

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**  
**FACULTAD DE CIENCIAS PURAS Y NATURALES**  
**CARRERA DE INFORMÁTICA**



**PROYECTO DE GRADO**  
**SISTEMA INTEGRADO WEB PARA EL CONTROL DE VENTAS,**  
**INVENTARIOS Y DISTRIBUCIÓN DE GARRAFAS DE GAS**  
**LICUADO DE PETRÓLEO (GLP)**  
**CASO: DISTRIBUIDORA DE GAS COMERCIALIZADORA**  
**“ÁNGELES”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA  
MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: CLAUDIA YUPANQUI ARUNI**  
**TUTOR METODOLÓGICO: M.SC. ALDO VALDEZ ALVARADO**  
**ASESOR: LIC. VÍCTOR POZO DIAZ**  
**LA PAZ – BOLIVIA**  
**2020**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA DE INFORMÁTICA**

**LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.**

## ***DEDICATORIA***

*A Dios por darme fortaleza y estar siempre conmigo.*

*A mis padres Juan y Josefa, por brindarme su apoyo incondicional, por nunca haberme dejado de apoyar, por su paciencia y comprensión, por sus buenos consejos y guiarme por un buen camino.*

## **AGRADECIMIENTOS**

Agradezco a Dios por guiarme en todo el camino recorrido, por darme la oportunidad de lograr un objetivo más, por cuidarme y siempre estar a mi lado.

A mis padres Juan, Josefa y a toda mi familia por todo el apoyo que me brindaron a lo largo de mi vida, por todo su apoyo y paciencia.

A mi tutor, M. Sc. Aldo Ramiro Valdez Alvarado, por la enseñanza y colaboración brindada, por sus constantes observaciones para mejorar el presente proyecto, los cuales ayudaron con la conclusión de este proyecto.

A mi asesor, Lic. Víctor Pozo Diaz, por dedicar tiempo a mi proyecto, dándole seguimiento, corrección y aporte para el desarrollo de este proyecto.

Al Lic. Ariel Carrasco Fernández representante de la Distribuidora Ángeles, por permitirme poner en práctica mis conocimientos y agradecer por toda la confianza depositada en mi persona.

A mi compañero Aldo, porque en todo momento fue un apoyo incondicional con palabras de motivación y consejos para seguir adelante.

## RESUMEN

La tecnología ha tenido un avance significativo donde va proporcionando herramientas informáticas fundamentales para la realización de aplicaciones web, estas herramientas cada vez son más indispensables para las empresas ya que tienen la necesidad de optimizar las tareas para un mejor manejo de la información y administrar correctamente la empresa.

El presente proyecto de grado pretende automatizar los procesos que la Distribuidora de Gas Comercializadora Ángeles que realiza la compra y venta de GLP (Gas Licuado de Petróleo), con el objetivo de mejorar el tiempo y esfuerzo en la realización de las tareas rutinarias en la empresa.

El sistema web almacena datos de la empresa, personal, camiones, registro de las compras, ventas realizadas y el saldo pendiente o saldo cancelado de las ventas o compras realizadas por cada chófer, también genera reportes diarios, mensuales o cuando así lo requiera el personal para un mejor manejo de la empresa.

Para el desarrollo del proyecto se utilizó la metodología ágil Scrum, que propone un modelo de proceso incremental, basado en iteraciones y revisiones continuas. En cada iteración se utilizó el lenguaje de modelado UWE (UML – Base Web Engineering), utilizando las siguientes fases: Diagrama de Casos de Uso, Modelo Conceptual, Modelo de Navegación y Modelo de Presentación.

Para evaluar y medir la calidad del proyecto se utilizó Web-Site QEM (Método de Evaluación de Calidad) y para la verificación de la seguridad del sistema se siguió el Top 10 de las vulnerabilidades del software de OWASP.

## **ABSTRACT**

The technology has had a significant advance where it is providing fundamental computer tools for the realization of web applications, these tools are increasingly essential for companies as they need to optimize tasks for better information management and correctly manage the business.

The present degree project intends to automate the processes that the Distributor of Gas Comercializadora Ángeles makes the purchase and sale of LPG (Liquefied Petroleum Gas), to improve the time and effort in performing routine tasks in the company.

The web system stores data of the company, personnel, trucks, record of purchases, sales made and the outstanding balance or canceled balance of sales or purchases made by each driver, also generates daily, monthly reports or when required by staff for better management of the company.

Agile Scrum methodology was used for the development of the project, which proposes an incremental process model, based on continuous iterations and revisions. In each iteration, the UWE modeling language (UML - Base Web Engineering) was used, which use the next phases: Use Diagram Cases, Conceptual Model, Navigation Model and Presentation Model.

To evaluate and measure the quality of the project, Web-Site QEM (Quality Evaluation Method) was used and for the verification of the security of the system, the Top 10 of the OWASP software vulnerabilities were followed.

## CONTENIDO

<b>CAPÍTULO I</b> .....	1
MARCO INTRODUCTORIO.....	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES .....	2
1.2.1 ANTECEDENTES INSTITUCIONALES.....	2
1.2.1.1 MISIÓN DE LA EMPRESA.....	3
1.2.1.2 VISIÓN DE LA EMPRESA .....	3
1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES .....	3
1.3 PLANTEAMIENTO DEL PROBLEMA.....	5
1.3.1 PROBLEMA CENTRAL.....	6
1.3.2 PROBLEMAS SECUNDARIOS .....	6
1.4 DEFINICIÓN DE OBJETIVOS.....	7
1.4.1 OBJETIVO GENERAL .....	7
1.4.2 OBJETIVOS ESPECÍFICOS .....	7
1.5 JUSTIFICACIÓN.....	7
1.5.1 JUSTIFICACIÓN ECONÓMICA.....	7
1.5.2 JUSTIFICACIÓN SOCIAL .....	8
1.5.3 JUSTIFICACIÓN TECNOLÓGICA .....	8
1.6 ALCANCES Y LÍMITES .....	9
1.6.1 ALCANCES .....	9
1.6.2 LÍMITES .....	10
1.7 APORTES .....	10
1.7.1 PRÁCTICO .....	10
1.7.2 TEÓRICO.....	11
1.8 METODOLOGÍA.....	11
<b>CAPÍTULO II</b> .....	13
MARCO TEÓRICO .....	13
2.1 INGENIERÍA DE SOFTWARE .....	13
2.1.1 EL PROCESO DE LA INGENIERÍA DE SOFTWARE.....	13
2.2 METODOLOGÍAS ÁGILES .....	14

2.3 METODOLOGÍA ÁGIL SCRUM.....	15
2.3.1 EQUIPO DE SCRUM.....	16
2.3.1.1 DUEÑO DEL PRODUCTO (PRODUCT OWNER).....	17
2.3.1.2 EQUIPO DE DESARROLLO (DEVELOPMENT TEAM) .....	17
2.3.1.3 SCRUM MASTER.....	18
2.3.2 ELEMENTOS DE SCRUM.....	19
2.3.2.1 PRODUCT BACKLOG.....	19
2.3.2.2 SPRINT BACKLOG.....	21
2.3.2.3 SPRINTS.....	22
2.3.2.4 INCREMENTO.....	22
2.3.3 EVENTOS DE SCRUM.....	22
2.3.3.1 REUNIÓN DE PLANIFICACIÓN DE SPRINT.....	23
2.3.3.2 OBJETIVO DEL SPRINT.....	23
2.3.3.3 REUNIÓN DIARIA DE SCRUM.....	24
2.3.3.4 REVISIÓN DEL SPRINT.....	25
2.3.3.5 SPRINT RETROSPECTIVA.....	26
2.3.4 FASES DE SCRUM.....	27
2.3.4.1 PRE-GAME.....	28
2.3.4.2 GAME.....	28
2.3.4.3 POST-GAME.....	29
2.4 INGENIERÍA WEB.....	29
2.4.1 PROCESO DE INGENIERÍA WEB.....	30
2.5 METODOLOGÍA UWE.....	31
2.5.1 ACTIVIDADES DE MODELADO DE UWE.....	32
2.5.1.1 ANÁLISIS DE REQUERIMIENTOS.....	32
2.5.1.2 MODELO DE CONTENIDO.....	35
2.5.1.3 MODELO NAVEGACIONAL.....	36
2.5.1.4 MODELO DE PRESENTACIÓN.....	37
2.6 INVENTARIOS.....	38
2.6.1 TÉCNICAS DE VALUACIÓN DE INVENTARIOS.....	40
2.7 COMPRA Y VENTA.....	41

2.8 HERRAMIENTAS.....	41
2.8.1 FRAMEWORKS.....	41
2.8.1.1 FRAMEWORK PARA FRONTEND.....	43
2.8.1.2 FRAMEWORK PARA BACKEND.....	43
2.8.1.3 BASE DE DATOS.....	46
<b>CAPÍTULO III</b> .....	47
MARCO APLICATIVO.....	47
3.1 INTRODUCCIÓN.....	47
3.2 PRE-GAME.....	48
3.2.1 EQUIPO SCRUM.....	48
3.2.2 RECOPIACIÓN DE LOS REQUERIMIENTOS.....	49
3.2.3 PILA DEL PRODUCTO (PRODUCT BACKLOG).....	49
3.2.4 HISTORIAS DE USUARIO.....	51
3.3 GAME.....	54
3.3.1 PLANIFICACIÓN DEL SPRINT: PRIMER SPRINT.....	54
3.3.2 DESARROLLO DEL SPRINT: PRIMER SPRINT.....	55
3.3.2.1 ANÁLISIS DE REQUERIMIENTOS.....	55
3.3.2.2 MODELO CONCEPTUAL.....	56
3.3.2.3 MODELO NAVEGACIONAL.....	56
3.3.2.4 MODELO DE PRESENTACIÓN.....	57
3.3.2.5 DISEÑO DE INTERFAZ.....	57
3.3.2.6 PRUEBAS UNITARIAS.....	58
3.3.3 PLANIFICACIÓN DEL SPRINT: SEGUNDO SPRINT.....	59
3.3.4 DESARROLLO DEL SPRINT: SEGUNDO SPRINT.....	59
3.3.4.1 ANÁLISIS DE REQUERIMIENTOS.....	60
3.3.4.2 MODELO CONCEPTUAL.....	60
3.3.4.3 MODELO NAVEGACIONAL.....	61
3.3.4.4 MODELO DE PRESENTACIÓN.....	61
3.3.4.5 DISEÑO DE INTERFAZ.....	62
3.3.4.6 PRUEBAS UNITARIAS.....	63
3.3.5 PLANIFICACIÓN DEL SPRINT: TERCER SPRINT.....	63

3.3.6 DESARROLLO DEL SPRINT: TERCER SPRINT .....	64
3.3.6.1 ANÁLISIS DE REQUERIMIENTOS .....	64
3.3.6.2 MODELO CONCEPTUAL.....	64
3.3.6.3 MODELO NAVEGACIONAL .....	65
3.3.6.4 MODELO DE PRESENTACIÓN.....	65
3.3.6.5 DISEÑO DE INTERFAZ.....	66
3.3.6.6 PRUEBAS UNITARIAS.....	67
3.3.7 PLANIFICACIÓN DEL SPRINT: CUARTO SPRINT.....	67
3.3.8 DESARROLLO DEL SPRINT: CUARTO SPRINT .....	68
3.3.8.1 ANÁLISIS DE REQUERIMIENTOS .....	68
3.3.8.2 MODELO CONCEPTUAL.....	70
3.3.8.3 MODELO NAVEGACIONAL .....	70
3.3.8.4 MODELO DE PRESENTACIÓN.....	71
3.3.8.5 DISEÑO DE INTERFAZ.....	73
3.3.8.6 PRUEBAS UNITARIAS.....	74
3.3.9 PLANIFICACIÓN DEL SPRINT: QUINTO SPRINT.....	74
3.3.10 DESARROLLO DEL SPRINT: QUINTO SPRINT .....	75
3.3.10.1 ANÁLISIS DE REQUERIMIENTOS .....	75
3.3.10.2 MODELO CONCEPTUAL.....	76
3.3.10.3 MODELO NAVEGACIONAL .....	76
3.3.10.4 MODELO DE PRESENTACIÓN.....	77
3.3.10.5 DISEÑO DE INTERFAZ.....	77
3.3.10.6 PRUEBAS UNITARIAS.....	78
3.4 POST - GAME .....	79
3.4.1 PRUEBAS DE STRESS.....	79
<b>CAPÍTULO IV</b> .....	<b>82</b>
CALIDAD Y SEGURIDAD .....	82
4.1 INTRODUCCIÓN.....	82
4.2 FACTORES DE CALIDAD DEL SOFTWARE .....	82
4.3 METODOLOGÍA WEBSITE QEM .....	82
4.3.1 CRITERIOS DE PREFERENCIA DE CALIDAD ELEMENTAL .....	83

4.4 CALIDAD DEL PROYECTO .....	84
4.4.1 USABILIDAD.....	84
4.4.2 FUNCIONALIDAD .....	87
4.4.3 CONFIABILIDAD .....	90
4.4.4 EFICIENCIA.....	91
4.4.5 RESULTADOS .....	93
4.5 SEGURIDAD .....	93
4.5.1 SEGURIDAD DEL PROYECTO .....	94
<b>CAPÍTULO V</b> .....	96
ANÁLISIS COSTO BENEFICIO .....	96
5.1 INTRODUCCIÓN.....	96
5.2 COSTO DEL SISTEMA .....	96
5.2.1 PUNTO FUNCIÓN.....	96
5.2.2 COCOMO II.....	99
5.3 COSTO DEL SOFTWARE.....	100
5.3.1 COSTO DE ELABORACIÓN DEL PROYECTO .....	101
5.3.2 COSTO DE IMPLEMENTACIÓN DEL SISTEMA.....	101
5.4 COSTO TOTAL DEL SOFTWARE.....	101
5.5 BENEFICIOS DEL SISTEMA .....	102
5.5.1 VALOR ACTUAL NETO (VAN) .....	102
5.5.2 TASA INTERNA DE RETORNO (TIR).....	103
5.6 COSTO / BENEFICIO .....	104
<b>CAPÍTULO VI</b> .....	106
CONCLUSIONES Y RECOMENDACIONES .....	106
6.1 CONCLUSIONES.....	106
6.2 RECOMENDACIONES .....	106
BIBLIOGRAFÍA .....	108

## ÍNDICE DE FIGURAS

Figura 2. 1.: Flujo de proceso Scrum .....	16
Figura 2. 2.: Roles del Scrum .....	19
Figura 2. 3.: Ejemplo de una Historia de Usuario .....	20
Figura 2. 4.: Ejemplo de Sprint Backlog .....	21
Figura 2. 5.: Diagrama del Scrum.....	23
Figura 2. 6.: Proceso de las fases de Scrum .....	27
Figura 2. 7.: Diagrama de Casos de Uso .....	33
Figura 2. 8.: Diagrama de Secuencia.....	34
Figura 2. 9.: Modelo de contenido.....	35
Figura 2. 10.: Estereotipos del Modelo Navegacional.....	36
Figura 2. 11.: Estereotipos del Modelo Navegacional.....	36
Figura 2. 12.: Estereotipos del Modelo de Presentación .....	37
Figura 2. 13.: Metamodelo para el modelo de escenarios web.....	38
Figura 3. 1.: Metodología Scrum Complementado con UWE .....	48
Figura 3. 2.: Modelo de Caso de Uso – Entrada y salida del sistema.....	55
Figura 3. 3.: Modelo de Conceptual –Entrada y salida del sistema.....	56
Figura 3. 4.: Modelo Navegacional –Entrada y salida del sistema .....	56
Figura 3. 5.: Modelo de Presentación –Entrada y salida del sistema .....	57
Figura 3. 6.: Diseño de Interfaz –Entrada y salida del sistema .....	57
Figura 3. 7.: Diseño de Interfaz – Registro de usuarios .....	58
Figura 3. 8.: Diagrama Casos de Uso - Administración del Personal .....	60
Figura 3. 9.: Modelo de Contenido - Administración del Personal .....	60
Figura 3. 10.: Modelo Navegacional - Administración del Personal .....	61
Figura 3. 11.: Modelo de Presentación - Administración del Personal .....	61
Figura 3. 12.: Diseño de Interfaz – Registro de Empleados.....	62
Figura 3. 13.: Diseño de Interfaz – Listado de Empleados.....	62
Figura 3. 14.: Diagrama de Casos de Uso – Administración de los camiones.....	64

Figura 3. 15.: Diagrama de Contenido – Administración de los camiones .....	64
Figura 3. 16.: Diagrama Navegacional – Administración de los camiones .....	65
Figura 3. 17.: Diagrama de Presentación – Administración de los camiones .....	65
Figura 3. 18.: Diseño de Interfaz – Administración de destinos .....	66
Figura 3. 19.: Diseño de Interfaz – Administración de camiones .....	66
Figura 3. 20.: Diagrama de Casos de Uso – Inventario .....	69
Figura 3. 21.: Modelo Conceptual – Inventario .....	70
Figura 3. 22.: Modelo Navegacional – Inventario.....	70
Figura 3. 23.: Modelo de Presentación – Inventario.....	71
Figura 3. 24.: Modelo de Presentación – Inventario.....	72
Figura 3. 25.: Diseño de Interfaz – Administración de Compras .....	73
Figura 3. 26.: Diseño de Interfaz – Administración de Ventas .....	73
Figura 3. 27.: Diagrama de Casos de Uso – Reportes .....	75
Figura 3. 28.: Modelo Conceptual – Reportes.....	76
Figura 3. 29.: Modelo Navegacional – Reportes .....	76
Figura 3. 30.: Modelo de Presentación– Reportes.....	77
Figura 3. 31.: Diseño de Interfaz– Reportes .....	77
Figura 3. 32.: Diseño de Interfaz– Reportes.....	78
Figura 3. 33.: Prueba de Stress – Tipo de prueba .....	80
Figura 3. 34.: Prueba de Stress – Parámetros de Simulación .....	80
Figura 3. 35.: Prueba de Stress – Resultado de la prueba.....	81
Figura 4. 1.: Taxonomía de tipos de criterio elementales.....	84
Figura 4. 2.: OWASP Top 10 - 2017 .....	94
Figura 4. 3.: Encriptado del password .....	95
Figura 5. 1.: Tabla de Resultados – Cocomo II.....	100

## ÍNDICE DE TABLAS

Tabla 3. 1.: Roles de Scrum.....	48
Tabla 3. 2.: Pila del Producto .....	50
Tabla 3. 3.: Historia de Usuario 1.....	52
Tabla 3. 4.: Historia de Usuario 2.....	52
Tabla 3. 5.: Historia de Usuario 3.....	53
Tabla 3. 6.: Historia de Usuario 4.....	53
Tabla 3. 7.: Historia de Usuario 5.....	54
Tabla 3. 8.: Sprint Backlog 1 .....	55
Tabla 3. 9.: Prueba Unitaria de login.....	58
Tabla 3. 10.: Sprint Backlog 2.....	59
Tabla 3. 11.: Prueba Unitaria de la administración del personal .....	63
Tabla 3. 12.: Sprint Backlog 3.....	63
Tabla 3. 13.: Prueba Unitaria de la administración de camiones .....	67
Tabla 3. 14.: Sprint Backlog 4.....	68
Tabla 3. 15.: Prueba Unitaria – Administración de compras y ventas .....	74
Tabla 3. 16.: Sprint Backlog 5.....	75
Tabla 3. 17.: Prueba Unitaria – Reportes.....	78
Tabla 4. 1.: Evaluación de Comprensibilidad.....	85
Tabla 4. 2.: Mecanismos de ayuda y retroalimentación en línea.....	86
Tabla 4. 3.: Aspectos de interfaces y estéticos .....	86
Tabla 4. 4.: Aspectos de interfaces y estéticos .....	87
Tabla 4. 5.: Aspectos de búsqueda y recuperación.....	88
Tabla 4. 6.: Aspectos de navegación y exploración .....	89
Tabla 4. 7.: Aspectos de navegación y exploración .....	89
Tabla 4. 8.: Aspectos de navegación y exploración .....	89
Tabla 4. 9.: Evaluación de Confiabilidad .....	90
Tabla 4. 10.: Evaluación de Confiabilidad .....	91

Tabla 4. 11.: Evaluación de Confiabilidad .....	92
Tabla 4. 12.: Evaluación de Eficiencia .....	92
Tabla 4. 13.: Calidad Global del Sistema .....	93
Tabla 5. 1: Punto Función Sin Ajuste.....	97
Tabla 5. 2: Factor de Complejidad Total .....	98
Tabla 5. 3: Conversión de Puntos de Función .....	99
Tabla 5. 4.: Costo de Elaboración del Proyecto .....	101
Tabla 5. 5.: Costo de Elaboración del Software II .....	101
Tabla 5. 6.: Interpretación del VAN .....	102
Tabla 5. 7 : Cantidad Nominal por Año .....	103
Tabla 5. 8.: Interpretación del TIR .....	104

# CAPÍTULO I

## MARCO INTRODUCTORIO

### 1.1 INTRODUCCIÓN

Actualmente la tecnología es muy importante ya que se usan sistemas para agilizar los trabajos más rápidos y eficientes gracias al invento de la tecnología como son las computadoras que se han ido modernizando, el internet, las redes inalámbricas, electrónicas, computadoras portátiles, radios, entre otros productos y servicios. Vemos que ahora debido a que la tecnología ha avanzado y se ha ido innovando se han creado programas para agilizar el trabajo del hombre, podemos ver que actualmente se tienen los trabajos en tiempo y forma, esto debido a que ya no es como antes que se tenía que arrastras el lápiz, habían errores, tener que verificar de principio a fin y si era urgente la información había la necesidad de quedarse a trabajar fuera del horario establecido en el área laboral. (García, E., 2014)

La distribuidora de gas comercializadora **Ángeles** es una empresa que realiza la distribución de Gas Licuado de Petróleo (GLP) a diferentes comunidades rurales de La Paz, con el pasar del tiempo la empresa expandió su área de comercialización, teniendo así una mayor cantidad de compra y venta de GLP.

La empresa tiene dos temporadas de ventas, una temporada baja y una temporada alta de ventas. La primera de estas, corresponde a la época de verano (octubre - abril), mientras que la temporada alta de ventas corresponden a la época de invierno (mayo - octubre), esto se debe a que en invierno hay hace más frio y se ocupan más estufas que usan gas, en verano la venta reduce drásticamente, registrándose ventas menores que en días de invierno.

Los datos de las ventas realizadas se requiere para el área contable, administración de la empresa, generar reportes con la información requerida por la Agencia Nacional de Hidrocarburos (ANH), actualmente la empresa efectúa estos procesos manualmente por lo

tanto no se tiene con exactitud las ganancias y pérdidas que tiene la empresa y para él envío de historias el encargado del área debe transcribir en una hoja de trabajo Excel lo cual genera desgaste de tiempo y esfuerzo ya que deben recurrir a los cuadernos para cualquier información que la distribuidora necesite, por todo lo mencionado se presenta este proyecto, ya que un sistema web permitiría acelerar las operaciones que la empresa requiera y tener una mejor administración.

## **1.2 ANTECEDENTES**

### **1.2.1 ANTECEDENTES INSTITUCIONALES**

La distribuidora de gas comercializadora **Ángeles** es una empresa legalmente establecida, dicha empresa queda ubicada en el km. 57 carretera La Paz – Oruro, provincia Aroma de la localidad de Calamarca, su principal función es la distribución de GLP en el área rural, específicamente en las provincias Aroma, Ingavi y Sud Yungas de La Paz. (Carrasco, A., 2019).

La distribuidora de gas comercializadora Ángeles fue fundada por el Lic. Rafael Paredes Chávez, el 20 de octubre de 1997, antes de iniciar con este emprendimiento se realizó muchos estudios de la zona de distribución y se vio que el mercado potencial estaría constituido principalmente por el público que vive en las zonas aledañas a la ciudad.

La distribuidora inicio sus actividades con dos camiones y perteneciendo a la categoría F de la clasificación que tiene la ANH, esto duro algunos años, pues su objetivo fue crecer más y ampliar sus zonas de distribución, ofreciendo un servicio de primera los 365 días del año, llevando las garrafas de GLP hasta las poblaciones más lejanas de las diferentes provincias de comercialización.

La empresa fue adquiriendo más vehículos para la distribución y traslado de GLP, personal administrativo, ayudantes, tanto fue el crecimiento que en este momento se encuentra en la categoría C en la clasificación de la ANH. Actualmente cuenta con un gerente general quien

se encarga del área administrativa y un jefe operario que se encarga del área operativa de la empresa, encargado de distribución.

### **1.2.1.1 MISIÓN DE LA EMPRESA**

Somos una empresa dedicada a la distribución y venta de Gas Licuado de Petróleo (GLP) a las familias del área rural realizando la entrega a domicilio, con respeto, eficiencia y responsabilidad, garantizando su seguridad en la manipulación del producto.

### **1.2.1.2 VISIÓN DE LA EMPRESA**

Somos una empresa líder en las provincias de Aroma, Ingavi, Sud Yungas y zonas aledañas con un producto de calidad, manteniendo los estándares internacionales en nuestro servicio de entrega a domicilio. De tal manera que somos la mejor alternativa del mercado en cuanto a este servicio se refiere.

### **1.2.2 ANTECEDENTES DE PROYECTOS SIMILARES**

Se pudo evidenciar que existen proyectos similares de la carrera de Informática de la Universidad Mayor de San Andrés, dichos proyectos fueron realizados para diferentes áreas e instituciones, por lo cual se citara las siguientes:

➤ **Título:** Sistema web de control de ventas e inventarios de insumos, caso: “La española”

**Autor:** Víctor Vladimir Quisbert Mendoza

**Año:** 2015

**Descripción:** El presente proyecto de grado titulado “Sistema web de control de ventas e inventarios de insumos” ha sido desarrollado en la empresa “La Española”. Esta es una empresa que se dedica a proveer de productos alimenticios tanto para la comunidad en general. Los problemas en esta empresa radicaban en que la gran cantidad de información mal almacenada producía una gran pérdida de tiempo al momento de dar reportes de las ventas realizadas y la actualización de productos. El

resultado de este proyecto de implementación de un sistema, será el de lograr que la empresa logre una gran evolución en el ámbito de la información, todo esto repercutirá para hacer que más empresas requieran de esta herramienta tan indispensable. Para el desarrollo del proyecto se utilizó la metodología ágil (AUP), que propone cuatro fases cada una de ellas se desarrolla basado en iteraciones, las cuales consiste en reproducir los ciclos de vida en cascada a menor escala. También se utilizó en cada una de las iteraciones la metodología UWE, que se especializa en el diseño de Aplicaciones Web.

➤ **Título:** Sistema de administración académico vía web, caso: Escuela de idiomas.

**Autor:** Francisco Carpio Palle

**Año:** 2017

**Descripción:** La escuela de idiomas del ejercito tiene como tarea principal dar una educación superior en idiomas extranjeros, con una formación profesional y adecuada para poder sobresalir y destacarse en diversos ambientes. Así este trabajo surge con el propósito de brindar un manejo eficaz y veloz de la información, para que pueda desempeñarse en diversos ámbitos, área académica (notas), económica (pensiones y gastos), inscripciones, horario e informes, cuyo objetivo fue desarrollar un sistema de administración académico donde contemple varias áreas dentro del instituto. En el desarrollo de proyecto se toma a la metodología UWE (Ingeniería Web Basado en UML) que proporciona técnicas y actividades para el modelado y construcción de las aplicaciones Web, tiene como herramienta principal al UML (Lenguaje de Modelado Unificado) para la recolección de requerimientos en el desarrollo Orientado a Objetos para un estudio personalizado a detalle.

➤ **Título:** Sistema web integrado para el registro y control de expedientes médicos, caso: NovaSalud S.R.L.

