

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



PROYECTO DE GRADO

**“SISTEMA WEB DE CONTROL Y SEGUIMIENTO DE SERVICIOS Y
GESTIÓN DE CLIENTES PARA LA EMPRESA CONSULTORA
CONTADORES PÚBLICOS & AUDITORES AYS S.R.L.”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

**POSTULANTE: WILSON MAMANI CONDORI
TUTOR METODOLÓGICO: M.SC. ALDO VALDEZ ALVARADO
ASESOR: LIC. FREDDY MIGUEL TOLEDO PAZ**

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

Al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando a punto de rendirme he estado; por ello, con toda la humildad que de mi corazón puede emanar, dedico primeramente mi trabajo a Dios.

A mi familia, particularmente a mi madre y mi abuela, pues este logro es un anhelo que siempre a estado en sus corazones.

Agradecimientos

*A Dios, por permitirme llegar a este momento de mi vida
y ser mi fortaleza para culminar este anhelado logro.*

*A mi Padre, por haberme forjado como la persona
que soy y haberme encaminado hacia este momento.*

*A mis amados hermanos, por el incondicional,
permanente e inmensurable apoyo.*

*A mi tutor, M. Sc. Aldo Valdez, por la colaboración
y predisposición y conocimiento compartido.*

*A mi Asesor, Lic. Miguel Toledo, por su importante
ayuda en el proceso de elaboración
del presente trabajo.*

RESUMEN

Tener los procesos automatizados de los servicios brindados y utilizar herramientas propias de las Tecnologías de la Información para presentar información de forma clara, sencilla y facilitar el diálogo razonable con los clientes genera mayor rentabilidad por cliente, información compartida a través de un centro de contactos, menor costo de adquisición, reducción de costos de ventas y el compromiso para negocios futuros de clientes rentables.

El presente trabajo de investigación es una aplicación Web que tiene como objetivo principal ofrecer a los empleados automatizar los procesos de registro y administración de servicios, gestión de usuarios y almacenamiento de información.

La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.”, empresa dedicada a servicios profesionales de contabilidad tributaria, teneduría de libros, y auditorías externas, asesoría en materia de impuestos a nivel nacional e internacional, sufre la deficiencia de no contar con un sistema informático el cual ayude a optimizar las tareas de controlar los trabajos recibidos en temporadas de alta demanda de servicios, Controlar pagos por servicio el cual se genera mucha desconfianza a momento de saldar cuentas con los clientes y por último la Administración y almacenamiento de documentos de clientes. En el desarrollo del sistema web se hace una combinación entre la metodología ágil Kanban y la metodología UWE, esta combinación es un aporte para futuros proyectos.

Kanban nos permite visualizar el trabajo en un tablero dividido en bloques, cada elemento del trabajo esta descrito en una tarjeta, utiliza columnas para ilustrar donde este cada elemento en el flujo de trabajo. Limita el trabajo en curso para realizar una cantidad de tareas.

UWE detalla el proceso de las aplicaciones, cuenta con cinco modelos que son el análisis de requerimientos, modelo de contenidos, modelo navegacional, modelo de presentación y modelo de procesos.

Palabras clave: aplicación web, Kanban, uwe, gestión de clientes

ABSTRACT

Having the automated processes of the services provided and using tools of Information Technology to present information in a clear, simple way and facilitate reasonable dialogue with customers generates greater profitability per customer, information shared through a contact center, lower acquisition cost, reduction of sales costs and the commitment to future business of profitable customers.

This research work is a Web application whose main objective is to offer employees to automate the processes of registration and administration of services, user management and information storage.

The Consultant Company “Contadores Públicos & Auditores AyS SRL”, a company dedicated to professional tax accounting, bookkeeping, and external auditing services, tax advice at national and international level, suffers from the lack of a computer system which helps to optimize the tasks of controlling the work received in seasons of high demand for services, Control payments for service which generates a lot of distrust when it comes to settle accounts with customers and finally the Administration and storage of customer documents. In the development of the web system a combination is made between the agile Kanban methodology and the UWE methodology, this combination is a contribution for future projects.

Kanban allows us to visualize the work on a board divided into blocks, each element of the work is described on a card, use columns to illustrate where each element is in the workflow. Limit work in progress to perform a number of tasks.

UWE details the application process, it has five models that are the analysis of requirements, content model, navigation model, presentation model and process model.

Keywords: web application, Kanban, uwe, customer management

ÍNDICE DE CONTENIDO

CAPÍTULO I	1
MARCO INTRODUCTORIO	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. ANTECEDENTES INSTITUCIONALES	2
1.2.1.1. MISIÓN	2
1.2.1.2. VISIÓN	3
1.3. ANTECEDENTES DE PROYECTOS SIMILARES	3
1.3.1. PROYECTO: “SISTEMA WEB PARA EL REGISTRO Y ADMINISTRACIÓN DE DOCUMENTOS DE NIÑOS PATROCINADOS CASO: CDI BO – 132”	3
1.3.2. PROYECTO: “SISTEMA WEB DE INSCRIPCIONES Y CONTROL DE PAGOS CASO: KUMON UNIDAD SEMILLA”	3
1.3.3. PROYECTO: “SISTEMA WEB DE CONTROL DE PAGOS, CITAS E HISTORIALES CLÍNICOS CASO: CLÍNICA DENTAL LAVADENT”	4
1.4. PLANTEAMIENTO DEL PROBLEMA	4
1.5. PROBLEMA CENTRAL	5
1.6. PROBLEMAS SECUNDARIOS	5
1.7. DEFINICIÓN DE OBJETIVOS	7
1.7.1. OBJETIVO GENERAL	7
1.7.2. OBJETIVOS ESPECÍFICOS	7
1.8. JUSTIFICACIÓN	7
1.8.1. JUSTIFICACIÓN ECONÓMICA	7
1.8.2. JUSTIFICACIÓN SOCIAL	8
1.8.3. JUSTIFICACIÓN TECNOLÓGICA	9
1.9. ALCANCES Y LÍMITES	10
1.9.1. ALCANCES	10

