (Fernández, 2019, pág. 57)

### **2.10.3 Tarjetas Crc (Tarjetas, Responsabilidades Y Colaboración)**

Una tarjeta CRC representa una clase en la programación orientada a objetos. Son utilizadas por la metodología XP para diseñar el sistema uniendo todas las funcionalidades. Ayudan a definir actividades mientras se está diseñando el sistema. Como se visualiza en la tabla 03 la tarjeta CRC representa: la clase a la que pertenece el objeto, las responsabilidades que deben realizar el objeto y las colaboraciones con otras clases para conocer cómo se comunican entre ellas y cumplen con cada responsabilidad del objeto. (Fernández, 2019, pág. 57)

**Figura 8***Modelo nombre de la clase*

<b>Modelo Nombre de la Clase</b>	
<b>Responsabilidades (Métodos o atributos de la clase)</b>	<b>Colaboradores (Clases con las que existe comunicación para realizar las responsabilidades)</b>

*Nota:* La clase a la que pertenece el objeto, las responsabilidades que deben realizar el objeto y las colaboraciones con otras clases para conocer cómo se comunican entre ellas y cumplen con cada responsabilidad del objeto. (Fernández, 2019, pág. 57)

## **2.11 Fases de la Metodología XP**

### **2.11.1 Planificación**

Permite obtener y recolectar la información necesaria para conocer lo que el cliente requiere; se debe estar al tanto de los objetivos del proyecto, el tiempo que tomará desarrollar el sistema, personas involucradas en el desarrollo, presentación de prototipos y la evaluación de calidad. (Fernández, 2019, pág. 58)

Esta fase lo componen las siguientes etapas:

#### ***Historias de Usuario***

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso, pero con algunas diferencias: Constan de 3 o 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucha firmeza en los detalles; no se debe

hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. (Fernández, 2019, pág. 59)

Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas. (Fernández, 2019, pág. 59)

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

Consiste en facilitar un cronograma como resultado de la información obtenida de las historias de usuario, que debe ser analizado por todos los actores del proyecto. Dicho plan puede ser modificado o actualizado de ser requerido, para lo cual se debe realizar una nueva reunión entre los actores. Este cronograma tiene que especificar el tiempo establecido para la implementación de las historias de usuario, prioridad de desarrollo de cada una y especificar la presentación de pequeñas versiones del proyecto. (Fernández, 2019, pág. 60)

Por medio del plan de entregas se debe obtener los siguientes factores:

Objetivos: Clasificar las historias de usuario.

Tiempo: Desarrollar y publicar los prototipos.

Personas involucradas: Especificar el número de personas necesarias.

Evaluaciones: Verificar la calidad del proyecto. (Fernández, 2019, pág. 60)

### **Velocidad del Proyecto**

Es una medida que representa la rapidez con la que se desarrolla el proyecto; estimarla es muy sencillo, basta con contar el número de historias de usuario que se pueden implementar en una iteración; de esta forma, se sabrá el cupo de historias que

se pueden desarrollar en las distintas iteraciones. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Es conveniente reevaluar esta medida cada 3 ó 4 iteraciones y si se aprecia que no es adecuada hay que negociar con el cliente un nuevo "Plan de entregas".

(Fernández, 2019, pág. 60)

### **Iteraciones**

Todo proyecto que siga la metodología X.P. se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Plan de entregas" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores. (Fernández, 2019, pág. 61)

### **Rotaciones**

Permite mantener un lazo de comunicación con todos los actores del proyecto para que los involucrados conozcan el diseño y la codificación del sistema en su totalidad. Esta característica permite que una persona rote por todas las funcionalidades durante el desarrollo del sistema, y dado que se crean patrones de codificación todos los individuos involucrados son aptos de entender el funcionamiento del mismo. Así la metodología XP trata de evitar cuellos de botella o códigos que no pueden ser reutilizados por otros programadores. (Fernández, 2019, pág. 61)

### **Reuniones**

Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto. (Fernández, 2019, pág. 62)



### **2.11.2 Diseño**

Fase en la que se crea estándares y patrones para realizar la codificación; de esta forma se obtendrá un código más eficiente, con calidad y comprensión para todos los involucrados en el desarrollo. Esta fase lo determinan los siguientes componentes: (Fernández, 2019, pág. 62)

#### **Metáfora del Sistema**

En la metodología XP es importante crear una metáfora entre los desarrolladores y el cliente. La metáfora consiste en narrar cómo funcionaría el sistema en el momento de la entrega final. Permite conocer el alcance, propósito y es utilizada como una guía para desarrollar el sistema. “Una metáfora para el sistema es una historia que todo el mundo puede contar acerca de cómo el sistema funciona”. Kent Beck. (Fernández, 2019, pág. 62)

Para un mejor desarrollo de la metáfora se debe especificar nombres (se debe definir las nomenclaturas de clases y métodos del sistema) y analizar posibles problemas que puedan generarse.

Las ventajas de crear una metáfora son las siguientes:

- Conocer la mayoría o todo lo que se va a implementar.
- Permite entender de forma clara el objetivo del sistema.
- El cliente pueda hablar cómodamente sobre el funcionamiento del sistema.
- Los desarrolladores conocen el mismo lenguaje y si hay nuevos desarrolladores les permite adaptarse más rápido y sin dificultades.
- Permite implementar los patrones de diseño que podrían ser utilizados durante el desarrollo del sistema. (Fernández, 2019, pág. 63)

### **Tarjetas CRC**

Las tarjetas Clase – Responsabilidad - Colaboración su técnica consiste en dibujar una tarjeta por cada clase u objeto, y dividirla en tres zonas:

- En la parte superior, el nombre de la clase.
- Debajo, en la parte izquierda, las responsabilidades de dicha clase. Son sus objetivos, a alto nivel.
- A la derecha de las responsabilidades, los colaboradores, que son otras clases que ayudan a conseguir cumplir a esta con sus responsabilidades (Fernández, 2019, pág. 63)

### **Soluciones Puntuales**

La metodología XP utiliza las soluciones puntuales llamadas Spike, para encontrar posibles respuestas a problemas (técnico, arquitectura del software, estimación de tiempo) que pueden aparecer durante el desarrollo del sistema. (Fernández, 2019, pág. 63)

Estos pequeños programas son utilizados para probar y evaluar soluciones; son considerados desechables, ya que una vez utilizados y valorados los resultados obtenidos de su ejecución se los puede descartar. La metodología XP da preferencia a la solución de los problemas de manera automática y no manual; de esta forma se trata de disminuir el riesgo y aumentar la confiabilidad en el desarrollo de alguna historia de usuario. (Fernández, 2019, pág. 64)

### **Funcionalidad Mínima**

La metodología XP indica que la funcionalidad extra consume recursos, representa una pérdida de tiempo y quizás nunca se utilice esas funcionalidades.

Por esta razón se da prioridad a una eficiente programación, focalizando en quien realiza la tarea programada para cada día, creando el código necesario y

verificando que realice su trabajo correctamente. De esta forma, se evita crear código que quizás no sea utilizado. Por lo tanto, nunca se debe añadir funcionalidad extra al programa, aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos. (Fernández, 2019, pág. 64)

### **Reciclaje**

Es una actividad que utiliza la metodología XP para obtener un mejor control cuando se programa. Consiste en eliminar la redundancia de código, simplificar y crear una estructura fácil de comprender y modificar para futuros cambios. (Fernández, 2019, pág. 65)

Es necesario revisar el código cada vez que se termina una iteración, tomando en cuenta los siguientes puntos:

- Desplazamiento fácil en el sistema. (Si/No)
- Existen varios objetos que realizan lo mismo. (Si/No)
- Existe suficiente espacio para el contenido del sistema. (Si/No)
- Existe código de funciones que fueron eliminadas. (Si/No)
- Comunica la visión principal u original del sistema. (Si/No)

Por lo tanto, refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común rehusar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. (Fernández, 2019, pág. 65)

### **2.11.3 Desarrollo**

La metodología XP indica que el código expresa tú interpretación del problema, así podemos utilizar el código para comunicar, para hacer más tus ideas, y por tanto para aprender y mejorar. (Fernández, 2019, pág. 66)

#### **Disponibilidad Del Cliente**

Para la metodología XP es necesario crear una comunicación fluida, personal y directa entre el cliente y el equipo de trabajo en todas las fases del proyecto; de este modo se reduce el tiempo de desarrollo del sistema y evita malos entendidos por cuanto se trata de una comunicación oral y no escrita. (Fernández, 2019, pág. 66)

Las funciones que realiza un cliente son:

- Realizar las historias de usuario y especificar el plan de entrega.
- Validar el sistema cuando se realicen las pruebas funcionales.
- Toma de decisiones sobre las funciones que realiza el sistema.
- Responder dudas, determinar prioridades, especificar mejoras y resolver posibles problemas.
- Tener un compromiso de trabajo en equipo.
- Estar presente antes de desarrollar cada historia de usuario y comunicar más detalladamente su funcionamiento.
- Determinar cuando el sistema puede ser colocado en producción.

(Fernández, 2019, pág. 66)

Por medio de la comunicación permanente se puede prevenir a tiempo cualquier situación o funcionalidad que no es deseable o necesaria para el proyecto, sin necesidad de esperar a estar en el ciclo de desarrollo para comenzar a realizar las correcciones, que a la postre puedan ser consideradas muy complicados. (Fernández, 2019, pág. 67)

### **Unidad de Pruebas**

La metodología XP recomienda crear las pruebas de unidad antes del código y de la implementación del sistema, para obtener un conocimiento global del comportamiento del sistema. Estas pruebas deben ser ejecutadas cada vez que exista una modificación en el sistema. Como resultado se obtiene un código conciso y simple ya que se implementa únicamente las características necesarias que requiere el sistema. Todas las características que tiene el sistema tienen que ser probadas constantemente, lo que permite adaptarse a nuevos cambios. (Fernández, 2019, pág. 67)

Las pruebas de unidad tienen las siguientes características:

- Se crean una vez y se ejecutan constantemente durante todo el desarrollo del proyecto.
- Verifican el funcionamiento correcto evitando redundancia.
- Al principio se obtiene una prueba y función simple; se van agregando nuevos requerimientos dependiendo de las etapas en la que se encuentre hasta obtener toda la funcionalidad requerida.
- Una funcionalidad se la determina como finalizada cuando paso correctamente todas las pruebas de unidad. (Fernández, 2019, pág. 67)

### **Programación por Parejas**

La metodología XP aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad. (Fernández, 2019, pág. 67)

## **Integración**

El objetivo de la metodología XP fomenta que la programación siempre debe ser en pareja, para desarrollar el código de manera continua, rotar funciones y escribir con los mismos estándares. (Fernández, 2019, pág. 68)

Existen dos tipos de integraciones que utiliza la metodología XP, que son:

### **Integración Continua**

Se realiza la integración de código cada vez que esté listo por los desarrolladores, para luego ser probado; esta debe ser realizada, de ser posible, varias veces en un día. Aquí todos trabajan con la última versión del proyecto. Al utilizar este tipo de integración se detecta o evade de manera eficiente los posibles problemas de compatibilidad que pueden surgir. (Fernández, 2019, pág. 69)

### **Integración Secuencial**

Es recomendable realizar la integración de código en una máquina para evitar posibles problemas o redundancia en el código; ya que los desarrolladores trabajan en paralelo en donde se debe elegir únicamente a una pareja de desarrolladores que integren, prueben y realicen cambios en el código. (Fernández, 2019, pág. 69)

#### **2.11.4 Pruebas**

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. El uso de los test en XP es el siguiente: (Fernández, 2019, pág. 70)

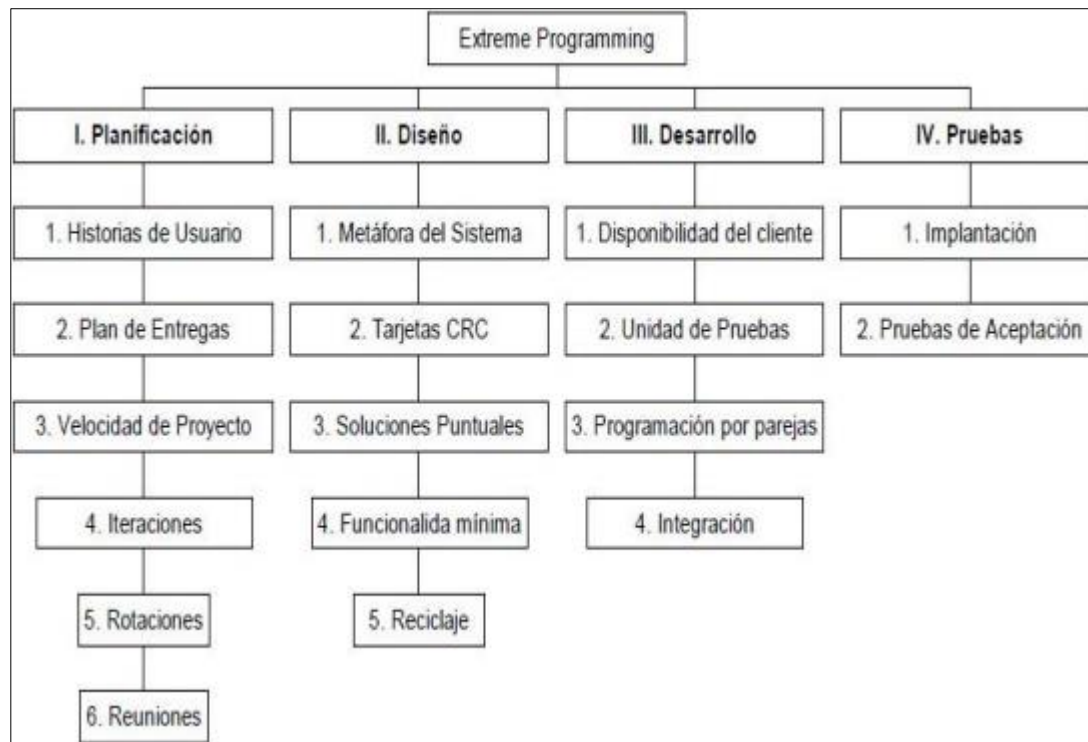
### **Implantación**

- Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.
- Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.

- Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.
- Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.
- Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican.
- Test de aceptación: Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.
- Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos. (Fernández, 2019, pág. 70)

### **Prueba de Aceptación**

Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario. (Fernández, 2019, pág. 70)

**Figura 9***Fases de la Metodología XP*

*Nota: Se observa las fases de la metodología XP para la creación de la aplicación.*

(Fernández, 2019, pág. 58)

## 2.12 Programación Laravel 5

Laravel es un framework para aplicaciones web con sintaxis expresiva y elegante. Creemos que el desarrollo debe ser una experiencia agradable y creativa para que sea verdaderamente enriquecedora. Laravel busca eliminar el sufrimiento del desarrollo facilitando las tareas comunes utilizadas en la mayoría de los proyectos web, como la autenticación, enrutamiento, sesiones y almacenamiento en caché.

(Organization, 2018, pág. 3)

Laravel es un framework para el lenguaje de programación PHP. Aunque PHP es conocido por tener una sintaxis poco deseable, es fácil de usar, fácil de desplegar y



se le puede encontrar en muchos de los sitios web modernos que usas día a día. Laravel no solo ofrece atajos útiles, herramientas y componentes para ayudarte a conseguir el éxito en tus proyectos basados en web, sino que también intenta arreglar alguna de las flaquezas de PHP. (Organization, 2018, pág. 3)

Laravel tiene una sintaxis bonita, semántica y creativa, que le permite destacar entre la gran cantidad de frameworks disponibles para el lenguaje. Hace que PHP sea un placer, sin sacrificar potencia y eficiencia. Es sencillo de entender, permite mucho el modularidad de código lo cual es bueno en la reutilización de código. (Organization, 2018, pág. 3)

### **2.12.1 Beneficios de Laravel**

- Incluye un ORM: A diferencia de CodeIgniter, Laravel incluye un ORM integrado. Por lo cual no debes instalar absolutamente nada.
- Bundles: existen varios paquetes que extienden a Laravel y te dan funcionalidades increíbles.
- Programas de una forma elegante y eficiente: No más código basura o espagueti que no se entienden, aprenderás a programar 'con clase' y ordenar tu código de manera de que sea lo más reutilizable posible.
- Controlas la BD desde el código: Puedes tener un control de versiones de lo que haces con ella. A esto se llaman migrations, es una excelente herramienta, porque puedes manejar todo desde tu IDE, inclusive montar datos en tus tablas.
- Da soporte a PHP 5.3.
- Rutas elegantes y seguras: Una misma ruta puede responder de distinto modo a un método GET o POST.
- Cuenta con su propio motor de plantillas HTML.

- Se actualiza fácilmente desde la línea de comandos: El framework es actualizable utilizando composer update y listo, nada de descargar un ZIP y estar reemplazando.
- Cuenta con una comunidad activa que da apoyo rápido al momento de que lo necesitas. (Organization, 2018, pág. 3)

### **2.13 Plantilla AdmiLTE**

Hoy día es muy importante cuidar no solo la funcionalidad sino también la estética de las aplicaciones. Los usuarios no buscan solo una aplicación funcional, también ha de ser una aplicación “bonita”, y todo ello debe realizarse en el menor tiempo posible. Además, las aplicaciones web deben poderse ejecutar en diferentes dispositivos como PC, móvil, Tablet, etc. sin perder funcionalidad. (ARNAU, 2017, pág. 18)

Es por esto y para no tener que desarrollar una aplicación para cada tipo de dispositivo que nacieron diferentes conjuntos de frameworks o lenguajes como pueden ser los tres presentados aquí. (ARNAU, 2017, pág. 18)

Mientras que Bootstrap es un framework para el diseño de sitios web, AdminLTE es una aplicación base que utiliza este framework para crear una interfaz adaptativa (responsive) que se adapte a cada tipo de dispositivo mediante un conjunto de reglas CSS y apoyándose en jQuery para la aplicación de efectos sobre los diferentes elementos que podemos encontrar de nuestra aplicación. (ARNAU, 2017, pág. 18)

Aunque AdminLTE incluye muchísimos componentes con una estética agradable, otros muchos componentes que iba a necesitar la aplicación no estarían disponibles y tendría que crearlos a mano. (ARNAU, 2017, pág. 18)

### **2.14 Case Powerdesigner**

Esta herramienta de diseño está más orientada al análisis y diseño de base de datos relacionales. Se puede crear el modelo físico y normalizar. Permite hacer

ingeniería reversa (reverse engineer) para desarrollar los modelos físicos y conceptuales a partir de la información de la base de datos. (Garcia, 2009, pág. 22)

Módulos

**2.14.1 Power Designer DataArchitect:**

Modelamiento físico y conceptual de datos integrados (PDM, CDM). (Garcia, 2009, pág. 22)

**2.14.2 Power Designer PhysicalArchitect:**

Ofrece la capacidad del modelo de datos físicos incluyendo la generación del código SQL e ingeniería reversa (reverse engineer) de bases de datos existentes (PDM). (Garcia, 2009, pág. 23)

**2.14.3 Power Designer Developer:**

Modelamiento físico de datos y orientados a objetos integrado. Para desarrolladores que requieran diseños de objetos (OOM, PDM). (Garcia, 2009, pág. 23)

**2.14.4 Power Designer ObjectArchitect:**

Combina la funcionalidad de la modelación de objetos, modelación de datos físicos integrados en una sola interfaz (PDM, CDM, OOM). (Garcia, 2009, pág. 23)

## 3 Desarrollo

### Introducción

El desarrollo de este sistema Software es con fines de automatizar el proceso de planificación de ventas y compras de la empresa Agrícola Moreplant, la cual será utilizada como una herramienta de trabajo misma que ayudará a mejorar el proceso de facturación en el área de contabilidad.

El objetivo principal para el desarrollo del sistema compone de una metodología ágil XP Extrem Programing, de acuerdo a sus características tenemos cuatro ciclos del software las cuales se realizará planificación, diseño, desarrollo, pruebas son los aspectos importantes para ir conjuntamente con el desarrollo.

A continuación, se presenta el proceso de desarrollo de software, aplicando la metodología XP Programación Extrema.

### 3.1 Planificación

#### 3.1.1 *Propósito*

El presente documento tiene como finalidad definir y cumplir con los requerimientos funcionales y no funcionales para desarrollar los módulos del sistema de control de facturación e inventarios de la empresa Agrícola Moreplant, permitiendo gestionar los procesos administrativos al área de ventas.

#### 3.1.2 *Alcance*

En el control de los procesos de facturación e inventarios, el cual permite la automatización de un sistema para los módulos contables de la empresa Agrícola Moreplant.

### 3.1.3 Personal involucrado

**Tabla 2**

*Personal involucrado 1*

<b>Nombre</b>	Taipe Rocha Carla Estefanía
<b>Rol</b>	Desarrollador
<b>Categoría profesional</b>	Tecnología en computación Estudiante ESPE
<b>Información de contacto</b>	cetaipe@espe.edu.ec / 0984294819
<b>Aprobación</b>	

*Nota:* Datos del personal involucrado en el desarrollo del sistema.

### 3.1.4 Perspectiva del producto

La empresa Agrícola Moreplant cuenta con varios procesos en el ámbito agrícola, los cuales engloban todas las actividades de ventas, compras, proveedores, clientes, entre otras. El presente documento se encargará en los procesos y actividades de automatización, lo que permite facilidad y eficacia en el ámbito laboral.

## 3.2 Requisitos Funcionales

En esta sección se dará una descripción global del funcionamiento del sistema el cual se ha desarrollado, el cual permite dar una visualización del contenido en general.

### 3.2.1 Modulo Inicilizacion

RF1. Para ingresar al sistema el administrador debe poseer su usuario y contraseña, el cual será generado en el desarrollo del sistema con opción de edición,

RF2. Para ingresar al sistema el vendedor debe poseer su usuario y contraseña, mismo que será generado por el administrador ya que es la persona que dará acceso a los vendedores.

RF3. El sistema permitirá elegir el tipo de acceso al sistema, sea administrador o vendedor.

### **3.2.2 Modulo Administrador**

RF1. El sistema editará al administrador su usuario y contraseña, el cual el sistema validará la información y finalmente podrá hacer uso del sistema.

RF2. Una vez validado la información correspondiente el usuario podrá visualizar la página principal en el cual contiene los módulos contables.

RF3. El sistema dará accesos al administrador a todos los módulos, con la finalidad de controlar los movimientos de los vendedores.

RF4. El sistema dará acceso al módulo de almacén, compras, ventas, cartera, caja, producción, acceso, reportes.

### **3.2.3 Modulo Vendedor**

RF5. El sistema editará al vendedor su usuario y contraseña, el cual el sistema validará la información y finalmente podrá hacer uso del sistema.

RF6. Una vez validado la información correspondiente el usuario podrá visualizar la página principal en el cual contiene los módulos contables.

RF7. El sistema tiene restricciones en los módulos con el vendedor, esto con finalidad de salvaguardar la información.

RF8. El sistema dará acceso al módulo de ventas, compras, inventarios, reportes para el manejo de su trabajo diario.

**Tabla 3***Requisito funcional 1*

		Historia de Usuario
Número: 1	Usuario: <b>RESPONSABLE GERENTE GENERAL</b>	
Nombre historia: <b>REGISTRO DE PERSONAS</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>REQUISITO</b>	
Puntos estimados: 2	Iteración asignada: 1	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá ingresar usuarios y contraseñas al administrador		
Observaciones: Para ingresar al sistema el administrador debe poseer su usuario y contraseña, el cual será generado en el desarrollo del sistema con opción de edición.		
<i>Nota:</i> Especificación del requerimiento funcional.		

**Tabla 4***Requisito Funcional 2*

		Historia de Usuario
Número: 2	Usuario: <b>RESPONSABLE GERENTE GENRAL</b>	
Nombre historia: <b>REGISTRO DE PERSONAS</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>REQUISITO</b>	
Puntos estimados: 2	Iteración asignada: 1	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá ingresar usuario y contraseña al vendedor		
Observaciones: Para ingresar al sistema el vendedor debe poseer su usuario y contraseña, mismo que será generado por el administrador ya que es la persona que dará acceso a los vendedores, con la opción de editar al momento de inicializar.		
<i>Nota:</i> Especificación del requerimiento funcional.		

**Tabla 5***Requisito Funcional 3*

		Historia de Usuario
Número: <b>3</b>	Usuario: <b>RESPONSABLE GERENTE GENERAL</b>	
Nombre historia: <b>REGISTRO DE PERSONAS</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito</b>	
Puntos estimados: 2	Iteración asignada: <b>1</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá elegir el tipo de usuario		
Observaciones: El sistema permitirá elegir el tipo de acceso al sistema sea administrador o vendedor		

*Nota:* Especificación del requerimiento funcional.

**Tabla 6***Requisito Funcional 4*

		Historia de Usuario
Número: <b>4</b>	Usuario: <b>ADMINISTRADOR</b>	
Nombre historia: <b>REGISTRO DE FUNCIONES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito</b>	
Puntos estimados: 2	Iteración asignada: <b>2</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá editar su usuario y contraseña del administrador.		
Observaciones: El sistema permitirá editar al administrador su usuario y contraseña, el cual fue creado en el desarrollo, el sistema validará la información finalmente podrá hacer uso del sistema.		

*Nota:* Especificación del requerimiento funcional.



**Tabla 7***Requisito Funcional 5*

		<b>Historia de Usuario</b>
Número: <b>5</b>	Usuario: <b>ADMINISTRADOR</b>	
Nombre historia: <b>REGISTRO DE FUNCIONES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito</b>	
Puntos estimados: 1	Iteración asignada: <b>2</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema validara la información del administrador.		
Observaciones: Una vez validado la información correspondiente el administrador podrá visualizar la página principal en el cual contiene los módulos contables.		

*Nota:* Especificación del requerimiento funcional.

**Tabla 8***Requisito Funcional 6*

		<b>Historia de Usuario</b>
Número: <b>6</b>	Usuario: <b>ADMINISTRADOR</b>	
Nombre historia: <b>REGISTRO DE FUNCIONES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito</b>	
Puntos estimados: 11	Iteración asignada: <b>2</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá dar funcionalidad al sistema, dando acceso a todos los módulos contables.		
Observaciones: El sistema le permitirá tener accesos al administrador a todos los módulos, con la finalidad de controlar los movimientos de los vendedores		

*Nota:* Especificación del requerimiento funcional.

**Tabla 9***Requisito Funcional 7*

		<b>Historia de Usuario</b>
Número: 7	Usuario: <b>ADMINISTRADOR</b>	
Nombre historia: <b>REGISTRO DE FUNCIONES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito</b>	
Puntos estimados: 8	Iteración asignada: <b>2</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá la funcionalidad en cada módulo.		
Observaciones: El sistema le permitirá tener acceso al módulo de almacén, compras, ventas, cartera, caja, producción, acceso, reportes.		

*Nota:* Especificación del requerimiento funcional.

**Tabla 10***Requisito Funcional 8*

		<b>Historia de Usuario</b>
Número: 8	Usuario: <b>VENDEDOR</b>	
Nombre historia: <b>REGISTRO DE ACTIVIDADES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito</b>	
Puntos estimados: 2	Iteración asignada: <b>3</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá editar su usuario y contraseña del vendedor.		
Observaciones: El sistema permitirá que el vendedor edite su usuario y contraseña, el cual el sistema validará la información y finalmente podrá hacer uso del sistema.		