**Autor:** Ximena Stefania Cordero Maydana

**Año:** 2018

**Descripción:** El proyecto ha sido desarrollado para el centro médico NovaSalud, con el objetivo de automatizar el registro y control de los historiales clínicos de los pacientes que registra la atención a pacientes asegurados, de campaña y externos. Ofreciendo diferentes servicios de especialidad aparte de consultas y exámenes médicos. Este registro se hace de forma manual, y el intercambio de los expedientes genera en muchos aspectos pérdida de tiempo o dualidad de la información, por este motivo y por todo lo mencionado anteriormente se formula el presente proyecto de grado cuyo fin es la implementación de un sistema integrado de información que mejore el manejo de la misma y acelere la toma de decisiones médicas.

➤ **Título:** Sistema web de control de inventarios, reservas y facturación, Caso: “Cafetería Lala’s”

**Autor:** Alejandra Lecoña Rivas

**Año:** 2018

**Descripción:** El sistema almacena los datos de la empresa, sus sucursales, almacenes, al personal, los productos catalogados, a sus proveedores, para luego realizar las compras, ventas y otros movimientos de inventario, también genera la factura por las ventas con factura computarizada, se publica el menú con anticipación para que el cliente pueda realizar su reserva. Genera los reportes de Kardex físico/valorado general e individual, ventas diarias, productos de mayor demanda, productos reservados, entre otros.

### 1.3 PLANTEAMIENTO DEL PROBLEMA

El problema principal que tiene la distribuidora de gas comercializadora Ángeles es que el control de inventarios y distribución se genera a causa de que no tienen registros exactos del ingreso y salida del producto, la administración es realizada de forma manual lo cual se vuelve tedioso, esto provoca que cuando la empresa necesite cierta información tendrán que recurrir a los cuadernos, generando pérdida de tiempo y dinero.

### **1.3.1 PROBLEMA CENTRAL**

¿Cómo realizar el control de ventas, inventarios y distribución de garrafas de gas licuado de petróleo (GLP) de la distribuidora de gas comercializadora Ángeles?

### **1.3.2 PROBLEMAS SECUNDARIOS**

La distribuidora de gas comercializadora Ángeles no cuenta con un buen control de ventas, inventario, distribución. Realizando una revisión detallada de los datos que se tienen en la empresa notamos las siguientes falencias:

- ✓ Escasa información del monto total de venta realizada en el mes, lo que provoca no saber con exactitud las ganancias o pérdidas de la empresa.
- ✓ No cuenta con un buen control de GLP distribuido y GLP que aún se encuentra en la distribuidora, esto ocasiona pérdidas de venta y tiempo al tener que revisar si existen garrafas con contenido, el control manual de compras y ventas del producto también genera riesgos de pérdida de información sobre el inventario.
- ✓ Al no existir un registro de choferes y camiones que serán designados para el recojo de GLP de la Planta Yacimientos Petrolíferos Fiscales Bolivianos (YPFB) se ocasiona una pérdida de tiempo al momento de ingresar a la planta YPFB.
- ✓ Cada mes se debe realizar un reporte con el detalle de chofer asignado, cantidad asignada, lugar de destino a la ANH, este reporte actualmente se realiza en una hoja electrónica Excel, generando una pérdida de tiempo porque se tiene que copiar del cuaderno.
- ✓ No existe un buen control de saldo por cobrar a los choferes de los camiones de distribución, por lo que existe la pérdida de dinero.
- ✓ Al no existir una manera rápida de generar cualquier detalle de la distribuidora hace que la administración sea ineficiente generando de esta manera pérdidas económicas a la empresa causando mal estar en la empresa.

## **1.4 DEFINICIÓN DE OBJETIVOS**

### **1.4.1 OBJETIVO GENERAL**

Desarrollar un sistema integrado web para el control de ventas, inventarios y distribución de GLP para la distribuidora de gas comercializadora Ángeles, mejorando así la administración de la empresa.

### **1.4.2 OBJETIVOS ESPECÍFICOS**

- ✓ Realizar el registro de las ventas realizadas y generar un total de venta para que la empresa pueda revisar las ventas en cualquier momento.
- ✓ Efectuar el registro de venta realizada y garrafas de gas licuado que se encuentran en la distribuidora.
- ✓ Clasificar a los choferes, ayudantes y camiones para la respectiva designación de lugares de venta.
- ✓ Generar un reporte detallado con los datos necesarios para realizar el envío a la ANH en un tiempo óptimo.
- ✓ Registrar la cantidad de garrafas vendidas y monto total de dinero por entregar de parte del chofer y generar un reporte para el respaldo del chofer.
- ✓ Generar una orden de saldos por cobrar y saldos pendientes por parte de los choferes.
- ✓ Obtener de manera más rápida información requerida por la distribuidora mediante el sistema.

## **1.5 JUSTIFICACIÓN**

### **1.5.1 JUSTIFICACIÓN ECONÓMICA**

La empresa implementará este nuevo sistema de control de garrafas de GLP, lo que implica la automatización de todos los procesos que la distribuidora requiere, donde se reducirá los errores más comunes, tanto el llenado como en la elaboración de reportes, de esta manera se

disminuirá los gastos destinados a material de escritorio y tiempo en la preparación de estos reportes, logrando la buena administración de la distribuidora.

Además, la empresa cuenta con el hardware necesario para implementar este sistema, por lo que no se realizara un costo adicional.

### **1.5.2 JUSTIFICACIÓN SOCIAL**

El presente proyecto ayudará al personal administrativo a realizar las diferentes tareas de registro de personal, camiones, venta de GLP de manera más rápida y ordenada, la distribuidora tendrá un mejor manejo de la información, brindando datos actualizados sobre la venta y actividades diarias que se realiza en la distribuidora. La gerencia ya no realizara el trabajo de preparar los reportes para el respectivo envío a la ANH ya que estas serán generadas de acuerdo a los datos registrados e información que sea necesaria, también tendrá el beneficio de estar bien informado en el momento que desee para así tomar decisiones adecuadas para la empresa.

### **1.5.3 JUSTIFICACIÓN TECNOLÓGICA**

Actualmente la distribuidora de gas comercializadora Ángeles cuenta con un equipo de computadora y una impresora para realizar diferentes tareas, sin embargo, el uso básico que le dan al equipo de computadora es para realizar reportes en el paquete Office. Este equipo cuenta con las herramientas informáticas necesarias para implementar el sistema que se desarrollara.

El equipo cuenta con las siguientes características:

- ✓ Procesador Intel(R) Core (TM) i5
- ✓ Memoria RAM de 8 GB
- ✓ Memoria ROM de 1 TB
- ✓ Sistema operativo Windows 8.1
- ✓ Acceso a internet

## **1.6 ALCANCES Y LÍMITES**

### **1.6.1 ALCANCES**

Este proyecto pretende implementar módulos que cubran las necesidades de la empresa y estas son las siguientes:

- **Módulo de autenticación**
  - Este módulo permitirá el acceso al sistema al personal administrativo.
  - Se realizará el registro de usuarios por roles asignados.
- **Módulo de administración de empleados**
  - Registro, edición y eliminación de empleados.
  - Registro, edición y eliminación de choferes
  - Vista de todos los empleados y choferes que fueron registrados.
- **Módulo de administración de camiones**
  - Registro, modificación y eliminación de camiones.
  - Vista de los camiones con choferes que fueron designados.
- **Módulo de inventario**
  - En este módulo se podrá conocer todo el movimiento que tuvo las garrafas de GLP, ya que se permitirá el control de entrada y salida de GLP. Para poder obtener saldos, donde se podrá saber si se está trabajando bajo pérdida o ganancia.
  - De acuerdo a la venta realizada se hará el cobro al chofer designado.
- **Módulo de Reportes**
  - Generar reportes con la información necesaria de manera automatizada.
  - Reporte del movimiento de garrafas de gas licuado de petróleo.
  - Se generará un reporte detallando el monto de entrega de dinero y saldo para el chofer que está realizando la entrega de dinero.

## **1.6.2 LÍMITES**

Entre los límites que tendrá el sistema que se realizara para la distribuidora de gas comercializadora Ángeles, se encuentran los siguientes:

- No se realizará ningún asiento contable por la venta o compra del producto, ni se generará ningún estado financiero.
- No se podrá realizar la revalorización de activos fijos.
- Solo el administrador podrá hacer el registro, modificación o eliminación de cualquier dato.
- Otros usuarios solo podrán ver detalle de los registros.
- El sistema es únicamente para el personal administrativo, para que pueda realizar el registro de todos los movimientos de la empresa.
- El sistema solo podrá ser usado para el control de garrafas de gas licuado de petróleo (GLP).
- No realiza planillas de sueldos, salarios, retrasos, o cualquier beneficio que tenga el personal de la empresa.

## **1.7 APORTES**

### **1.7.1 PRÁCTICO**

El sistema para la distribuidora de gas comercializadora Ángeles ofrece el control de la venta, inventarios y distribución de GLP, también podemos obtener información detallada de las ventas realizadas ya sea diariamente o cuando la distribuidora lo necesite, de manera confiable y oportuna.

El sistema es acertado tanto para el dueño de la empresa como para el personal administrativo ya que podrán realizar un mejor control de toda la empresa, la implementación de este sistema le dará a la empresa una ventaja importante respecto a sus competidores.

## **1.7.2 TEÓRICO**

Para el progreso de este proyecto se implementará la metodología ágil Scrum ya que esta metodología permite ejecutar los resultados de manera más rápida, flexible y productiva. Por otra parte, para el análisis y modelado del sistema de la distribuidora de gas comercializadora **Ángeles** se usará el modelado UWE.

UWE es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización. UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. (Galiano, L., 2012).

## **1.8 METODOLOGÍA**

El desarrollo del siguiente proyecto se apoya en el método científico, el cual es un método de estudio sistemático de la naturaleza que incluye las técnicas de observación, mediante los mismos se cubrirá los requerimientos necesarios para llegar al cumplimiento de los objetivos trazados.

En la investigación se tomará en cuenta la investigación exploratoria y la investigación descriptiva; la exploratoria es considerada el primer acercamiento científico a un problema, esta se utiliza cuando este aún no ha sido abordado o no ha sido suficientemente estudiado y las condiciones existentes no son aún determinantes, mientras que la descriptiva se efectúa cuando se desea describir todos sus componentes principales.

El proyecto se desarrollará con la metodología ágil SCRUM, ya que se requerirá de un proceso de revisión y modificación constante. En cuanto a la herramienta de diseño se utilizará la propuesta de Ingeniería Web basada en UWE, esta es una herramienta detallada para el proceso de modelado de aplicaciones web, prestando una especial atención a la sistematización y personalización.

Este sistema proporciona diferentes modelos para el diseño del software entre los que se puede mencionar:

- Análisis de requerimientos.
- Modelo conceptual.
- Modelo navegacional.
- Modelo de escenarios web.
- Modelo de interacción temporal.

Las herramientas que se usaran para el desarrollo e implementación del sistema web son las siguientes:

- Plataforma: Windows.
- Software de control de versiones: Git en su versión estable.
- Lenguaje de programación: JavaScript, Php
- Sistema de Base de Datos: MySQL
- Editor de código: Visual Studio Code.
- Herramientas de maquetación: Lavarel, Bootstrap
- Motor de plantillas: Laravel 5.2.4

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 INGENIERÍA DE SOFTWARE

La ingeniería de software es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo. Operación y mantenimiento de software; es decir, la aplicación de la ingeniería dirigida al software. 2) El estudio de enfoques segundo el punto 1. (Pressman, R., 2010).

La ingeniería de software es una tecnología con varias capas. En el que cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad.

Los métodos de la ingeniería de software proporcionan la experiencia técnica para elaborar software. Incluyen un conjunto amplio de tareas, como comunicación, análisis de los requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo. Los métodos de la ingeniería de software se basan en un conjunto de principios fundamentales que gobiernan cada área de a tecnología e incluyen actividades de modelación y otras técnicas descriptivas. (Pressman, R., 2010).

##### 2.1.1 EL PROCESO DE LA INGENIERÍA DE SOFTWARE

El proceso de ingeniería de software es un conjunto de actividades, acciones y tareas que se ejecutan cuando se crea algún producto de trabajo.

Según (Pressman, R., 2010), una estructura de proceso general para la ingeniería de software consta de cinco actividades:

- **Comunicación:** Antes de comenzar cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente. Se busca entender los objetivos de los

participantes respecto del proyecto, y reunir los requerimientos que ayuden a definir las características y funciones del software.

- **Planeación:** Un proyecto de software es un viaje difícil y la actividad de planeación crea un mapa que guía al equipo mientras viaja. El mapa llamado plan del proyecto de software define el trabajo de ingeniería de software al describir las tareas técnicas por realizar.
- **Modelado:** Crea un bosquejo del objeto por hacer a fin de entender el panorama general como se verá arquitectónicamente, como ajustan entre si las partes constituyentes y muchas características más.
- **Construcción:** Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores de este.
- **Despliegue:** El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

## 2.2 METODOLOGÍAS ÁGILES

Las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno. (Rosselló, V., 2019).

Las metodologías ágiles mejoran la satisfacción del cliente ya que permite involucrarlo y comprometerlo en todo el proceso del proyecto y en cada etapa se informará al cliente de los progresos y logros del mismo, también permite a todos los miembros del equipo conocer el estado del proyecto en cualquier momento. Cabe recalcar que optar por la aplicación de una gestión ágil permite ahorrar tiempo y costes y se trabaja con mayor velocidad y eficiencia. Uno de los beneficios de la aplicación es que se trabaja a través de entregas parciales del producto.

Según (Citón, M., 2006), lo ágil se define como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Las metodologías ágiles varían en su forma de responder al cambio, pero en general comparten las siguientes características:

- ✓ Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- ✓ El software que funciona es más importante que la documentación exhaustiva.
- ✓ La colaboración con el cliente en lugar de la negociación de contratos.
- ✓ La respuesta al cambio en lugar de aferrarse a un plan.

### 2.3 METODOLOGÍA ÁGIL SCRUM

Esta metodología se basa en una estructura de desarrollo incremental, es decir, cualquier ciclo de desarrollo del producto se desgrana en pequeños proyectos que están divididos en distintas etapas. Para el desarrollo del proyecto se usará esta metodología, ya que es ideal debido al método iterativo e incremental.

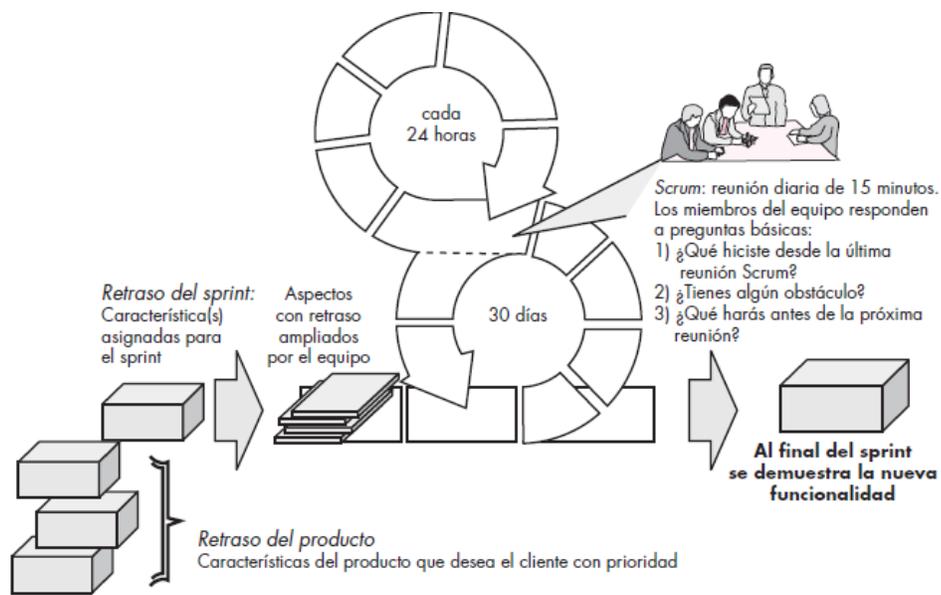
Scrum es un marco de trabajo que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar, emplea un enfoque iterativo e incremental para optimizar el control del riesgo. (Sutherland, J., Schwaber, K., 2013).

Según (Sutherland, J., Schwaber, K., 2013), tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

- **Transparencia:** Los aspectos significativos del proceso debe ser visibles para aquellos que son responsables del resultado.
- **Inspección:** Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo.

- **Adaptación:** Si uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso debe ser ajustado. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Según (Pressman, R., 2010), el uso de un conjunto de patrones de proceso del software que han demostrado ser eficaces para proyectos con plazos de entrega muy apretados, requerimientos cambiantes y negocios críticos. Cada uno de estos patrones de proceso define un grupo de acciones de desarrollo:



**Figura 2. 1.: Flujo de proceso Scrum**

Fuente: Pressman, R., 2010

### 2.3.1 EQUIPO DE SCRUM

Para desarrollar el proyecto se hizo uso de la metodología ágil SCRUM. Se utilizó esta metodología principalmente por sus modelos organizativos, también permite la construcción de proyectos de manera iterativa enfocándose principalmente en los módulos, funcionalidades y características que agregan más valor.

Según (Sutherland, J., Schewaber, K., 2013), el Equipo Scrum consiste en un Dueño de Producto (*Product owner*), el Equipo de Desarrollo (*Development Team*) y un Scrum *Master*. Los equipos Scrum son autoorganizados y multifuncionales, eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad, entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación, asegurando que siempre estará disponible una versión potencialmente útil y funcional del producto.

### **2.3.1.1 DUEÑO DEL PRODUCTO (PRODUCT OWNER)**

Según (Sutherland, J., Schewaber, K., 2013), para simplificar la comunicación y toma de decisiones es necesario que este rol recaiga en una única persona, este es el responsable de maximizar el valor del producto y trabajo del equipo de desarrollo, es la única persona responsable de gestionar la lista del producto (*Product Backlog*). La lista del producto incluye:

- ✓ Expresar claramente los elementos de la lista de producto.
- ✓ Ordenar los elementos en la lista del producto para alcanzar los objetivos y misiones de la mejor manera posible.
- ✓ Optimizar el valor del trabajo desempeñado por el equipo de desarrollo.
- ✓ Asegurar que la lista del producto es visible, transparente y clara para todos y muestra aquello en lo que el equipo trabajara a continuación.
- ✓ Asegurar que el equipo de desarrollo entiende los elementos de la lista del producto al nivel necesario.

### **2.3.1.2 EQUIPO DE DESARROLLO (DEVELOPMENT TEAM)**

Según (Sutherland, J., Schewaber, K., 2013), el equipo de desarrollo consiste en que los profesionales que desempeñan el trabajo de entregar un incremento de producto, deben potencializar la producción, al final de cada Sprint y estas tienen las siguientes características:

- ✓ Son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un incremento de producto.
- ✓ Son autoorganizados, nadie indica al equipo de desarrollo como convertir elementos de la lista del producto en incrementos de funcionalidad potencialmente desplegables.
- ✓ Scrum no reconoce sub equipos en los equipos de desarrollo.
- ✓ Scrum no reconoce títulos para los miembros de un equipo de desarrollo, todos son desarrolladores, independientemente del trabajo que realice cada persona.

Es muy recomendable para los miembros del equipo de desarrollo trabajar a tiempo completo en un proyecto, para mantenerse centrados y ágiles. La composición del equipo de desarrollo no debe cambiar a menudo y si hubiera la necesidad de hacer algunos cambios en los miembros del equipo, no deberá suceder durante un Sprint.

### **2.3.1.3 SCRUM MASTER**

Según (Sutherland, J., Schewaber, K., 2013), el Scrum *Master* es quien vela por esos valores estén presentes en todos los eventos y artefactos, también es responsable de asegurar que el equipo trabaja ajustándose a la teoría, práctica y reglas de Scrum, es un líder que está al servicio del equipo Scrum.

El Scrum *Master* ayuda al dueño de producto de las siguientes maneras:

- ✓ Entender y practicar la agilidad.
- ✓ Busca técnicas para gestionar la lista de producto de manera efectiva.
- ✓ Entender la planificación del producto en un entorno empírico.
- ✓ Asegurar que conozca como ordenar la lista de producto para maximizar el valor.

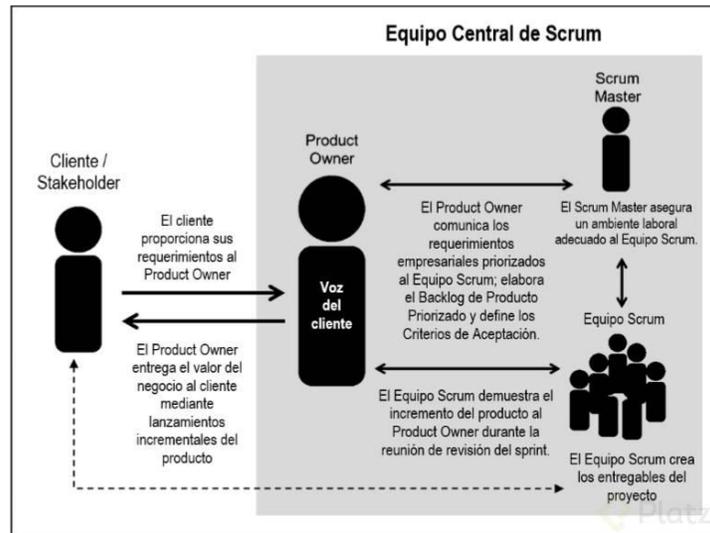
El Scrum *Master* ayuda al equipo de desarrollo de las siguientes maneras:

- ✓ Facilitar los eventos de Scrum según se requiera o necesite.
- ✓ Ayudar a crear productos de alto valor.
- ✓ Guiar al equipo de desarrollo en ser autoorganizado y multifuncional.

- ✓ Eliminar impedimentos para el progreso del equipo de desarrollo.

### 2.3.2 ELEMENTOS DE SCRUM

El proceso de *Scrum* posee una cantidad necesaria de elementos formales para llevar adelante un proyecto de desarrollo. A continuación, se describirá cada uno de ellos.



**Figura 2. 2.: Roles del Scrum**

Fuente: Platzi, 2017

#### 2.3.2.1 PRODUCT BACKLOG

Según (Trigas, M., 2012), *Product Backlog* se define como el documento en que se contemplan todas las funcionalidades o requisitos en forma de lista por prioridades, estos requisitos serán los que deba cumplir el producto o que ir adquiriendo en las diferentes iteraciones.

En la ingeniería de software tradicional, los requisitos del sistema forman parte del proceso de adquisición, siendo por tanto responsabilidad del cliente la definición del problema y de las funcionalidades que debe aportar la solución.

Las características principales de esta lista de objetivos son:

- ✓ La lista debe contener los objetivos del producto.
- ✓ La lista debe contener también los posibles riesgos e incluir las tareas necesarias para solventarlos.
- ✓ En la lista deben indicarse las posibles iteraciones que se han indicado al cliente.
- ✓ En cada objetivo se debe indicar el valor que le da el cliente y el coste estimado.

**a) HISTORIAS DE USUARIO**

Las historias de usuario son las descripciones de las funcionalidades que tendrá el software, estas se componen de tres fases denominadas “Las 3 C”

- ✓ **Card:** Breve descripción escrita a modo de recordatorio.
- ✓ **Conversation:** Conversación que sirve para asegurar que se ha entendido bien todo y así terminar el objetivo.
- ✓ **Confirmation:** Pruebas funcionales que sirven para fijar detalles que sean relevantes e indicar cuál será el límite.

Según (Scrum Manager, 2014), la estructura de una historia de usuario está formada por:

- Nombre breve y descriptivo
- Descripción de la funcionalidad en forma de diálogo o monologo del usuario describiendo la funcionalidad que desea realizar.
- Criterio de validación y verificación que determina para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Cambiar dirección de envío	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Pérez	
Descripción: Quiero cambiar la dirección de envío de un pedido.	
Validación: El cliente puede cambiar la dirección de entrega de cualquiera de los pedidos que tiene pendientes de envío.	

**Figura 2. 3.: Ejemplo de una Historia de Usuario**  
Fuente: Scrum Manager, 2014

### 2.3.2.2 SPRINT BACKLOG

Según (Quijano, J., 2012), el *Sprint Backlog* es un listado de tareas que proviene del desglose de las historias de usuario que conforman la pila de Producto (*Product Backlog*). Entendiendo como tarea, divisiones del trabajo que se puedan realizar en un plazo máximo de dos días y un mínimo de una hora; que sean atómicas en el sentido que se puedan finalizar sin necesidad de que sean dependientes de otras; que estén asignadas y así el equipo pueda saber de forma sencilla quien o quienes las están realizando.

Para cada Sprint se registra la siguiente información:

- Descripción breve.
- Persona que la tiene asignada.
- Esfuerzo pendiente para terminarla.
- Persona responsable de cada tarea, estado en el que se encuentra y tiempo de trabajo que queda para completarla.
- Debe servir de medio para registrar en cada reunión diaria del sprint, el tiempo que le queda a cada tarea.
- Facilitar la consulta y la comunicación diaria y directa del equipo.

A continuación, se mostrará un ejemplo.

SPRINT	INICIO	DURACIÓN
1	1-mar-07	12

SPRINT BACKLOG			ESFUERZO																
Tarea	Estad	Responsal	J	V	L	M	X	J	V	L	M	J	V	L	M	J	V	L	M
Descripción de la tarea 1	Terminada	Luis	16	16	16	16	16	16											
Descripción de la tarea 2	Terminada	Luis	12	8															
Descripción de la tarea 3	Terminada	Luis	4	4	4	4	4												
Descripción de la tarea 4	Terminada	Elena	8	4															
Descripción de la tarea 5	Terminada	Elena	16	16	4														
Descripción de la tarea 6	Terminada	Elena	6	6	2														
Descripción de la tarea 7	Terminada	Antonio	16	4															
Descripción de la tarea 8	Terminada	Antonio	16	16	20	12	4												
Descripción de la tarea 9	Terminada	Antonio	12	2															
Descripción de la tarea 10	En curso	Luis	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
Descripción de la tarea 11	Pendiente	Luis	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Descripción de la tarea 12	Terminada	Luis	14	14	14	14	14	14											
Descripción de la tarea 13	En curso	Antonio	8	8	8	8	8	8	6										

Figura 2. 4.: Ejemplo de Sprint Backlog

Fuente: Scrum Manager, 2016

### **2.3.2.3 SPRINTS**

Según (Sutherland, J., Schwaber, K., 2013), el núcleo de *Scrum* es el Sprint, este es el procedimiento de adaptación de las cambiantes del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Los *Sprints* contienen y consisten de la reunión de planificación del Sprint, los *scrums* diarios, el trabajo de desarrollo, la revisión del sprint y la retrospectiva del sprint.

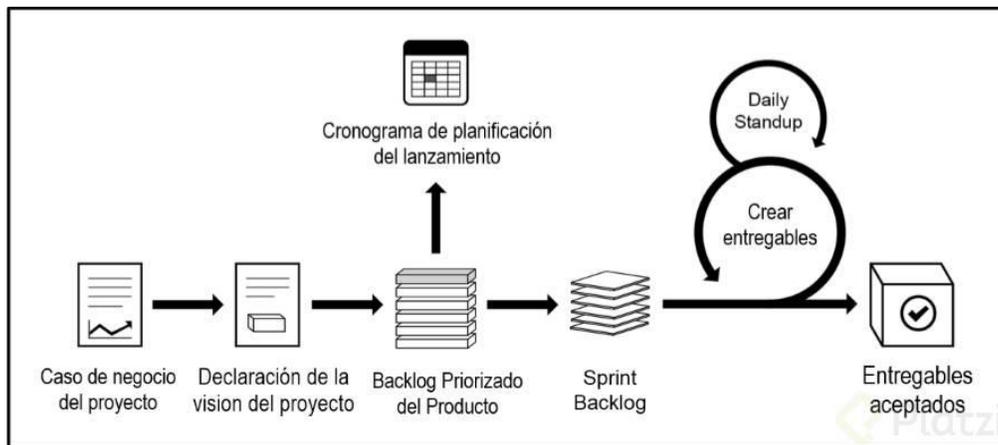
Cada Sprint tiene una definición de que se va a construir, un diseño y un plan flexible que guíara la construcción, el trabajo y resultado.