1.9.2. LÍMITES	10
1.10. APORTES	10
1.10.1. PRÁCTICO	10
1.10.2. TEÓRICO.....	11
1.11. METODOLOGÍA.....	11
CAPÍTULO II.....	13
MARCO TEÓRICO	13
2.1. INTRODUCCIÓN.....	13
2.2. INGENIERÍA DE SOFTWARE	13
2.3. METODOLOGÍAS DE DESARROLLO ÁGILES	14
2.4. METODOLOGÍA ÁGIL KANBAN	15
2.4.1. REGLAS DE KANBAN	16
2.4.1.1. REGLA NRO. 1: MOSTRAR EL PROCESO (VISUALIZAR)	16
2.4.1.2. REGLA NRO. 2: LIMITAR EL TRABAJO EN CURSO (WIP)	17
2.4.1.3. REGLA NRO. 3: OPTIMIZAR EL FLUJO DE TRABAJO	18
2.5. INGENIERÍA WEB BASADA EN UML (UWE).....	18
2.6. PATRONES DE DISEÑO.....	19
2.6.1. OBJETIVOS DE LOS PATRONES	20
2.7. PATRONES DE ARQUITECTURA	20
2.8. MVC (MODELO-VISTA-CONTROLADOR).....	21
2.9. COMPONENTES PRINCIPALES DE MVC.....	22
2.9.1. MODELO	22
2.9.2. VISTA	22
2.9.3. CONTROLADOR	23
2.10. CRM (<i>CUSTOMER RELATIONSHIP MANAGEMENT</i>).....	23
2.10.1. TIPOS DE CRM SOFTWARE	25
2.11. HERRAMIENTAS PARA LA IMPLEMENTACIÓN.....	25
2.11.1. PHP.....	25
2.11.2. JAVASCRIPT	26

2.11.3. CSS	27
2.11.4. HTML	27
2.11.5. MYSQL	28
2.11.6. CODEIGNITER	29
CAPÍTULO III	30
MARCO APLICATIVO.....	30
3.1. INTRODUCCIÓN.....	30
3.2. APLICACIÓN DE LAS METODOLOGÍAS KANBAN-UWE.....	30
3.3. ESPECIFICACIONES	32
3.4. RECOPIACIÓN DE REQUERIMIENTOS	32
3.5. APLICACIÓN DE REGLAS KANBAN	34
3.5.1. MOSTRAR EL PROCESO (VISUALIZAR)	34
3.5.2. LIMITANDO EL WIP (WORK IN PROGRESS).....	34
3.5.3. OPTIMIZAR EL FLUJO DE TRABAJO	35
3.5.4. ROLES KANBAN	35
3.6. MÓDULO DE ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES	36
3.6.1. FASE DE ANÁLISIS	36
3.6.1.1. ANÁLISIS DE REQUERIMIENTOS.....	36
3.6.1.2. CASOS DE USO	36
3.6.1.3. MODELOS DE CONTENIDOS	40
3.6.2. FASE DE DISEÑO	41
3.6.2.1. MODELO NAVEGACIONAL	41
3.6.2.2. MODELO DE PRESENTACIÓN.....	42
3.6.3. FASE DE DESARROLLO.....	44
3.6.3.1. MODELO DE FLUJO DE PROCESO	44
3.6.4. FASE DE IMPLEMENTACIÓN	45
3.7. MÓDULO DE CONTROL Y SEGUIMIENTO DE SERVICIOS	45
3.7.1. FASE DE ANÁLISIS	45
3.7.1.1. ANÁLISIS DE REQUERIMIENTOS.....	45

3.7.1.2. CASOS DE USO	46
3.7.1.3. MODELOS DE CONTENIDOS	48
3.7.2. FASE DE DISEÑO	49
3.7.2.1. MODELO NAVEGACIONAL	49
3.7.2.2. MODELO DE PRESENTACIÓN.....	50
3.7.3. FASE DE DESARROLLO.....	50
3.7.3.1. MODELO DE FLUJO DE PROCESO	50
3.7.4. FASE DE IMPLEMENTACIÓN	51
3.8. MÓDULO DE CONTROL Y SEGUIMIENTO DE PAGOS POR SERVICIOS	52
3.8.1. FASE DE ANÁLISIS.....	52
3.8.1.1. ANÁLISIS DE REQUERIMIENTOS.....	52
3.8.1.2. CASOS DE USO	52
3.8.1.3. MODELOS DE CONTENIDOS	54
3.8.2. FASE DE DISEÑO	55
3.8.2.1. MODELO NAVEGACIONAL	55
3.8.2.2. MODELO DE PRESENTACIÓN.....	56
3.8.3. FASE DE DESARROLLO.....	57
3.8.3.1. MODELO DE FLUJO DE PROCESO	57
3.8.4. FASE DE IMPLEMENTACIÓN	57
3.9. PRUEBA DE ESFUERZO	58
CAPÍTULO IV	60
CALIDAD Y SEGURIDAD	60
4.1. INTRODUCCIÓN.....	60
4.2. CALIDAD DE SOFTWARE	60
4.3. NORMA ISO 9126.....	61
4.4. METODOLOGÍA WEB QEM.....	61
4.4.1. FASES DE WEB QEM	61
4.4.1.1. DEFINICIÓN DE LAS METAS DE EVALUACIÓN Y SELECCIÓN DEL PERFIL DE USUARIO.....	61

4.4.1.2. DEFINICIÓN DE LOS REQUERIMIENTOS DE CALIDAD Y/O COSTO	62
4.4.1.3. DEFINICIÓN DE CRITERIOS BÁSICOS Y PROCEDIMIENTOS DE MEDICIÓN.....	62
4.4.1.4. DEFINICIÓN DE ESTRUCTURAS DE AGREGACIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN GLOBAL	62
4.4.1.5. ANÁLISIS DE RESULTADOS Y RECOMENDACIONES	63
4.4.2. DESARROLLO DE CARACTERÍSTICAS DE WEB QEM.....	63
4.4.2.1. FUNCIONALIDAD	64
4.4.2.2. CONFIABILIDAD	66
4.4.2.3. USABILIDAD.....	67
4.4.2.4. MANTENIBILIDAD	68
4.4.2.5. PORTABILIDAD.....	69
4.4.2.6. CALIDAD GLOBAL	69
4.5. SEGURIDAD	70
4.5.1. ALGORITMO DE ENCRIPCIÓN SHA-2	70
4.5.2. VARIANTES DEL SHA-2	71
4.5.3. CAPA DE CONEXIÓN SEGURA SSL	71
4.5.3.1. IMPLEMENTACIÓN DE CERTIFICADO SSL CON LET'S ENCRYPT Y CERBOT	72
CAPÍTULO V.....	74
COSTO BENEFICIO	74
5.1. INTRODUCCIÓN.....	74
5.2. COCOMO II.....	74
5.3. COSTO DEL SISTEMA	76
5.4. ESTIMACIÓN DEL COSTO.....	80
5.5. CÁLCULO BENEFICIO VAN Y TIR	84
5.5.1. VALOR ACTUAL NETO (VAN)	84
5.5.2. TASA INTERNA DE RETORNO (TIR).....	85
5.6. COSTO BENEFICIO	86