*Nota:* Especificación del requerimiento funcional.

**Tabla 11***Requisito Funcional 9*

		Historia de Usuario
Número: <b>9</b>	Usuario: <b>VENDEDOR</b>	
Nombre historia: <b>REGISTRO DE ACTIVIDADES</b>		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito</b>	
Puntos estimados: 2	Iteración asignada: <b>3</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá editar su usuario y contraseña del vendedor.		
Observaciones: El sistema permitirá que el vendedor edite su usuario y contraseña, el cual el sistema validará la información y finalmente podrá hacer uso del sistema.		

*Nota:* Especificación del requerimiento funcional.

**Tabla 12***Requisito Funcional 10*

		Historia de Usuario
Número: <b>10</b>	Usuario: <b>VENDEDOR</b>	
Nombre historia: <b>REGISTRO DE ACTIVIDADES</b>		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito, Restricción</b>	
Puntos estimados: 4	Iteración asignada: <b>3</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá ingresar al vendedor		
Observaciones: Una vez validado la información correspondiente el usuario podrá visualizar la página principal en el cual contiene los módulos contables de acceso		

*Nota:* Especificación del requerimiento funcional.

**Tabla 13***Requisito Funcional 11*

		Historia de Usuario
Número: <b>11</b>	Usuario: <b>VENDEDOR</b>	
Nombre historia: <b>REGISTRO DE ACTIVIDADES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito, Restricción</b>	
Puntos estimados: 2	Iteración asignada: 4	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá ingresar al vendedor		
Observaciones: Una vez validado la información correspondiente el usuario podrá visualizar la página principal en el cual contiene los módulos contables de acceso		
<i>Nota:</i> Especificación del requerimiento funcional.		

**Tabla 14***Requisito Funcional 12*

		Historia de Usuario
Número: <b>12</b>	Usuario: <b>VENDEDOR</b>	
Nombre historia: <b>REGISTRO DE ACTIVIDADES</b>		
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito, Restricción</b>	
Puntos estimados: 2	Iteración asignada: 4	
Programador responsable: Carla Taipe		
Descripción: El sistema permitirá dar acceso al vendedor.		
Observaciones: El sistema tiene restricciones en los módulos con el vendedor, esto con finalidad de salvaguardar la información.		
<i>Nota:</i> Especificación del requerimiento funcional.		

**Tabla 15***Requisito Funcional 13*

<b>Historia de Usuario</b>	
Número: <b>13</b>	Usuario: <b>VENDEDOR</b>
Nombre historia: <b>REGISTRO DE ACTIVIDADES</b>	
Prioridad en negocio: <b>ALTA</b>	Tipo: <b>Requisito, Restricción</b>
Puntos estimados: 2	Iteración asignada: <b>4</b>
Programador responsable: Carla Taipe	
Descripción: El sistema mostrará y dará funcionalidad a los módulos de vendedor	
Observaciones: El sistema permitirá tener acceso al módulo de ventas, cartera, caja, reportes para el manejo de su trabajo diario	

*Nota:* Especificación del requerimiento funcional.

### 3.3 Requisitos no Funcionales

RNF1. El sistema garantizara el rendimiento de procesos del sistema.

RNF2. El sistema tendrá seguridad para salvaguardar la información económica que posee la empresa.

RNF3. El sistema tendrá un interfaz intuitivo y sencillo para la adaptación del usuario.

RNF4. El sistema tendrá disponibilidad 24/7 para el uso del administrador y cumpla con las funciones.

RNF5. El sistema dispondrá operatividad para realizar todas las funciones necesarias tanto como para el administrador y el vendedor.

RNF6. El sistema dispondrá portabilidad siempre y cuando estemos conectados a un dispositivo con internet.

**Tabla 16***Requisito no Funcional 1*

		Historia de Usuario
Número: 1	Usuario: <b>Sistema</b>	
Nombre historia: Seguridad		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito,</b>	
Puntos estimados: 1	Iteración asignada: 1	
Programador responsable: Carla Taipe		
Descripción: El sistema tendrá seguridad para salvaguardar la información económica que posee la empresa.		
Observaciones: El sistema permitirá que la seguridad evite las vulnerabilidades de seguridad contra amenazas, tales como algún tipo de modificación y accesos no permitidos.		

*Nota:* Especificación del requerimiento no funcional.

**Tabla 17***Requisito no Funcional 2*

		Historia de Usuario
Número: 2	Usuario: <b>Sistema</b>	
Nombre historia: Fiabilidad		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito,</b>	
Puntos estimados: 1	Iteración asignada: 1	
Programador responsable: Carla Taipe		
Descripción: El sistema garantizara el rendimiento de procesos del sistema.		
Observaciones: El sistema permitirá que sea fiable el software, proporcionando las especificaciones originales del software, presentando sin errores ningún módulo.		

*Nota:* Especificación del requerimiento no funcional.

**Tabla 18***Requisito no Funcional 3*

		Historia de Usuario
Número: <b>3</b>	Usuario: <b>Sistema</b>	
Nombre historia: <b>Fiabilidad</b>		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito,</b>	
Puntos estimados: <b>1</b>	Iteración asignada: <b>1</b>	
Programador responsable: <b>Carla Taípe</b>		
Descripción: El sistema garantizara el rendimiento de procesos del sistema.		
Observaciones: El sistema permitirá que sea fiable el software, proporcionando las especificaciones originales del software, presentando sin errores ningún módulo.		
<i>Nota:</i> Especificación del requerimiento no funcional.		

**Tabla 19***Requisito no Funcional 4*

		Historia de Usuario
Número: <b>4</b>	Usuario: <b>Sistema</b>	
Nombre historia: <b>Disponibilidad</b>		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito,</b>	
Puntos estimados: <b>1</b>	Iteración asignada: <b>1</b>	
Programador responsable: <b>Carla Taípe</b>		
Descripción: El sistema garantizara el rendimiento de procesos del sistema.		
Observaciones: El sistema permitirá que sea fiable el software, proporcionando las especificaciones originales del software, presentando sin errores ningún módulo.		
<i>Nota:</i> Especificación del requerimiento no funcional.		

**Tabla 20****Requisito no Funcional 5**

		Historia de Usuario
Número: 5	Usuario: <b>Sistema</b>	
Nombre historia: Operatividad		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito,</b>	
Puntos estimados: 1	Iteración asignada: <b>1</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema dispondrá operabilidad para realizar todas las funciones necesarias tanto como para el administrador y el vendedor.		
Observaciones: El sistema permitirá que las funciones del sistema sean accesibles para la funcionalidad de todos los módulos empleados.		

*Nota:* Especificación del requerimiento no funcional.

**Tabla 21****Requisito no Funcional 6**

		Historia de Usuario
Número: 6	Usuario: <b>Sistema</b>	
Nombre historia: Portabilidad		
Prioridad en negocio:	Tipo:	
<b>ALTA</b>	<b>Requisito,</b>	
Puntos estimados: 1	Iteración asignada: <b>1</b>	
Programador responsable: Carla Taipe		
Descripción: El sistema dispondrá portabilidad siempre y cuando estemos conectados a un dispositivo con internet.		
Observaciones: El sistema permitirá la funcionalidad del sistema siempre y cuando tenga navegabilidad a internet, de esa manera podrá ser visualizada desde cualquier dispositivo.		

*Nota:* Especificación del requerimiento no funcional.



### 3.4 Planificación de los Lanzamientos

Después de establecer los requerimientos funcionales se realiza la planificación de lanzamiento o entregas, que con ello se establecerá el plan de lanzamiento.

En decir que la evaluación, riesgos y prioridades realizando estimaciones basadas en los spike (investigar).

**Tabla 22**

*Plan de Lanzamiento*

N	NOMBRE DE HISTORIA	ITERACION				
		1	2	3	4	5
1	REGISTRO DE PERSONAS	X				
2	REGISTRO DE FUNCIONES		X	X		
3	REGISTRO DE ACTIVIDADES				X	
4	<b>SISTEMA</b>					<b>X</b>

*Nota:* Plan de Lanzamiento.

### 3.5 Iteraciones

A continuación, presentamos el Plan de Iteraciones que se ha ido elaborando conjuntamente con el cliente, dando así una fecha límite en 20 semanas para la finalización del proyecto.

**Figura 10***Plan de Iteraciones*

I T E R A C I O N E S	N - H I S T O R I A	SEMANAS																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	■	■	■																	
	2	■	■	■																	
	3	■	■	■																	
2	4				■	■	■	■	■												
	5				■	■	■	■	■												
3	6									■	■	■	■								
	7									■	■	■	■								
4	8													■	■	■	■				
	9													■	■	■	■				
	10													■	■	■	■				
5	11																		■	■	■
	12																		■	■	■
	13																		■	■	■

Nota: Plan de Iteraciones.

El plan de lanzamiento ha servido para que sea agrupadas con las iteraciones ya que de esa forma hay acuerdo a la complejidad y afinidad en los requerimientos a desarrollarse.

### 3.6 Diseño

Se considera el diseño que tiene la metodología como propuesta y poner en practica

#### 3.6.1 Modelación de la Base De Datos

En la modelación de nuestra base de datos se determina los requerimientos funcionales, la importancia es que se pueda cumplir con las necesidades planteadas por parte del cliente.

### **3.6.2 *Modelo Entidad Relación***

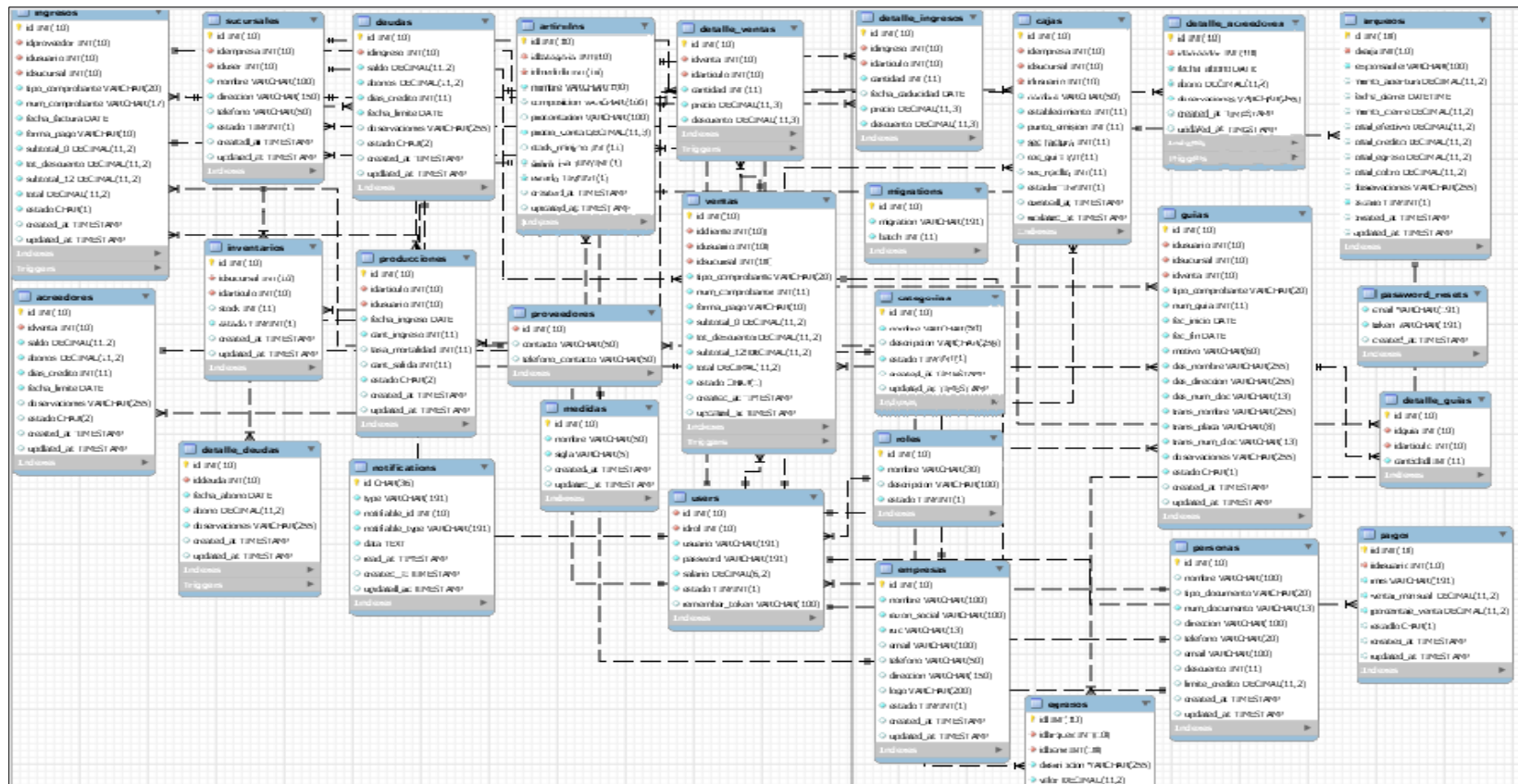
El modelo entidad relación permite visualizar la relación que tiene con otras entidades, estas entidades son reales, distinguibles y definidas por su atributo la cual almacena información.

### **3.6.3 *Modelo Relacional***

Este modelo es la representación del modelo Entidad Relación para su implementación sobre un sistema gestor de datos relaciones, a menudo cambia con respecto al modelo E-R. Como podemos ver permite el muestreo el detalle de la estructura de las tablas y las relaciones con los datos que tiene cada tabla.

Figura 11

Relación de Tablas



Nota: Se observa la relación de tablas mismas que tienen los respectiva información y datos.