### **2.3.2.4 INCREMENTO**

Según (Sutherland, J., Schwaber, K., 2013), el incremento es la suma de todos los elementos de la lista de producto completados durante un Sprint y el valor de los incrementos de todos los Sprint anteriores. Al final de un Sprint, el nuevo incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado que cumple la definición de “Terminado” del equipo Scrum. El incremento debe estar en condiciones de utilizarse sin importar si el dueño de producto decide liberarlo o no.

### **2.3.3 EVENTOS DE SCRUM**

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (*time-boxes*), de tal modo que todos tienen una duración máxima. Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso. (Sutherland, J., Schwaber, K., 2013).



**Figura 2. 5.: Diagrama del Scrum**

Fuente: Platzi, 2017

### 2.3.3.1 REUNIÓN DE PLANIFICACIÓN DE SPRINT

El trabajo a realizar durante el Sprint se planifica en la reunión de planificación de Sprint, tiene un máximo de tiempo de ocho horas para un Sprint de un mes. Para *Sprints* más cortos, el evento es usualmente más corto. El *Scrum Master* se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El *Scrum Master* enseña al equipo Scrum a mantenerse dentro del bloque de tiempo. (Sutherland, J., Schwaber, K., 2013),

La reunión de planificación de Sprint responde a las siguientes preguntas:

- ✓ ¿Qué puede entregarse en el incremento resultante del Sprint que comienza?
- ✓ ¿Cómo se conseguirá hacer el trabajo necesario para entregar el incremento?

### 2.3.3.2 OBJETIVO DEL SPRINT

El objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la lista de producto. Proporciona una guía al equipo de desarrollo acerca de por qué está construyendo el incremento. El objetivo del Sprint ofrece al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint. Los

elementos de la lista de los productos seleccionados ofrecen una función coherente, que puede ser el objetivo del Sprint.

A medida que el equipo de desarrollo trabaja, se mantiene el objetivo del Sprint en mente. Con el fin de satisfacer el objetivo se implementa la funcionalidad y la tecnología, si el trabajo resulta ser diferente de lo que el equipo de desarrollo espera, ellos colaboran con el dueño del producto para negociar el alcance de la lista de pendientes del Sprint (*Sprint Backlog*).

### **2.3.3.3 REUNIÓN DIARIA DE SCRUM**

Según (Sutherland, J., Schwaber, K., 2013), el Scrum diario es una reunión con un bloque de tiempo de 15 minutos para que el equipo de desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.

El Scrum diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del equipo de desarrollo explica:

- ✓ ¿Qué hice ayer que ayudo al equipo de desarrollo a lograr el objetivo del Sprint?
- ✓ ¿Qué hare hoy para ayudar al equipo de desarrollo a lograr el objetivo del Sprint?
- ✓ ¿Veó algún impedimento que evite que el equipo de desarrollo o yo logremos el objetivo del Sprint?

El equipo de desarrollo usa el Scrum diario para evaluar el progreso hacia el objetivo del Sprint y para evaluar que tendencia sigue este progreso hacia la finalización del trabajo contenido en la lista del Sprint. El Scrum diario optimiza las posibilidades de que el equipo de desarrollo cumpla el objetivo del Sprint. Cada día, el equipo de desarrollo debería entender como intenta trabajar en conjunto como un equipo auto organizado para lograr el objetivo del Sprint y crear el incremento esperado hacia el final del Sprint. El equipo de desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Scrum

diario, para tener discusiones detalladas, o para adaptar, o re planificar el resto del trabajo del Sprint.

Los Scrum diarios mejoran la comunicación, eliminar la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del equipo de desarrollo. El Scrum diario constituye una reunión clave de inspección y adaptación.

#### **2.3.3.4 REVISIÓN DEL SPRINT**

Según (Sutherland, J., Schwaber, K., 2013), al final del Sprint se lleva a cabo una revisión de Sprint para inspeccionar el incremento y adaptar la lista de producto si fuese necesario, el equipo Scrum y los interesados colaboraran acerca de lo que se hizo durante el Sprint. Durante la revisión de Sprint, el equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la lista de producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento y la presentación del incremento tienen como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

La revisión de Sprint incluye los siguientes elementos:

- ✓ Los asistentes son el equipo *Scrum* y los interesados clave invitados por el dueño de producto.
- ✓ El dueño de producto explica que elementos de la lista de producto se han Terminado y cuales no se han Terminado.
- ✓ El equipo de desarrollo habla acerca de que fue bien durante el *Sprint*, que problemas aparecieron y como fueron resueltos esos problemas.
- ✓ El equipo de desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del incremento.

- ✓ El dueño de producto habla acerca de la lista de producto en el estado actual, proyecta fecha de finalización probable en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario).
- ✓ El grupo completo colabora acerca de que hacer a continuación, de modo que la revisión del Sprint proporcione información de entrada valiosa para reuniones de planificación de *Sprints* subsiguientes.

El resultado de la revisión de *Sprint* es una lista de producto revisada, que define los elementos de la lista de producto posibles para el siguiente Sprint. Es posible además que la lista de producto reciba un ajuste general para enfocarse en nuevas oportunidades.

### **2.3.3.5 SPRINT RETROSPECTIVA**

Según (Peralta, A., 2003), la retrospectiva de *Sprint* tiene lugar después de la revisión de Sprint y antes de la siguiente reunión de planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para *Sprints* de un mes. Para *Sprints* más cortos se reserva un tiempo proporcionalmente menor. El *Scrum Master* se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El *Scrum Master* participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso Scrum recae sobre él.

El propósito de la retrospectiva de Sprint es:

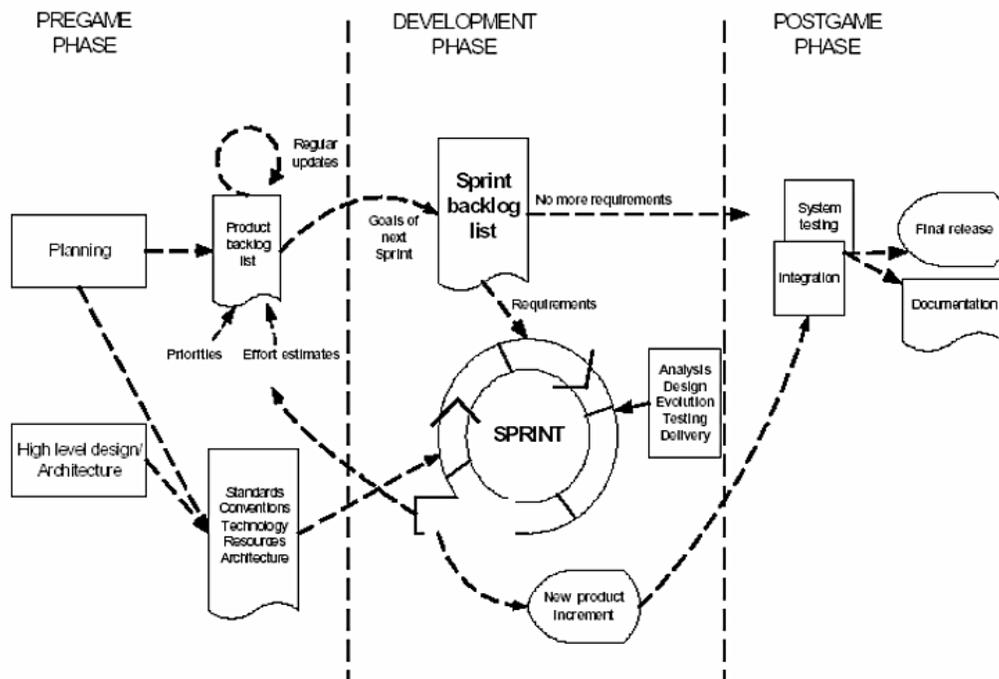
- ✓ Inspeccionar como él fue último Sprint en cuanto a personas, relaciones, procesos y herramientas.
- ✓ Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras.
- ✓ Crear un plan para implementar las mejoras a la forma en la que el equipo Scrum desempeña su trabajo.

El *Scrum Master* alienta al equipo para que mejore, dentro del marco de proceso *Scrum*, su proceso de desarrollo y su práctica para hacerlos más efectivos y amenos para el siguiente Sprint.

Para el final de la retrospectiva de *Sprint*, el equipo *Scrum* debería haber identificado mejoras que implementara en el próximo *Sprint*. El hecho de implementar estas mejoras en el siguientes *Sprint*, constituye la adaptación subsecuente a la inspección del equipo de desarrollo así mismo. Aunque las mejoras pueden implementarse en cualquier momento, la retrospectiva de *Sprint* ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

### 2.3.4 FASES DE SCRUM

Scrum consta de tres fases: Pregame, Game y Postgame.



**Figura 2. 6.: Proceso de las fases de Scrum**

Fuente: Peralta, A., 2003

#### 2.3.4.1 PRE-GAME

Según (Peralta, A., 2003), la fase de *Pre-Game* incluye dos subfases:

- ✓ **Planificación:** Consiste en la definición del sistema que será construido. Para esto se crea la lista *Product Backlog* a partir del conocimiento que actualmente se tiene del sistema. En ella se expresan los requerimientos priorizados y a partir de ella se estima el esfuerzo requerido. La lista *Product Backlog* es actualizada constantemente con ítems nuevos y más detallados, con estimaciones más precisas y cambios en la prioridad de los ítems.
- ✓ **Arquitectura:** El diseño de alto nivel del sistema se planifica a partir de los elementos existentes en la lista *Product Backlog*. En caso de que el producto a construir sea una mejora a un sistema ya existente, se identifican los cambios necesarios para implementar los elementos que aparecen en la lista *Product Backlog* y el impacto que pueden tener estos cambios. Se sostiene una *Design Review Meeting* para examinar los objetivos de la implementación y tomar decisiones a partir de la revisión.

#### 2.3.4.2 GAME

Según (Peralta, A., 2003), en esta fase se espera que ocurran cosas impredecibles. Para evitar el caos Scrum define prácticas para observar y controlar las variables técnicas y del entorno, así también como la metodología de desarrollo que hayan sido identificadas y puedan cambiar, Este control se realiza durante los *Sprints*, Dentro de variables de entorno encontramos: tiempo, calidad, requerimientos, recursos, tecnologías y herramientas de implementación.

- ✓ **Sprint:** Es un *TimeBox* de 2-3 semanas en el que realizamos una reunión diaria de Scrum.
- ✓ **Reunión:** Tres preguntas son hechas y respondidas por todos los miembros del equipo.

- ¿Qué hiciste desde la última reunión del equipo?
- ¿Qué obstáculos te encuentras?
- ¿Qué planeas lograr en la próxima reunión del equipo?

Estas reuniones diarias ayudan al equipo a descubrir posibles problemas lo antes posible y llevar a la socialización del conocimiento y, por lo tanto, a promover una estructura de equipo auto organizado.

#### **2.3.4.3 POST-GAME**

Según (Peralta, A., 2003), contiene el cierre del *release*. Para ingresar a esta fase se debe llegar a un acuerdo respecto a las variables del entorno por ejemplo que los requerimientos fueron completados.

- ✓ **Demostración:** Entrega el incremento de software al cliente para que la funcionalidad que se ha implementado pueda ser demostrada y evaluada por el propietario del producto y el interesado.
- ✓ **Cierre:** Preparación para la publicación incluida la documentación final, las pruebas preliminares y la versión.

#### **2.4 INGENIERÍA WEB**

La ingeniería web define un tipo especial de aplicaciones cliente/servidor, las cuales pueden definirse, a su vez, como una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor. Los servidores son en general potentes ordenadores dedicados a gestionar unidades de disco, mientras que los clientes son máquinas menos potentes que usan los recursos que ofrecen los servidores. (Mora, S., 2003).

La ingeniería web toma prestado muchos de los conceptos y principios básicos de la ingeniería de software, dando importancia a estas actividades. Existen diferencias delgadas en la forma en que se llevan a cabo estas actividades, pero el concepto principal es idéntico dado que dicta un enfoque disciplinado para el desarrollo de un sistema.

En el desarrollo de aplicaciones basadas en Web, se aplican herramientas de Ingeniería de Software, sin tener en cuenta que contienen características especiales, tales como usabilidad, navegabilidad, seguridad, mantenimiento y escalabilidad. (Pinzón, O., s.f.).

#### **2.4.1 PROCESO DE INGENIERÍA WEB**

Según (Pressman, R., 2003), las actividades que formarían parte del marco de trabajo incluirían las tareas abajo mencionadas. Estas tareas serían aplicables a cualquier aplicación Web, independientemente del tamaño y complejidad de la misma.

- ✓ **Comunicación con el cliente:** La comunicación con el cliente se caracteriza por medio de dos grandes tareas: el análisis del negocio y la formulación. El análisis del negocio define el contexto empresarial-organizativo para las WebApps y otras aplicaciones de negocio. La formulación es una actividad de recopilación de requisitos que involucran a todos los participantes.
- ✓ **Planeación:** Se crea el plan del proyecto para el incremento de la WebApp. El plan consiste de una definición de tareas y un calendario de plazos respecto al periodo establecido para el desarrollo del proyecto.
- ✓ **Modelado:** Las labores convencionales de análisis diseño de la ingeniería del software se adaptan al desarrollo de las WebApp, se mezclan y luego se funden en una actividad de modelado de la IWeb. El intento es desarrollar análisis rápido y modelos de diseño que definan requisitos y al mismo tiempo representen una WebApp que los satisfará.
- ✓ **Construcción:** Las herramientas y la tecnología IWeb se aplican para construir la WebApp que se ha modelado. Una vez que se construye el incremento de WebApp se dirige a una serie de pruebas rápidas para asegurar que se descubran los errores en el diseño.
- ✓ **Implementación:** Las WebApps se configura para su ambiente operativo, se entrega a los usuarios finales y luego comienza un periodo de evaluación. La retroalimentación acerca de la evaluación para realizar los procesos respectivos.

## 2.5 METODOLOGÍA UWE

UWE (*UML Web Engineering*, en español Ingeniería web basada en UML) es una metodología que permite especificar de mejor manera una aplicación Web, para el proceso de creación de aplicaciones detalla esta, con una gran cantidad de definiciones, en el proceso de diseño lista que debe utilizarse. Procede de manera iterativa e incremental, coincidiendo con UML, incluyendo flujos de trabajo y puntos de control. (Pérez, H., 2010).

UWE se especializa en la especificación de aplicaciones que se adaptan, y por eso hace énfasis especial en las características de personalización, y la definición de los modelos de usuario o en un patrón de características de navegación basado en preferencias, tareas o conocimiento. Otros aspectos de interés de la metodología UWE es la orientación a objetos, usuarios y la definición de un modelo de referencia que da soporte a la metodología y formaliza los modelos por el grado de restricciones y definiciones que proporciona.

UWE es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemática, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones web.

UWE define vistas especiales representadas gráficamente por diagramas en UML. Además, UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos.

### **2.5.1 ACTIVIDADES DE MODELADO DE UWE**

Las actividades base de modelado de UWE son:

- ✓ Análisis de requerimientos
- ✓ Modelo conceptual
- ✓ Modelo navegacional
- ✓ Modelo de presentación

#### **2.5.1.1 ANÁLISIS DE REQUERIMIENTOS**

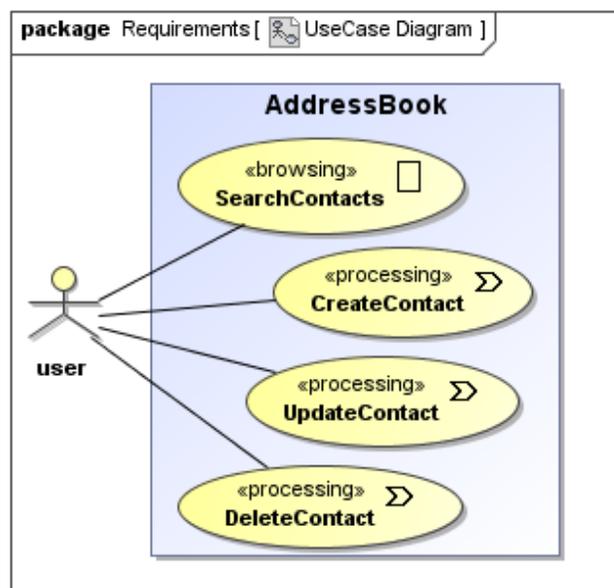
En el análisis de requerimientos, debe mencionarse un tipo de modelo denominado Diagrama de Casos de Uso.

- a) **Diagrama de Casos de Uso:** Los casos de uso se emplean para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar como se implementa ese comportamiento. Los casos de uso proporcionan un medio para que los desarrolladores, los usuarios finales del sistema y los expertos del dominio lleguen a una comprensión común del sistema.

Como afirma (Booch et al., 2006) un caso de uso es una descripción, de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor. Gráficamente, un caso de uso se representa como una elipse. Un diagrama de casos de uso es un diagrama que muestra un conjunto de casos de uso, actores y sus relaciones.

Según (Cevallos, K., 2015), se tiene lo siguiente:

- ✓ **Sistema:** El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del sistema.
- ✓ **Caso de uso:** Se representan con elipses. La etiqueta en la elipse indica la función.
- ✓ **Actor:** Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.
- ✓ **Relaciones:** Es entre un actor y un caso de uso, entre ellos se encuentran “*include*”, donde una acción es prerequisite de otra y “*extends*” es una acción que deriva de otra.

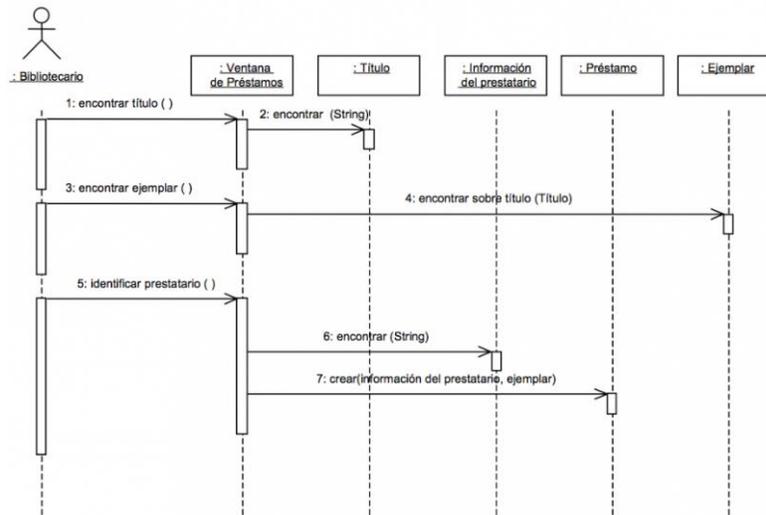


**Figura 2. 7.: Diagrama de Casos de Uso**

Fuente: LMU, s.f.

- b) **Diagramas de secuencia:** Se utilizan para modelar los aspectos dinámicos de los sistemas. Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes; un diagrama de interacción destaca la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de secuencia es un tipo de diagrama de interacción porque describe como y en que orden, un grupo de objetos que funcionan en conjunto.

A continuación, se muestra una figura de referencia.



**Figura 2. 8.: Diagrama de Secuencia**

Fuente: Cillero, M., 2009

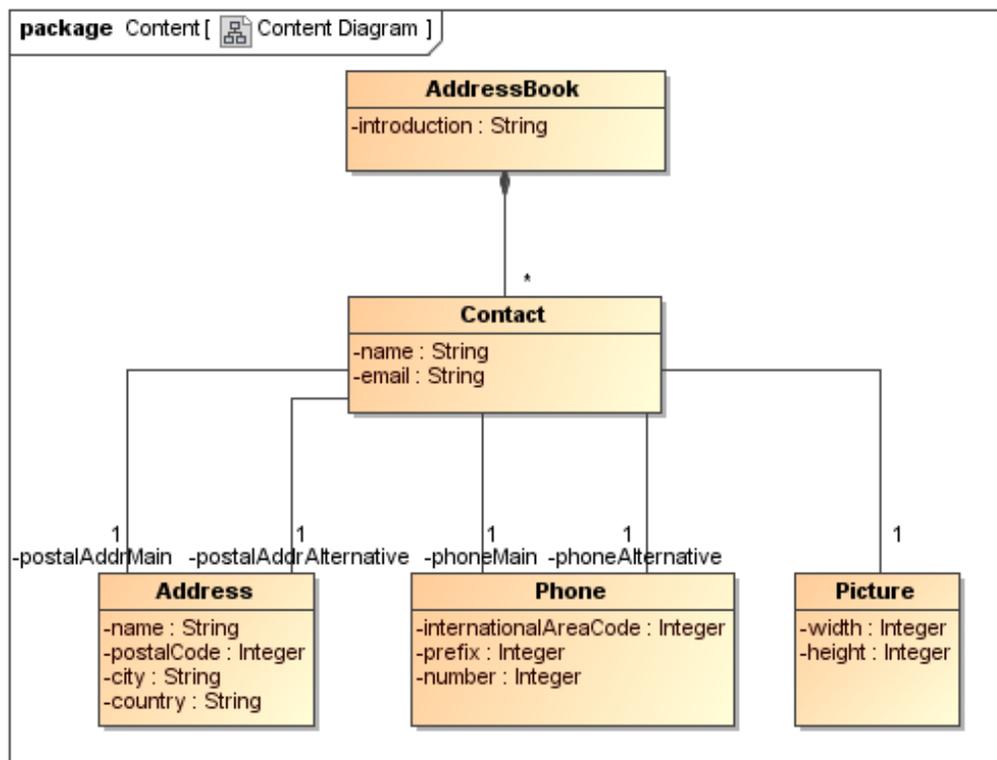
Según (Cevallos, K., 2015) los elementos principales de los diagramas de secuencia son:

- ✓ **Rol de la Clase:** el rol de la clase describe la manera en que un objeto se va a comportar en el contexto. No se listan los atributos del objeto.
- ✓ **Activación:** los cuadros de activación representan el tiempo que un objeto necesita para completar una tarea.
- ✓ **Mensajes:** los mensajes son flechas que representan comunicaciones entre objetos. Las medias flechas representan mensajes asincrónicos. Los mensajes asincrónicos son enviados desde un objeto que no va a esperar una respuesta del receptor para continuar con sus tareas.
- ✓ **Líneas de Vida:** las líneas de vida son verticales y en línea de puntos, ellas indican la presencia del objeto durante el tiempo.
- ✓ **Destrucción de objetos:** los objetos pueden ser eliminados tempranamente usando una flecha etiquetada “<>” que apunta a una X.
- ✓ **Loops:** una repetición o *loop* en un diagrama de secuencias, es representado como un rectángulo. La condición para abandonar el *loop* se coloca en la parte inferior

### 2.5.1.2 MODELO DE CONTENIDO

El diseño del modelo lógico-conceptual debe realizarse según los modelos de casos de uso que ya se hayan establecido y que se definen en la especificación de requerimientos. El modelo de contenido debe incluir aquellos objetos que estén implicados en las tareas y actividades comunes que los usuarios finales realizarán en la aplicación web, esto significa diferenciar los elementos de mayor importancia para la realización de una tarea o que resultan como el efecto de otra.

En la siguiente figura se dispone una clase agenda “*AddressBook*” conteniendo un conjunto de contactos. Cada contacto debe contener un nombre, y puede contener una dirección de correo, dos números de teléfono y dos direcciones postales.



**Figura 2. 9.: Modelo de contenido**

Fuente: LMU, s.f.

### 2.5.1.3 MODELO NAVEGACIONAL

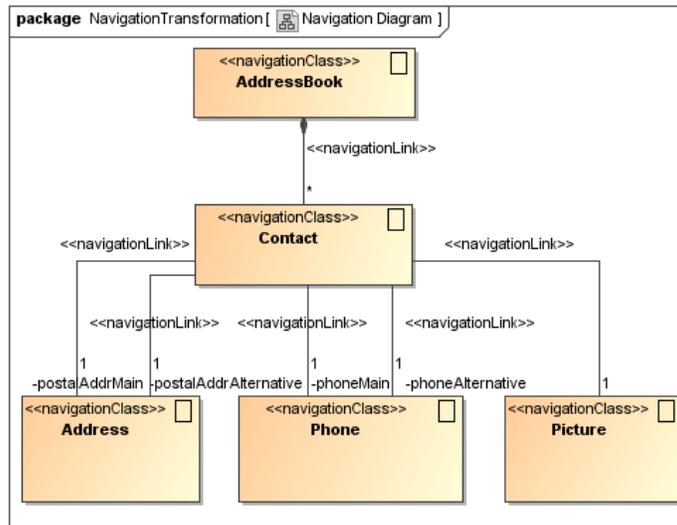
El modelo de navegación debe comprender la especificación de que objetos pueden ser visitados mediante la navegación a través de la aplicación web y todas sus relaciones. Este modelo puede destacarse como el más importante de los modelos dentro del marco UWE debido a que con este pueden representarse diferentes elementos estáticos al mismo tiempo que se pueden incorporar lineamientos semánticos de referencia para las funcionalidades dinámicas de la aplicación web. UWE provee diferentes estereotipos para el modelado de navegación, en la figura se presentan los más usados.



**Figura 2. 10.: Estereotipos del Modelo Navegacional**

Fuente: LMU, s.f.

En la siguiente figura, se muestra la aplicación de los iconos que pertenecen al diagrama navegacional.



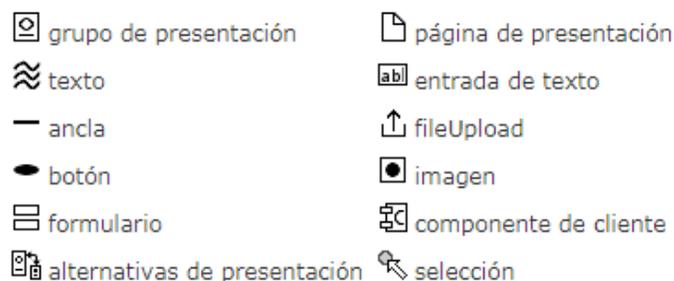
**Figura 2. 11.: Estereotipos del Modelo Navegacional**

Fuente: LMU, s.f.

### 2.5.1.4 MODELO DE PRESENTACIÓN

El modelo de presentación ofrece una visión abstracta de la interfaz de usuario de una aplicación web. Se basa en el modelo navegacional y en los aspectos concretos de la interfaz de usuario (IU). Describe la estructura básica de la IU, (por ejemplo, texto, imágenes, enlaces, formularios). Su ventaja es que es independiente de las técnicas actuales que se utilizan para implementar un sitio web, lo que permite a las partes interesadas discutir la conveniencia de la presentación antes de que realmente se aplique. (Cítlali, G, et. al., 2014).

A continuación, se presenta los estereotipos más usados para el modelo de presentación.



**Figura 2. 12.: Estereotipos del Modelo de Presentación**

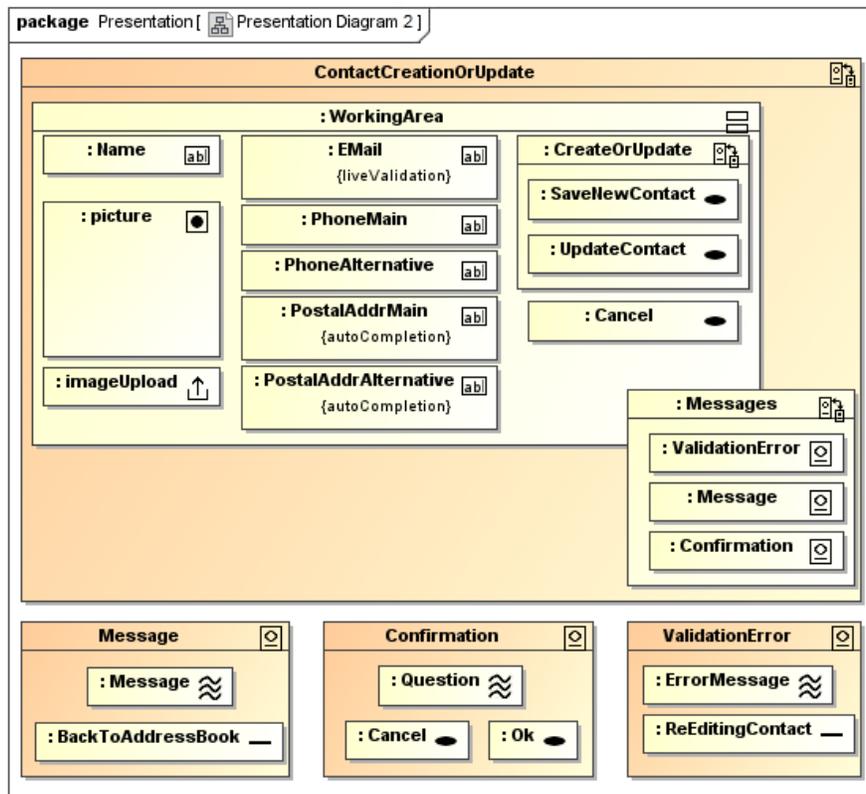
Fuente: LMU, s.f.