CAPÍTULO VI	88
CONCLUSIONES Y RECOMENDACIONES	88
6.1. CONCLUSIONES.....	88
6.2. RECOMENDACIONES	89
BIBLIOGRAFÍA	90
ANEXOS	93
ANEXO A – ÁRBOL DE PROBLEMAS	93
ANEXO B – ÁRBOL DE OBJETIVOS	94
ANEXO C – MARCO LÓGICO.....	95
ANEXO D – CRONOGRAMA DE ACTIVIDADES	96

ÍNDICE DE TABLAS

Tabla 1: Tabla de Problema-Causa-Efecto.....	6
Tabla 2: Patrones de Diseño	22
Tabla 3: Lista de requerimientos	33
Tabla 4: Tabla de roles	35
Tabla 5: Especificaciones del Administrador de usuarios.....	37
Tabla 6: Especificaciones de la gestión de clientes.....	39
Tabla 7: Especificaciones del control y seguimiento de servicios	47
Tabla 8: Especificaciones del control de pagos y reportes.....	53
Tabla 9: Preguntas para obtener el grado de usabilidad.....	67
Tabla 10: Evaluación de mantenibilidad	68
Tabla 11: Calidad global.....	70
Tabla 12: Tabla de modos de desarrollo.....	75
Tabla 13: Puntos fusión COCOMO II.....	77
Tabla 14: Tabla modo de conductores de costo	79
Tabla 15: Costo total desarrollo de software.....	83
Tabla 16: Flujo de caja por años.....	84

ÍNDICE DE FIGURAS

Figura 1: Tablero ficticio de Kanban	17
Figura 2: Modelo Vista Controlador	23
Figura 3: Tabla Kanban-UWE	31
Figura 4: Tabla Kanban-UWE	34
Figura 5: Caso de uso para administración de roles de usuario	36
Figura 6 : Casos de uso de la gestión de Clientes.....	38
Figura 7: Modelo de Contenidos de administración de usuarios y gestión de clientes.....	40
Figura 8: Modelo Navegacional de la Administración de Usuarios y Gestión de Clientes	41
Figura 9: Modelo de Presentación del Administrador de usuarios	42
Figura 10: Modelo de presentación para editar y guardar usuarios	43
Figura 11: Modelo de presentación para listar y desplegar la información	43
Figura 12: modelo de presentación para registrar y editar clientes	44
Figura 13: Modelo de flujo de proceso del módulo de usuarios y clientes.....	44
Figura 14: Captura de la interfaz de Administrador de usuarios.....	45
Figura 15: Caso de uso para la gestión de servicios.....	46
Figura 16: Modelo de Contenidos del control y seguimiento de servicios	48
Figura 17: Modelo Navegacional del control y seguimiento de servicios	49
Figura 18: Modelo de Presentación del control y seguimiento de servicios.....	50
Figura 19: Modelo de flujo del control y seguimiento de servicios.....	51
Figura 20: Captura del entorno grafico del control y seguimiento de servicios	51
Figura 21: Caso de uso para control de pagos y reportes.....	52
Figura 22: Modelo de Contenidos del control de pagos y reportes.....	54
Figura 23: Modelo Navegacional del control de pagos y generación de reportes.....	55
Figura 24: Modelo de Presentación del control de pagos	56
Figura 25: Modelo de Presentación del entorno de generación de reportes.....	56
Figura 26: Modelo de flujo del control de pagos y reportes	57
Figura 27: Modelo de flujo del control de pagos y reportes	58

Figura 28: Resultado las pruebas de esfuerzo	59
Figura 29: Resultado de la prueba con 100 usuarios en cada página.....	59
Figura 30: Fases de WebQEM	63
Figura 31: Factor de ajuste de complejidad.	64
Figura 32: Certificación <i>Let's Encrypt</i> activa.....	73
Figura 33: Líneas de código software COCOMO.....	80
Figura 34: Estimación de costo software COCOMO	81
Figura 35: Punto fusión software COCOMO.....	82
Figura 36: Estimación de costo software COCOMO	82
Figura 37: Factores de escala software COCOMO	83

CAPÍTULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

El poderoso auge de las TIC ha cambiado los paradigmas y estrategias reconocidas y establecidas por muchos años como válidas. Dentro de las TIC, la industria del software alcanza una posición relevante, por su característica de controlar o hacer accesible, en la mayoría de los casos, los adelantos electrónicos. Por citar un ejemplo podemos analizar los modernos sistemas web que permiten que el trabajo a distancia se realice con mayor facilidad en dispositivos requieren de un software que proporcione la interfaz que las personas necesitan para centralizar distintas áreas de trabajo.

Una aplicación web es una aplicación o herramienta informática accesible desde cualquier navegador, bien sea a través de internet (lo habitual) o bien a través de una red local. A través del navegador se puede acceder a toda la funcionalidad (Mateo, 2018).

Bolivia ha comenzado a incursionar en esta industria del desarrollo de software hace aproximadamente diez años y hoy es uno de los países sudamericanos que oferta este producto a costos más reducidos que otros. Los primeros años fueron de prueba para muchas empresas bolivianas, y sobrevivieron aquellas que sobre todo se enfocaron para atender a grandes empresas del mercado local (Cueto, 2009).

Las regiones que destacan más emprendimientos en este rubro corresponden al eje troncal: La Paz, Cochabamba y Santa Cruz, acompañados en los últimos años de Potosí y Sucre como resultado de la demanda de producción de software (Suaznabar, 2017).

La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.”, empresa dedicada a servicios profesionales de contabilidad tributaria, teneduría de libros, y auditorías externas,

asesoría en materia de impuestos a nivel nacional e internacional, sufre la deficiencia de no contar con un sistema informático el cual ayude a optimizar las tareas de controlar los trabajos recibidos en temporadas de alta demanda de servicios, Controlar pagos por servicio el cual se genera mucha desconfianza a momento de saldar cuentas con los clientes y por último la Administración y almacenamiento de documentos de clientes. La pregunta central del trabajo es ¿de qué manera se puede mejorar el control de trabajos, pagos y administración de documentación de clientes para La empresa Contable “Contadores Públicos & Auditores AyS S.R.L.”?