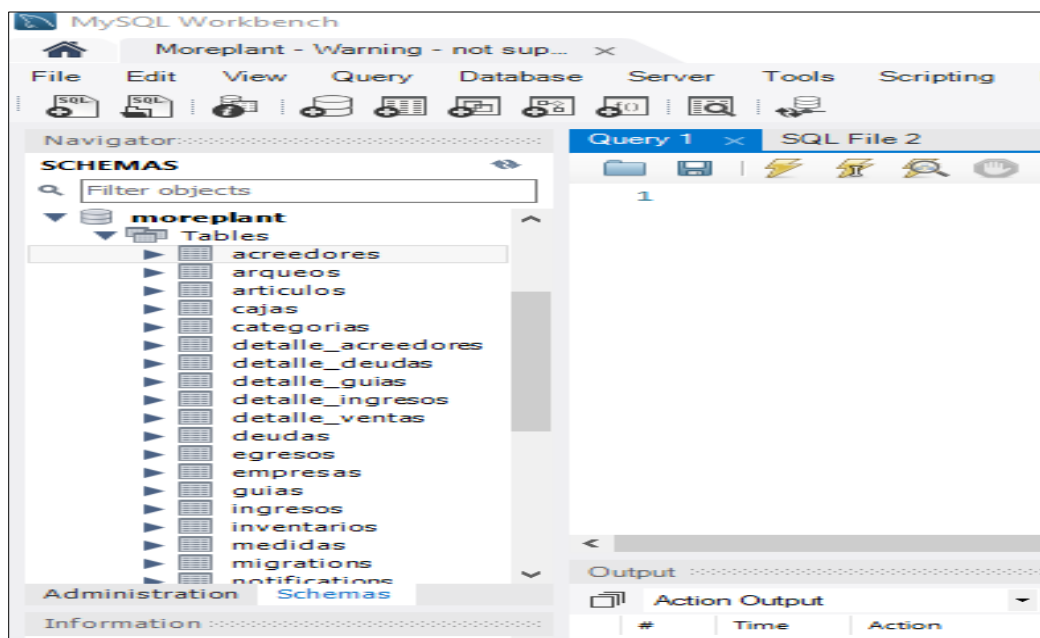
### 3.7 Codificación

#### 3.7.1 Implementación en el Sistema Gestor de Base De Datos

El sistema gestor base de datos nos permite realizar funciones como la creación de tablas, columnas, tipo de datos, restricciones (claves primarias, claves foráneas) para esto hemos ocupado el sistema gestor MySQL.

#### Figura 12

Creación de Tablas



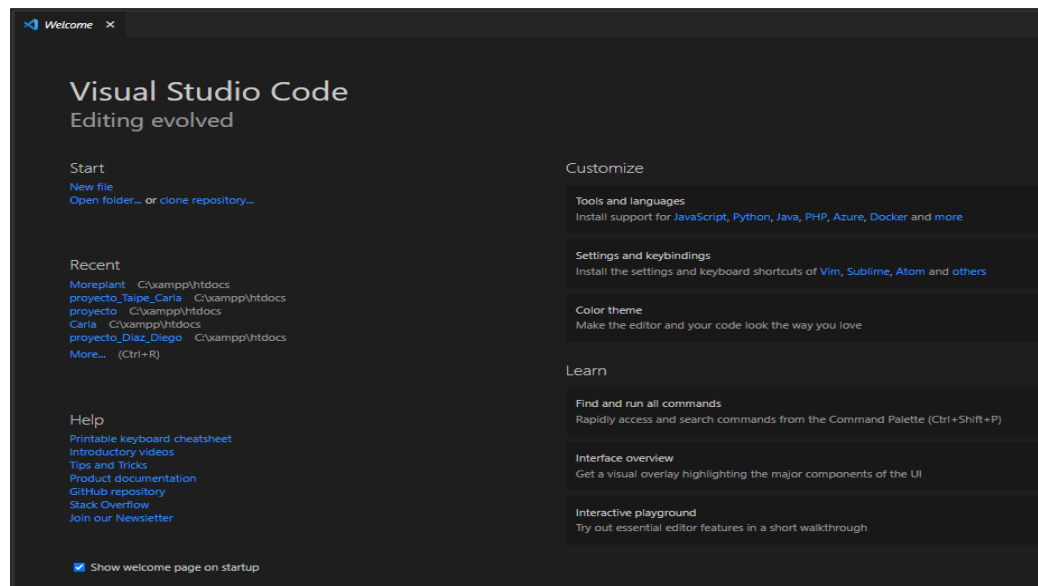
*Nota:* Tenemos la creación de nuestra Base de Datos.

#### 3.7.2 Visual Studio Code

Es un editor de código fuente que permite trabajar con diversos lenguajes de programación, gestiona tus propios atajos de teclado y refactorizar el código. Es gratuito y de código abierto.

**Figura 13**

*Interfaz Visual Studio Code*



*Nota:* Editor de Texto.

### **3.7.3 Modelo, Vista, Controlador.**

Es un tipo de tipo de diseño que separa en capas muy bien definidas el desarrollo de una aplicación las cuales son tres el modelo encargado de la lógica la aplicación y la aplicación y la conexión con la base de datos.

### **3.7.4 Modelo**

Es la representación de datos que el sistema maneja, su lógica de negocio y mecanismos de persistencia. (Alicante, 2021)

Figura 14

*Modelo de Empresa*

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Empresa extends Model
8  {
9      protected $table = 'empresas';
10     //protected $primaryKey = 'id';
11     protected $fillable = ['nombre', 'razon_social', 'ruc', 'email', 'telefono', 'direccion', 'logo'];
12
13     public function sucursales(){
14         return $this->hasMany('App\Sucursal');
15     }
16     public function cajas()
17     {
18         return $this->hasMany('App\Caja');
19     }
20 }
21

```

*Nota: Código Empresa.*

Figura 15

*Modelo de Ventas*

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Venta extends Model
8  {
9      protected $fillable = [
10         'idcliente',
11         'idusuario',
12         'idsucursal',
13         'tipo_comprobante',
14         'num_comprobante',
15         'forma_pago',
16         'subtotal_0',
17         'tot_descuento',
18         'subtotal_12',
19         'total',
20         'estado'
21     ];
22     public function usuario()
23     {
24         return $this->belongsTo('App\User');
25     }
26     public function cliente()
27     {
28         return $this->belongsTo('App\Persona');
29     }
30 }

```

*Nota: Código Ventas.*

**Figura 16**

*Modelo de Artículo*

```
Articulo.php X
app > Articulo.php > ...
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Articulo extends Model
8  {
9      protected $fillable = [
10         'idcategoria', 'idmedida', 'nombre', 'composicion', 'presentacion', 'precio_venta', 'stock_minimo', 'grava_iva', 'estado'
11     ];
12     public function categoria()
13     {
14         return $this->belongsTo('App\Categoria');
15     }
16     public function medida()
17     {
18         return $this->belongsTo('App\Medida');
19     }
20 }
21
```

*Nota:* Código Artículo.

### **3.7.5 Vista**

Es la vista o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con este. (Alicante, 2021)



Figura 17

Vista Bienvenido

```

welcome.blade.php X
resources > views > welcome.blade.php > ...
1  <!doctype html>
2  <html lang="{{ app()->getLocale() }}">
3  <head>
4      <meta charset="utf-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7
8      <title>Laravel</title>
9
10     <!-- Fonts -->
11     <link href="https://fonts.googleapis.com/css?family=Raleway:100,600" rel="stylesheet" type="text/css">
12
13     <!-- Styles -->
14     <style>
15         html, body {
16             background-color: #fff;
17             color: #636b6f;
18             font-family: 'Raleway', sans-serif;
19             font-weight: 100;
20             height: 100vh;
21             margin: 0;
22         }
23
24         .full-height {
25             height: 100vh;
26         }
27
28         .flex-center {
29             align-items: center;
30             display: flex;

```

*Nota:* Aquí se detalla la Bienvenida del Interfaz.

Figura 18

Vista Inicio

```

contenido.blade.php X
resources > views > auth > contenido.blade.php > ...
1  <!DOCTYPE html>
2  <html lang="es">
3
4  <head>
5
6      <meta charset="utf-8">
7      <meta http-equiv="X-UA-Compatible" content="IE=edge">
8      <title>MorePlant | Inicio</title>
9      <!-- Tell the browser to be responsive to screen width -->
10     <meta name="viewport" content="width=device-width, initial-scale=1">
11     <!-- Font Awesome -->
12     <link rel="stylesheet" href="css/plantilla.css">
13     <!-- Id for Channel Notification -->
14     <meta name="userId" content="{{ Auth::check() ? Auth::user()->id : '' }}">
15     <meta name="csrf-token" content="{{ csrf_token() }}">
16
17 </head>
18
19 <body class="hold-transition login-page">
20     <div class="login-box">
21         @yield('login')
22     </div>
23
24     <!-- Bootstrap and necessary plugins -->
25     <script src="js/plantilla.js"></script>
26
27 </body>
28
29 </html>

```

*Nota:* Inicio de Login para el Usuario.

Figura 19

Vista Login

```

login.blade.php x
resources > views > auth > login.blade.php > ...
1  @extends('auth.contenido')
2
3  @section('login')
4  <div class="login-logo">
5    <a href="#"><b>Moreplant</b></a>
6  </div>
7  <!-- /.login-logo -->
8  <div class="login-box-body">
9    <form class="was-validated" method="POST" action="{{ route('login')}}">
10     {{ csrf_field() }}
11     <h1>Acceder</h1>
12     <p class="text-muted">Control de acceso al sistema</p>
13     <div class="form-group mb-3{{ $errors->has('usuario' ? 'is-invalid' : '')}}">
14       <span class="input-group-addon bg-light-blue"><i class="fas fa-user"></i></span>
15       <input type="text" value="{{ old('usuario') }}" name="usuario" id="usuario" class="form-control" placeholder="
16       {!! $errors->first('usuario', '<span class="invalid-feedback">:message</span>') !!}
17     </div>
18     <div class="form-group mb-4{{ $errors->has('password' ? 'is-invalid' : '')}}">
19       <span class="input-group-addon bg-light-blue"><i class="fas fa-lock"></i></span>
20       <input type="password" name="password" id="password" class="form-control" autocomplete="on" placeholder="Pa
21       {!! $errors->first('password', '<span class="invalid-feedback">:message</span>') !!}
22     </div>
23
24     <div class="col-6">
25       <button type="submit" class="btn btn-primary px-4">Acceder</button>
26     </div>
27
28   </form>
29 </div>
30 @endsection

```

*Nota:* Ingreso de Usuario y contraseña.

### 3.7.6 Controlador

Actúa como el intermediario entre el modelo y la vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. (Alicante, 2021)

Figura 20

*Controlador Persona*

```

Persona.php X
app > Persona.php > ...
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Persona extends Model
8  {
9      protected $fillable = ['nombre', 'tipo_documento', 'num_documento', 'direccion', 'telefono', 'email', 'descuento', 'limite_credito'];
10
11     public function proveedor()
12     {
13         return $this->hasOne('App\Proveedor');
14     }
15
16     public function user()
17     {
18         return $this->hasOne('App\User');
19     }
20 }
21

```

*Nota:* Tipos de datos de los usuarios.

Figura 21

*Controlador Ingreso*

```

Ingreso.php X
app > Ingreso.php > ...
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Ingreso extends Model
8  {
9      protected $fillable = [
10         'idproveedor',
11         'idusuario',
12         'idsucursal',
13         'tipo_comprobante',
14         'num_comprobante',
15         'fecha_factura',
16         'forma_pago',
17         'subtotal_0',
18         'descuento',
19         'subtotal_12',
20         'total',
21         'estado'
22     ];
23     public function usuario()
24     {
25         return $this->belongsTo('App\User');
26     }
27     public function proveedor()
28     {
29         return $this->belongsTo('App\Proveedor');
30     }
31 }

```

*Nota:* Tipo de Datos del Controlador Ingresos.

Figura 22

*Controlador Proveedor*

```
Proveedor.php X
app > Proveedor.php > ...
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Proveedor extends Model
8  {
9      protected $table = 'proveedores';
10     protected $fillable = [
11         'id', 'contacto', 'telefono_contacto'
12     ];
13
14     public $timestamps = false;
15
16     public function persona()
17     {
18         return $this->belongsTo('App\Persona');
19     }
20
21
22 }
23
```

*Nota:* Tipo de Datos del Controlador Proveedor.