Según (Salas, M., 2017), el modelo de presentación define la apariencia que pueden tomar los modelos de navegación. Este modelo describe estructuras de objetos en base a clases especializadas que conforman las páginas Web. Las clases de presentación representan páginas Web que están compuestas de elementos de IU y otras clases de presentación. Por otro lado, también en estas clases pueden definirse varias propiedades físicas de apariencia que incluyen ordenamiento de los elementos del modelo y la definición de propiedades CSS (*Cascading Style Sheets*).

El modelo permite una visión amplia de los procesos de la página web que se representan en los diagramas de navegación; pueden interpretarse también con las interfaces del sistema

web. El diagrama de presentación de la metodología UWE, permite al usuario comprender y analizar, sobre el área de trabajo al que se someterá con la implantación del sistema.

En la siguiente figura, se muestra la aplicación de los iconos que pertenecen a los diagramas de presentación.



**Figura 2. 13.: Metamodelo para el modelo de escenarios web**  
Fuente: LMU, s.f.

## 2.6 INVENTARIOS

Inventario es un activo y se define como el volumen del material disponible en un almacén: insumos, producto elaborado o producto semielaborado. Cuando la demanda es mayor que el volumen disponible y los tiempos de aprovisionamiento no permiten cubrir el déficit, se considera inventario agotado; es decir es el artículo que normalmente se tiene un inventario

pero que no está disponible para satisfacer la demanda en el momento justo. Una situación inversa, sería un inventario en exceso o sobre stock. (Zapata, J., 2014).

La realización de un inventario depende de la empresa o la persona que se encargue de la actividad, así como el momento en que se lo está realizando, por lo que estos factores generan distintos tipos de inventario que son los siguientes:

- **Inventarios iniciales:** Realizados en una empresa al inicio del periodo fiscal, que suele ser el 1ro de enero de cada año.
- **Inventarios periódicos:** Realizados cada determinado tiempo (semanal, quincenal, mensual, trimestral y demás).
- **Inventarios finales:** Realizados siempre que finaliza el periodo fiscal, que suele ser el 31 de diciembre de cada año.
- **Materias primas:** Son aquellas que no han sufrido ningún cambio previo al proceso de producción y son utilizadas directamente en el mismo.
- **Materias en proceso:** Se utilizan en la elaboración del producto y su aspecto ha cambiado por resultado del proceso de producción.
- **Productos terminados:** Son los que están terminados para el almacenamiento y venta posterior.
- **Materiales y suministros:** No forman parte del producto, pero son piezas fundamentales para el buen funcionamiento de las maquinas utilizadas en el proceso productivo.

La mayoría de los inventarios se ajusta a una de estas categorías, aunque el tamaño de cada una de ellas varía considerablemente, dependiendo de las características específicas del sector o el negocio a tratarse.

Los negocios de distribución suelen dedicarse principalmente a la reventa de producto terminado, mientras que una fábrica generalmente tiene menos productos terminados y más materias primas y productos en proceso.

Aunque hayan cambiado las formas de realizar un inventario, sus características siempre serán las mismas y deberán estar presentes en todo momento, para poder lograr un inventario a la perfección. Las características que posee un inventario perfecto son las siguientes:

- c) Es ordenado porque agrupa todos los elementos que conforman el patrimonio en sus cuentas correspondientes y estas a su vez en sus masas patrimoniales.
- d) Es detallada, ya que este se especifica todas y cada una de las características de todos los bienes que conforman el patrimonio de la empresa.
- e) Es valorada, ya que cada uno de los elementos se les da un valor expresado en unidades monetarias.

### **2.6.1 TÉCNICAS DE VALUACIÓN DE INVENTARIOS**

Los métodos más importantes para valuar los inventarios son los siguientes:

- **Método PEPS o FIFO:** Este método se basa en lo primero que entra es lo primero que sale. Su apreciación se adapta más a la realidad del mercado, ya que emplea una valoración basada en costos más recientes.
- **Método UEPS o LIFO:** Este método contempla que toda aquella mercancía que ingresa de último es la primera en salir. Su ventaja se basa en que el inventario mantiene su valor estable cuando ocurre alguna alza en los precios.
- **Método del costo promedio aritmético:** El resultado lo dará la media aritmética de los precios unitarios de los artículos.
- **Método del promedio armónico o ponderado:** Este se calculará ponderando los precios con las unidades compradas, para luego dividir los importes totales entre el total de las unidades.
- **Método del costo básico:** Por medio de este método se atribuyen valores fijos a las existencias mínimas, este método es bastante parecido al LIFO con la diferencia de que se aplica solamente a la cantidad de inventario mínimo.

- **Método del costo promedio móvil o del saldo:** Calcula el valor de la mercancía, de acuerdo con las variaciones producidas por las entradas y salidas (compras o ventas) obteniéndose promedios sucesivos.
- **Método del precio de venta al detalle:** Permite la estimación de inventarios con la frecuencia que se desee. El inventario físico se practicará, basándose en los precios de venta marcados en los artículos.
- **Costo de mercado o el más bajo:** Se toma como base el precio inferior de las existencias, manteniendo el principio contabilístico del conservatismo el cual no anticipa beneficios y prevé posibles pérdidas.

## 2.7 COMPRA Y VENTA

Compra es obtener algo a cambio de dinero. La compra implica necesariamente otra operación: la venta. Estas son actividades opuestas: la persona que hace la compra entrega dinero para recibir un bien o servicio, mientras que la persona que vende entrega el bien o servicio a cambio de dinero.

Por otra parte, la palabra venta posee diferentes acepciones según el contexto en que se aplique. Venta puede referirse a un objeto o servicio que se encuentra a disposición del público, lo cual quiere decir que aún no está vendido, por lo tanto, se encuentra en venta.

## 2.8 HERRAMIENTAS

### 2.8.1 FRAMEWORKS

Un *framework* se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se considera como una aplicación genérica incompleta y configurable a la que se añade las últimas piezas para construir una aplicación concreta. (Gutiérrez, J., s.f.)

Los objetivos principales de un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente, promover buenas prácticas de desarrollo como el uso de patrones,

facilitar las cosas a la hora de desarrollar una aplicación, haciendo énfasis en el verdadero problema y olvidar de implementar funcionalidades que son de uso común como ser el registro de un usuario, establecer conexión con la base de datos, manejo de sesiones.

Según AcensTechnologies (s. f.), el uso de un *framework* a la hora de realizar un proyecto, ofrece importantes ventajas, no sólo al facilitar la tarea de la creación de la aplicación, sino otras como en el mantenimiento del código, realizar ampliaciones, uso de patrones de diseño, estructura predefinida de la aplicación, código altamente testeado, comunidad de usuarios detrás de cada *framework*, trabajo en equipo.

- a) **Estructura predefinida de la aplicación:** El programador no necesita la estructura global de la aplicación, ya que esta es proporcionada por el propio *framework*. Esto tiene la ventaja de que, pasado un tiempo, si tenemos que tocar algo en la aplicación, sabremos donde encontrar el archivo en cuestión de forma rápida.
- b) **Código altamente testeado:** Todo el código que forma parte del *framework* está altamente probado, lo que garantiza el buen funcionamiento del mismo. Nosotros podríamos desarrollar esas mismas funcionalidades, pero nunca podremos garantizar ese nivel de testeo que ofrecen los *frameworks*.
- c) **Comunidad de usuarios detrás de cada *framework*:** La gran mayoría de los *frameworks* tienen detrás a una amplia comunidad de usuarios, de los cuales muchos ayudan en su desarrollo o creando extensiones con funcionalidades extra que podremos utilizar de forma sencilla sin tener que desarrollarlas por nuestra cuenta.
- d) **Trabajo en equipo:** El uso de *frameworks* facilita el trabajo en equipo, ya que, si todos conocen el *framework* utilizado, conocerán la estructura de directorios y sabrán dónde tienen que ir para realizar una determinada acción.
- e) **Menos código y mas desarrollo:** El uso de *frameworks* ayuda a escribir menor código, lo cual significa menos tiempo gastado escribiendo.

### 2.8.1.1 FRAMEWORK PARA FRONTEND

Estas son tecnologías que corren del lado del cliente, todas aquellas tecnologías que corren del lado del navegador web, *frontend* es la parte de un programa o dispositivo a la que un usuario accede directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

HTML, CSS y *JavaScript* son los lenguajes principales del *frontend*, de los que se desprenden una cantidad de *frameworks* y librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios

Los *Frameworks* y librerías usadas para el *frontend* se describen a continuación:

- ***JavaScript*** es un lenguaje de los denominados lenguajes de scripting. Los scripts (script se traduce como guion, literalmente) son archivos de órdenes, programas por lo general simples. Es por esto que no podemos definir *JavaScript* como un lenguaje de programación en un sentido estricto, pero sin embargo sí nos permite crear páginas dinámicas, con algunos efectos realmente interesantes y que mejoren considerablemente su aspecto. Nos permite tener cierta interacción con el usuario de nuestras páginas, reconocer determinados eventos que se puedan producir y responder a éstos adecuadamente. Tratándose de un lenguaje de script, los programas que realicemos, no necesitarán ser compilado. Los lenguajes de scripting son lenguajes interpretados. Resumiendo: trabajar con *JavaScript* es sencillo y rápido, los resultados pueden ser muy satisfactorios y aunque el lenguaje tenga algunas limitaciones, permite al programador controlar lo que ocurre en la página. (Ribes, A., 2011).

### 2.8.1.2 FRAMEWORK PARA BACKEND

Según (Chapaval, N., 2018) *Backend* es la capa de acceso a datos de un software, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que

maneja dichos datos; accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas.

Backend está enfocado en hacer que todo lo que está detrás de un sitio web funcione correctamente; toma los datos, los procesa y los envía al usuario, además de encargarse de las consultas o peticiones a la base de datos, la conexión con el servidor, cuenta con una serie de lenguajes y herramientas que le ayudan a cumplir con su trabajo como *Python*, *PHP* y demás lenguajes.

### **A) PHP**

Php identifica a un lenguaje de programación que nació como Personal Home Page (PHP), PHP suele procesarse directamente en el servidor, aunque también puede usarse a través de software 27 capas de ejecutar comandos y para el desarrollo de otra clase de programas, sin embargo, en la actualidad esa vinculado a PHP Hypertext Pre-Processor. Es un lenguaje de programación interpretado de tipado dinámico cuya filosofía hace hincapié en una sintaxis que favorezca un código legible, orientado a objetos, que también puede utilizarse para el desarrollo web.

Es un lenguaje de programación poderosa y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos.

Las características de Php se resumen a continuación:

- ✓ Gran extensión de documentación: Php tiene la ventaja de ser usado por millones de usuarios en muchos proyectos.
- ✓ Es multiplataforma, lo cual es ventajoso para hacer ejecutable su código fuente entre varios sistemas operativos.
- ✓ Es un lenguaje de programación multiparadigma, el cual soporta varios paradigmas de programación como orientación a objetos, estructurada, programación imperativa y, en menor medida, programación funcional.

- ✓ Es libre y de código abierto, lo que implica que existe una gran comunidad que lo respalda, además, de que existen una gran variedad de librerías que facilitan su uso.
- ✓ Se puede separar la estructura: Esto quiere decir que se puede tener de manera independiente el código que se encarga de mover los datos del que se enlaza a la interfaz.

## **B) LARAVEL**

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web.

Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el “código espagueti”, tiene una gran influencia de *frameworks* como *Ruby on Rails*, Sinatra y ASP.NET.

Laravel tiene como objetivo ser un *framework* que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Intenta aprovechar lo mejor de otros *frameworks* y aprovechar las características de las últimas versiones de PHP.

Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

Según (Sierra, K., 2018) las características más notables de *Laravel* son:

- **Blade:** Blade es un sistema de plantillas para crear vistas en Laravel. Este permite extender plantillas creadas y secciones en otras vistas en las cuales también tendremos accesibles las variables y con posibilidad de utilizar código PHP en ellas, además, ligado al uso de Bootstrap u otro framework HTML genera resultados optimizados a los diferentes dispositivos móviles, tablets, PC's.
- **Eloquent:** Es el ORM que incluye Laravel para manejar de una forma fácil y sencilla los procesos correspondientes al manejo de bases de datos de nuestro proyecto. Transforma las consultas SQL a un sistema MVC lo que permite procesar consultas SQL directamente y así protegernos de la inyección SQL.

- **Routing:** Laravel proporciona un sistema de organización y gestión de rutas que nos permite controlar de manera exhaustiva las rutas de nuestro sistema.
- **Middlewares:** Son una especie de controladores que se ejecutan antes y después de una petición al servidor, lo que nos permite insertar múltiples controles, validaciones o procesos en estos puntos del flujo de la aplicación.
- **Escalable:** Algunos de los sitios más concurridos del planeta utilizan la capacidad de *Laravel* para escalar de manera rápida y flexible para satisfacer las demandas de tráfico pesadas.

### 2.8.1.3 BASE DE DATOS

- **MySQL:** Es un sistema de gestión de base de datos relacional, es un sistema de administración y almacenamiento de base de datos que sirve para almacenar, agregar y procesar información de manera inmediata.

Esto agrega velocidad y flexibilidad, las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos.  
(Peña, 2014)

## CAPÍTULO III

### MARCO APLICATIVO

#### 3.1 INTRODUCCIÓN

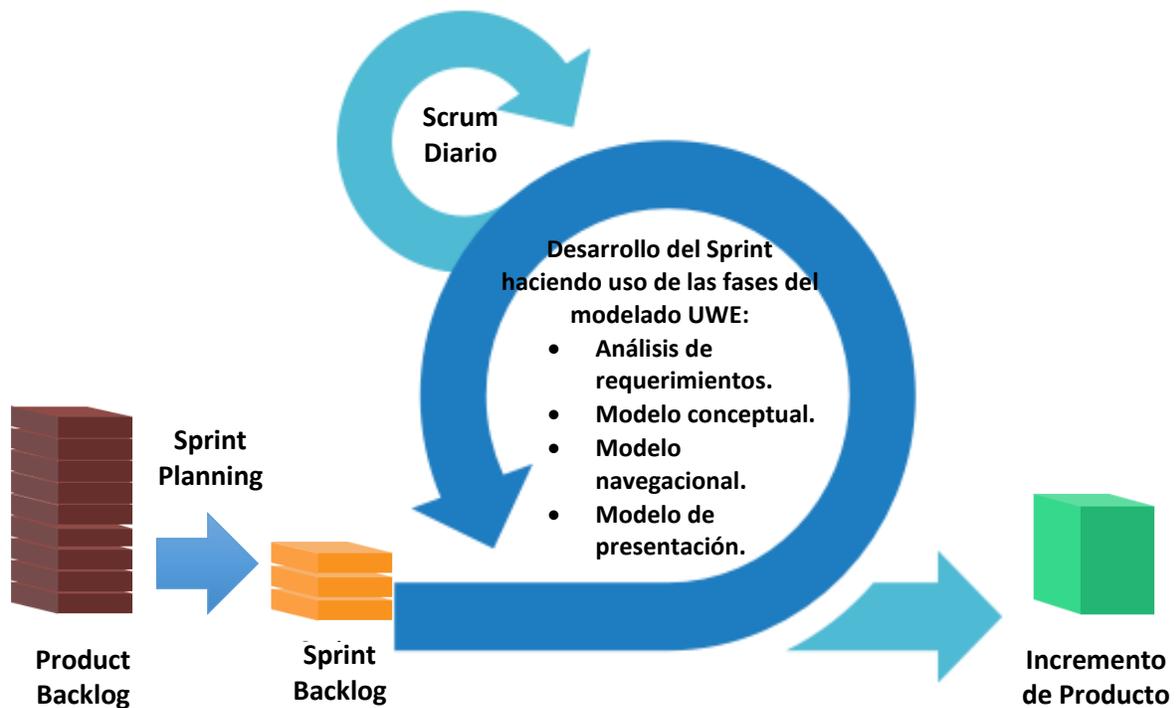
En este capítulo se abordará todo el ciclo de vida de la aplicación web de control de ventas, compras e inventarios de la distribuidora de gas comercializadora Ángeles, para lo cual se utilizará la metodología ágil Scrum y la metodología de ingeniería web basada en UML (UWE), estas ya fueron descritas en el capítulo anterior, dichas herramientas sirven para la ejecución de cada uno de los *Sprints* del proyecto.

En relación de los *Sprints*, estos se definieron uno por cada módulo del sistema por lo que se consideraron seis *Sprints* para el desarrollo total del mismo. Se propone una metodología Scrum porque es una metodología ágil y esta puede adaptarse al cambio, también existe comunicación directa entre el cliente y los desarrolladores, ya que se mantienen reuniones frecuentes, esto quiere decir que el cliente obtiene productos funcionales en corto tiempo.

En cada una de las reuniones para la definición del *Product Backlog*, se pudo observar que, un sistema de control de ventas, compras e inventarios era una necesidad para la distribuidora Ángeles debido al mal manejo de los registros físicos.

En la siguiente figura se muestra la manera en que interactúan dichas metodologías, Scrum se encarga del desarrollo del proyecto mediante las iteraciones y en cada una de estas se hace uso de las fases de la metodología UWE, estas fases son las siguientes:

- El análisis de requerimientos
- El modelo conceptual
- El modelo navegacional
- El modelo de presentación



**Figura 3. 1.: Metodología Scrum Complementado con UWE**

Fuente: Elaboración Propia

## 3.2 PRE-GAME

### 3.2.1 EQUIPO SCRUM

La siguiente tabla describe el equipo Scrum para el desarrollo del proyecto.

ROL	NOMBRE
<b>Dueño del Producto (Product Owner)</b>	Distribuidora de Gas Comercializadora Ángeles
	Lic. Ariel Adalid Carrasco Fernández
<b>Responsable del Scrum (Scrum Master)</b>	M. Sc. Aldo Ramiro Valdez Alvarado
	Lic. Víctor Pozo Diaz
<b>Equipo de desarrollo (Development Team)</b>	Claudia Yupanqui Aruni

**Tabla 3. 1.: Roles de Scrum**

Fuente: Elaboración Propia

### **3.2.2 RECOPIACIÓN DE LOS REQUERIMIENTOS**

En esta fase de requerimientos se tuvo entrevistas con las áreas involucradas para tener los requerimientos y características que debería tener el sistema, en estas entrevistas se les pregunto qué es lo que esperan del sistema además de preguntar que procesos necesitan para que puedan llegar a ser automatizados.

La distribuidora de gas comercializadora Ángeles es una empresa cuya actividad principal es la distribución, venta y compra de GLP, los datos de las ventas realizadas se requiere para el área contable, administración de la empresa, generar reportes con la información requerida por la ANH, actualmente la empresa efectúa estos procesos manualmente por lo tanto no se tiene con exactitud las ganancias y pérdidas que tiene la empresa y para él envió de historias el encargado del área debe transcribir en una hoja de trabajo Excel lo cual genera desgaste de tiempo y esfuerzo ya que deben recurrir a los cuadernos para cualquier información que la distribuidora necesite, por todo lo mencionado se presenta este proyecto, ya que un sistema web permitiría acelerar las operaciones que la empresa requiera y tener una mejor administración.

Una vez comprendida las dificultades que tiene la distribuidora Ángeles, se establecen sus requerimientos.

Controlar los inventarios mediante la automatización de las compras y ventas del producto, esto para determinar si se está trabajando bajo pérdida o ganancia y así generar reportes que serían enviados a la ANH. Realizar el registro computarizado para poder llevar un control adecuado de las ventas, conocer los lugares de venta de mayor demanda.

### **3.2.3 PILA DEL PRODUCTO (PRODUCT BACKLOG)**

La siguiente tabla especifica la lista completa de tareas que debían ser alcanzadas con la realización del sistema, esta tabla cuenta con número de lista, descripción de las tareas a ser realizada, prioridad dentro del sistema, numero de Sprint y el estado de la tarea.

ID	TAREA	PRIORIDAD	MODULO	ESTADO
1	Planificación del proyecto	Alta		Terminado
2	Creación de la base de datos para el proyecto	Alta		Terminado
3	Autenticación de usuarios	Alta	Módulo de autenticación	Terminado
4	Registro de usuarios al sistema	Alta	Módulo de autenticación	Terminado
5	Administrar información personal del usuario	Media	Módulo de autenticación	Terminado
6	Administrar datos de la empresa	Media	Módulo de administración	Terminado
7	Administrar datos de los empleados.	Alta	Módulo de administración	Terminado
8	Administrar datos de los choferes.	Alta	Módulo de administración	Terminado
9	Administrar datos de los camiones.	Alta	Módulo de administración	Terminado
10	Visualización de camiones, empleados y choferes.	Alta	Módulo de administración	Terminado
11	Registro de compras y ventas	Alta	Módulo de inventario	Terminado
12	Administrar datos de compra	Alta	Módulo de inventario	Terminado
13	Administrar datos de venta	Alta	Módulo de inventario	Terminado
14	Generar reportes de ventas diarias, semanales, mensuales y anuales	Alta	Módulo de reportes	Terminado
15	Generar orden de pago de saldos pendientes o pagados por los choferes de las ventas o compras realizadas	Alta	Módulo de reportes	Terminado

**Tabla 3. 2.: Pila del Producto**

Fuente: Elaboración Propia

### 3.2.4 HISTORIAS DE USUARIO

Según Scrum Manager (2014), la descripción de una funcionalidad que debe incorporar un sistema de software y cuya implementación aporta valor al cliente.

La estructura de una historia de usuario está formada por:

- ✓ Nombre breve y descriptivo
- ✓ Descripción de la funcionalidad en forma de dialogo o monologo del usuario describiendo la funcionalidad que desea realizar.
- ✓ Criterio de validación y verificación que determinara para considerar terminando y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Considerando los objetivos del *Product Backlog*, se pudo construir un conjunto de historias de usuario, las cuales contaron con número de historia, nombre de historia, descripción, prioridad en el negocio, usuario, programador responsable, descripción, validación y Sprint asignado.

- ✓ **Numero de historia.** Numeración incremental asignada a cada historia.
- ✓ **Nombre de historia.** Propio de cada historia de usuario y diferente entre las demás historias.
- ✓ **Usuario.** Persona que se relacionara con la parte del sistema correspondiente a la historia.
- ✓ **Prioridad en el negocio.** Describe la importancia que cada una de las historias de usuario posee, existen tres grados de prioridad para el desarrollo e implementación de las historias que son: alta, media, baja.
- ✓ **Programador responsable.** Encargado de alcanzar el objetivo de dicha historia.
- ✓ **Sprint asignado.** Numero de Sprint en el que se alcanzaran los objetivos de dicha historia, existen cinco posibilidades de asignación que son: 1, 2, 3, 4 y 5.
- ✓ **Descripción.** Objetivo que el usuario desea alcanzar.

- ✓ **Validación.** También conocido como comentario de lo que debe alcanzarse con la historia de usuario.

Cada una de estos puntos serán desglosados en las siguientes tablas.

### H1 Historia de usuario Login.

<b>HISTORIA DE USUARIO</b>			
<b>Numero</b>	1	<b>Usuario</b>	Administrador
<b>Nombre</b>		Login	
<b>Prioridad en el negocio: Alta</b>		<b>Iteración</b>	1
<b>Programador responsable</b>		Claudia Yupanqui Aruni	
<b>Descripción</b>		Permite en ingreso al sistema	
<b>Validación</b>		Permite que el usuario registrado pueda ingresar al sistema, realizar registros de compra, venta, dependiendo del rol asignado al usuario.	

**Tabla 3. 3.: Historia de Usuario 1**

Fuente: Elaboración Propia

### H2 Administración del personal en el sistema.

<b>HISTORIA DE USUARIO</b>			
<b>Numero</b>	2	<b>Usuario</b>	Administrador
<b>Nombre</b>		Administración del personal en el sistema	
<b>Prioridad en el negocio: Alta</b>		<b>Iteración</b>	2
<b>Programador responsable</b>		Claudia Yupanqui Aruni	
<b>Descripción</b>		Registro del personal en el sistema	
<b>Validación</b>		Permite el registro y la modificación de datos de información personal de los empleados por el administrador del sistema.	

**Tabla 3. 4.: Historia de Usuario 2**

Fuente: Elaboración Propia

### H3 Administración de camiones en el sistema.

<b>HISTORIA DE USUARIO</b>			
<b>Numero</b>	3	<b>Usuario</b>	Administrador
<b>Nombre</b>	Administración de camiones en el sistema		
<b>Prioridad en el negocio: Alta</b>	<b>Iteración</b>		3
<b>Programador responsable</b>	Claudia Yupanqui Aruni		
<b>Descripción</b>	Registro y modificación de camiones en el sistema		
<b>Validación</b>	Permite el registro y modificación de datos de información de los camiones por el administrador del sistema.		

**Tabla 3. 5.: Historia de Usuario 3**

Fuente: Elaboración Propia

### H4 Gestión de ventas y compras.

<b>HISTORIA DE USUARIO</b>			
<b>Numero</b>	4	<b>Usuario</b>	Administrador, despachador
<b>Nombre</b>	Gestión de compras y venta realizadas		
<b>Prioridad en el negocio: Alta</b>	<b>Iteración</b>		4
<b>Programador responsable</b>	Claudia Yupanqui Aruni		
<b>Descripción</b>	Gestión de compras y ventas realizadas en el sistema.		
<b>Validación</b>	Permite gestionar las compras y ventas realizadas por la empresa.		

**Tabla 3. 6.: Historia de Usuario 4**

Fuente: Elaboración Propia

## H5 Generar reportes.

HISTORIA DE USUARIO			
<b>Numero</b>	5	<b>Usuario</b>	Administrador, despachador
<b>Nombre</b>	Generar reportes		
<b>Prioridad en el negocio:</b> Media	<b>Iteración</b>	5	
<b>Programador responsable</b>	Claudia Yupanqui Aruni		
<b>Descripción</b>	Realizar diferentes acciones dentro de los módulos.		
<b>Validación</b>	Permite generar reportes de los movimientos de la empresa.		

**Tabla 3. 7.: Historia de Usuario 5**

Fuente: Elaboración Propia

### 3.3 GAME

En esta etapa se desarrolla todos los requerimientos de nuestra pila de productos, para la planificación se utilizó *Sprint Backlog*, y para el análisis, diseño, implementación y pruebas que se hizo el uso de UWE.

#### 3.3.1 PLANIFICACIÓN DEL SPRINT: PRIMER SPRINT

Debido a que el proyecto se dividió en cinco módulos, lo que se realizó en esta parte del proceso fue la selección de la lista de actividades del *Sprint Backlog*, las actividades fueron extraídas de la lista *Product Backlog* y desarrolladas durante el tiempo establecida para el desarrollo del Sprint.