El Objetivo central es el de desarrollar un Sistema Web para el control de trabajos, pagos y administración de documentación de clientes para La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.”.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES INSTITUCIONALES

La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.”, dedicada a brindar servicios profesionales de contabilidad tributaria, teneduría de libros, y auditorías externas, asesoría en materia de impuestos a nivel nacional e internacional, empezó su arduo trabajo empresarial un 18 de agosto de 2017, en inmediaciones de su oficina central ubicada en la avenida 20 de octubre nro. 2144, edificio Europa en la zona de Sopocachi.

1.2.1.1. MISIÓN

Contribuir al éxito de las empresas a través de servicios profesionales de contabilidad tributaria, teneduría de libros, y auditorías externas, asesoría en materia de impuestos a nivel nacional e internacional. Ser una empresa avocada a múltiples disciplinas que se encuentra mejorando sus procedimientos, con el objeto de anticiparse a las necesidades de todos los clientes, sin descuidar el manejo profesional en la solución de sus problemas.

1.2.1.2. VISIÓN

Como firma de auditoría, contabilidad y consultoría, es lograr destacarse entre las firmas de auditoría más importantes del departamento de La Paz. Consolidarse como una firma de Auditores y Consultores especialistas de alta competitividad, con clientes satisfechos y con colaboradores comprometidos y en permanente proceso de capacitación.

1.3. ANTECEDENTES DE PROYECTOS SIMILARES

En inmediaciones de la biblioteca de la carrera de Informática en la Universidad Mayor de San Andrés (UMSA), se hallaron como referencia proyectos similares, los cuales se detallan a continuación:

1.3.1. PROYECTO: “SISTEMA WEB PARA EL REGISTRO Y ADMINISTRACIÓN DE DOCUMENTOS DE NIÑOS PATROCINADOS CASO: CDI BO – 132”

El proyecto fue desarrollado en el Centro de Desarrollo Integral de Bolivia Bo-132 en área de patrocinio, que realiza los procesos de registro de postulante, registro de patrocinado, control de documentación requerida, las actualizaciones de la información registrada.

El área de patrocinio no cuenta con un sistema de información automático es decir todo lo realizan manualmente lo que ocasiona volúmenes de papelería con información. Por lo mencionado anteriormente se desarrolló un Sistema Web para el registro y administración de documentos de niños patrocinados CDI BO-132, optimizando el trabajo en el tiempo de procesos y además llevar un control adecuado de la información. (Cortez Cusi, 2014).

1.3.2. PROYECTO: “SISTEMA WEB DE INSCRIPCIONES Y CONTROL DE PAGOS CASO: KUMON UNIDAD SEMILLA”

El proyecto tiene como finalidad facilitar el proceso de inscripciones y control de pago de mensualidades mediante la implementación de un Sistema Web, el cual permitirá la automatización de dichos procesos.

Para el desarrollo del proyecto se aplicó la metodología de desarrollo ágil SCRUM apoyándose en la metodología de desarrollo UWE para el modelado del diseño.

El Sistema es un producto de calidad de acuerdo a la metodología de evaluación de calidad de sitios web llamada Web-site QEM (Condori Mendoza, 2015).

1.3.3. PROYECTO: “SISTEMA WEB DE CONTROL DE PAGOS, CITAS E HISTORIALES CLÍNICOS CASO: CLÍNICA DENTAL LAVADENT”

Este proyecto tiene como finalidad el control de pagos, citas e historias clínicas que son los tratamientos que realiza la clínica dental a cada paciente.

La metodología utilizada para el análisis, diseño y desarrollo del sistema es la metodología SCRUM, conjuntamente con la metodología UWE para el modelado del entorno web. En cuanto al desarrollo del sistema, se utilizó el framework CodeIgniter. Finalmente se puede concluir que el sistema desarrollado, por medio de la integración de los módulos planteados, puede obtenerse información clave para brindar servicios a los clientes de acuerdo a las necesidades y preferencias hacia los tratamientos estéticos que ofrece la empresa, dejando a los clientes satisfechos (Huanca Cantuta, 2015).

1.4. PLANTEAMIENTO DEL PROBLEMA

En la actualidad las empresas cualquiera sea su tamaño (micro, pequeñas, medianas o grandes), tienen la necesidad de adoptar tecnologías de punta, las cuales permitan que las empresas tengan mayor alcance, faciliten el trabajo que realizan y logren llegar a nuevos mercados.

En el mercado empresarial de Bolivia, se observa que las grandes empresas se apoyan en sistemas Informáticos para realizar sus tareas de manera eficiente. La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.” sufre la deficiencia de no contar con un sistema informático el cual ayude a optimizar las tareas de:

- ✓ En temporadas de alta demanda de servicios, se genera un conflicto con la organización de los trabajos pendientes, donde es necesario saber los detalles de dichos trabajos como ser el avance en el que se encuentra o es estado en el que este se encuentra, de tal manera que el encargado pueda hacer seguimiento a cada uno de los trabajos pendientes y que estos hayan sido concluidos exitosamente en la fecha comprometida.

- ✓ Es difícil controlar los pagos por servicios brindados, y mantener la sumatoria de ingresos, gastos, y cuentas por cobrar, problema por el cual se genera mucha desconfianza a momento de saldar cuentas con los clientes, ya que en ese momento no existe un registro seguro y confiable acerca de los adelantos y detalle de servicios brindados a los clientes que realizaron a determinados empleados de la empresa.

- ✓ La información o documentación recibida y generada no se encuentra centralizada, ordenada no almacenada correctamente para su consulta, una importante deficiencia encontrada en la institución, la cual se genera a causa de que en el proceso de brindarle los servicios a los clientes se genera documentos digitales sumamente importantes los cuales no están siendo guardados de tal manera de que estos estén seguros y correctamente ordenados, para acceder a ellos.

1.5. PROBLEMA CENTRAL

¿De qué manera se puede mejorar el control y seguimiento de trabajos y pagos por servicios, además del almacenamiento de documentación y la gestión de clientes de la empresa consultora “Contadores Públicos & Auditores AyS S.R.L.”?