### 3.8 Pruebas

Figura 23

*Login de usuarios*



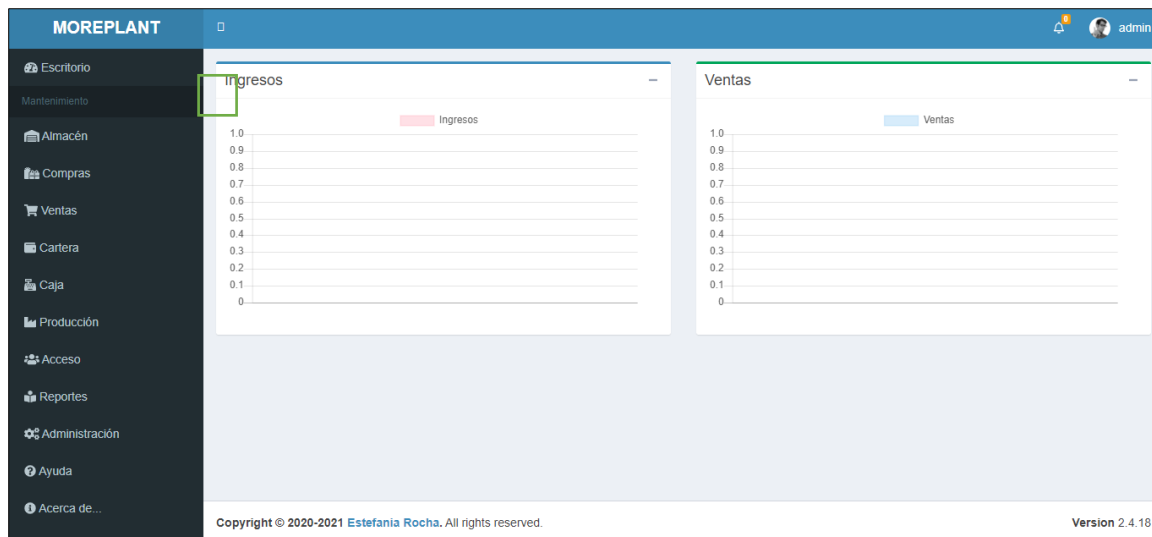
**Moreplant**

# Acceder

Control de acceso al sistema

**Acceder**

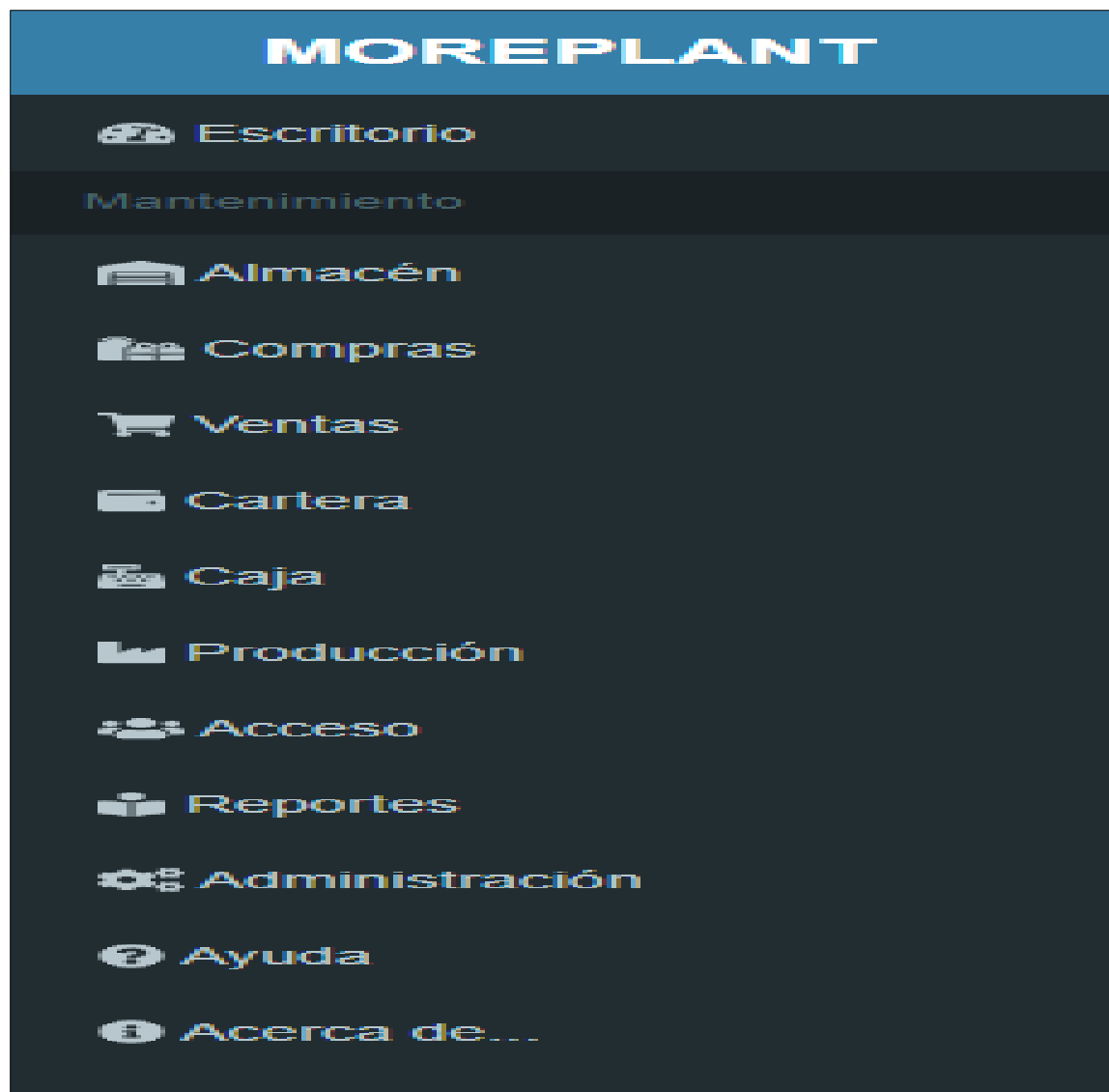
*Nota:* El Login autentica a los usuarios que requieren ingresar.

**Figura 24***Vista página principal*

**Nota:** El Login permitirá autenticarse a los usuarios que requieran ingresar.

Figura 25

Selección del menú



*Nota:* Menú muestra módulos contables.

**Figura 26**

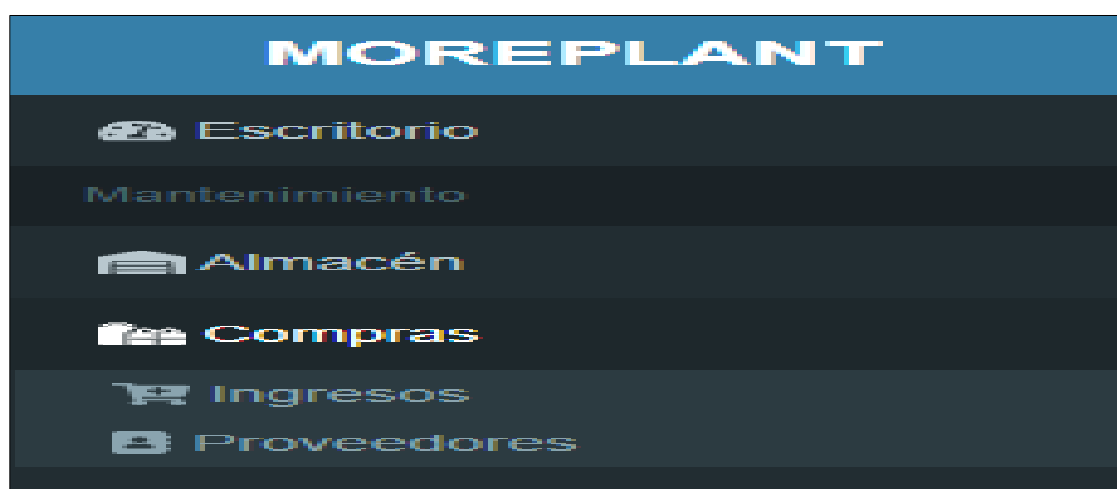
*Submenú de Almacén*



*Nota:* Submódulos Categoría, Artículo, U. Medida, Existencia.

**Figura 27**

*Submenú de Compras*

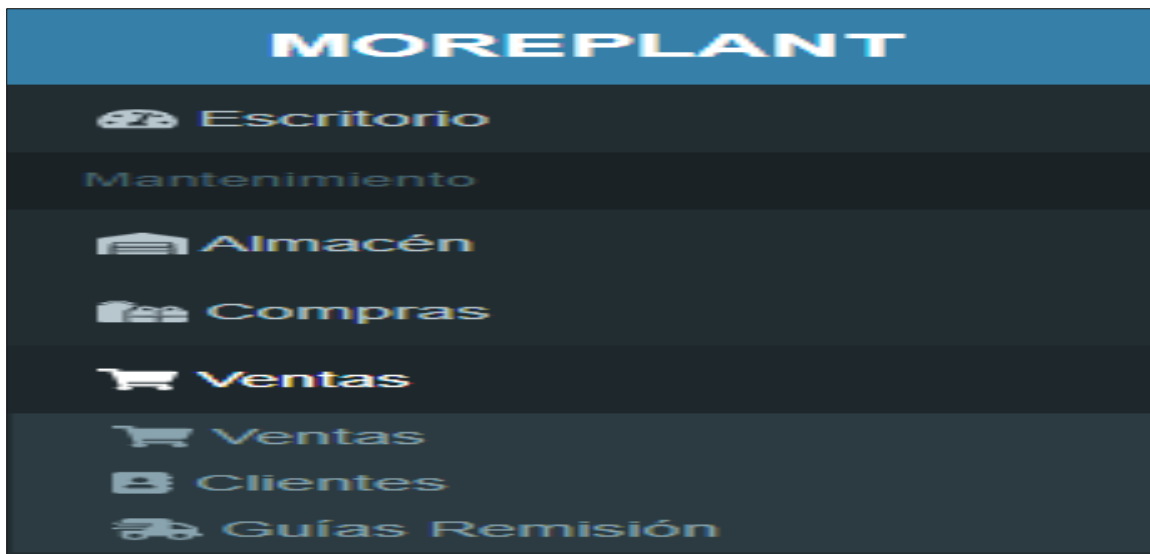


*Nota:* Submódulo Ingresos, Proveedores.



**Figura 28**

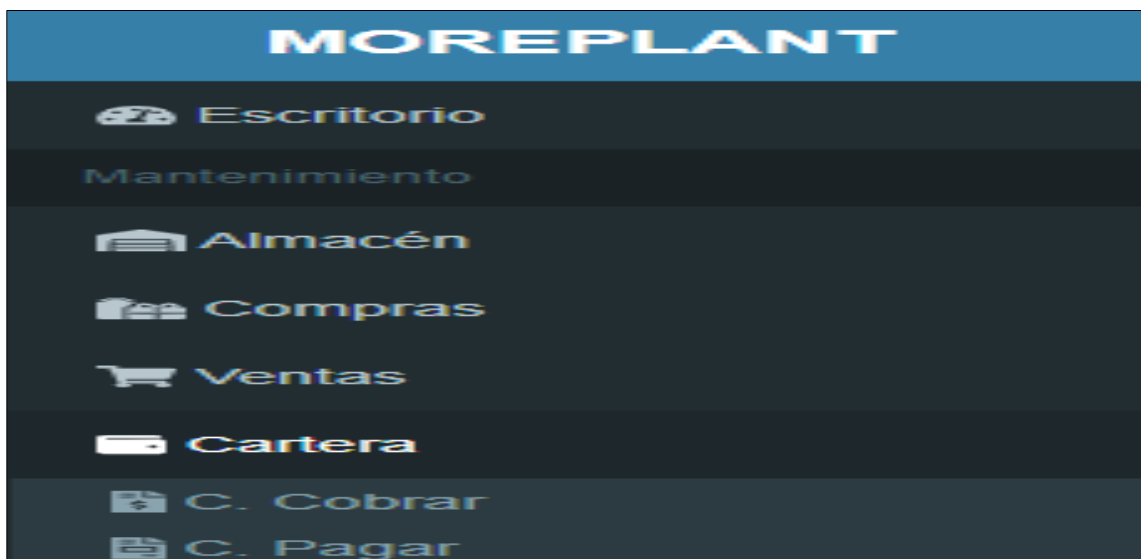
*Submódulo Ventas*



*Nota:* Submódulo Ventas, Clientes, Guía Remisión.

**Figura 29**

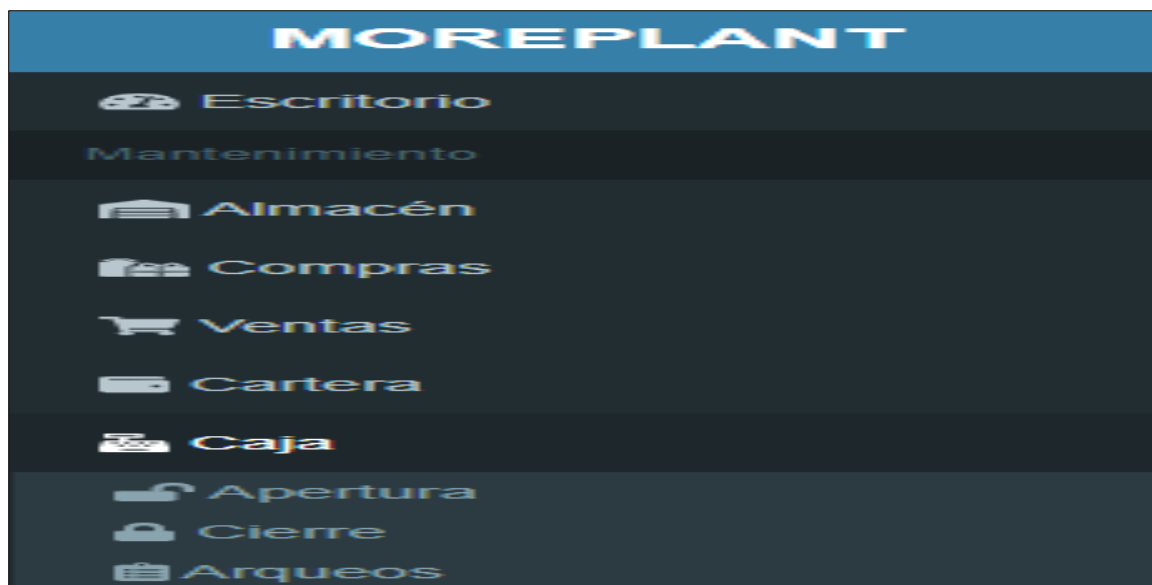
*Submódulo Cartera*



*Nota:* Submódulo Cuentas por Cobrar, Cuentas por Pagar.