ID	TAREA	PRIORIDAD	ESTADO
1	Autenticación de usuarios en el sistema	Alta	Completado
2	Registro de usuarios del sistema	Alta	Completado

3	Edición de la información personal del usuario	Alta	Completado
4	Asignación de roles por usuario	Alta	Completado

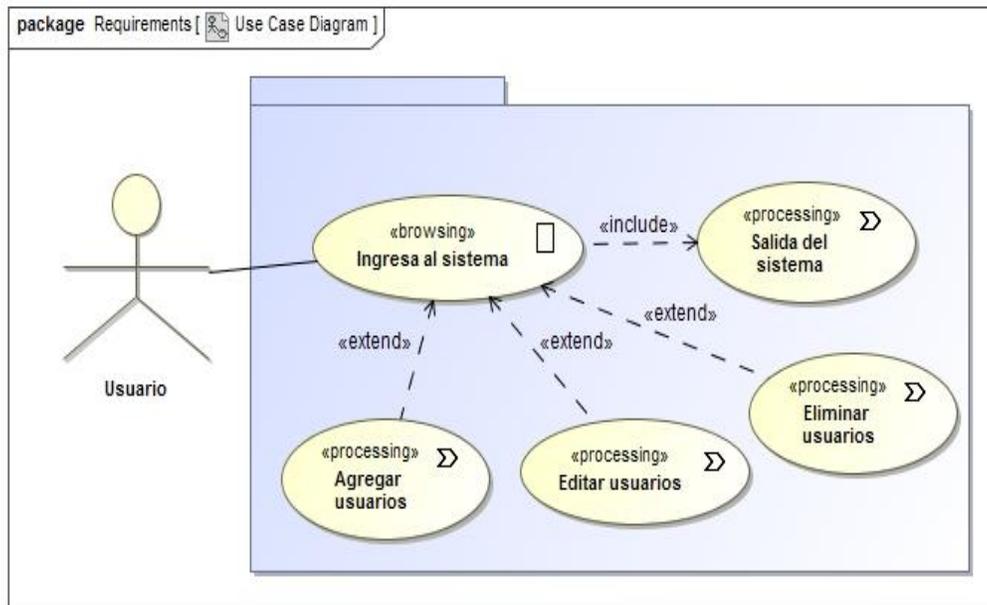
**Tabla 3. 8.: Sprint Backlog 1**  
Fuente: Elaboración Propia

### 3.3.2 DESARROLLO DEL SPRINT: PRIMER SPRINT

Para el desarrollo del primer Sprint se diseñaron los modelos correspondientes a la metodología UWE, estos modelos fueron enfocados a responder las necesidades y alcanzar los objetivos de la lista del Sprint.

#### 3.3.2.1 ANÁLISIS DE REQUERIMIENTOS

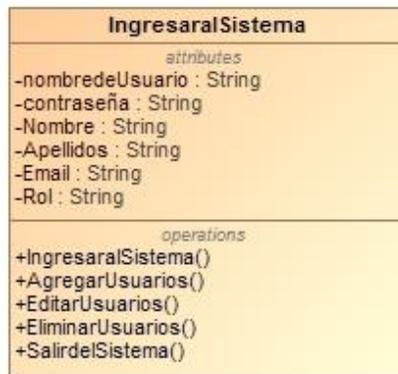
En la figura se presenta el modelo de caso de uso genérico que se diseñó para el primer Sprint.



**Figura 3. 2.: Modelo de Caso de Uso – Entrada y salida del sistema**  
Fuente: Elaboración Propia

### 3.3.2.2 MODELO CONCEPTUAL

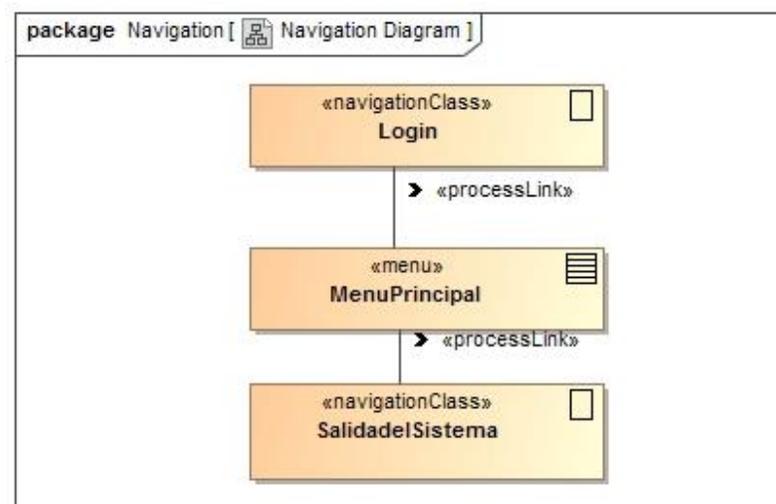
El modelo conceptual se realizó considerando que el primer Sprint es el elemento más importante y este es el modelo de entrada y salida del sistema.



**Figura 3. 3.: Modelo de Conceptual –Entrada y salida del sistema**  
Fuente: Elaboración Propia

### 3.3.2.3 MODELO NAVEGACIONAL

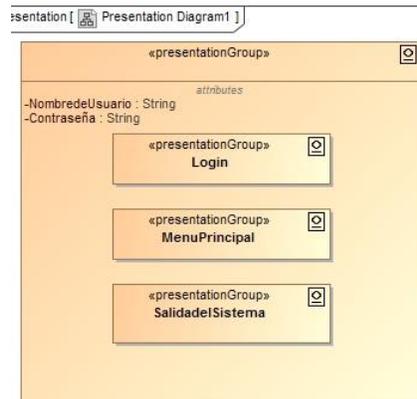
A continuación, se muestra el modelo navegacional del login del sistema.



**Figura 3. 4.: Modelo Navegacional –Entrada y salida del sistema**  
Fuente: Elaboración Propia

### 3.3.2.4 MODELO DE PRESENTACIÓN

El modelo de presentación que se diseñó para el primer Sprint muestra los procesos principales de entrada y salida del sistema.



**Figura 3. 5.: Modelo de Presentación –Entrada y salida del sistema**  
Fuente: Elaboración Propia

### 3.3.2.5 DISEÑO DE INTERFAZ

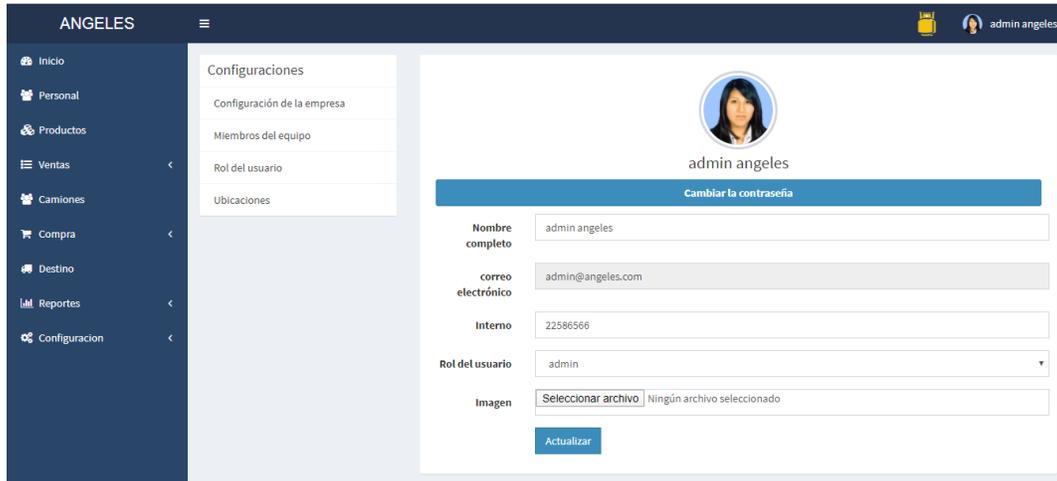
El usuario final interactúa con el sistema web y para realizar una inspección del sistema se realizaron varias capturas de pantalla para observar el sistema final.

A continuación, se muestra el diseño de interfaz del ingreso al sistema.



**Figura 3. 6.: Diseño de Interfaz –Entrada y salida del sistema**  
Fuente: Elaboración Propia

La siguiente figura muestra el registro de usuarios, diferenciando según el rol, esto para dar los permisos correspondientes a cada usuario.



**Figura 3. 7.: Diseño de Interfaz – Registro de usuarios**  
Fuente: Elaboración Propia

### 3.3.2.6 PRUEBAS UNITARIAS

Las pruebas nos ayudan a corregir cualquier error inesperado, para que el sistema se ejecute correctamente al momento de ingresar los datos o realizar cualquier tipo de consulta.

PRUEBA UNITARIA:	LOGIN
<b>Descripción:</b>	Permite el ingreso y salida del sistema.
<b>Objetivos:</b>	- Ingresar el nombre de usuario y contraseña. - Verifica si el usuario existe en el sistema. - Ingresa al sistema de administración.
<b>Condiciones:</b>	Tener acceso a internet desde cualquier dispositivo.
<b>Resultado esperado:</b>	Que el usuario registrado ingrese al sistema.
<b>Resultado obtenido:</b>	El usuario puede ingresar al sistema, realizar el registro de nuevos usuarios, estos son diferenciados por rol, editar usuarios y eliminar usuarios.

**Tabla 3. 9.: Prueba Unitaria de login**  
Fuente: Elaboración Propia

### 3.3.3 PLANIFICACIÓN DEL SPRINT: SEGUNDO SPRINT

En este sprint se planifico el desarrollo del módulo de registro y gestión del personal, el cual se encargará de todo lo referente a los datos del personal, como ser, tipo de empleado, dirección y datos personales básicos, estas tareas fueron sustraídas de la lista de deseos principal, en cada requerimiento se realiza el análisis, diseño, implementación y pruebas utilizando las fases de la metodología UWE.

La siguiente tabla, muestra el Sprint Backlog con las tareas a realiza en el segundo Sprint.

ID	TAREA	PRIORIDAD	ESTADO
1	Registro y modificación del personal	Alta	Completado
2	Administración del personal	Alta	Completado
3	Registro y edición de los choferes	Alta	Completado
4	Administración de los choferes	Alta	Completado
5	Listar al personal	Media	Completado

**Tabla 3. 10.: Sprint Backlog 2**

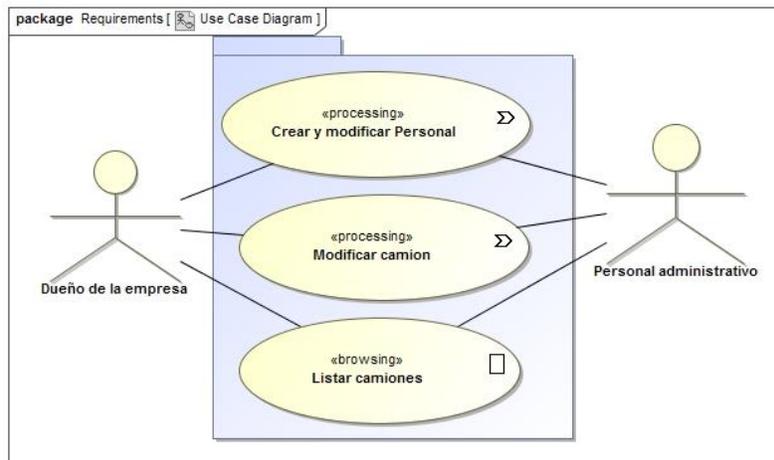
Fuente: Elaboración Propia

### 3.3.4 DESARROLLO DEL SPRINT: SEGUNDO SPRINT

Para el desarrollo del primer Sprint se diseñaron los modelos correspondientes a la metodología UWE, estos modelos fueron enfocados a responder las necesidades y alcanzar los objetivos de la lista del Sprint.

### 3.3.4.1 ANÁLISIS DE REQUERIMIENTOS

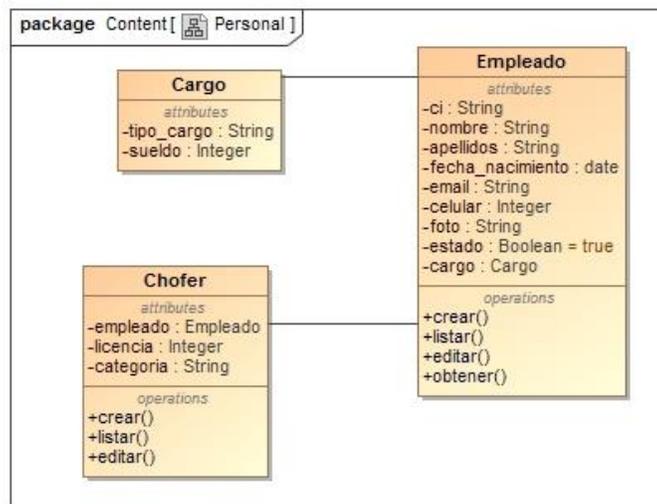
En la figura se presenta el análisis de requerimientos para el Segundo Sprint.



**Figura 3. 8.: Diagrama Casos de Uso - Administración del Personal**  
Fuente: Elaboración Propia

### 3.3.4.2 MODELO CONCEPTUAL

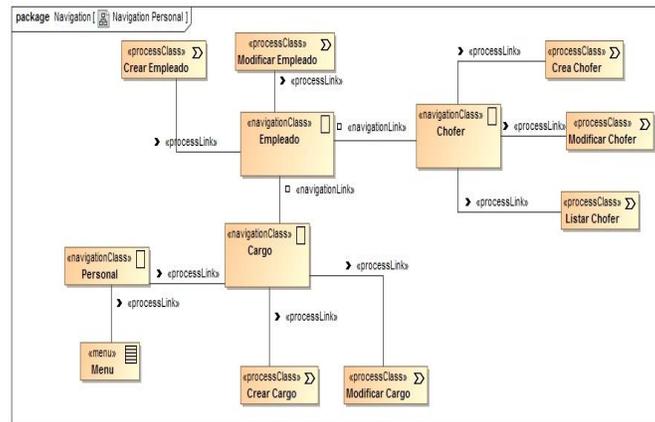
En modelo conceptual se realizó con los elementos más importantes.



**Figura 3. 9.: Modelo de Contenido - Administración del Personal**  
Fuente: Elaboración Propia

### 3.3.4.3 MODELO NAVEGACIONAL

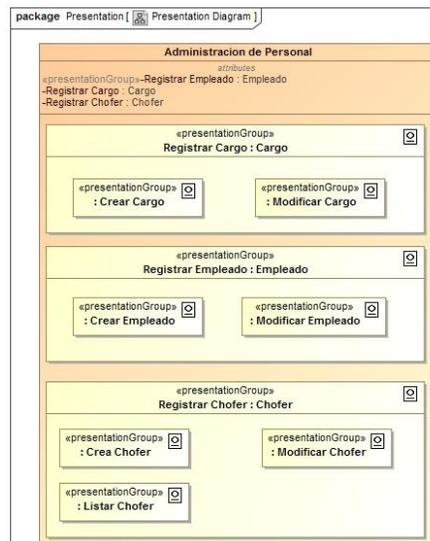
El modelo navegacional especifica los objetos que pueden ser visitados mediante la navegación a través del sistema web.



**Figura 3. 10.: Modelo Navegacional - Administración del Personal**  
Fuente: Elaboración Propia

### 3.3.4.4 MODELO DE PRESENTACIÓN

El modelo de presentación se diseñó para el segundo Sprint.



**Figura 3. 11.: Modelo de Presentación - Administración del Personal**  
Fuente: Elaboración Propia

### 3.3.4.5 DISEÑO DE INTERFAZ

El usuario final interactúa con el sistema web y para realizar una inspección del sistema se realizaron varias capturas de pantalla para observar la administración del personal.

Nombre completo	Email	Telefono	Estado
Juan Yupanqui Mamani	juanyupanqui67@hotmail.com	71251032	Inactivo
Bibiano Vilca Cachi	bibianovilca@gmail.com	76353525	Activo
Aldo Carrasco Fernandez	aldocarras123@gmail.com	69884263	Activo
Claudia Yupanqui Aruni	yaczo@gmail.com	69884173	Activo

**Figura 3. 12.: Diseño de Interfaz – Registro de Empleados**  
Fuente: Elaboración Propia

En la siguiente pantalla se mostrará las ventas y compras que realiza el empleado en el sistema.

Perfil Ordenes de venta Facturas Pagos Entregas

Claudia Yupanqui Aruni

Ajustes generales Personal Asignar contraseña

Nombre: Claudia Yupanqui Aruni

Email: yaczo@gmail.com

Interno: 69884173

Tipo Empleado: Administrador

Estado: Active

Cancelar Enviar

**Figura 3. 13.: Diseño de Interfaz – Listado de Empleados**  
Fuente: Elaboración Propia

### 3.3.4.6 PRUEBAS UNITARIAS

La prueba unitaria para la administración del personal de la empresa.

PRUEBA UNITARIA:	ADMINISTRACIÓN DEL PERSONAL
<b>Descripción:</b>	Permite realizar cambios del personal en el sistema.
<b>Objetivos:</b>	<ul style="list-style-type: none"> <li>- Registro de empleados al sistema.</li> <li>- Realizar las modificaciones necesarias en los empleados.</li> <li>- Listar a los empleados.</li> </ul>
<b>Condiciones:</b>	Tener acceso a internet desde cualquier dispositivo.
<b>Resultado esperado:</b>	Que el usuario realice actividades de los empleados en el sistema.
<b>Resultado obtenido:</b>	El usuario puede realizar el registro de empleados en el sistema y realizar los diferentes cambios que el usuario lo requiera.

**Tabla 3. 11.: Prueba Unitaria de la administración del personal**

Fuente: Elaboración Propia

### 3.3.5 PLANIFICACIÓN DEL SPRINT: TERCER SPRINT

La siguiente tabla, muestra el Sprint Backlog con las tareas a realiza en el tercer Sprint.

ID	TAREA	PRIORIDAD	ESTADO
1	Registro y modificación de los camiones de distribución y traslado	Alta	Completado
2	Administración de los camiones	Alta	Completado
3	Asignación de choferes a cada camión	Alta	Completado
4	Vista de los camiones registrados	Alta	Completado

**Tabla 3. 12.: Sprint Backlog 3**

Fuente: Elaboración Propia

### 3.3.6 DESARROLLO DEL SPRINT: TERCER SPRINT

Para el desarrollo del tercer Sprint se diseñaron los modelos correspondientes a la metodología UWE.

#### 3.3.6.1 ANÁLISIS DE REQUERIMIENTOS

A continuación, se muestra el caso de uso para el módulo de camiones.

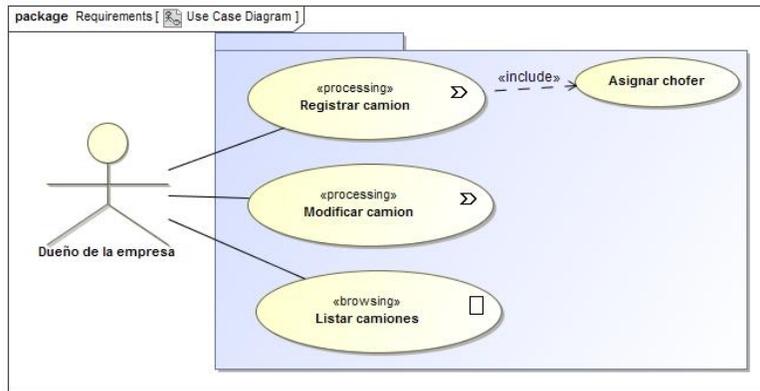


Figura 3. 14.: Diagrama de Casos de Uso – Administración de los camiones  
Fuente: Elaboración Propia

#### 3.3.6.2 MODELO CONCEPTUAL

El modelo conceptual proporciona la información relevante de la administración de camiones.

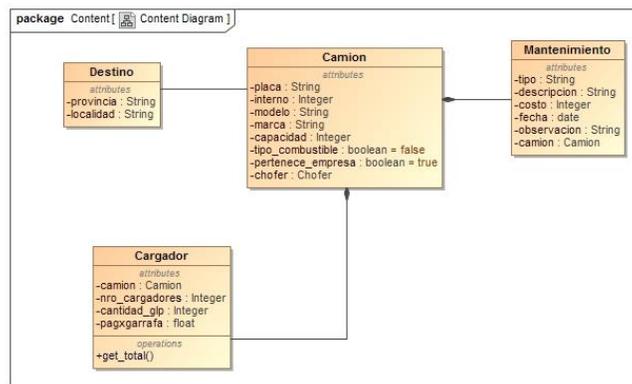
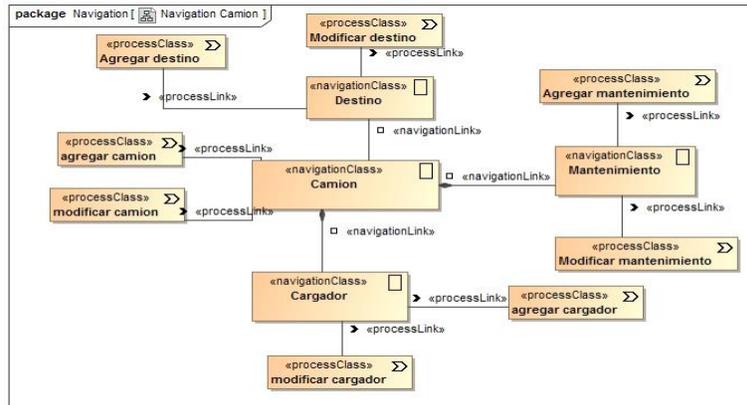


Figura 3. 15.: Diagrama de Contenido – Administración de los camiones  
Fuente: Elaboración Propia

### 3.3.6.3 MODELO NAVEGACIONAL

El modelo navegacional especifica los objetos que pueden ser visitados mediante la navegación a través del sistema web.

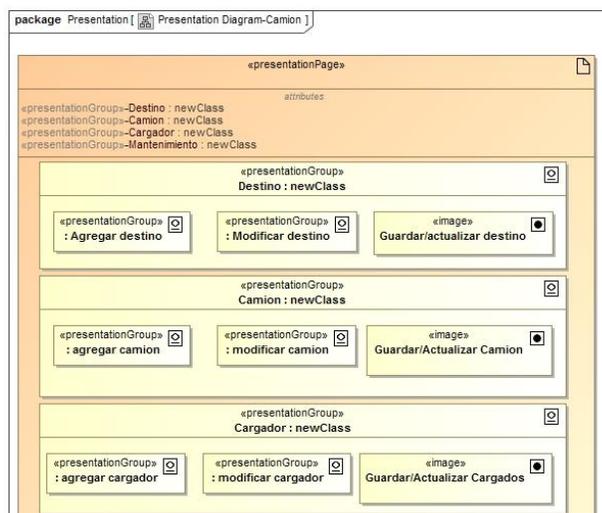


**Figura 3. 16.: Diagrama Navegacional – Administración de los camiones**

Fuente: Elaboración Propia

### 3.3.6.4 MODELO DE PRESENTACIÓN

El modelo de presentación nos ofrece una visión abstracta de la interfaz de usuario, a continuación, se muestran los modelos de presentación del módulo institucional.



**Figura 3. 17.: Diagrama de Presentación – Administración de los camiones**

Fuente: Elaboración Propia

### 3.3.6.5 DISEÑO DE INTERFAZ

El interfaz son las pantallas con las que el usuario final interactúa con el sistema web, para suministrar información a la base de datos. A continuación, se muestra las interfaces diseñadas para el módulo.

El diseño de la pantalla destino permite registrar, modificar y listar los datos.

Nombre del lugar	Código de localización	Dirección de entrega	Default Location	Interno	Acción
Calamarca	PL	Calamarca carretera La Paz - Oruro	No	Primary Location	
Lahuachaca	7	Carretera La Paz - Oruro	No	73484585	
Patacamaya	PAT	Carretera La Paz - Oruro	No	73484844	
Umala	U	Carretera La Paz - Oruro	No	76353525	
Villa Remedios	VR	Carretera La Paz - Oruro	No	73635521	

**Figura 3. 18.: Diseño de Interfaz – Administración de destinos**  
Fuente: Elaboración Propia

Pantalla camión permite registrar, modificar y listar los datos de los camiones.

Placa	Interno	Informacion camion	Estado
626-PXU	9	Nissan, Wollswagen, 400, BO, Amarillo	Activo
2314-KBD	5	Nissan, Nissan, 350, BO, Amarillo	Activo
3631-BLL	7	Volvo, Volvo, 350, BO, Amarillo	Activo
1236-SKT	4	Nissan, Nissan, 400, BO, Amarillo	Activo

**Figura 3. 19.: Diseño de Interfaz – Administración de camiones**  
Fuente: Elaboración Propia

### 3.3.6.6 PRUEBAS UNITARIAS

Se realizan las pruebas unitarias para el módulo de administración del personal.

PRUEBA UNITARIA:	ADMINISTRACIÓN DEL PERSONAL
<b>Descripción:</b>	Permite realizar cambios de los camiones en el sistema.
<b>Objetivos:</b>	<ul style="list-style-type: none"><li>- Registro de destinos al sistema.</li><li>- Registro de Camiones al sistema.</li><li>Registro de cargadores al sistema.</li><li>- Registro de Mantenimientos realizados a los camiones pertenecientes a la empresa.</li><li>- Realizar las modificaciones necesarias en los camiones, destinos.</li><li>- Listar a los camiones, destinos y mantenimientos existentes en el sistema.</li></ul>
<b>Condiciones:</b>	Tener acceso a internet desde cualquier dispositivo.
<b>Resultado esperado:</b>	Que el usuario realice actividades de los camiones en el sistema.
<b>Resultado obtenido:</b>	El usuario puede realizar el registro de camiones, destinos y mantenimientos y cargadores en el sistema y realizar los diferentes cambios que el usuario lo requiera.

**Tabla 3. 13.: Prueba Unitaria de la administración de camiones**  
Fuente: Elaboración Propia

### 3.3.7 PLANIFICACIÓN DEL SPRINT: CUARTO SPRINT

En este sprint se planifico el desarrollo del módulo de inventarios, para obtener la cuarta versión del sistema, el cual se encargará de todo el movimiento que tenga el GLP, como ser compras, ventas y la entrega de dinero de parte de los choferes, para tener el control de los inventarios y así generar los reportes físicos.

En cada requerimiento se realiza el análisis, diseño, implementación y pruebas unitarias utilizando las fases de la metodología UWE.

ID	TAREA	PRIORIDAD	ESTADO
1	Registro de las compras de GLP.	Alta	Completado
2	Administración las compras de GLP.	Alta	Completado
3	Administración de garrafas compradas y vendidas.	Alta	Completado
4	Registro de la venta de GLP.	Alta	Completado
5	Administración de la venta de GLP	Alta	Completado

**Tabla 3. 14.: Sprint Backlog 4**

Fuente: Elaboración Propia

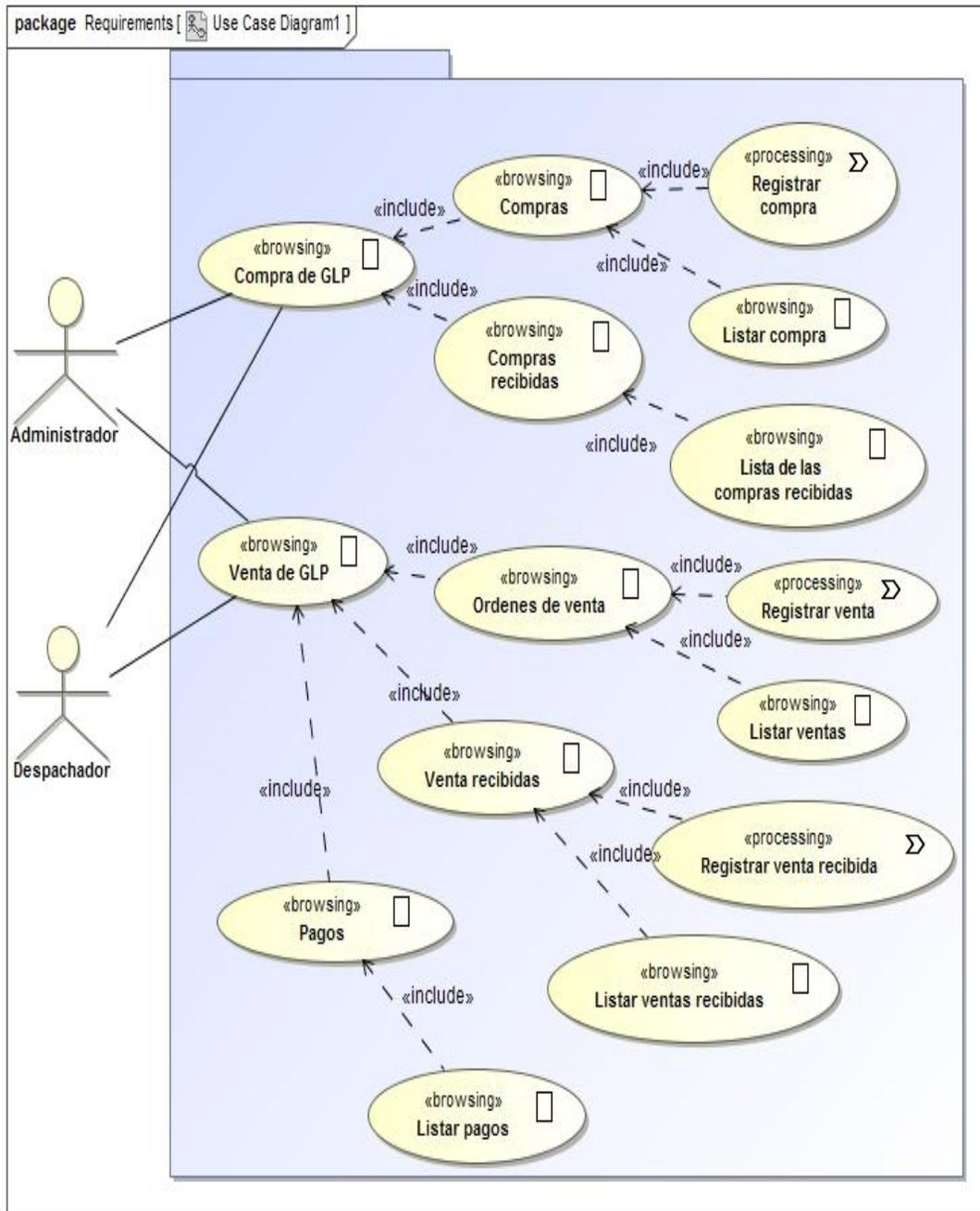
### 3.3.8 DESARROLLO DEL SPRINT: CUARTO SPRINT

Para el desarrollo del tercer Sprint se diseñaron los modelos correspondientes a la metodología UWE.