1.6. PROBLEMAS SECUNDARIOS

Los problemas encontrados dentro en el proceso de brindar un adecuado servicio al usuario en la empresa AyS S.R.L. son los siguientes:

	PROBLEMA	CAUSA	EFEECTO
1) Control de trabajos recepcionados	Se tiene un registro bastante extenso en documento Excel el cual se encuentra desordenado y pesado.	La cantidad de trabajos recepcionados en temporadas de alta demanda es alta.	Desorden y no es posible hacer el seguimiento al avance de los trabajos.
2) Control y seguimiento de pagos por servicios	No existe un registro centralizado de los pagos realizados por los clientes y muchas veces los ingresos de la empresa no cuadran.	Los pagos por servicios de clientes se realizan con adelantos y saldos pendientes, los cuales se cancelan a medida que el usuario recibe el servicio.	Susceptibilidad a momento cobrar saldos al cliente cuando los ingresos no cuadran.
3) Administración y Almacenamiento de archivos y documentación generada	No se cuenta con alguna herramienta que organice la documentación final del servicio para el usuario	Cuando la empresa termina de brindar el servicio al cliente, se genera documentación (producto) la cual debe estar lista y organizada a momento de la entrega y guardada para las posteriores gestiones.	Perdida de documentación, o realizar el trabajo nuevamente
4) Reportes acerca de la información de trabajos realizados a los clientes	La información de los trabajos realizados (historial) para cada cliente en el transcurso de los años no está centralizada.	Cuando se brinda servicios a los clientes es necesario el detalle de los servicios realizados en gestiones pasadas como referencia.	Pérdida de tiempo al buscar información del cliente.

Tabla 1: Tabla de Problema-Causa-Efecto

Fuente: (Elaboración propia, 2019)

1.7. DEFINICIÓN DE OBJETIVOS

1.7.1. OBJETIVO GENERAL

Desarrollar un sistema web de control y seguimiento de servicios y gestión de clientes para La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.”

1.7.2. OBJETIVOS ESPECÍFICOS

- ✓ Realizar el control y seguimiento de trabajos recepcionados (servicios), donde el usuario ingresará los trabajos recibidos y podrá hacer seguimiento acerca del avance y también registrar observaciones.
- ✓ Realizar el control y seguimiento de pagos por servicios, donde se registrará cualquier pago que el cliente realice y donde se podrá verificar el saldo a cobrar a momento de terminar el servicio.
- ✓ Almacenar documentos en un repositorio de archivos clasificados por clientes, donde los archivos se encuentren seguros y se registre cualquier observación acerca de cada uno de estos. Facilitar la administración de documentación la cual se genera cuando el servicio al usuario culminó, el cual sería el producto a entregar y la documentación estará disponible al igual que los de las gestiones anteriores para su uso.
- ✓ Generar reportes sobre la información de los clientes, se almacenará toda la información requerida de los clientes que ya trabajaron con la empresa, generando un historial de servicios brindados al usuario.

1.8. JUSTIFICACIÓN

1.8.1. JUSTIFICACIÓN ECONÓMICA

El sistema Web permitirá que la empresa “Contadores Públicos & Auditores AyS S.R.L.” reduzca sus pérdidas económicas que resultan de la falta de automatización en los procesos

de control y administración. Actualmente la empresa cuenta con 3 personas encargadas del control de trabajos decepcionados y su correspondiente seguimiento de forma manual y repetitiva, pero con el sistema será suficiente con una persona la cual interactúe con el sistema, por ende, tendrá una significativa reducción en los pagos de sueldos a personal.

La implementación del sistema web reducirá gastos realizados por la empresa en cuanto a discos ópticos de almacenamiento CD's, los cuales ya no serán necesarios pues estarán debidamente almacenados en el sistema de forma confiable.

El usuario del sistema tendrá la información, documentación y la administración de los trabajos las 24 horas al día desde cualquier ubicación al instante, evitando los gastos de transporte, cuando se requiere información que se encuentran en otras sucursales de la empresa. De la misma manera reducirá gastos en impresiones pues actualmente todos los documentos generados por el servicio son impresos para el archivo de la empresa, pero el sistema web almacenara toda la documentación en documentos digitales.

En todas las tareas que actualmente están involucradas en la implementación del sistema web, existe demora a la hora a causa de organización y control, pero el sistema ayudara a que los usuarios logren obtener más resultados en menos tiempo.

1.8.2. JUSTIFICACIÓN SOCIAL

El sistema web tendrá un gran impacto entre la empresa y los clientes pues será de gran aporte para que el cliente reciba un servicio óptimo y eficiente, como es común mientras más rápido sea atendido el cliente en cuanto al servicio, solicitado más satisfecho quedara, siendo el tiempo un factor determinante, esto sin descuidar la calidad del servicio.

De esta manera la empresa brindara servicios de manera eficaz, fácil, confiable y dinámica, donde tanto el cliente como los usuarios de la empresa facilitarán sus actividades, las cuales estarán al alcance de un celular, computadora o algún dispositivo con acceso a internet.

1.8.3. JUSTIFICACIÓN TECNOLÓGICA

La empresa cuenta con el equipamiento necesario para la implementación del sistema pues actualmente cuenta con un servidor con las características necesarias el cual solo aloja a un sistema web de publicación y promoción de servicios.

También cuenta con un cableado estructurado, servicio de internet de alta velocidad y equipos de última generación donde todos cuentan con navegadores correctamente actualizados y conectados a la red.

- Características técnicas del hardware del servidor
 - ✓ Marca: Dell.
 - ✓ Modelo: PowerEdge 2800.
 - ✓ Procesador: Intel®Xeon™.
 - ✓ Memoria ram: 12 GB DDR2.
 - ✓ Disco duro: 2 TB.

- Requerimientos del software en el servidor web
 - ✓ Servidor http: Apache.
 - ✓ Gestor de base de datos: MySQL.
 - ✓ Lenguaje de programación: PHP.

- Requerimientos en equipos de usuarios
 - ✓ Procesadores core2 dúo o superior.
 - ✓ Memoria RAM de 4 gigas o superior.
 - ✓ velocidad de internet superior a los 2 Mb/s con conexión constante.
 - ✓ Navegadores web como ser Google Chrome, Mozilla Firefox entre otros.

1.9. ALCANCES Y LÍMITES

1.9.1. ALCANCES

Para lograr cumplir los objetivos planteados se desarrollarán los siguientes módulos:

- ✓ Módulo de administración de usuarios y gestión de clientes/empleados, donde se tendrá toda la información sobre los servicios brindados al cliente y la asignación de trabajos a los empleados.
- ✓ Módulo de control y seguimiento de servicios, donde se controlará la recepción, estado de avance, conclusión u observaciones de los trabajos pendientes (servicios) y almacenamiento de documentación de clientes con respecto al servicio brindado.
- ✓ Módulo de Control y seguimiento de pagos por servicios, donde se controlará los pagos que el cliente realice por el servicio hasta su pago total.
- ✓ Módulo de información centralizada y reportes, donde se encontrará toda la información de los servicios y se podrán generar reportes de diferentes gestiones.