**Figura 30**

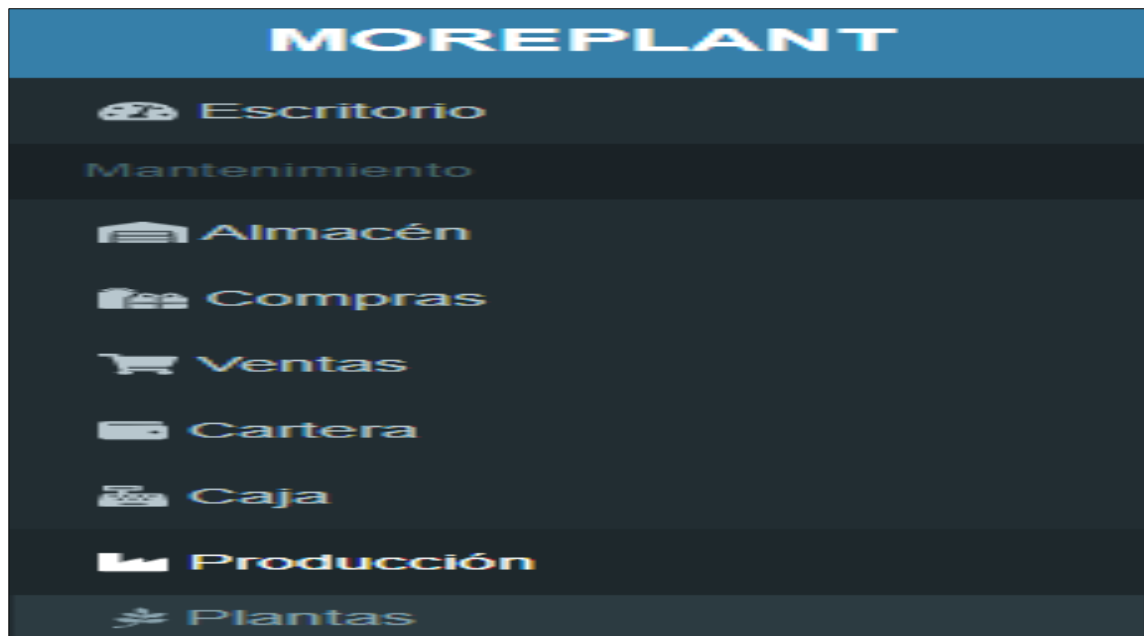
*Submódulo de Caja*



*Nota:* Submódulo Apertura, Cierre, Arqueos.

**Figura 31**

*Submenú Producción*



*Nota:* Submódulo Plantas.

**Figura 32**

*Submódulo Acceso*



*Nota:* Permite seleccionar Usuario, Roles.

Figura 33

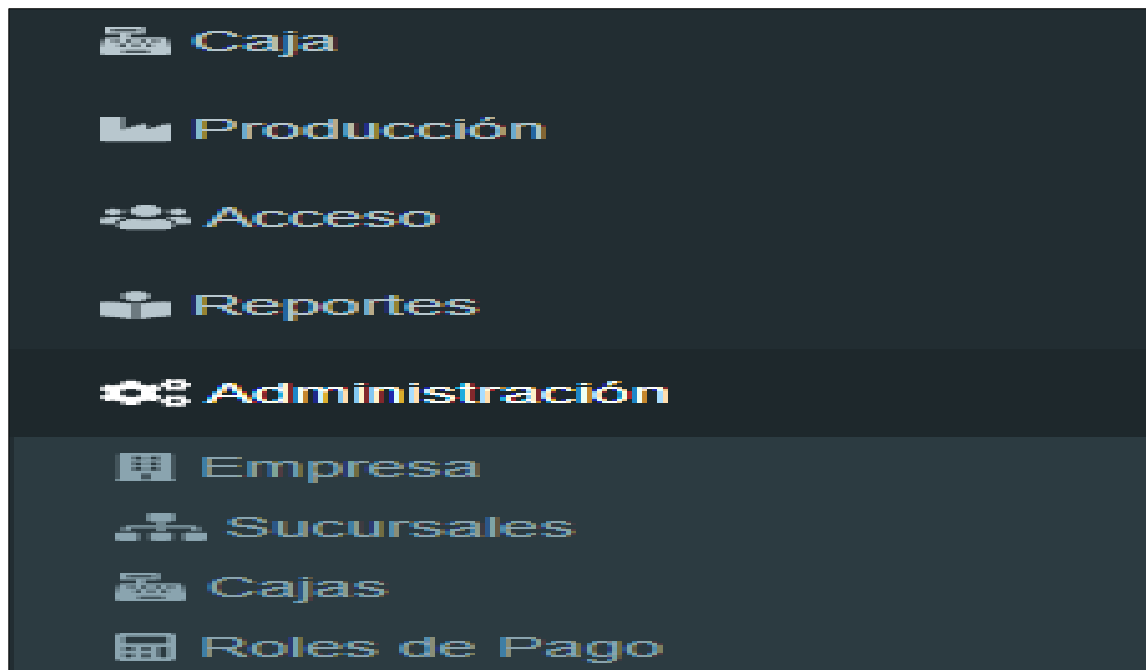
Submódulo Reportes



Nota: Submódulo Reporte Ingreso, Reporte Ventas.

**Figura 34**

*Submódulo Administración*



*Nota:* Submódulo Empresa, Sucursales, Cajas, Roles de Pago.

#### **4.4 Creación e Inicialización del Sistema**

#### 4.4.1 Acceso - Usuarios

**Figura 35**

*Registro de Usuario*

Registrar Usuario ✕

Nombre:	AGRICOLA MOREPLANT		
Tipo documento:	RUC	Número documento:	1804474375001
Dirección:	San Francisco Camilo Cienfuegos s/n		
Teléfono:	0983597700	Email:	christin199203@live.com
Usuario:	Christian	Contraseña:	.....
Salario:	400	Rol:	Administrador

Cerrar Guardar

*Nota:* Muestra Datos de Información cual es el tipo de usuario si administrador y usuario.

**Figura 36**

*Acceso Administración - Empresa*

Registrar Empresa

Logo  
Seleccionar archivo module\_...\_top.png

**Nombre:** AGRICOLA MOREPLANT

**Razón Social:** MANGUI ICHINA CHRISTIAN ORLANDO

**RUC:** 1804474375001 **Teléfonos:** 0983597700

**Email:** Christian19903@live.com

**Dirección:** SAN FRANCISCO CAMILO CIENFUEGOS S/N

Cerrar Guardar

*Nota:* Registro de información para la creación de la Empresa.

**Figura 37**

*Acceso Administración - Sucursales*

Registrar Sucursal

**Empresa:** Agricola Moreplant **Nombre:** Agricola Moreplant 2

**Dirección:** LATACUNGA - SALCEDO

**Teléfono:** 0979035680 **Responsable:** Jessica Tatiana Moreta Telenchana

Cerrar Guardar

*Nota:* Información de Sucursales y quien es responsable.



**Figura 38***Administración - Caja*

<b>Empresa:</b>	<b>Sucursal:</b>	<b>Responsable:</b>
Agricola Moreplant	Agricola Moreplant	Carla Estefania Taij
<b>Nombre Caja:</b>	<b>No. Establecimiento:</b>	<b>No. Punto Emisión:</b>
Cuchibamba	2	2
<b>Sec. Factura:</b>	<b>Sec. Guía Remisión:</b>	<b>Sec. Recibo:</b>
270	1	101

Cerrar Guardar

*Nota:* Muestra la información que se va registrar en la caja.

**Figura 39***Registro de Medida*

<b>Nombre</b>	Kilogramo
<b>Sigla</b>	Kg

Cerrar Guardar

*Nota:* Información del registro de medida que se va manejar.

**Figura 40***Almacén - Categoría*

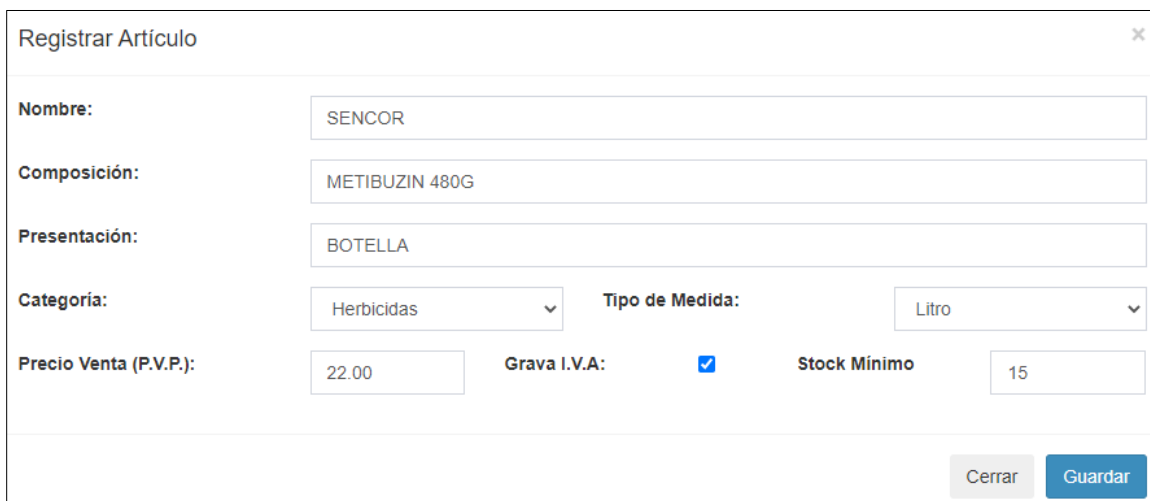
Registrar Categoría

**Nombre** Desparasitante

**Descripción** Cero Diaz

Cerrar Guardar

*Nota:* Registro de información para la clasificación de productos.

**Figura 41***Almacén – Artículo*

Registrar Artículo

**Nombre:** SENCOR

**Composición:** METIBUZIN 480G

**Presentación:** BOTELLA

**Categoría:** Herbidas **Tipo de Medida:** Litro

**Precio Venta (P.V.P.):** 22.00 **Grava I.V.A.:**  **Stock Mínimo:** 15

Cerrar Guardar

*Nota:* Permite el registro del artículo y sus cantidades.

**Figura 42***Registro de Proveedor*

Registrar Proveedor ✕

---

**Nombre Proveedor:**

**Tipo Documento:**  **Número:**

**Dirección:**

**Teléfono:**  **Email:**

**Nombre Contacto:**  **Teléfono del contacto**

---

*Nota:* Información necesaria y requerida del proveedor.

**Figura 43***Compras - Ingresos*

Registrar Ingreso

**Proveedor:**  **Empresa:**  **Sucursal:**

**Tipo Comprobante:**  **Número Comprobante:**  **Fecha Emisión Factura:**

**Artículo (\*Seleccione)**

Acción	Artículo	Cantidad	Fec. Cad.	Precio	Descuento %	Subtotal
<input type="button" value="✕"/>	Sencor	<input type="text" value="2"/>	<input type="text" value="30/04/2021"/>	<input type="text" value="25"/>	<input type="text" value="5"/>	\$ 47.50
<b>Subtotal 0:</b>						\$ 0.00
<b>Descuento:</b>						\$ 2.50
<b>Subtotal 12:</b>						\$ 47.50
<b>Iva 12%:</b>						\$ 5.70
<b>Total Neto:</b>						\$ 50.70

**Forma de Pago:**  **Valor de Crédito:**  **Días de Crédito:**  **Fecha Límite:**

**Observaciones:**

Copyright © 2020-2021 Estefanía Rocha. All rights reserved. Version 2.4.

*Nota:* Permite el registro de compras.

**Figura 44***Caja - Apertura*

Apertura de Caja

Caja: Cuchibamba      Responsable: Stefy      Monto de Apertura: 0

*Nota:* Para realizar una Venta primero deberemos abrir la caja para que se vaya registrando constantemente.

**Figura 45***Ventas - Ventas*

Registrar Venta

Cliete(\*) Daniela Taibe      Tipo Comprobante(\*) Factura      Número Comprobante(\*) 001-004-000000165

No. Identificación: 1720119716      Límite de Crédito: 2000.00      Saldo Pendiente:      Disponible a Facturar: 2000.00

Artículo (\*) Seleccione)

Acción	Artículo	Cantidad	Precio	Descuento %	Subtotal	
<input type="button" value="✖"/>	Sencor	15	22,000	5	\$ 313.50	
					Subtotal 0:	\$ 0.00
					Descuento:	\$ 16.50
					Subtotal 12:	\$ 313.50
					Iva 12%:	\$ 37.62
					Total Neto:	\$ 334.62

Forma de Pago: Crédito      Valor de Crédito: 334.62      Días de Crédito: 30      Fecha Límite: 7/03/2021

Observaciones:



Copyright © 2020-2021 Estefanía Rocha. All rights reserved.      Version 2.0

*Nota:* Esta factura fue diseñada con aspectos y necesidades del cliente las cuales tiene crédito, abonos, observaciones y como resultado su monto límite de crédito.

**Figura 46***Cartera- Cuentas por Cobrar*

☰ Cuentas por Cobrar

Show 10 entries Search:

Acción	Usuario	Proveedor	V. Pagar	Abonos	Fec. Limite	Observaciones	Estado
 	admin	Daniela Taipe	30.24	0.00	2021-03-12	cuentas por pagar	<span>Pendiente</span>

Showing 1 to 1 of 1 entries



Previous 1 Next

*Nota:* Visualiza la venta que se hizo el crédito y se registra directamente en cuentas por cobrar.

**Figura 47***Cartera – Cuentas por Pagar*

☰ Cuentas por Pagar

Show 10 entries Search:

Acción	Usuario	Proveedor	V. Pagar	Abonos	Fec. Limite	Observaciones	Estado
 	admin	Elsa Rocha	50.70	0.00	2021-03-26	Pago para el abono	

Showing 1 to 1 of 1 entries

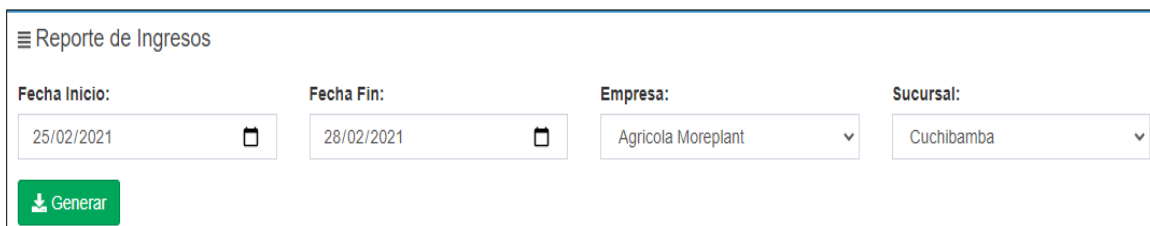
Previous 1 Next

*Nota:* Al inicio se hizo una compra la cual quedo con crédito y automáticamente se pone en las cuentas por pagar.