#### 3.3.8.1 ANÁLISIS DE REQUERIMIENTOS

Se presentará el caso de uso de inventarios donde los actores involucrados son: personal administrativo donde puede realizar las diferentes tareas con el producto, despachador que al igual que el administrador podrá hacer el registro de compra y venta, pero este no podrá realizar la modificación o eliminación de cualquier registro existente en el sistema. También se realizó una orden de pagos pendientes y pagos realizados por los choferes por las ventas o compras realizadas.

A continuación, se muestra el caso de uso para el cuarto sprint para el módulo de inventarios.

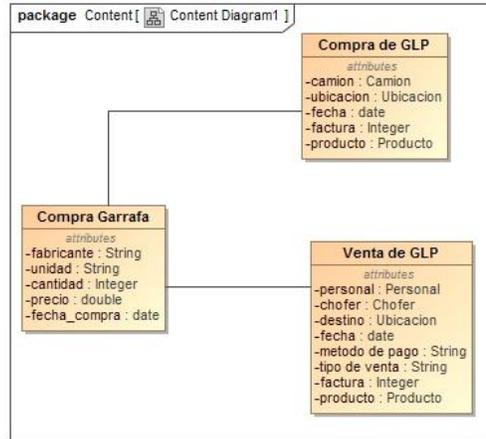


**Figura 3. 20.: Diagrama de Casos de Uso – Inventario**

Fuente: Elaboración Propia

### 3.3.8.2 MODELO CONCEPTUAL

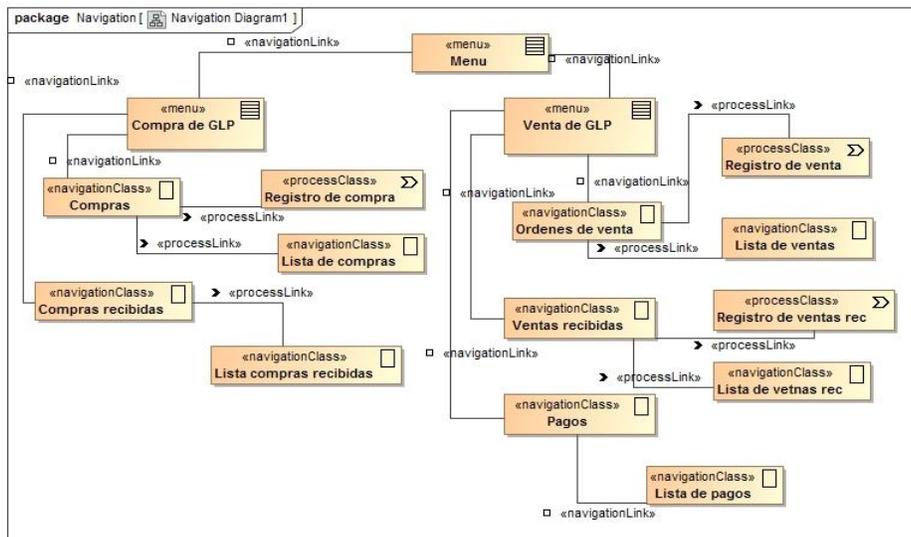
El diseño para el modelo conceptual del cuarto Sprint puede verse en la siguiente figura.



**Figura 3. 21.: Modelo Conceptual – Inventario**  
Fuente: Elaboración Propia

### 3.3.8.3 MODELO NAVEGACIONAL

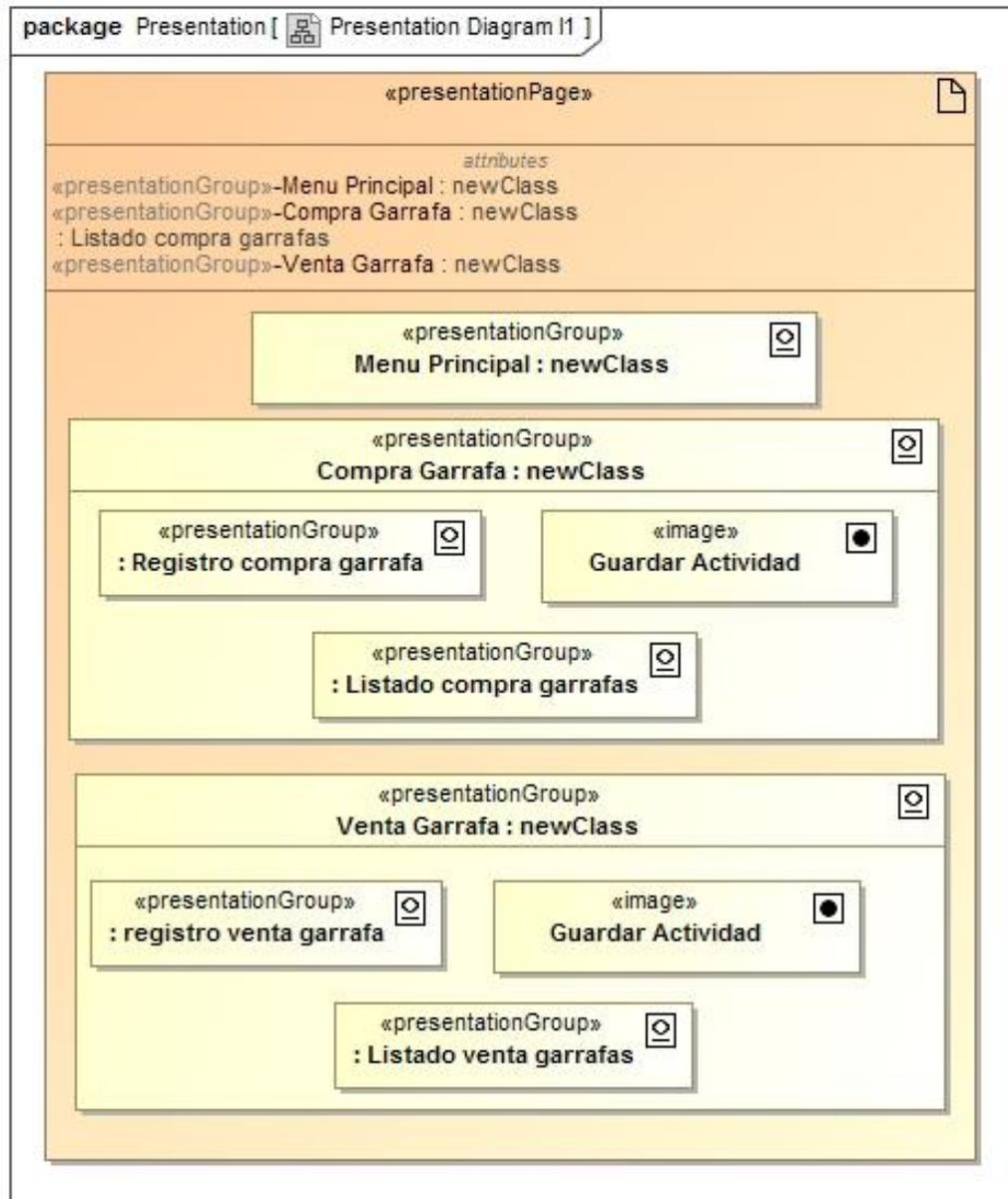
El modelo navegacional especifica los objetos que pueden ser visitados mediante la navegación a través del sistema web.



**Figura 3. 22.: Modelo Navegacional – Inventario**  
Fuente: Elaboración Propia

### 3.3.8.4 MODELO DE PRESENTACIÓN

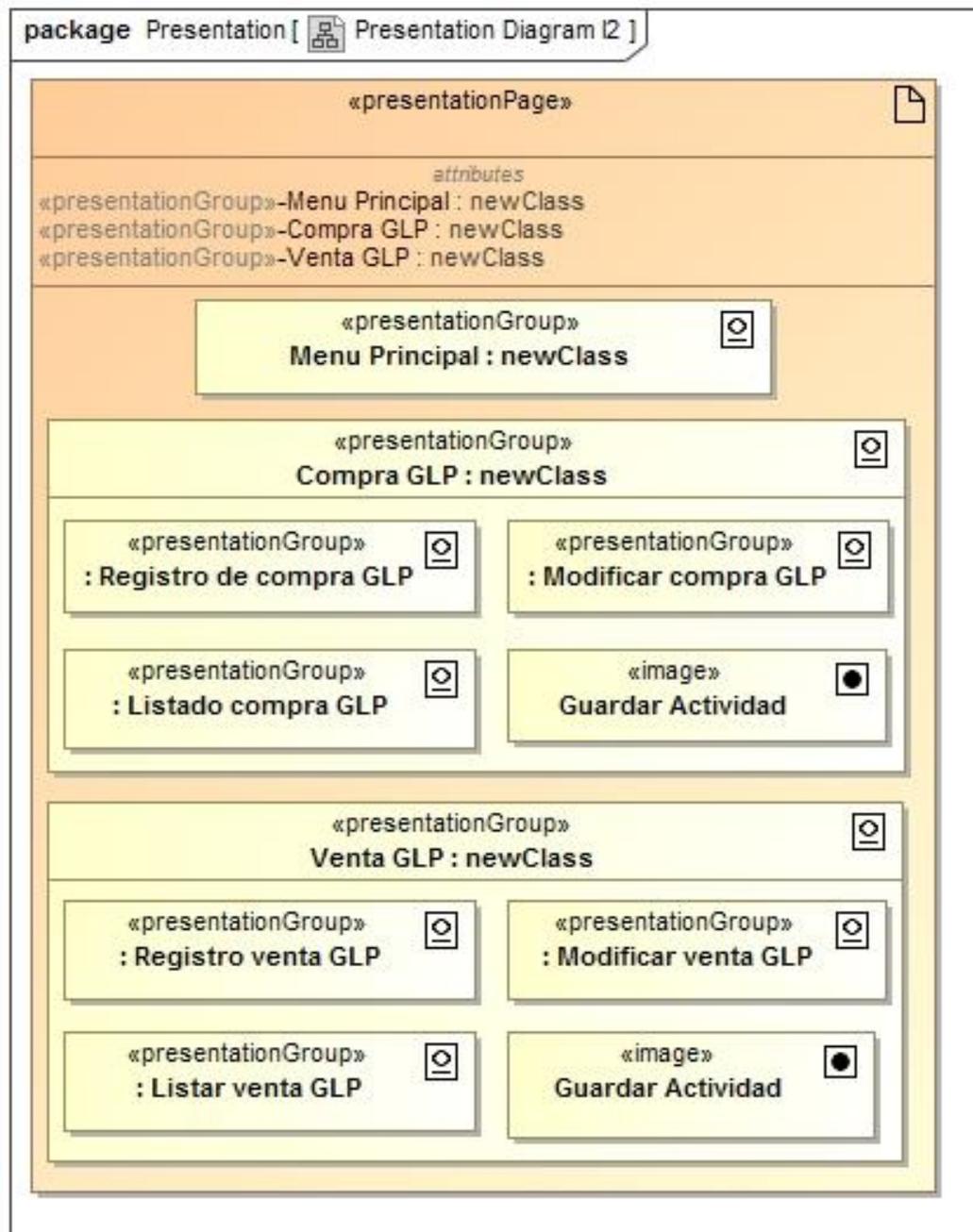
El modelo de presentación nos ofrece una visión abstracta de la interfaz de usuario, a continuación, se muestra el modelo del módulo de inventario.



**Figura 3. 23.: Modelo de Presentación – Inventario**

Fuente: Elaboración Propia

Se define el modelo de presentación de la actividad de compra y venta de GLP.

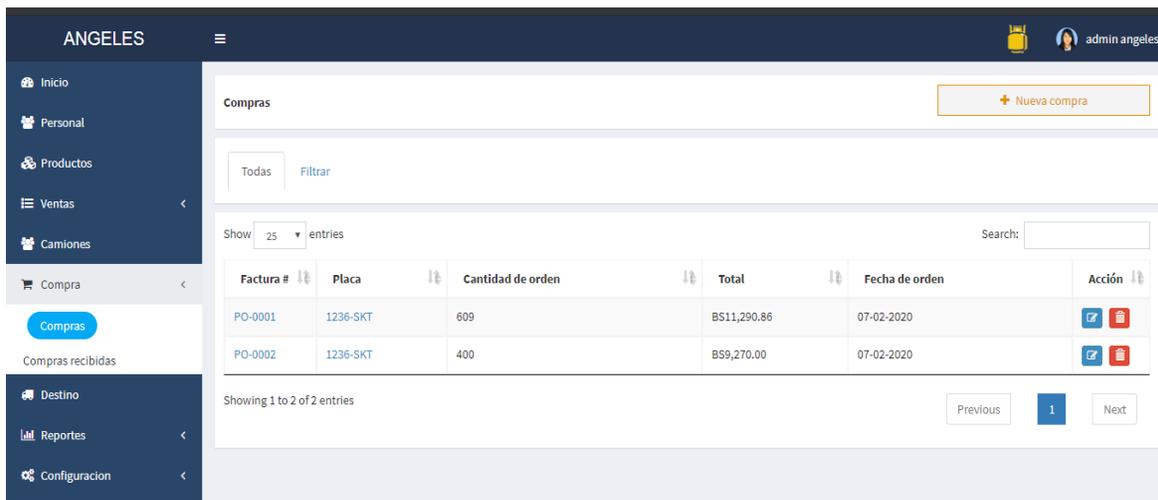


**Figura 3. 24.: Modelo de Presentación – Inventario**

Fuente: Elaboración Propia

### 3.3.8.5 DISEÑO DE INTERFAZ

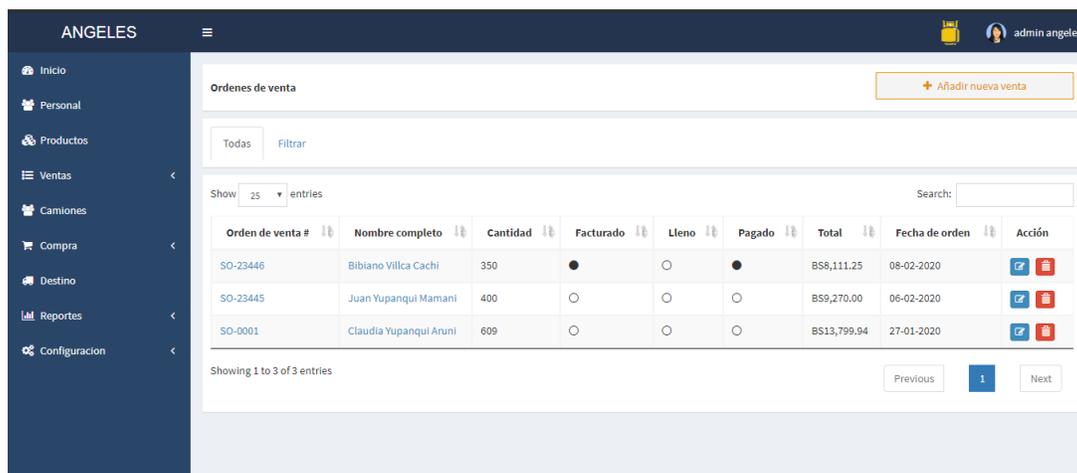
El usuario final interactúa con el sistema web y para realizar una inspección del sistema se realizaron capturas de pantalla para observar el control de compras y ventas del producto.



**Figura 3. 25.: Diseño de Interfaz – Administración de Compras**

Fuente: Elaboración Propia

La siguiente captura de pantalla muestra la administración de ventas.



**Figura 3. 26.: Diseño de Interfaz – Administración de Ventas**

Fuente: Elaboración Propia

### 3.3.8.6 PRUEBAS UNITARIAS

Se realizan las pruebas unitarias para el módulo de Inventarios.

PRUEBA UNITARIA:	ADMINISTRACIÓN DE COMPRAS Y VENTAS
<b>Descripción:</b>	Permite realizar el registro de todas las compras y ventas realizadas por la empresa.
<b>Objetivos:</b>	<ul style="list-style-type: none"> <li>- Registro de compra de garrafas al sistema.</li> <li>- Registro de venta de garrafas al sistema.</li> <li>- Registro de compra de GLP al sistema.</li> <li>- Registro de venta de GLP al sistema.</li> <li>- Realizar las modificaciones necesarias en los registros de compra y venta de GLP.</li> <li>- Listar la compra de garrafas, venta de garrafas.</li> <li>- Listar la compra y venta de GLP.</li> <li>- Mostrar un total de GLP vendido, comprado, stock actualizado, total de dinero a entregar por cada chofer</li> </ul>
<b>Condiciones:</b>	Tener acceso a internet desde cualquier dispositivo.
<b>Resultado esperado:</b>	Que el usuario realice actividades de distribución, compra y ventas de GLP en el sistema.
<b>Resultado obtenido:</b>	El usuario puede realizar el registro de las compras y ventas de GLP también obtiene los totales y dinero a entregar por cada chofer.

**Tabla 3. 15.: Prueba Unitaria – Administración de compras y ventas**

Fuente: Elaboración Propia

### 3.3.9 PLANIFICACIÓN DEL SPRINT: QUINTO SPRINT

Al igual que en cada Sprints, realizamos la planificación correspondiente del ultimo Sprint, obteniendo la lista de deseos procedentes del Product Backlog, en cada requerimiento se realiza el análisis, diseño, y pruebas utilizando las fases de la metodología UWE.

ID	TAREA	PRIORIDAD	ESTADO
1	Reportes de compras y ventas	Alta	Completado
2	Reporte de saldo por cobrar a choferes	Alta	Completado
3	Reporte de datos de los empleados	Alta	Completado

**Tabla 3. 16.: Sprint Backlog 5**

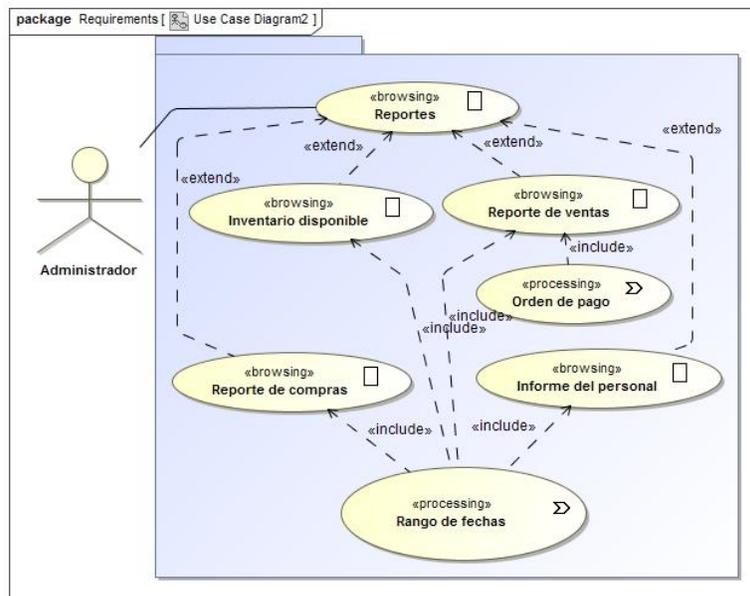
Fuente: Elaboración Propia

### 3.3.10 DESARROLLO DEL SPRINT: QUINTO SPRINT

El ultimo Sprint se realizado de acuerdo a los reportes requeridos para el sistema, siguiendo los lineamientos de la metodología.

#### 3.3.10.1 ANÁLISIS DE REQUERIMIENTOS

A continuación, se muestra el caso de uso del quinto Sprint.

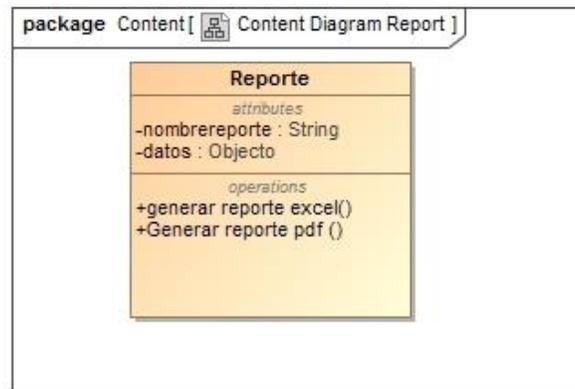


**Figura 3. 27.: Diagrama de Casos de Uso – Reportes**

Fuente: Elaboración Propia

### 3.3.10.2 MODELO CONCEPTUAL

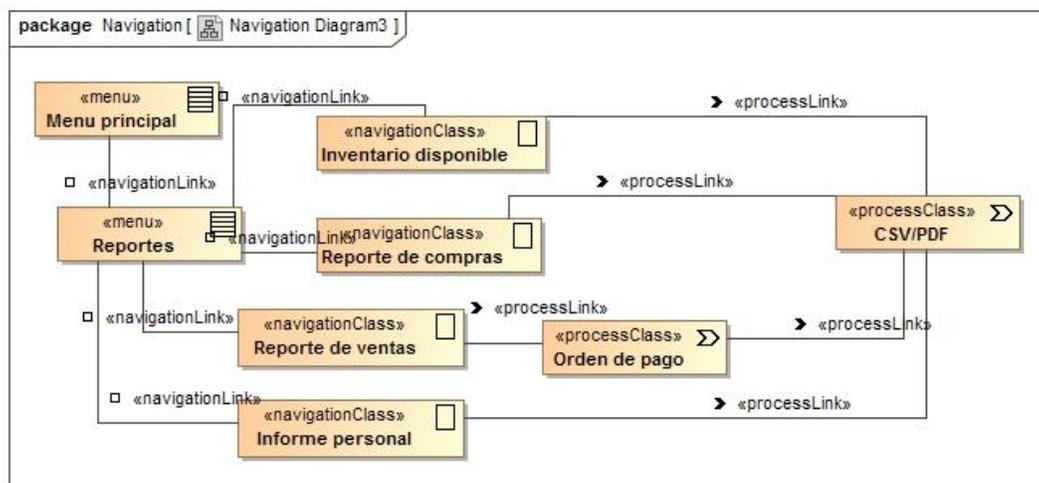
A continuación, se presenta el último modelo conceptual del proyecto, en este caso relacionado al módulo de generación de reportes.



**Figura 3. 28.: Modelo Conceptual – Reportes**  
Fuente: Elaboración Propia

### 3.3.10.3 MODELO NAVEGACIONAL

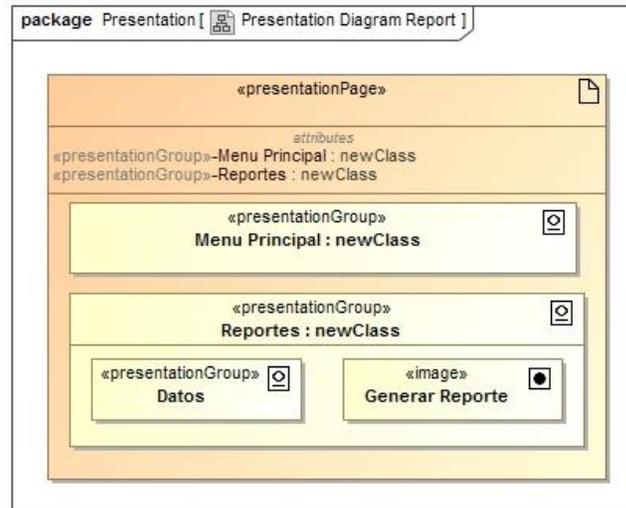
La siguiente figura representa el modelo navegacional del quinto Sprint del proyecto.



**Figura 3. 29.: Modelo Navegacional – Reportes**  
Fuente: Elaboración Propia

### 3.3.10.4 MODELO DE PRESENTACIÓN

El modelo de presentación nos ofrece una visión abstracta de la interfaz de usuario, a continuación, se muestra el modelo para el módulo de reportes.



**Figura 3. 30.: Modelo de Presentación– Reportes**  
Fuente: Elaboración Propia

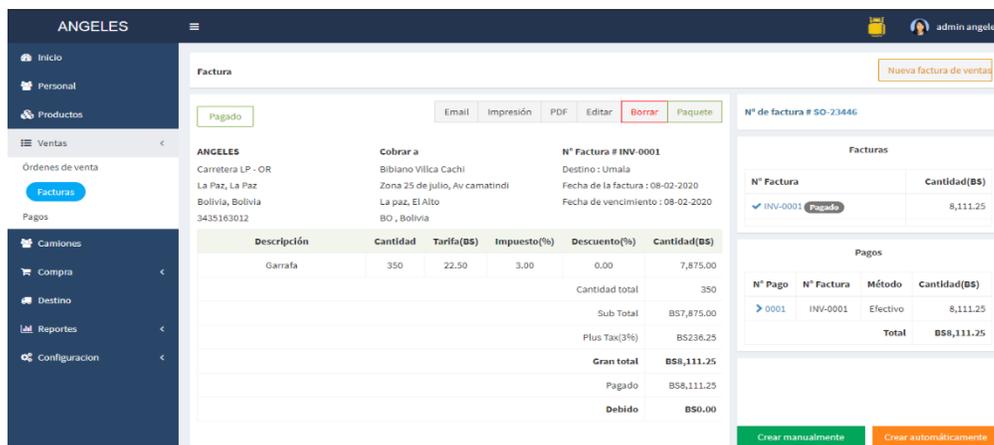
### 3.3.10.5 DISEÑO DE INTERFAZ

Las interfaces son las pantallas con las que el usuario final interactúa con el sistema web.

Fecha	N° Factura	Personal	Cantidad	Valores de venta(BS)	Valor de costo(BS)	Impuesto(BS)	Ganancias(BS)	Margen de beneficio(%)
08-01-2020	INV-7188	Juan Yupanqui Mamani	200	4,600.00	3,904.00	0.00	696.00	17.83
07-01-2020	INV-7187	Juan Yupanqui Mamani	409	9,202.50	7,983.68	0.00	1,218.82	15.27
06-01-2020	INV-7182	Juan Yupanqui Mamani	609	13,702.50	11,887.68	411.08	1,814.82	15.27
06-01-2020	INV-7186	Bibiano Vilca Cachi	400	9,200.00	7,808.00	0.00	1,392.00	17.83

**Figura 3. 31.: Diseño de Interfaz– Reportes**  
Fuente: Elaboración Propia

La siguiente figura muestra la orden de pago pendientes o pagos realizados por parte de los choferes, esta orden se genera al momento de realizar una venta o una compra.