1.9.2. LÍMITES

- ✓ El sistema web será implementado en la sucursal central de la Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.” ubicado en la zona de Sopocachi.
- ✓ El Sistema Web no emitirá, ni realizará la tarea de impresión de facturas.
- ✓ No efectuará el control del personal.
- ✓ El sistema web solo estará disponible para los servicios de contabilidad tributaria, teneduría de libros, y auditorías externas y asesoría en materia de impuestos a nivel nacional e internacional.

1.10. APORTES

1.10.1. PRÁCTICO

El sistema web, será de mucha utilidad para la empresa Contable “Contadores Públicos & Auditores AyS S.R.L.” porque brindará información segura a los usuarios y clientes,

logrando tener un correcto control y seguimiento a todas las actividades realizadas en el proceso de brindar un óptimo servicio al cliente.

1.10.2. TEÓRICO

El presente proyecto claramente será una referencia para el desarrollo de proyectos orientados al desarrollo web pues en el sistema se aplicará la metodología Kanban el cual tiene la cualidad de ser flexible, especialmente en la entrada de tareas, así como en el seguimiento de estas, la priorización, la supervisión del equipo de trabajo y los informes de dedicación.

En cuanto al modelado y análisis del sistema se empleará UML basado en Ingeniería Web (UWE).

1.11. METODOLOGÍA

Se aplicará el Método Científico, el cual es un método de estudio sistemático de la naturaleza que incluye las técnicas de observación, reglas para el razonamiento y la predicción, ideas sobre la experimentación planificada y los modos de comunicar los resultados experimentales y teóricos.

Los aspectos de la investigación:

- ✓ La investigación exploratoria: que es considerada como el primer acercamiento científico a un problema. Se utiliza cuando este aún no ha sido abordado o no ha sido suficientemente estudiado y las condiciones existentes no son aún determinantes; para luego pasar a la Investigación.
- ✓ Descriptiva: que se efectúa cuando se desea describir, en todos sus componentes principales, una realidad.

En el proceso de desarrollo del sistema se aplicará la metodología **Kanban**, la cual pertenece a la categoría de metodologías ágiles, aquellas que buscan gestionar de manera generalizada cómo se van completando las tareas. Kanban es una palabra japonesa que se compone de dos partes:

Kan, que significa visual, y Ban, que hace referencia a tarjeta, de modo que como podemos deducir la metodología utiliza tarjetas para gestionar, de manera visual, la realización de determinados procesos y tareas.

Hay dos objetivos que rigen este método productivo: por un lado, lograr un producto de calidad, al obligar a cada fase del proyecto a finalizar su tarea correctamente, y acabar con el caos, saturación o cuello de botella que puede darse en una fase del proyecto en condiciones normales en las que prima la rapidez por encima de la calidad del producto.

En cuanto al modelado y análisis del sistema utilizaremos UWE el cual nos facilitara el modelado de la aplicación web, El principal objetivo del enfoque UWE es proporcionar: un lenguaje de modelado específico del dominio basado en UML; una metodología dirigida por modelos; herramientas de soporte para el diseño sistemático; y herramientas de soporte para la generación semi-automática de Aplicaciones Web.

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos, diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación.

CAPÍTULO II

MARCO TEÓRICO

2.1. INTRODUCCIÓN

En el desarrollo de este capítulo se describe la teoría sobre las metodologías, técnicas y herramientas que se utiliza para el desarrollo del sistema web, con el fin de tener una referencia sólida para la comprensión correcta de la documentación.

2.2. INGENIERÍA DE SOFTWARE

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, el estudio de las aplicaciones de la ingeniería al software. Integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.

Se citan las definiciones más reconocidas, formuladas por prestigiosos autores:

“Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software” (Zelkovitz, 1978).

“Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software” (Bohem, 1976).

“La ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales” (Bauer, 1972).

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software.

En 2004, la U. S. *Bureau of Labor Statistics* (Oficina de Estadísticas del Trabajo de Estados Unidos) contó 760 840 ingenieros de software de computadora.

El término Ingeniero de Software, sin embargo, se utiliza de manera genérica en el ambiente empresarial, y no todos los que se desempeñan en el puesto de ingeniero de software poseen realmente títulos de ingeniería de universidades reconocidas.

Algunos autores consideran que Desarrollo de Software es un término más apropiado que Ingeniería de Software para el proceso de crear software. Personas como Pete McBreen (autor de *Software Craftmanship*) cree que el término IS implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software.

Indistintamente se utilizan los términos Ingeniería de Software o Ingeniería del Software; aunque menos común también se suele referenciar como Ingeniería en Software. En Hispanoamérica los términos más comúnmente usados son los dos primeros.

La creación del software es un proceso intrínsecamente creativo y la ingeniería del software trata de sistematizar este proceso con el fin de acotar el riesgo de fracaso en la consecución del objetivo, por medio de diversas técnicas que se han demostrado adecuadas sobre la base de la experiencia previa.

2.3. METODOLOGÍAS DE DESARROLLO ÁGILES

Las metodologías de desarrollo ágiles nacen en respuesta a los problemas de las metodologías tradicionales y se basa en dos aspectos, el retrasar las decisiones y la planificación adaptativa; potenciando más el desarrollo del software a gran escala (Cabrera, Figueroa, y Solís, 2008).

Como resultado de esta nueva teoría se crea un Manifiesto Ágil cuyos principios son:

- ✓ Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- ✓ Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja al cliente.
- ✓ Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- ✓ Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- ✓ La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
- ✓ Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- ✓ El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- ✓ El software funcionando es la medida principal de progreso.
- ✓ Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- ✓ La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- ✓ Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
- ✓ A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

2.4. METODOLOGÍA ÁGIL KANBAN

Kanban se basa en un sistema de producción que dispara trabajo solo cuando existe capacidad para procesarlo. El disparador de trabajo es representado por tarjetas Kanban de las cuales se dispone de una cantidad limitada.

Cada tarjeta Kanban acompaña a un ítem de trabajo durante todo el proceso de producción, hasta que éste, es empujado fuera del sistema, liberando una tarjeta. Un nuevo ítem de trabajo, solo podrá ser ingresado/aceptado si se dispone de una tarjeta Kanban libre.