**Figura 48***Producción - Plantas*

The screenshot shows a web form titled "Ingreso Producción" with a close button (X) in the top right corner. The form contains three input fields: a dropdown menu for "Planta:" with the text "Seleccione" and a downward arrow; a date input field for "Fecha Ingreso:" with the placeholder "dd/mm/aaaa" and a calendar icon; and a text input field for "Cantidad Ingreso:" with the value "0". At the bottom right, there are two buttons: "Cerrar" (grey) and "Guardar" (blue).

*Nota:* Es el registro de plantas para poder hacer el ingreso al almacén.

**Figura 49***Reportes – Reporte Ingresos*

The screenshot shows a web form titled "Reporte de Ingresos" with a hamburger menu icon on the left. The form contains four input fields: "Fecha Inicio:" with the value "25/02/2021" and a calendar icon; "Fecha Fin:" with the value "28/02/2021" and a calendar icon; "Empresa:" with the value "Agricola Moreplant" and a downward arrow; and "Sucursal:" with the value "Cuchibamba" and a downward arrow. At the bottom left, there is a green button with a download icon and the text "Generar".

*Nota:* Aquí escogemos la sucursal, fecha que es lo principal y nos arroja los reportes.

**Figura 50***Reportes – Reporte Ventas*

☰ Reporte de Ventas

**Fecha Inicio:**  
**Fecha Fin:**  
**Empresa:** 
**Sucursal:**

*Nota:* Escogemos la sucursal, fecha que es lo principal y nos arroja los reportes.

**Figura 51***Administración – Roles de Pago*

Generar Rol de Pago ✕

---

**Mes:**

**Empleado:**



**Porcentaje de Utilidad:**

<b>Salario</b>	\$ 480.00
<b>Venta Mensual</b>	\$ 0
<b>Utilidad/Venta</b>	\$ 0.00
<b>Total a Recibir</b>	\$ 480.00

*Nota:* Se escoge lo ítems para poder exportar el rol de pago.

**Figura 52**

*Registro de Empresa del almacén principal con sus Sucursales.*

Empresas <span style="float: right;">+ Nuevo</span>								
Show 10 entries		Search: <input type="text"/>						
Acción	Empresa	Razón Social	RUC	Email	Teléfono/os	Dirección	Logo	Estado
 	Agrícola Moreplant	Christian Orlando Mangui Ichina	1804474375001	Christian199203@live.com	0983597700	San Francisco Camilo Cienfuegos S/n		<span style="background-color: green; color: white; border-radius: 10px; padding: 2px 5px;">Activo</span>

Showing 1 to 1 of 1 entries

Previous 1 Next

*Nota:* Se visualiza la información de las sucursales y sus direcciones.

**Figura 53**

*Consolidación de Caja.*










Sucursales <span style="float: right;">+ Nuevo</span>						
Show 10 entries		Search: <input type="text"/>				
Acción	Empresa	Nombre Sucursal	Dirección	Teléfono/os	Responsable	Estado
 	Agrícola Moreplant	Izamba	Izamba	3042384	Christian Orlando Mangui Ichina	<span style="background-color: green; color: white; border-radius: 10px; padding: 2px 5px;">Activo</span>
 	Agrícola Moreplant	Salcedo	Salcedo Y La Que Cruza	3216547	Jessica Tatiana Moreta Telenchana	<span style="background-color: green; color: white; border-radius: 10px; padding: 2px 5px;">Activo</span>
 	Agrícola Moreplant	Cuchibamba	Ambato Cuchibamba	0984294819	Carla Estefania Taipe Rocha	<span style="background-color: green; color: white; border-radius: 10px; padding: 2px 5px;">Activo</span>

*Nota:* Registro de Encargado de Caja.



**Figura 54**

Guardar documentos de la matriz consolidada

Cajas <span style="float: right;">+ Nuevo</span>									
Show 10 entries		Search: <input type="text"/>							
Acción	Empresa/Sucursal	Nombre	No. Esta.	No. Pto. Emi.	Sec. Factura	Sec. Guía	Sec. Recibo	Responsable	Estado
  	Agricola Moreplant/Izamba	1	1	1	350	250	150	Christian	Activo
  	Agricola Moreplant/Salcedo	Caja 12	1	4	165	156	789	admin	Activo
  	Agricola Moreplant/Cuchibamba	Cuchibamba	2	2	270	1	101	Stefy	Activo

*Nota:* Permite guardar los documentos descargados del sistema.

**Figura 55**







Consolidación de Roles de pagos.

Roles de Pago <span style="float: right;">+ Nuevo</span>							
Show 10 entries		Search: <input type="text"/>					
Acción	Empresa / Sucursal	Nombre	Mes	Salario	T. Ventas	% Ventas	Estado
 	Agricola Moreplant/Salcedo	Jessica Tatiana Moreta Telenchana	2	480.00	0.00	0.00	Registrado
 	Agricola Moreplant/Izamba	Christian Oriando Mangui Ichina	2	500.00	0.00	0.00	Registrado
 	Agricola Moreplant/Cuchibamba	Carla Estefania Talpe Rocha	1	452.00	0.00	0.00	Registrado

*Nota:* Se genera un el documento de pago registrando como un egresa de la empresa.

**Figura 56**




*Consolidación de listado de Almacén - Categorías.*

Categorías <span style="float: right;">+ Nuevo</span>			
Show <span>10</span> entries		Search: <input type="text"/>	
Acción	Nombre	Descripción	Estado
 	Desparasitante	Cero Diaz	<span>Activo</span>
 	Herbicidas	Sencor	<span>Activo</span>
 	Plantas	Piminieto	<span>Activo</span>

*Nota:* Listado de las categorías de productos.

**Figura 57**

*Consolidación de listado de Almacén - Artículos.*

Artículos <span style="float: right;">+ Nuevo</span>									
Show <span>10</span> entries		Search: <input type="text"/>							
Acción	Nombre	Composición	Categoría	Presentación	U. Medida	I.V.A.	P.V.P.	Stock Min.	Estado
 	Sencor	Metibuzin 480g	Herbicidas	Botella	Litro	Si	22.000	15	<span>Activo</span>
 	Pimiento	Pimiento Cacho	Plantas	Unidad	Unidad	No	0.120	200	<span>Activo</span>

Showing 1 to 2 of 2 entries

*Nota:* Listado de artículo con edición y eliminación.

**Figura 58**

*Consolidación de listado de Almacén – U. Medidas.*

Acción	Nombre	Sigla
	Kilogramo	kg.
	Libra	lb.
	Litro	Lt.
	Mililitro	ml.
	Unidad	Uni

*Nota:* Listado de U. Medidas con edición y eliminación.

**Figura 59**

*Consolidación de listado de Almacén – Existencia.*

Producto	Categoría	Stock	Estado
Sencor	Herbicidas	2	Activo

Showing 1 to 1 of 1 entries

Previous 1 Next

*Nota:* Se podrá verificar los productos que se encuentran eligiendo la sucursal.

**Figura 60***Consolidación de Compras - Proveedor.*

☰ Proveedores + Nuevo

Show  entries Search:

Acción	Nombre Prov.	T. Doc.	No. Doc.	Dirección	Teléfono/s	Email	Nombre Contacto	Tel. Contacto
	Agripac S.a	Cedula	0503398752	Ambato - Quero	0984701667	luis192004@hotmail.com	Ing Luis Rocha	3042384
	Elsa Rocha	Ruc	0501575401001	Quito	0958823913	confeccionesy@hotmail.com	Elsa Rocha	0958823913

*Nota:* Listado de Proveedores.**Figura 61***Ventas – Guía de Remisión.*

☰ Registrar Guía

---

**Datos de la Guía**

Comp.:  # Guía:  Inicio:  Fin:

Motivo:  Comp. Factura:

---

**Datos del Destinatario**


Nombre / Razón Social:  RUC / C.I.:  Dirección:

**Datos del Transportista**

Nombre / Razón Social:  RUC / C.I.:  Placa:

---

**Artículos a Transportarse:**

Acción	Cantidad	Unidad	Artículo
	<input type="text" value="1"/>	Litro	Sencor - Botella



Observaciones:

Copyright © 2020-2021 Estefania Rocha. All rights reserved. Version 2.4.11

*Nota:* Registro de información para llevar mercadería a domicilio.

**Figura 62**

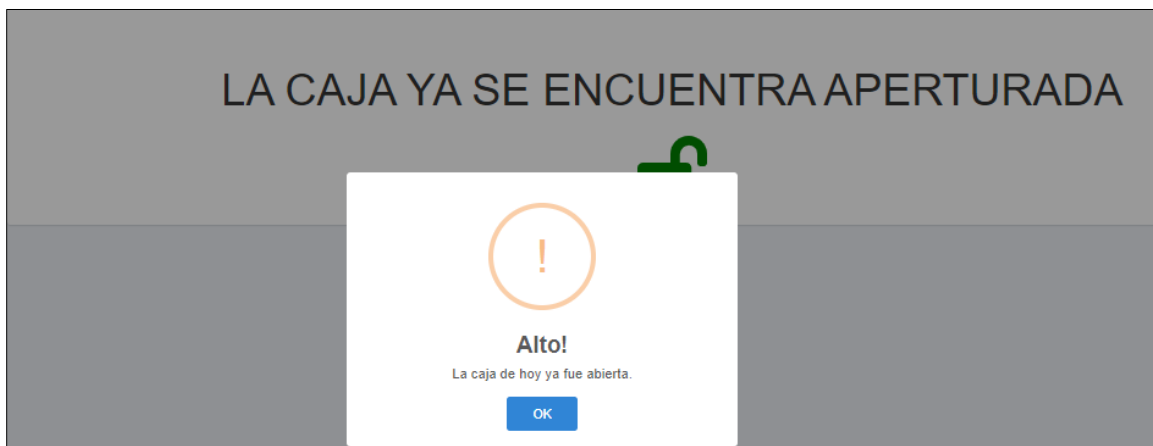
*Consolidación Cartera – Cuentas por Cobrar.*

Cuentas por Cobrar							
Show 10 entries		Search: <input type="text"/>					
Acción	Usuario	Proveedor	V. Pagar	Abonos	Fec. Limite	Observaciones	Estado
 	admin	Daniela Taipe	30.24	0.00	2021-03-12	cuentas por pagar	<span>Pendiente</span>

*Nota:* Listado y permite la impresión del reporte de cuentas por cobrar.

**Figura 63**

*Consolidación Caja - Apertura.*



*Nota:* La caja solo permite abrir una vez por día para salvaguardar la información.

**Figura 64**

*Consolidación Caja – Cierre de Caja.*

**Cierre de Caja**

Caja:  Responsable:  Monto de Apertura:

---

**Ingresos**

Efectivo:

Crédito:

**Total Ventas: \$ 29.10**

---

**Egresos**

Egresos Diarios:

**Total Egreso: \$ 0.00**

---

**Egresos Diarios de Caja** + Nuevo

Acción	Descripción	Valor	Beneficiario
<b>TOTAL A ENTREGAR:</b>			
<b>\$ 100.00</b>			

**Observaciones:**

Ingrese las observaciones necesarias

Cerrar

*Nota:* La caja solo permite abrir una vez por día para salvaguardar la información.

**Figura 65**

*Consolidación Caja – Arqueo.*

**Arqueos**

Show  entries Search:

Acción	Responsable	F. Apertura	V. Apertura	F. Cierre.	V. Cierre	T. Efectivo	T. Crédito	T. Egresos	T. Cobro	Estado
	admin	2021-02-25 10:27:48	100.00	2021-02-25 12:42:37	100.00	0.00	29.10	0.00		<span>Cerrado</span>

*Nota:* El arqueo se muestra siempre y cuando se haya cerrado la caja.

#### 4.5 Validación y verificación.

##### Introducción

En esta sección presentaremos las validaciones de los requerimientos funcionales que el cliente solicito y el sistema espera demostrar.

##### Pruebas unitarias a los requerimientos del sistema.

A continuación, mediante el sistema, se valida el funcionamiento de cada requerimiento que se espera obtener del aplicativo web.

Para ello tomaremos en cuenta cada información que se tiene del requerimiento y la interacción de cada entidad, el cual nos permite validar su funcionamiento o si existe algún error en el desarrollo.

**Tabla 23**

*Pruebas de funcionalidad Administrador*

PRUEBAS DE FUNCIONABILIDAD			
REQUISITOS FUNCIONALES	ITEMS DE VERIFICACION		OBSERVACIÓN
	SI	NO	
Permite e ingreso al interfaz.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Valida la información de tipo de usuario.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Ingreso al interfaz del sistema.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso a módulos contables.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Almacén.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Almacén Categorías, Artículo, U. Medida, Existencias.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Compras.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Ingresos, Proveedores	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso modulo Ventas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Ventas, Clientes, Guía Remisión	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso modulo Cartera.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulos Cuentas por Cobrar, Cuentas por Pagar.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Caja	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Apertura, Cierre, Caja.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Producción.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Plantas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple

PRUEBAS DE FUNCIONABILIDAD			
Acceso a módulo de Accesos tipos de usuarios	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Usuarios, Roles	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Reportes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo a Reportes Ingresos, Reportes Ventas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Administración.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo a Empresa, Sucursales, Cajas, Roles de Pagos.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple

*Nota:* Muestra los resultados de funcionalidad.