**Figura 3. 32.: Diseño de Interfaz– Reportes**  
Fuente: Elaboración Propia

### 3.3.10.6 PRUEBAS UNITARIAS

Se realizan las pruebas unitarias para el módulo de Reportes.

PRUEBA UNITARIA:	REPORTES
<b>Descripción:</b>	Permite generar reportes de diferentes acciones dentro del sistema.
<b>Objetivos:</b>	Genera reportes cuando sea necesario, teniendo en cuenta las fechas.
<b>Condiciones:</b>	Tener acceso a internet desde cualquier dispositivo.
<b>Resultado esperado:</b>	Que el usuario logre realizar todos objetivos en el Sprint.
<b>Resultado obtenido:</b>	El usuario puede generar reportes al momento de realizar registros de compras, ventas u otra actividad.

**Tabla 3. 17.: Prueba Unitaria – Reportes**  
Fuente: Elaboración Propia

### **3.4 POST - GAME**

#### **3.4.1 PRUEBAS DE STRESS**

Una evaluación de este tipo poner a prueba la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan bugs (defectos) potencialmente peligrosos. (Sánchez, J., 2018).

Dentro de las pruebas de rendimiento de software se realizan o se encuentran las pruebas de stress son las que evalúan y ponen a prueba la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremo. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan defectos potencialmente peligrosos. La importancia de detectar errores a tiempo es tal, que las pruebas de estrés suelen realizarse en las primeras instancias del plan de aseguramiento de la calidad (QA). (Sánchez, J., 2018).

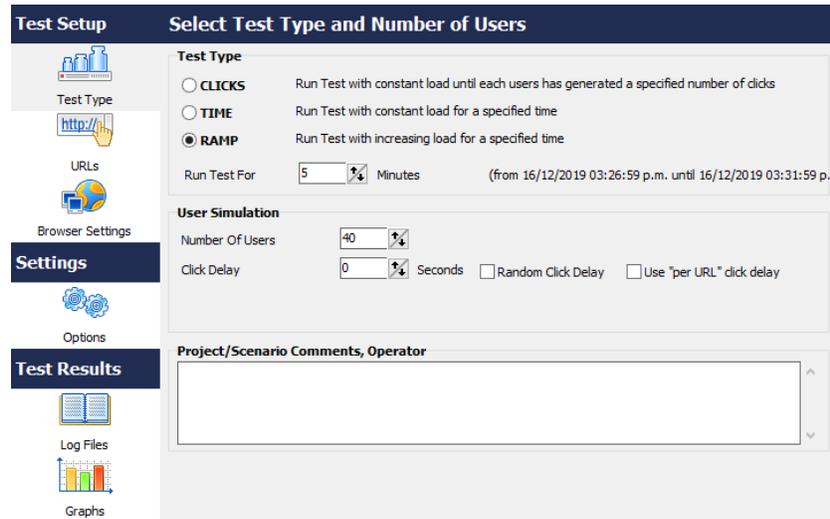
Se mide el tiempo de respuesta, el número de soluciones atendidas, cantidad de memoria consumida, medición del procesador y acceso al disco, número de transacciones y evaluar los límites de los componentes del sistema.

La herramienta que realizara la prueba de stress es WebServer Stress Tool, el cual es una herramienta para simular altas cargas de trabajo en un servidor web. WebServer Stress Tool puede simular una cantidad de 10000 usuarios de forma independiente a través de un conjunto de direcciones URL. Las direcciones URL simples son compatibles, así como también los patrones URL complejos.

Para usar Webserver Stress Tool es importante que se maximice el tiempo de actividad y se resuelva problemas críticos de rendimiento en el servidor web, también es importante que

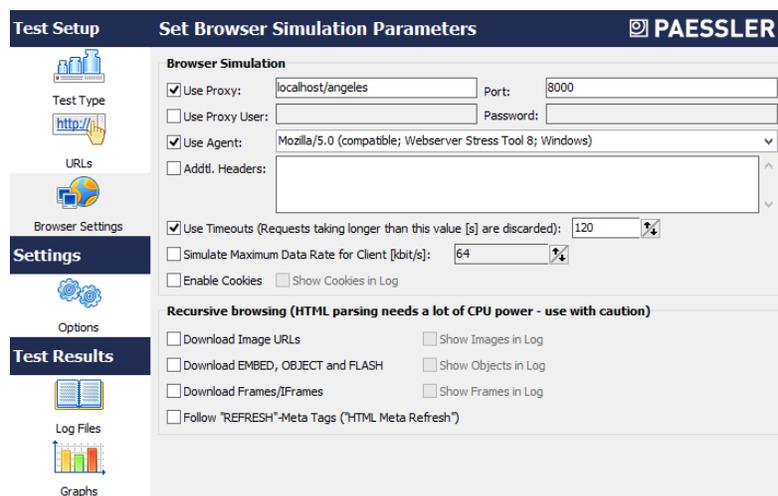
reciban los recursos del servidor que necesitan cuando lo necesitan para garantizar una experiencia de usuario de alta calidad.

A continuación, se realiza las pruebas de stress, en el programa antes mencionado.



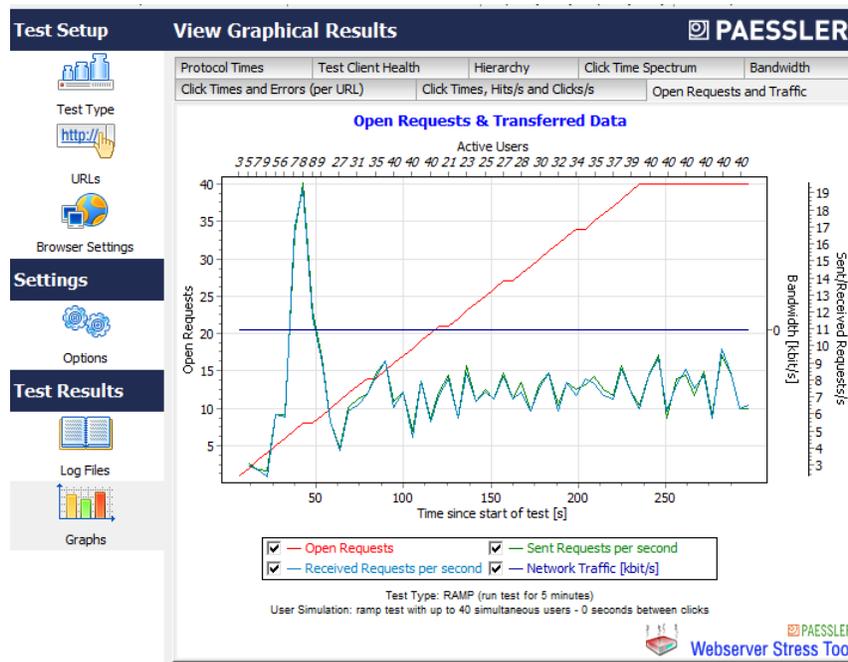
**Figura 3. 33.: Prueba de Stress – Tipo de prueba**  
Fuente: (WebServer Stress Tool, 2019)

A continuación, se establece los parámetros de simulación del navegador, donde podremos observar en la siguiente figura.



**Figura 3. 34.: Prueba de Stress – Parámetros de Simulación**  
Fuente: (WebServer Stress Tool, 2019)

Por último, se ve los resultados en la siguiente figura.



**Figura 3. 35.: Prueba de Stress – Resultado de la prueba**

Fuente: (WebServer Stress Tool, 2019)

Como podemos observar en el resultado obtenido por WebServer Stress, se observa que el rendimiento del sistema donde existen mas de 40 usuarios conectados indica que el sistema funciona y responde bastante bien cuando existe esta saturación de usuarios conectados, el resultado nos sirve para mejorar el sistema y de esta forma cumplir con las expectativas del usuario.

## **CAPÍTULO IV**

### **CALIDAD Y SEGURIDAD**

#### **4.1 INTRODUCCIÓN**

En este capítulo se determina la calidad del sistema web en base a parámetro de medición y medidas de seguridad que tiene el sistema, esto consiste en aquellos procedimientos, técnicas e instrumentos aplicados por entes capacitados para garantizar que un producto cumpla o supere un nivel mínimo aceptable para su comercialización.

Si es que así se lo ha planteado, lo que hasta el momento no se tiene estándares del software, no es lo mismo que las pruebas del sistema, estos son implícitos al momento de hacer las pruebas de integración final del sistema.

#### **4.2 FACTORES DE CALIDAD DEL SOFTWARE**

Según (Pressman, R., 2010), la calidad del software es “la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.

Entre las diferentes metodologías para medir la calidad de software, existe el método Website QEM, el cual se describirá a continuación.

#### **4.3 METODOLOGÍA WEBSITE QEM**

Según (Olsina, L., 1999), el propósito era el de aportar una estrategia eficaz para evaluar y analizar la calidad de sitios o aplicaciones Web. Está basada en un modelo jerárquico de requerimientos de calidad, partiendo de las características de alto nivel prescritas en la norma ISO-9126, las cuales son las siguientes:

- ✓ **Usabilidad:** Esta definido como un conjunto de atributos que otorgan el esfuerzo necesario para su uso, y en la evaluación individual de dicho uso, mediante un conjunto de usuarios declarados implícitos.
- ✓ **Funcionalidad:** Se define como el conjunto de atributos que otorgan la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen conjuntos de usuarios declarados implícitos.
- ✓ **Confiabilidad:** Se define como un conjunto de atributos de habilidad del software para mantener condiciones de establecer su propio nivel de desempeño por un periodo determinado.
- ✓ **Eficiencia:** Se define como un conjunto de atributos que otorgan la relación entre el nivel de rendimiento del software y la cantidad de recursos usados por el usuario, bajo las condiciones establecidas.
- ✓ **Mantenibilidad:** Se define como un conjunto de atributos que otorgan el esfuerzo necesario para hacer modificaciones específicas.
- ✓ **Portabilidad:** Se define como el conjunto de atributos que otorgan la habilidad de software para ser transferido de un entorno a otro.

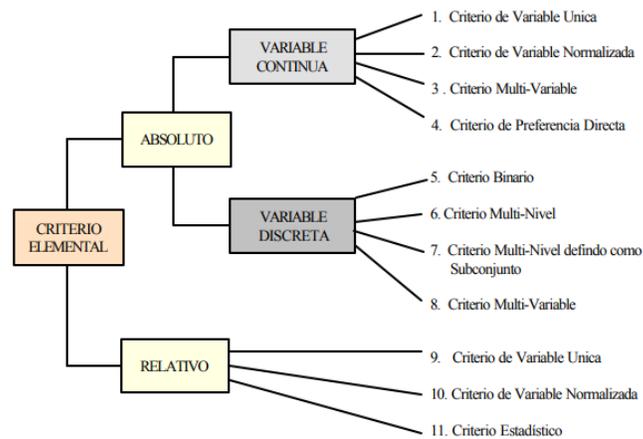
#### **4.3.1 CRITERIOS DE PREFERENCIA DE CALIDAD ELEMENTAL**

Según (Olsina, L., 1999), un criterio de evaluación elemental ayuda a comprender y especificar como medir atributos cuantificables. Nosotros estamos interesados en la determinación de la preferencia de calidad elemental para cada atributo de manera que por medio de un proceso de agregación podamos obtener un valor numérico global para el producto a evaluar.

La elección del tipo de criterio de evaluación elemental resulta de importancia en consideración de los niveles de precisión, objetividad y facilidad de uso. El nivel de precisión depende del grado de criticidad de alguno o de todos los componentes del producto en un proyecto de evaluación.

Un criterio de evaluación elemental absoluto es aquél que se emplea para determinar la preferencia absoluta de un atributo de un artefacto, y que no está relacionado con indicadores de otros sistemas comparativos. Un criterio absoluto se diferencia de uno relativo en que la meta de este último consiste solamente en la determinación de los indicadores relativos de los sistemas comparados sin evaluar la calidad (o la característica que corresponda) de cada sistema de un modo individual e independiente. Por lo tanto, una comparación relativa entre sistemas produce un orden o ranking relativo, que no puede ser interpretado individualmente dado que no representan el grado de cumplimiento de los requerimientos absolutos.

En la siguiente figura se pretende mostrar la taxonomía de los criterios elementales.



**Figura 4. 1.: Taxonomía de tipos de criterio elementales**

Fuente: Olsina, L., 1999

## 4.4 CALIDAD DEL PROYECTO

El análisis de calidad del software que se realizó para el presente proyecto, se realizó el uso del modelo de agregación y puntaje.

### 4.4.1 USABILIDAD

La usabilidad es la capacidad del software de ser entendido, aprendido y usado en forma fácil y atractiva. La usabilidad está determinada por los usuarios finales y los usuarios indirectos

del software, dirigidos a todos los ambientes, a la preparación del uso y el resultado obtenido. (ISO/IEC9126, 2010).

Para evaluar la usabilidad se debe considerar las siguientes características descritas en la tabla.

- **Comprensión global del sitio:** Representa a todas aquellas facilidades que permiten a la audiencia tener una rápida comprensión tanto de la estructura organizativa, como del contenido del sitio web, facilitando el acceso rápido y el recorrido del mismo y también de sus componentes.

COMPENSIBILIDAD GLOBAL DEL SITIO		
Nº	Subcaracterística	Resultado
1	Esquema de Organización Global	0,80
2	Calidad en el sistema de etiquetado	1,00
3	Visita guiada orientada al usuario	1,00
4	Mapa de imagen	0,95
<b>TOTAL</b>		0,93

**Tabla 4. 1.: Evaluación de Compensibilidad**

Fuente: Elaboración Propia

- **Mecanismos de ayuda y retroalimentación en línea:** Representa a un conjunto de preguntas que se realizan con mayor frecuencia y que están ya publicadas en el sitio web con sus respectivas respuestas.

MECANISMOS DE AYUDA Y RETROALIMENTACIÓN EN LÍNEA			
Nº	Subcaracterística		Resultado
1	Calidad de la ayuda		1,00
1.1	Ayuda Explicada orientada al usuario	1,00	
1.2	Ayuda de la búsqueda	1,00	

2	Indicador de última actualización		0,98
2.1	Global	1,00	
2.2	Restringido	1,00	
3	Directorio de direcciones		0,88
4	Facilidad FAQ		0,95
5	Retroalimentación		0,50
<b>TOTAL</b>			0,82

**Tabla 4. 2.: Mecanismos de ayuda y retroalimentación en línea**

Fuente: Elaboración Propia

- **Aspectos de interfaces y estéticos:** Se refiere al diseño de las paginas creadas en la aplicación para hacerla más agradable y vistosa al usuario.

ASPECTOS DE INTERFACES Y ESTÉTICOS			
N°	Subcaracterística	Resultado	
1	Cohesión al agrupar los objetos de control principales		1,00
2	Permanencia y estabilidad en la presentación de los controles principales		1,00
2.1	Permanencia de los controles directos	1,00	
2.2	Permanencia de los controles indirectos	1,00	
3	Aspectos de estilo		1,00
3.1	Uniformidad en el color de enlaces	1,00	
3.2	Uniformidad en el estilo global	1,00	
4	Preferencia estética		1,00
<b>TOTAL</b>			1,00

**Tabla 4. 3.: Aspectos de interfaces y estéticos**

Fuente: Elaboración Propia

La usabilidad de la aplicación web se determinará por el promedio de las características que se mencionará anteriormente como se muestra en la siguiente tabla:

USABILIDAD		
Nº	Características	Resultado
1	Comprensibilidad global del sitio	0,93
2	Mecanismos de ayuda y retroalimentación en línea	0,82
3	Aspectos de interfaces y estéticos	1,00
<b>TOTAL</b>		0,91

**Tabla 4. 4.: Aspectos de interfaces y estéticos**

Fuente: Elaboración Propia

**USABILIDAD = 91 %**

De acuerdo a los datos obtenidos en la tabla de usabilidad, se concluye que el sistema tiene una usabilidad del 0,91, es decir 91 %, que de cada 100 personas que lleguen a usar el sistema 91 personas indican que el sistema es fácil de manejar y comprensible.

#### **4.4.2 FUNCIONALIDAD**

Se refiere a la capacidad del producto para suministrar un conjunto de funciones que satisfagan las necesidades implícitas o explícitas de los usuarios, al ser utilizado bajo condiciones específicas. Se relaciona directamente con aquello que el software hace para satisfacer necesidades, mientras que las demás características se refieren al cómo y al cuándo. Evalúa el cumplimiento de requerimientos, la exactitud de los resultados, la seguridad del producto y la interacción con otros sistemas. (ISO/IEC9126, 2010).

Para evaluar la funcionalidad se debe considerar las siguientes características descritas en la tabla.

- **Aspectos de búsqueda y recuperación:** Especifica los mecanismos de búsqueda y recuperación para encontrar la información:

ASPECTOS DE BÚSQUEDA Y RECUPERACIÓN			
Nº	Subcaracterística		Resultado
1	Mecanismo de búsqueda en el sitio web		0,90
1.1	Búsqueda restringida	1,00	
1.2	Búsqueda global	1,00	
2	Mecanismos de recuperación		0,50
2.1	Nivel de personalización	1,00	
2.2	Nivel de retroalimentación en la recuperación	0,00	
<b>TOTAL</b>			0,70

**Tabla 4. 5.: Aspectos de búsqueda y recuperación**

Fuente: Elaboración Propia

- **Aspectos de navegación y exploración:** Describe la facilidad con la que el usuario puede navegar por todas las páginas que contiene el sitio web.

ASPECTOS DE NAVEGACIÓN Y EXPLORACIÓN			
Nº	Subcaracterística		Resultado
1	Navegabilidad		1,00
1.1	Orientación	1,00	
1.2	Promedio de enlaces por pagina	1,00	
2	Objetos de control navegacional		0,90
2.1	Permanencia y estabilidad	0,95	
2.2	Estabilidad	0,95	
3	Nivel de desplazamiento		1,00
3.1	Desplazamiento vertical	1,00	

3.2	Desplazamiento horizontal	1,00	
4	Predicción navegacional		0,80
<b>TOTAL</b>			0,70

**Tabla 4. 6.: Aspectos de navegación y exploración**

Fuente: Elaboración Propia

- **Aspectos de dominio orientados al usuario:** Se refieren a la capacidad de los temas, pero no limitan directamente su contenido.

ASPECTOS DE DOMINIO ORIENTADOS AL USUARIO		
Nº	Subcaracterística	Resultado
1	Relevancia de contenido	0,90
2	Servicios online	0,90
<b>TOTAL</b>		0,90

**Tabla 4. 7.: Aspectos de navegación y exploración**

Fuente: Elaboración Propia

La aplicación web se determina por el promedio de las características que se mencionaron anteriormente como se muestra en la siguiente tabla:

FUNCIONALIDAD		
Nº	Características	Resultado
1	Aspectos de búsqueda y recuperación	0,70
2	Aspectos de navegación y exploración	0,70
3	Aspectos del dominio orientados al usuario	0,90
<b>TOTAL</b>		0,80

**Tabla 4. 8.: Aspectos de navegación y exploración**

Fuente: Elaboración Propia

## FUNCIONALIDAD = 80 %

Interpretando la tabla de funcionalidad, el sistema tiene una funcionalidad o utilidad del 80 % para la empresa, lo que nos indica que el sistema cumple con los requisitos funcionales de forma satisfactoria.

### 4.4.3 CONFIABILIDAD

Esta característica principal se refiere a la capacidad del producto de software para conservar su nivel de desempeño bajo condiciones específicas durante un determinado periodo de tiempo.

La característica de la confiabilidad:

- **No deficiencia:** Representa la ausencia de los enlaces rotos es decir que conducen a nodos inaccesibles.

CONFIABILIDAD			
Nº	Subcaracterística		Resultado
1	No deficiencia		1,00
1.1	Errores de enlaces	0,00	
1.1.1	Enlaces rotos	0,00	
1.1.2	Enlaces inválidos	0,00	
1.1.3	Enlaces no implementados	0,00	
1.2	Errores o deficiencias varias	0,00	
1.2.1	Deficiencias p cualidades ausentes debido a diferentes navegadores	0,00	
1.2.2	Deficiencias o resultados inesperados	0,00	
1.2.3	Nodos destino en construcción	0,00	
<b>TOTAL</b>			<b>1,00</b>

**Tabla 4. 9.: Evaluación de Confiabilidad**

*Fuente: Elaboración Propia*

## CONFIABILIDAD = 100 %

El sistema tiene un grado de confiabilidad del 100 %, esto quiere decir que de cada 100 ejecuciones existen 0 fallas de elección del sistema.

### 4.4.4 EFICIENCIA

Esta característica principal se refiere a la capacidad del producto de software para proporcionar un desempeño apropiado, en relación con la cantidad de recurso utilizado, bajo condiciones establecidas en determinado momento del tiempo. Para determinar qué tan eficiente es un producto, se ha subdividido en tres sub características que abarcan los aspectos más relevantes que se deben tener en cuenta al respecto: comportamiento en el tiempo, consumo de recursos y conformidad en la eficiencia.

Para evaluar la eficiencia se debe considerar las siguientes características descritas en la tabla.

- **Desempeño:** El desempeño del software se considera que el esperado, de acuerdo con los niveles de servicio pactados para el mismo, teniendo en cuenta las necesidades de los usuarios. Se mide el tamaño de todas las páginas del sitio web, considerando todos sus componentes gráficos, tabulares y textuales.

DESEMPEÑO		
Nº	Subcaracterística	Resultado
1	Páginas de acceso rápido	1,00
TOTAL		1,00

**Tabla 4. 10.: Evaluación de Confiabilidad**

Fuente: Elaboración Propia

- **Accesibilidad:** Este atributo representa la accesibilidad a la información que está en las páginas. Es de relevancia que el sitio entero sea editado.

<b>ACCESIBILIDAD</b>		
<b>Nº</b>	<b>Subcaracterística</b>	<b>Resultado</b>
1	Accesibilidad de la información	1,00
1.1	Soporte a versión solo texto	1,00
1.2	Imagen con titulo	1,00
1.3	Legibilidad global	1,00
2	Accesibilidad de ventanas	1,00
2.1	Número de visitas considerando marcos	1,00
2.2	Versión sin marcos	1,00
<b>TOTAL</b>		<b>1,00</b>

**Tabla 4. 11.: Evaluación de Confiabilidad**

Fuente: Elaboración Propia

La eficiencia se determinará por el promedio de las características que se mencionaron anteriormente como se muestra en la tabla.

<b>EFICIENCIA</b>		
<b>Nro</b>	<b>Características</b>	<b>Resultado</b>
1	Desempeño	1,00
2	Accesibilidad	1,00
<b>TOTAL</b>		<b>1,00</b>

**Tabla 4. 12.: Evaluación de Eficiencia**

Fuente: Elaboración Propia

**EFICIENCIA = 100 %**

Para concluir decimos que el sistema tiene una eficiencia de 100 %, es decir que de cada 100 ejecuciones existen 0 fallas.

#### 4.4.5 RESULTADOS

Una vez realizadas las pruebas de Usabilidad, Funcionalidad, Confiabilidad y Eficiencia, propuesto por el modelo Website QEM, la calidad total del sistema web se determina por el promedio de estos como se muestra en la siguiente tabla:

N <sup>o</sup>	CARACTERISTICA DE ALTO NIVEL	CALCULO	PORCENTAJE
1	Usabilidad	0,91	91 %
2	Confiabilidad	1,00	100 %
3	Funcionalidad	0,80	80 %
4	Eficiencia	1,00	100 %
<b>CALIDAD GLOBAL</b>		0,93	93 %

**Tabla 4. 13.: Calidad Global del Sistema**

Fuente: Elaboración Propia

Por lo tanto, el nivel de calidad total del sistema integrado web para el control de ventas, inventarios y distribución de GLP, es de un 93%, lo cual es aceptable porque está dentro del rango establecido.

#### 4.5 SEGURIDAD

Los problemas de seguridad para sitios y sistemas web deben ser contemplados desde el diseño lógico. Los peligros se identifican al principio del proceso de desarrollo de software y las características de su diseño se especifican de modo que los eliminen o controlen.

OWASP es una organización sin ánimo de lucro a nivel mundial dedicada a mejorar la seguridad de las aplicaciones web y del software en general, pretender ser el centro de referencia de toda la información de seguridad enfocada a las aplicaciones web. (Vindel, R.,2016).

OWASP, cada cierto tiempo realiza un informe recogiendo las vulnerabilidades más comunes dentro de las aplicaciones web. Con este informe podemos estar al tanto de dichas vulnerabilidades y podemos emplearlo para comprobar si nuestro proyecto tiene alguna de las que aparecen en el informe.

OWASP Top 10 - 2017
A1:2017-Injection
A2:2017-Broken Authentication
A3:2017-Sensitive Data Exposure
A4:2017-XML External Entities (XXE)
A5:2017-Broken Access Control
A6:2017-Security Misconfiguration
A7:2017-Cross-Site Scripting (XSS)
A8:2017-Insecure Deserialization
A9:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

**Figura 4. 2.: OWASP Top 10 - 2017**

Fuente: OWASP, 2017

#### 4.5.1 SEGURIDAD DEL PROYECTO

Para la realización del proyecto se hizo uso de diferentes librerías relacionadas a la seguridad mediante las cuales puede responder a la lista de vulnerabilidades de OWASP de la siguiente manera:

- **Inyección:** El ORM elocuente de Laravel utiliza el enlace PDO que protege de las inyecciones de SQL.
- **Perdida de autenticación:** El ingreso al sistema se realiza mediante Tokens, los cuales se generan en base a una clave secreta y la encriptación del *password*.

id	user_id	password
1	admin angeles	\$2y\$10\$w0tv1wcfBOohPiKdtdQp.OU9MaMg1xZJP.XDJ8rDQLe...

**Figura 4. 3.: Encriptado del password**

Fuente: Elaboración propia

- **Exposición de datos sensibles:** El ingreso al sistema se realiza mediante Tokens, los cuales también se realiza el cifrado de datos.
- **Procesamiento de Entidad Externa:** El sistema utiliza la librería Blade de Laravel para prevenir errores y ataques de tipo XSS.
- **Pérdida de Control de Acceso:** Los controles de acceso de hacen primeramente ejecutando la creación del token correspondiente al usuario y luego diferenciando los accesos según su rol, además el token se invalida una vez que se termina la sesión.
- **Configuración de Seguridad Incorrecta:** El sistema tiene una arquitectura segmentada la cual proporciona una separación efectiva y segura entre componente y accesos a terceros.
- **Secuencia de Comando en Sitios Cruzados:** El sistema no relaciona comandos ni datos sensibles con ningún otro tipo de sistema web, ya que, incluso los reportes se hacen mediante un servidor de informes de ejecución que funciona con Laravel.
- **Deserialización Insegura:** El token que genera el sistema asegura que solamente si un usuario pertenece a la colección de usuarios validos las consultas con la base de datos se ejecuten, de lo contrario se bloquean.

## CAPÍTULO V

### ANÁLISIS COSTO BENEFICIO

#### 5.1 INTRODUCCIÓN

En este capítulo se consideran los costos de desarrollo dado que no podría hablar de los costos de equipos para la implementación del sistema ya que la institución cuenta con una tecnología que le permite satisfacer los requerimientos de software y hardware.

#### 5.2 COSTO DEL SISTEMA

##### 5.2.1 PUNTO FUNCIÓN

Es una técnica de estimación de software desarrollada originalmente por Allan Albrecht, quien definió conceptos para medir el software a partir de valoraciones de funcionalidades entregadas al usuario y no a partir de aspectos técnicos, esto con la intención de producir valoraciones independientes de la tecnología y fases del ciclo de vida utilizado. (O'Farrill, L., s.f.).

Los puntos función permiten traducir el tamaño de funcionalidades de software a un número, a través de la suma ponderada de las características que este tiene. Una vez que se han obtenido los puntos función, estos pueden ser traducidos en horas hombre o días de trabajo, según factor de conversión que depende de mediciones históricas de la productividad. Así se puede determinar el costo y presupuesto de los proyectos.