Este proceso de producción, donde un trabajo se introduce al sistema solo cuando existe disponibilidad para procesarlo, se denomina pull (tirar) en contrapartida al mecanismo push (empujar), donde el trabajo se introduce en función de la demanda. En el desarrollo de Software, Kanban fue introducido por David Anderson de la Unidad de Negocios XIT de Microsoft, en 2004, reemplazando el sistema de tarjetas por un tablero visual similar al de Scrum, pero con características extendidas que veremos a continuación.

2.4.1. REGLAS DE KANBAN

Kanban demuestra ser una de las metodologías adaptativas que menos resistencia al cambio presenta. Dichas reglas son:

2.4.1.1. REGLA NRO. 1: MOSTRAR EL PROCESO (VISUALIZAR)

Consiste en la visualización de todo el proceso de desarrollo, mediante un tablero físico, generalmente, públicamente asequible. El objetivo de mostrar el proceso, consiste en:

- ✓ Entender mejor el proceso de trabajo actual.
- ✓ Conocer los problemas que puedan surgir y tomar decisiones.
- ✓ Mejorar la comunicación entre todos los interesados/participantes del proyecto.
- ✓ Hacer los futuros procesos más predecibles.

Un tablero Kanban, se divide en columnas las cuales representan un proceso de trabajo. Un ejemplo clásico de columnas para dividir un tablero Kanban, sería el siguiente:

Cola de entrada | Análisis | Desarrollo | Test | *Deploy* | Producción

La cantidad y nombre de las columnas, varía de acuerdo a las necesidades de cada equipo y en la mayoría de los casos, estas son subdivididas en dos columnas: cola de espera y en curso.

2.4.1.2. REGLA NRO. 2: LIMITAR EL TRABAJO EN CURSO (WIP)

Los límites del WIP (work in progress- trabajo en curso) consisten en acordar anticipadamente, la cantidad de ítems que pueden abordarse por cada proceso (es decir, por columnas del tablero).

El principal objetivo de establecer estos límites, es el de detectar cuellos de botella.

Los cuellos de botella representan el estancamiento de un proceso determinado. Viendo el siguiente tablero ficticio (Figura 1), se puede comprender mejor:



Figura 1: Tablero ficticio de Kanban

Fuente: (Bahit,2011)

En el tablero anterior, podemos visualizar claramente, que en la columna **pruebas** se produce un cuello de botella, pues el límite WIP está cubierto, mientras que el proceso siguiente (*deploy*), está totalmente libre. Esto, claramente marca un problema a resolver en el proceso correspondiente a las pruebas.

Es un valor a tener en cuenta, que la resolución de cuellos de botella, la mayoría de las veces, motiva la colaboración del equipo entre los diferentes procesos. Pues mientras existen procesos colapsados, existen a la vez, procesos libres para aceptar nuevos ítems. El cuello de botella ha generado un estancamiento, y los procesos libres, pueden ayudar a desestancar a los procesos colapsados.

2.4.1.3. REGLA NRO. 3: OPTIMIZAR EL FLUJO DE TRABAJO

El objetivo una la producción estable, continua y previsible. Midiendo el tiempo que el ciclo completo de ejecución del proyecto demanda (por ejemplo, cantidad de días desde el inicio del análisis hasta el fin del *deploy*, según el ejemplo del tablero anterior), se obtiene el *CycleTime*.

Al dividir, el *CycleTime* por el WIP, se obtiene el rendimiento de trabajo, denominado *Throughput*, es decir, la cantidad de ítems que un equipo puede terminar en un determinado período de tiempo.

$$\textit{Throughput} = \textit{CycleTime}/\textit{WIP}$$

Con estos valores, la optimización del flujo de trabajo consistirá en la búsqueda de:

- ✓ Minimizar el *CycleTime*
- ✓ Maximizar el *Throughput*
- ✓ Lograr una variabilidad mínima entre *CycleTime* y *Throughput*.

2.5. INGENIERÍA WEB BASADA EN UML (UWE)

La Ingeniería Web basada en UML (UWE, por sus siglas en inglés UML Based Web Engineering) es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado.

Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Otras características relevantes del proceso y método de autoría de UWE son el uso del paradigma orientado a objetos, su orientación al usuario, la definición de un metamodelo (modelo de referencia) que da soporte al método y el grado de formalismo que alcanza debido al soporte que proporciona para la definición de restricciones sobre los modelos.

La metodología UWE define vistas especiales representadas gráficamente por diagrama en UML. Los principales aspectos en los que se fundamenta UWE son los siguientes:

- ✓ Notación estándar: el uso de la metodología UML para todos los modelos.
- ✓ Métodos definidos: pasos definidos para la construcción de cada modelo.
- ✓ Especificación de restricciones: recomendables de manera escrita, para que la exactitud en cada modelo aumente.

2.6. PATRONES DE DISEÑO

Los patrones de diseño son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es

que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

2.6.1. OBJETIVOS DE LOS PATRONES

Los patrones de diseño pretenden:

- ✓ Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- ✓ Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- ✓ Formalizar un vocabulario común entre diseñadores y desarrolladores.
- ✓ Estandarizar el modo en que se realiza el diseño.
- ✓ Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Asimismo, no pretenden:

- ✓ Imponer ciertas alternativas de diseño frente a otras.
- ✓ Eliminar la creatividad inherente al proceso de diseño.

No es obligatorio utilizar los patrones, solo es aconsejable en el caso de tener el mismo problema o similar que soluciona el patrón, siempre teniendo en cuenta que en un caso particular puede no ser aplicable. "Abusar o forzar el uso de los patrones puede ser un error".

2.7. PATRONES DE ARQUITECTURA

Los patrones arquitectónicos, o patrones de arquitectura, también llamados arquetipos ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas,

sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor.

Aunque un patrón arquitectónico comunica una imagen de un sistema, no es una arquitectura como tal. Un patrón arquitectónico es más un concepto que captura elementos esenciales de una arquitectura de software.

Muchas arquitecturas diferentes pueden implementar el mismo patrón y por lo tanto compartir las mismas características. Además, los patrones son a menudo definidos como una cosa estrictamente descrita y comúnmente disponible. Por ejemplo, la arquitectura en capas es un estilo de llamamiento-y-regreso, cuando define uno un estilo general para interactuar. Cuando esto es descrito estrictamente y comúnmente disponible, es un patrón.