**Tabla 24**

*Pruebas de funcionalidad Vendedor*

PRUEBAS DE FUNCIONABILIDAD			
REQUISITOS FUNCIONALES	ITEMS DE VERIFICACION		OBSERVACIÓN
	SI	NO	
Permite e ingreso al interfaz.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Valida la información de tipo de usuario.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Ingreso al interfaz del sistema.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso a módulos contables.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso modulo Ventas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Ventas, Clientes, Guía Remisión	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso modulo Cartera.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulos Cuentas por Cobrar, Cuentas por Pagar.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Caja	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo Apertura, Cierre, Caja.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Acceso al módulo Reportes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple
Submódulo a Reportes Ingresos, Reportes Ventas	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Si cumple

*Nota:* Muestra los resultados de funcionalidad.



#### 4.6 Tabla de costos primarios y secundarios.

**Tabla 25**

*Costos primarios y secundarios*

<b>Ord.</b>	<b>Descripción</b>	<b>Costo</b>
<b>Costos primarios</b>		
1	Dominio y hosting	\$400.00
2	Personal	\$400,00
<b>Costos secundarios</b>		
1	Internet	\$50.00
2	Memoria USB	\$10,00
3	CD'S	\$3,00
	Costo Total	\$863,00

*Nota:* Muestra los resultados de los costos primarios y secundarios utilizados

## 4. Conclusiones y Recomendaciones

### 4.1 Conclusiones.

- Se automatizó el control y la administración de la Agrícola Moreplant, con las necesidades establecidas por parte de cliente, donde se determinaron requerimientos para el funcionamiento la ejecución y el control de la contabilidad, evitando que se siga llevando una contabilidad antigua, brindando un mejor servicio para el usuario.
- La metodología que he implementado es la una herramienta ágil XP Xtreme Programming, la cual nos da un ciclo de vida de 4 etapas en la que se debe implementar en el desarrollo del software, la misma que la metodología permite tener una documentación secuencial al desarrollo es efectiva y satisface los requerimientos de software.
- El diseño con la herramienta AdminLTE permite tener un interfaz amigable e interactivo con el usuario, es muy utilizada en para este tipo de software o por decir un punto de venta.
- La implementación del software para el control, facturación e inventarios tiene la capacidad positiva de la situación inicial al disminuir los procesos manuales de trabajo y de dar una respuesta pronta, con ello disminuye una redundancia o mitigando que la información sea la correcta.

## 4.2 Recomendaciones

- Para el desarrollo del software lo primero que se debe hacer es reconocer los requerimientos que el cliente necesita para la implementación del software, con la finalidad de evitar malos entendidos en el proceso del desarrollo.
- Tomar en cuenta los lineamientos de la Norma IEEE- 830 con esto la clasificación de los requerimientos serán claros y precisos, considerando que cualquier organización o entidad puede crearlos dependiendo a las necesidades del desarrollo.
- La ingeniería de software es de ayuda en el proceso del desarrollo, ya que por teoría y conocimiento de desarrollo nos muestra la manera de utilizar herramientas, técnicas y métodos.
- Realizar pruebas de funcionamiento de software nos ayudara ir identificando y corrigiendo errores con prontitud y evitar que al final esos errores vayan acarreado con problemas en el software.

## 5. BIBLIOGRÁFICAS

(s.f.).

Académica, L. d. (15 de Octubre de 2010). *UNIVERSIDAD ICESI*. Recuperado el 15 de 07 de 2020, de UNIVERSIDAD ICESI:

[https://www.icesi.edu.co/departamentos/tecnologias\\_informacion\\_comunicacion/proyectos/lisa/home/analisis/srs/srs#:~:text=El%20est%C3%A1ndar%20IEEE%20830%2D1998,el%20grupo%20de%20desarrollo%20para](https://www.icesi.edu.co/departamentos/tecnologias_informacion_comunicacion/proyectos/lisa/home/analisis/srs/srs#:~:text=El%20est%C3%A1ndar%20IEEE%20830%2D1998,el%20grupo%20de%20desarrollo%20para)

Alicante, U. d. (2021). *Servicio de Informática* . Obtenido de Servicio de Informática :

[https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20\(MVC\)%20es,control%20en%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz%20de,los%20mecanismos%20interacci%C3](https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html#:~:text=Modelo%20Vista%20Controlador%20(MVC)%20es,control%20en%20tres%20componentes%20distintos.&text=La%20Vista%2C%20o%20interfaz%20de,los%20mecanismos%20interacci%C3)

ARNAU, F. L. (Febrero de 2017). *UNIVERSIDAD DE MÁLAGA*. Obtenido de UNIVERSIDAD DE MÁLAGA:

<https://riuma.uma.es/xmlui/bitstream/handle/10630/15463/FabianluisplazaarnauMemoria.pdf?sequence=1&isAllowed=y>

Atom. (22 de Agosto de 2016). *Análisis y desarrollo de Sistemas de Información*.

Recuperado el 26 de julio de 2020, de Análisis y desarrollo de Sistemas de Información: <http://nata3ablog.blogspot.com/2016/08/requisitos-funcionales-y-no-funcionales.html>

Bibiana Gonzales, J. Q. (27 de Agosto de 2010). *Repositorio UNIVERSIDAD*

*TECNÓLOGICA DE PEREIRA*. Recuperado el 15 de agosto de 2020, de CIENCIAS DE LA COMPUTACIÓN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN:

<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/1333/6584038011G643.pdf?sequence=1>

Castro, M. L. (s.f.). Ingeniería en Sistemas. *Implementacion de una aplicacion web de matriculacion y cobros de pension para la unidad educativa Guayasamin*. Universidad Politecnica Salesiana sede Guayaquil, Guayaquil.

Drake, J. M. (2008). *Proceso de Desarrollo de Aplicacion Software*. Obtenido de Proceso de Desarrollo de Aplicacion Software:

[https://www.ctr.unican.es/asignaturas/MC\\_OO/Doc/OO\\_08\\_I2\\_Proceso.pdf](https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf)

Ecuatoriano, E. (s.f.). *Ejercito Ecuatoriano | Reseña Historica*. Recuperado el 18 de julio de 2020, de Ejercito Ecuatoriano | Reseña Historica:

<https://ejercitoecuadoriano.mil.ec/institucion/fftt/resena-historica>

EcuRed,colaboradores. (11 de Julio de 2011). *EcuRed*. Recuperado el 12 de enero de 2020, de EcuRed: [https://www.ecured.cu/CASE#Visual\\_Paradigm](https://www.ecured.cu/CASE#Visual_Paradigm)

Farias, M. .. (s.f.). *Modelo y Modelados de Sistemas*. Obtenido de Modelo y Modelados de Sistemas: [https://users.dcc.uchile.cl/~nbaloian/DSS-DCC/Esp/1\\_3.pdf](https://users.dcc.uchile.cl/~nbaloian/DSS-DCC/Esp/1_3.pdf)

Fernández, J. L. (19 de Diciembre de 2019). *UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO*. Obtenido de UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO:

<http://repositorio.unprg.edu.pe/bitstream/handle/UNPRG/1948/BC-TES-TMP-803.pdf?sequence=1&isAllowed=y>

Fuentes Laínez, J. R. (2015). *Desarrollo de Software Agil. Extremme Programming y Scrum 2ª Edición*. España: IT Campus Academy.

Gallego Manuel, T. (s.f.). *openaccess.uoc.edu*. Obtenido de Universitat Oberta de Catalunya:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>

- García, M. E. (Febrero de 2009). *Escuela Politécnica Nacional*. Obtenido de Escuela Politécnica Nacional: <https://bibdigital.epn.edu.ec/bitstream/15000/1250/1/CD-2057.pdf>
- Jiménez, R. E. (Diciembre de 2015). *REVISTA TECNOLÓGICA*. Obtenido de REVISTA TECNOLÓGICA: <https://core.ac.uk/download/pdf/80296686.pdf>
- Jiménez, R. E. (29 de 01 de 2016). *Escuela Especializada en Ingeniería*. Obtenido de Escuela Especializada en Ingeniería: <https://core.ac.uk/download/pdf/80296686.pdf>
- Lopez, J. S. (11 de 2013). *Red Circulos*. Obtenido de Red Circulos: <https://proyectocirculos.files.wordpress.com/2013/11/software.pdf>
- Manrique Daniel. (7 de Febrero de 2018). *Desarrollando sistemas sin limites*. Recuperado el 21 de Junio de 2020, de Nortware: [https://www.northware.mx/2018/02/07/poceso-de-diseno-ui-ux-en-desarrollo-de-software-y-apps/#:~:text=UX%20NO%20ES%20DISE%C3%91O%20GR%C3%81FICO&text=A%20diferencia%20de%20UI%20\(Interfaz,Desarrollo%20de%20Software%20y%20Apps](https://www.northware.mx/2018/02/07/poceso-de-diseno-ui-ux-en-desarrollo-de-software-y-apps/#:~:text=UX%20NO%20ES%20DISE%C3%91O%20GR%C3%81FICO&text=A%20diferencia%20de%20UI%20(Interfaz,Desarrollo%20de%20Software%20y%20Apps).
- Manuel trigas, G. (13 de Enero de 2014). *Universidad Oberta Catalunya*. Recuperado el 25 de Agosto de 2020, de OPENACCES: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- Mendez. (22 de Octubre de 2008). *Especificacion de Requisitos segun el estandar de IEEE 830*. Obtenido de Especificacion de Requisitos segun el estandar de IEEE 830: <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

Organization, R. (19 de 02 de 2018). *Libro Laravel 5 Conceptos básicos y ejemplos*.

Obtenido de Libro Laravel 5 Conceptos básicos y ejemplos: <https://www.pdf-manual.es/programacion-web/php/175-libro-laravel-5-conceptos-basicos-y-ejemplos.html>

Peñalvo, D. F. (2018). *Ciencia de la Computación e Inteligencia Artificial*. Obtenido de Ciencia de la Computación e Inteligencia Artificial:

<https://repositorio.grial.eu/bitstream/grial/1228/1/07-rep.pdf>

Rojó, S. d. (1 de Junio de 2013). ELICITACION Y ESPECIFICACION DE

REQUERIMIENTOS. *Tesis presentada para obtener el grado de Magister*. La Lata, Buenos Aires, Argentina: Repositorio.

SARA, C. A. (s.f.). *EMPRESA FICTICIA S.A* . Obtenido de EMPRESA FICTICIA S.A :

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64827/8/scuervoTFM0617memoria.pdf>

Simon Pickin, M. G. (2012). *Universidad Carlos III de Madrid*. Obtenido de Universidad

Carlos III de Madrid: [http://www.it.uc3m.es/pbasanta/SOFTCOM/new/1\\_IS.pdf](http://www.it.uc3m.es/pbasanta/SOFTCOM/new/1_IS.pdf)

Software., I. d. (23 de Mayo de 2010). *IEEE-UCA*. Recuperado el 27 de julio de 2020, de Rama estudiantil EL SALVADOR:

[https://ewh.ieee.org/sb/el\\_salvador/uca/historia.html](https://ewh.ieee.org/sb/el_salvador/uca/historia.html)

Sommerville. (2004). *Requerimientos de Software*. Obtenido de Requerimientos de

Software: <https://lsi2.ugr.es/~mvega/docis/requeintro>

Sommerville, I. (2005). *Ingeniería de Software*. Madrid: PERARSON EDUCACIÓN S.A.

Subra, J. P., & Vannieuwenhuyse, A. (2018). *Scrum Un método ágil para sus proyectos*.

Barcelona: Ediciones ENI.

Terrestre, C. d. (14 de 09 de 2005). *Comando de Educación y Doctrina Militar Terrestre*.

Recuperado el 10 de julio de 2020, de Comando de Educación y Doctrina Militar

Terrestre: <https://cedeejercito.mil.ec/resena-historica/>

The Industry-leading, d. m. (10 de Marzo de 2019). *POWERDESIGNER*. Recuperado el

16 de Agosto de 2020, de POWERDESIGNER:

[https://www.powerdesigner.biz/ES/powerdesigner/probar-powerdesigner-](https://www.powerdesigner.biz/ES/powerdesigner/probar-powerdesigner-source_adw847a.html?gclid=CjwKCAjwNf6BRAwEiwAkt6UQIS3obfpBxoeZxE0q7pkhYzJ4INApXNOs4z6P05Mv4evPqYBOaQlxhoCixYQAvD_BwE)

[source\\_adw847a.html?gclid=CjwKCAjwNf6BRAwEiwAkt6UQIS3obfpBxoeZxE0q](https://www.powerdesigner.biz/ES/powerdesigner/probar-powerdesigner-source_adw847a.html?gclid=CjwKCAjwNf6BRAwEiwAkt6UQIS3obfpBxoeZxE0q7pkhYzJ4INApXNOs4z6P05Mv4evPqYBOaQlxhoCixYQAvD_BwE)

[7pkhYzJ4INApXNOs4z6P05Mv4evPqYBOaQlxhoCixYQAvD\\_BwE](https://www.powerdesigner.biz/ES/powerdesigner/probar-powerdesigner-source_adw847a.html?gclid=CjwKCAjwNf6BRAwEiwAkt6UQIS3obfpBxoeZxE0q7pkhYzJ4INApXNOs4z6P05Mv4evPqYBOaQlxhoCixYQAvD_BwE)

*Universidad Politecnica Salesiana Sede Guayaquil*. (s.f.). Obtenido de Universidad

Politecnica Salesiana Sede Guayaquil.

Wilmer Ramiro Valle Bastidas, T. C. (2017). *Universidad Israel*. Recuperado el 5 de

Julio de 2020, de Repositorio Universisda Israel:

<http://157.100.241.244/handle/47000/1471>

YURY, Y. S. (marzo de 2018). *UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS*

*ANDES*. Obtenido de UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS

ANDES: <http://TUAEXCOMSIS017-2018.pdf.com>



# ANEXOS