Se consideran cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla, estos valores se definen de la siguiente manera:

- ✓ **Número de entradas de usuario:** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación.

- ✓ **Número de salidas de usuario:** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación.
- ✓ **Número de peticiones del usuario:** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva.
- ✓ **Numero de archivos lógicos internos:** Se cuenta cada archivo maestro lógico, este es un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente.
- ✓ **Numero de archivos de interfaz externo:** Se cuenta todas las interfaces legibles por la máquina.

Para calcular el Punto Función, se debe hacer uso de la siguiente ecuación:

$$PF = PFSA * (0,65 + (0,01 * FCT))$$

Donde:

PF: Punto Función; PFSA: Punto Función Sin Ajuste; FCT: Factor de Complejidad Total

A continuación, se presenta el cálculo del PFSA:

Parametros	Cuenta	Complejidad	Total
Nº de Entradas de Usuario	14	4	56
Nº de Salidas de Usuario	14	5	70
Nº de Peticiones de Usuario	7	4	28
Nº de Archivos	5	10	50
Nº de Interfaces Externas	0	7	0
<b>TOTAL</b>			204

**Tabla 5. 1: Punto Función Sin Ajuste**

Fuente: Elaboración Propia

A continuación, se realizará el cálculo de FCT:

$$FCT = \sum(F_i)$$

Donde:

$F_i$ : Es asignada por la escala de la tabla a continuación.

<b>Importancia</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Comunicación de datos						X
Procesamiento distribuido				X		
Rendimiento					X	
Volumen de transacciones					X	
Entrada de datos Online	X					
Eficiencia del usuario final					X	
Reutilización				X		
Facilidad de instalación						X
Facilidad de Operación					X	
Múltiples locales					X	
Facilidad de cambio				X		
<b>SUBTOTALES</b>	0	0	0	9	20	10

**Tabla 5. 2: Factor de Complejidad Total**

Fuente: Elaboración Propia

Realizamos el cálculo de  $F_i$ :

$$FCT = \sum(F_i) = 39$$

Por último, hacemos uso de los resultados obtenidos anteriormente

$$PF = PFSA * (0,65 + (0,01 * FCT))$$

$$PF = 204 * (0,65 + (0,01 * 39))$$

$$PF = 212,16$$

Para la conversión de los puntos de función a KLDC se convierte los puntos función a miles de líneas de código, teniendo en cuenta lo que se tiene en la siguiente tabla:

Lenguaje	Nivel	Factor LDC/PF
C	2.50	128
ANSI BASUC	5.00	6464
JAVA	6.00	53
PL/I	4.00	80
ANSI COBOL 74	3.00	107
VISUAL BASIC	7.00	46
ASP	9.00	46
<b>PHP</b>	<b>11.00</b>	<b>29</b>
VISUAL C++	9.50	34

**Tabla 5. 3: Conversión de Puntos de Función**

Fuente: Pressman, R., 2003

Ahora se calculará las líneas de código (LDC) con la siguiente ecuación:

$$\text{LDC} = \text{PF} * (\text{Factor LDC/PF})$$

$$\text{LDC} = 212,16 * (29)$$

$$\text{LDC} = 6152,64$$

Por lo tanto, se tiene 6 mil líneas de código en el proyecto.

### 5.2.2 COCOMO II

Según (Pressman, R., 2002), COCOMO II es un modelo que cuenta con características como: costo de elaboración del proyecto, costos del software desarrollado y costos de implantación del sistema. Esta estimación de costo resulta de gran importancia debido a que se ve a grandes rasgos el esfuerzo y tiempo, los cuales serán plasmados en índices monetarios. Está orientado a la magnitud del producto final, midiendo el tamaño del proyecto, en función de la cantidad de línea de código.

### 5.3 COSTO DEL SOFTWARE

Para el cálculo del proyecto se hará uso de la aplicación de COCOMO II, el cual es un método orientado a los puntos función, donde el costo total del sistema estará dado por el costo de elaboración del proyecto el costo del software desarrollado y el costo de la implementación del sistema.

Hay que considera que:

- El sueldo promedio de un programador esta entre el rango de Bs. 2060 a Bs. 4000, por lo que en este caso se tomara el promedio que vienes a ser Bs. 3030 o su equivalente en \$us 435.

Haciendo los cálculos respectivos con la aplicación se obtuvieron los siguientes resultados:

The screenshot shows the USC-COCOMO II.2000.4 application window. The project name is 'proyecto', the scale factor is 18.97, and the development model is 'Early Design'. The main table displays the following data:

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NCM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	Proyecto	6000	435.00	1.00	Non-Specified	21.1	21.1	284.5	9174.23	1.5	2.2	0.0

Summary statistics at the bottom of the window:

	Estimated	Effort Sched	PROD	COST	INST	Staff	RISK
Optimistic	14.1	8.5	424.6	6146.73	1.0	1.7	
Most Likely	21.1	9.7	284.5	9174.23	1.5	2.2	0.0
Pessimistic	31.6	11.0	189.7	13761.34	2.3	2.9	

Additional parameters: Total Lines of Code: 6000, Hours/PM: 152.00.

**Figura 5. 1.: Tabla de Resultados – Cocomo II**

Fuente: Elaboración Propia

- **Horas/Personas/Mes:** 152
- **Tiempo de Desarrollo:** 9.7 semanas
- **Número de Programadores:** 2,2 personas, redondeando 2

- **Costo del Proyecto:** \$us 9174.23 lo que equivale a 63944.3831 Bs

### 5.3.1 COSTO DE ELABORACIÓN DEL PROYECTO

El costo calculado para la elaboración del proyecto, incluyendo las fases de planificación, de diseño y de desarrollo, se reflejan a continuación:

DETALLE	IMPORTE (\$US)
Material de escritorio	20
Energía eléctrica	100
Internet	200
Otros	30
<b>TOTAL</b>	<b>350</b>

**Tabla 5. 4.: Costo de Elaboración del Proyecto**  
Fuente: Elaboración Propia

### 5.3.2 COSTO DE IMPLEMENTACIÓN DEL SISTEMA

El costo total del software desarrollado esta dado por la suma de los costos mencionados anteriormente que son los costos de desarrollo de elaboración del proyecto y costo de elaboración del software.

### 5.4 COSTO TOTAL DEL SOFTWARE

El costo total del software se muestra la siguiente tabla:

DETALLE	IMPORTE (\$US)
Costo de elaboración del proyecto	350
Costo de desarrollo del software	9174.23
<b>TOTAL</b>	<b>9524.23</b>

**Tabla 5. 5.: Costo de Elaboración del Software II**  
Fuente: Elaboración Propia

Por lo tanto, el costo total del proyecto es igual a \$us 9524.23, realizando la conversión a moneda nacional es 66383.8831 Bs

## 5.5 BENEFICIOS DEL SISTEMA

En un proyecto es importante analizar la rentabilidad del mismo y sobre todo si es viable o no. Existen dos métodos muy usados para calcular la viabilidad de un proyecto, estos son: el VAN (Valor Actual Neto) y el TIR (Tasa Interna de Retorno). El VAN y la TIR son herramientas financieras que permiten evaluar la rentabilidad de un proyecto de inversión, con estos dos métodos, es como los que se calcula el beneficio del proyecto.

### 5.5.1 VALOR ACTUAL NETO (VAN)

La VAN es un procedimiento que permite calcular el valor presente de un determinado número de flujos de futuros ingresos y egresos que podría tener el proyecto. El método consiste únicamente en descontar el monto actual, es decir actualizar por medio de una tasa de flujos de caja futuros del proyecto, a ese valor se le resta la inversión inicial, así el valor obtenido es el valor actual neto.

$$VAN = \sum [Ganancias/(1+k)^n] - I_0$$

Donde:

**VAN:** Valor Actual Neto; **Ganancias:** Ingreso de flujo de caja anual; **I<sub>0</sub>:** Valor del desembolso inicial de la inversión; **k:** tasa de descuento o tasa de interés de un préstamo; **n:** Numero de periodos considerados.

Valor del VAN	Interpretación del VAN
VAN > 0	El proyecto es rentable
VAN = 0	El proyecto es rentable
VAN < 0	El proyecto no es rentable

**Tabla 5. 6.: Interpretación del VAN**

Fuente: Elaboración Propia

Los valores de ganancia esperados para el presente proyecto se calculan a cinco años, en este caso se utilizará una tasa de 13%, ya que es la tasa de interés de préstamo actual en las entidades financieras y una inversión de Bs. 66383.88

$$\text{Inversión} = \text{Bs. } 66383.88$$

$$\text{TD} = 13\%$$

Los valores de ganancia esperados se detallan en la siguiente tabla:

AÑO	1	2	3	4	5
FLUJO DE CAJA EN BS	25.000	37.000	49.000	61.000	73.000

**Tabla 5. 7 : Cantidad Nominal por Año**

Fuente: Elaboración Propia

Para hallar la VAN tenemos:

$$VAN = \frac{25.000}{(1 + 0,13)^1} + \frac{37.000}{(1 + 0,13)^2} + \frac{49.000}{(1 + 0,13)^3} + \frac{61.000}{(1 + 0,13)^4} + \frac{73.000}{(1 + 0,13)^5}$$

$$VAN = \frac{25.000}{1,13} + \frac{37.000}{1,28} + \frac{49.000}{1,44} + \frac{61.000}{1,63} + \frac{73.000}{1,84}$$

$$VAN = 22123.89 + 28906.25 + 24027.78 + 37423.31 + 39673.91 - 66383.88$$

$$VAN = 85771,26$$

Se tiene que el valor obtenido es mayor a cero por lo que la inversión es factible en un principio y el proyecto es rentable.

### 5.5.2 TASA INTERNA DE RETORNO (TIR)

El TIR es una tasa de descuento de un proyecto de inversión para que sea rentable. Cuando la VAN toma un valor igual a cero, k pasa a llamarse TIR. En términos generales, las mejores inversiones son las que proporcionan mayor TIR.

$$TIR = -I_0 + \sum [Ganancias / (1 + k)^n]$$

A continuación, se describe la interpretación del valor del TIR.

Valor del TIR	Interpretación del TIR
TIR > k	Se recomienda invertir
TIR = k	Es indiferente invertir
TIR < k	Se recomienda no invertir

**Tabla 5. 8.: Interpretación del TIR**

Fuente: Elaboración Propia

$$TIR = -I_0 + \frac{Q_1}{(1+k)^1} + \frac{Q_2}{(1+k)^2} + \frac{Q_3}{(1+k)^3} + \frac{Q_4}{(1+k)^4} + \frac{Q_5}{(1+k)^5}$$

Donde:

**TIR** = Tasa Interna de Retorno; **I<sub>0</sub>** = Valor del desembolso inicial de la inversión; **k** = tasa de interés de ahorro; **n** = es el número de periodos considerados

$$TIR = -66383.88 + \frac{25.000}{(1+0,13)^1} + \frac{37.000}{(1+0,13)^2} + \frac{49.000}{(1+0,13)^3} + \frac{61.000}{(1+0,13)^4} + \frac{73.000}{(1+0,13)^5}$$

$$TIR = 52 \%$$

Como vemos que el TIR es mayor que cero, se recomienda invertir en el sistema.

## 5.6 COSTO / BENEFICIO

Para hallar el costo / beneficio de un proyecto se aplica la siguiente ecuación:

$$C / B = \sum [Ganancias] / \sum [Costos]$$

Reemplazando los valores previamente calculados en la ecuación, tenemos:

$$C = 85771,26 \text{ Bs}$$

$$B = 66383,88 \text{ Bs}$$

$$C / B = 85771,26 / 66383,88$$

$$C / B = 1,29$$

Por lo tanto, por cada boliviano invertido, la empresa tiene una ganancia de 0.29 ctvs

## CAPÍTULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 CONCLUSIONES

Se logro almacenar a todos los empleados y camiones dentro de una base de datos, asegurando de ese modo su permanencia y seguridad además de la eliminación de la vulnerabilidad de perdidas en situaciones de traslado y con la automatización de las compras y ventas de GLP se consiguió tener un control de los inventarios, también se pudo obtener saldos de dinero de compra y venta de los choferes.

El sistema garantiza la seguridad de la información de la empresa, al tener registrado las compras y ventas también se pudo obtener las ventas diarias en el momento oportuno y al tener automatizado gran parte de sus procesos se puede obtener los reportes que ayudaran a la administración y a la gerencia en la toma de decisiones para planificar estrategias de crecimiento.

Por lo que podemos concluir que se desarrolló un sistema integrado web para el control de ventas, inventarios y distribución de garrafas de gas licuado de petróleo optimo ya que se mejoró el tiempo y uso de recursos y así minimizar el esfuerzo en la realización de las tareas rutinarias en la distribuidora Ángeles, obteniendo como resultado un producto totalmente funcional.

#### 6.2 RECOMENDACIONES

De acuerdo con los logros obtenidos en el presente proyecto, surgen algunas recomendaciones para mejorar el sistema.

- ✓ Suministrar información en el momento en que se realiza la transacción para tener siempre actualizado la información en el sistema.

- ✓ En el futuro se recomienda implementar e integrar al sistema el pedido del producto mediante la plataforma para una mejor interacción con el cliente.
- ✓ Implementar e integrar al sistema el módulo contable es recomendable, ya que este maneja la situación financiera de toda la empresa.
- ✓ Realizar un mantenimiento al sistema para prevenir posibles fallas en el futuro.

## BIBLIOGRAFÍA

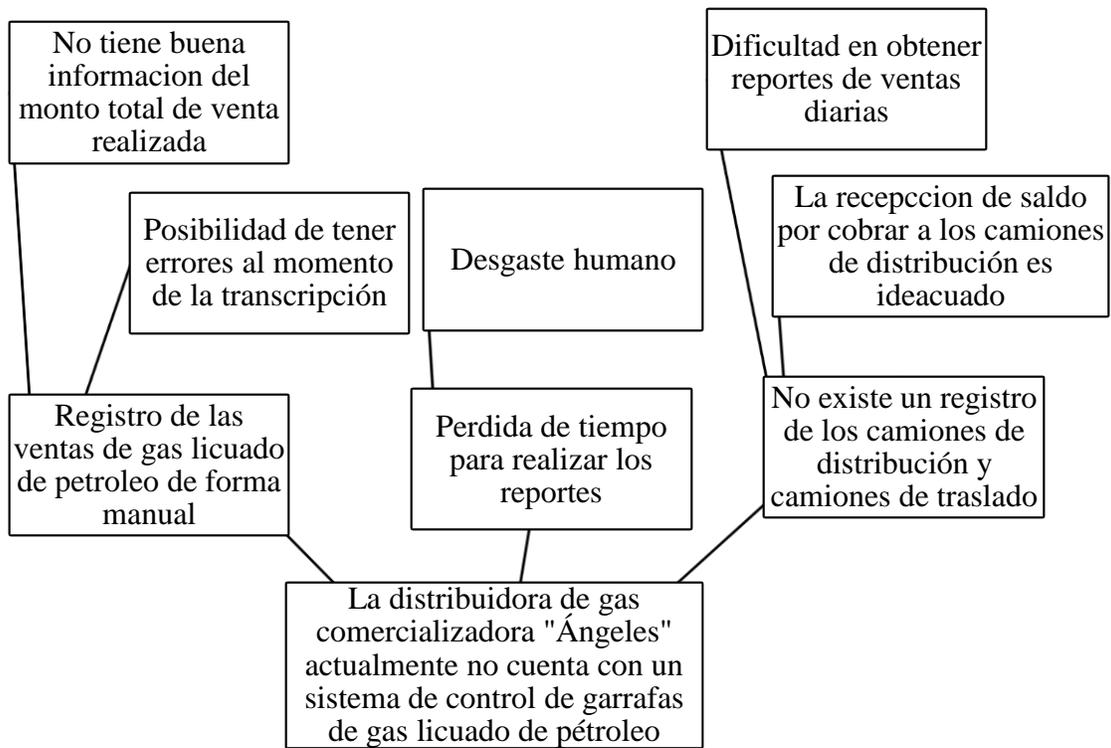
- AcensTechnologies. (s. f.). *Framework para el desarrollo ágil de aplicaciones*. Recuperado de: <https://www.acens.com/wp-content/images/2014/03/frameworks-white-paper-acens-.pdf>
- Aquino, H. (2018). *Aplicación web para el registro y control de compras, ventas e inventarios. Caso: Asociación de productores en metalmecánica sur “Metal Sur”*
- Booch, G., Rumbaugh, J., & Jacobson, I. (2006). *El Lenguaje Unificado de Modelado (2da ed.)*. Madrid: Pearson Educación.
- Carpio, F. (2017). *Sistema de administración académico vía web, caso: Escuela de idiomas*.
- Carrasco, A. (2019). *Antecedentes Institucionales*. Recuperado de: Distribuidora de Gas Comercializadora Ángeles.
- Casillas, L. et al. (2018). *Base de datos en MySQL*. Recuperado de: [https://s3.amazonaws.com/academia.edu.documents/54703109/05-M05-Bases\\_de\\_datos\\_en\\_MySQL.pdf](https://s3.amazonaws.com/academia.edu.documents/54703109/05-M05-Bases_de_datos_en_MySQL.pdf)
- Cevallos, K. (2015). *UML: Casos de Uso*. Recuperado de: <https://ingsoftwarekarlacevallos.wordpress.com/2015/06/04/uml-casos-de-uso/>
- Chapaval, N. (2018). Qué es Frontend y Backend. Recuperado 30 de mayo de 2019, de platzi.com website: <https://platzi.com/blog/que-es-frontend-y-backend/>
- Cillero, M. (2009). *Diagrama de secuencia*. Recuperado de: <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-interaccion/diagrama-de-secuencia/>
- Cordero, X. (2018). *Sistema web integrado para el registro y control de expedientes médicos, caso: NovaSalud S.R.L.*

- Php. (2019). *Página de Php*. Recuperado de: <https://www.php.net/manual/en/intro-what-is.php>
- Fundanet, (2019). *Pruebas unitarias y de integración, un doble test que marca la diferencia*. Recuperado de: <https://www.fundanet.es/blog/pruebas-unitarias-y-de-integracion/>
- Galiano, L. (2012). *Informe de la metodología aplicada en mi solución en mi solución informática de mi proyecto*. Recuperado de: <http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html>
- García, E. (2014). *Administración tecnológica e innovación en la estrategia de la empresa*. Recuperado de: <https://www.gestiopolis.com/administracion-tecnologica-e-innovacion-en-la-estrategia-de-la-empresa/>
- Gutiérrez, J. (s. f.). *¿Qué es un framework web?* Recuperado de: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)
- Infante, S. (2012). *Curso Django: Entendiendo como trabaja Django*. Recuperado de: [maestrosdelweb.com website: http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/](http://www.maestrosdelweb.com/curso-django-entendiendo-como-trabaja-django/)
- Lecoña, A. (2018). *Sistema web de control de inventario, reservas y facturación, caso: “cafetería Lala’s”*.
- LUM. *Ludwing-Maximilian-Universität München* Institute for Informatics. Research Unit of Programming and Software Engineering. Recuperado de: <http://uwe.pst.ifi.lmu.de/teachingTutorialSpanish.html>
- Molina, A. (2016). *Sistema Web Integrado de Control de PProduccion, Ventas y Personal para la empresa “Altaplast”*

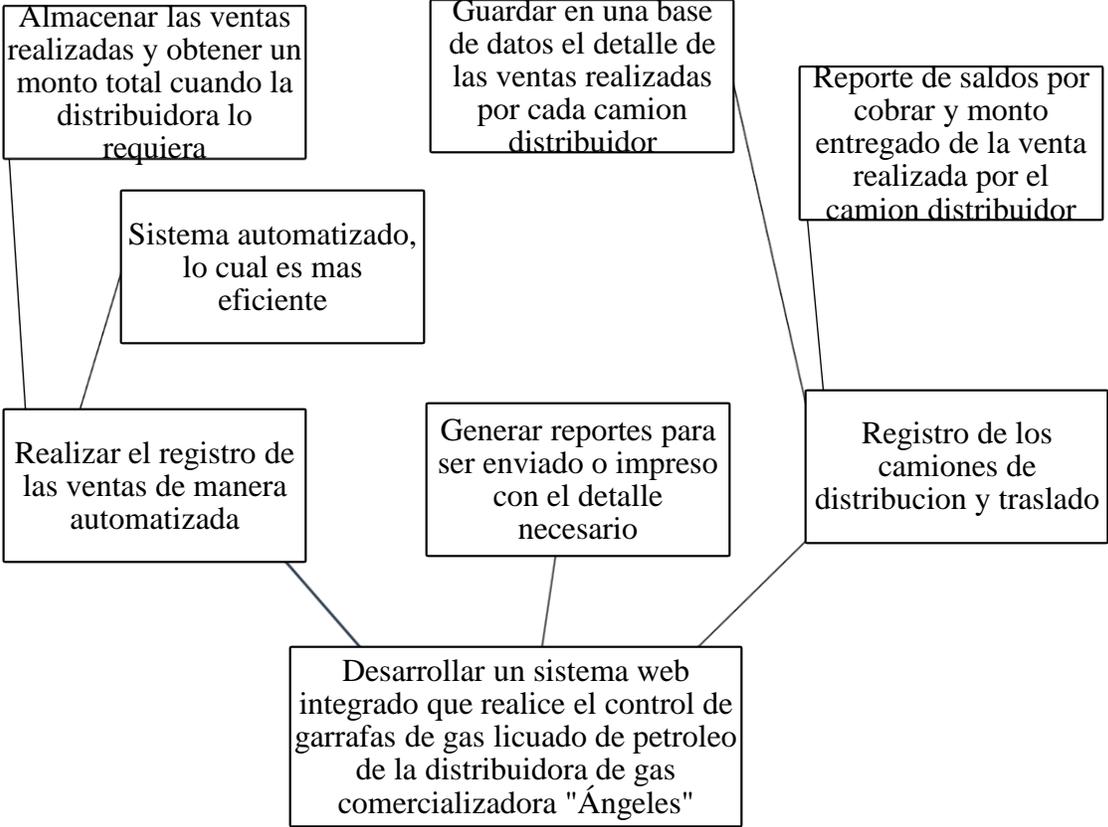
- Mora, S. (2003). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Recuperado de: <https://laingenieriaweb.wordpress.com/la-ingenieria-web/>
- O’Farrill, L. *Estimación de tiempo y esfuerzo en proyectos de software*. Recuperado de: <http://www.pmoinformatica.com/2015/04/estimacion-puntos-funcion-introduccion.html>
- Peralta, A., (2003), *Metodología Scrum*. Recuperado de: <https://fi.ort.edu.uy/innovaportal/file/2021/1/scrum.pdf>
- Pérez, H. (2010). *Propuesta de análisis y diseño basada en UML y UWE para la migración de arquitectura de software centralizada hacia internet*. Recuperado de: [http://biblioteca.usac.edu.gt/tesis/08/08\\_0470\\_CS.pdf](http://biblioteca.usac.edu.gt/tesis/08/08_0470_CS.pdf)
- Pinzón, O. *Ingeniería Web: Una Metodología para el Desarrollo de Aplicaciones Web Escalables y Sostenibles*. Universidad de Panamá. Recuperado de: [http://www.laccei.org/LACCEI2017-BocaRaton/student\\_Papers/SP277.pdf](http://www.laccei.org/LACCEI2017-BocaRaton/student_Papers/SP277.pdf)
- Pressman, R. (2010). *Ingeniería del Software. Un Enfoque Práctico*. 7ª ed. México D.F. 805 p.
- Platzi. (2017). *Qué es SCRUM y los roles en SCRUM*. Recuperado de: <https://platzi.com/blog/que-es-scrum-y-los-roles-en-scrum/>
- Quisbert, V. (2015). *Sistema web de control de ventas e inventarios de insumos, caso: “La Española”*.
- Quijano, J. (2012). *Cuando todos son ventajas. Sprint Backlog. Hablando de Scrum*. Recuperado de: <https://www.genbeta.com/desarrollo/cuando-todos-son-ventajas-sprint-backlog-hablando-de-scrum>.
- Redrován Fausto, L. N. (2017). *Estado del Arte: Métricas de Calidad para el Desarrollo de Aplicaciones Web*. Ecuador: 3C Tecnología. Recuperado de: <https://www.3ciencias.com/wp-content/uploads/2017/12/Art1-1.pdf>

- Ribes, A. (2011). *Manual JavaScript*. Madrid: CEP S.L.
- Rosselló, V., (2019). *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa*. Recuperado de: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- Rossum, G. Van. (2017). El tutorial de Python. *docs.python.org.ar*. Recuperado de <http://docs.python.org.ar/tutorial/pdfs/TutorialPython3.pdf>
- Salas, M., (2017). *Ingeniería Web Basada en UML*. Recuperado de: <http://marcelosalasvargas.blogspot.com/2017/06/>
- Sánchez, J., (2015). *Pruebas de Software. Fundamentos y Técnicas*. Recuperado de: [http://oa.upm.es/40012/1/pfc\\_jose\\_manuel\\_sanchez\\_peno\\_3.pdf](http://oa.upm.es/40012/1/pfc_jose_manuel_sanchez_peno_3.pdf)
- Sierra, K., (2018). *¿Qué es Laravel y para qué sirve?* Recuperado de: <https://www.synergyweb.es/blog/laravel-desarrollo-medida/>
- Scrum Manager, (2014). *Historias de Usuario*. Recuperado de: [https://www.scrummanager.net/bok/index.php/Historia\\_de\\_usuario](https://www.scrummanager.net/bok/index.php/Historia_de_usuario)
- Sutherland, J., K. S. (2013). *La guía Definitiva de Scrum: Las Reglas del Juego*. Estados Unidos. Recuperado de: <https://www.scrumguides.org/docs/scrumguide/v1/scrum-guidees.pdf>
- Trigas, M. (2012). *Metodología Scrum*. Recuperado de: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612 memoria.pdf>
- Vindel, R. (2016). *Introducción a OWASP*. Recuperado de: <https://www.adictosaltrabajo.com/2016/03/07/introduccion-a-owasp/>
- Zapata, J. (2014). *Fundamentos de la gestión de inventarios*. Medellín, Colombia: Editorial L. Vieco S.A.S.

## ANEXO A – ÁRBOL DE PROBLEMAS



**ANEXO B- ÁRBOL DE OBJETIVOS**



## ANEXO C – MARCO LÓGICO

Resumen narrativo de objetivos	Indicadores objetivamente verificables:	Medición de verificación	Supuestos
<b>Fin:</b> Automatizar el control de gas licuado de petróleo (GLP).	<b>Medidas del logro del fin:</b> Disminuir el tiempo de trabajo y ya no realizar el registro de ventas de gas licuado de petróleo de forma manual.	Observación de las operaciones que se realizan en la distribuidora.	Capacitación al personal administrativo e instalar el software en el equipo de computadora de la distribuidora
<b>Propósito:</b> Implementar un sistema web para realizar un mejor el control de garrafas de GLP.	Registro de todo el entorno de garrafas de GLP gas licuado de petróleo, como ser: choferes, camiones, ayudantes.	La parte operaria notara los cambios realizados mediante la observación	Adaptabilidad de los cambios que se realicen en la distribuidora.
<b>Componentes:</b> Almacenar todos los datos proporcionados a la empresa en una base de datos	<b>Dimensión de productos necesarios para lograr el proyecto:</b> Módulo de administración, módulo de movimiento.	Documento aval de aceptación por parte de la distribuidora de gas comercializadora “Ángeles”.	Apoyo por parte de la distribuidora para realizar las respectivas pruebas.
<b>Actividades:</b> Llevar a cabo una recopilación de datos sobre las ventas que se realizan.	<b>Nivel de esfuerzo por actividad:</b> El tiempo de demora del proyecto esta previamente establecida.	Documentos de análisis, diseño y desarrollo del sistema.	Apoyo con el hardware necesario para la instalación del sistema. Disponibilidad con la información proporcionada por parte de la distribuidora.