Uno de los aspectos más importantes de los patrones arquitectónicos es que encarnan diferentes atributos de calidad. Por ejemplo, algunos patrones representan soluciones a problemas de rendimiento y otros pueden ser utilizados con éxito en sistemas de alta disponibilidad. A primeros de la fase de diseño, un arquitecto de software escoge qué patrones arquitectónicos mejor ofrecen las calidades deseadas para el sistema.

2.8. MVC (MODELO-VISTA-CONTROLADOR)

El patrón de diseño MVC de las siglas en inglés (*Model-Views-Controllers*) fue propuesto en 1994 por cuatro autores Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en su libro titulado "*Design Patterns - Elements of Reusable Object-Oriented Software*."

Inicialmente al desarrollar aplicaciones web creamos código espagueti como varios lo han denominado, la cual consiste en la combinación de código HTML, PHP y JAVASCRIPT en un mismo archivo, lo cual tiene muchos inconvenientes al realizar modificaciones u actualizaciones en la forma como esta visualizado nuestro contenido debido a que debemos

ir archivo por archivo realizando los ajustes necesarios. En respuesta a este inconveniente se emplea un modelo MVC.

El patrón MVC está basado al desarrollo Orientado a Objetos donde destacan actualmente 23 Modelos MVC que se encuentran clasificados en 3 categorías principales: Estructural, Creacional y Comportamiento.

CATEGORÍA	DESCRIPCIÓN
PATRÓN CREACIONAL	Este patrón de diseño proporciona al programador crear instancias de objetos de una forma más flexible y de esta forma administre que tipo de objetos deben crearse para cada caso.
PATRÓN ESTRUCTURAL	Su diseño esta principalmente basado en la herencia aplicada a la creación de interfaces para la creación de nuevos objetos y obtener nuevas funcionalidades
PATRÓN DE COMPORTAMIENTO	Se caracteriza por ser un patrón de diseño que permite la comunicación entre objetos.

Tabla 2: Patrones de Diseño

Fuente: (Elaboración propia, 2019)

2.9. COMPONENTES PRINCIPALES DE MVC

2.9.1. MODELO

Es la capa donde alojaremos y manipularemos nuestra base de datos la cual contendrá toda la información que generemos en la aplicación y para lo cual podremos acceder a la información a través de las sentencias básicas de SQL como son: *selects*, *updates*, *inserts*, *delete*, etc.

2.9.2. VISTA

Las vistas generalmente creadas en HTML y CSS constarán de interfaces con las cuales el usuario final interactúa, realiza las peticiones al sistema e introduce los datos que posteriormente serán almacenados en el modelo.

Es por ello que podemos denominar a la vista con un objeto de entrada y salida de datos, sin embargo, cabe mencionar que a pesar de existir una dependencia entre la vista y el modelo estos no trabajan directamente necesitan de un intermediario que es el controlador.

2.9.3. CONTROLADOR

Es la capa que se encarga de ser el intermediario entre la vista y el modelo, en esta capa se llevan a cabo todas las operaciones funcionales de la aplicación es decir aquí se llevaran a cabo todas las instrucciones llámense buscar información en el modelo, visualizar dicha información o cualquier otra acción que le soliciten al sistema.

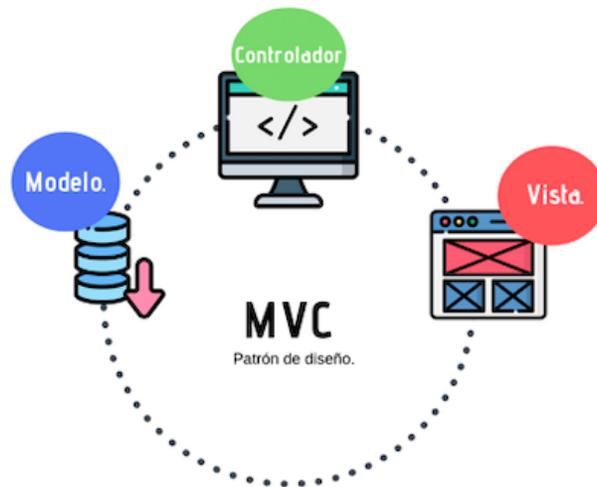


Figura 2: Modelo Vista Controlador

Fuente: (García, 2019)

2.10. CRM (*CUSTOMER RELATIONSHIP MANAGEMENT*)

La *Customer Relationship Management*, más conocida por sus siglas CRM, puede tener varios significados:

- ✓ Administración basada en la relación con los clientes, un modelo de gestión de toda la organización, basada en la satisfacción del cliente (u orientación al mercado según otros autores). El concepto más cercano es marketing relacional (según se usa en

España) y tiene mucha relación con otros conceptos como: *clienting*, *marketing 1x1*, *marketing* directo de base de datos, etc.

- ✓ Software para la administración de la relación con los clientes. Sistemas informáticos de apoyo a la gestión de las relaciones con los clientes, a la venta y al marketing,² y que se integran en los llamados Sistemas de Gestión Empresarial (SGE), y que incluyen CRM, ERP, PLM, SCM y SRM. El software de CRM puede comprender varias funcionalidades para gestionar las ventas y los clientes de la empresa: automatización y promoción de ventas, tecnologías *data warehouse* («almacén de datos») para agregar la información transaccional y proporcionar capa de *reporting*, *dashboards* e indicadores claves de negocio, funcionalidades para seguimiento de campañas de marketing y gestión de oportunidades de negocio, capacidades predictivas y de proyección de ventas.

Customer relationship management (CRM) es un enfoque para gestionar la interacción de una empresa con sus clientes actuales y potenciales. Utiliza el análisis de datos de la historia de los clientes con la empresa y para mejorar las relaciones comerciales con dichos clientes, centrándose específicamente en la retención de los mismos y, en última instancia, impulsando el crecimiento de las ventas.

Un aspecto importante del enfoque de CRM son los sistemas informáticos de CRM que recopilan datos de una variedad de canales de comunicación diferentes, incluidos el sitio web, el teléfono, el correo electrónico, el chat en vivo, los materiales de marketing y, más recientemente, las redes sociales de la compañía. A través del enfoque de CRM y los sistemas utilizados para facilitarlos, las empresas aprenden más sobre sus audiencias objetivo y cómo atender mejor sus necesidades.

Sin embargo, la adopción del enfoque de CRM también puede ocasionalmente generar favoritismo entre una audiencia de consumidores, lo que resulta en insatisfacción entre los clientes y en derrotar el propósito de CRM.

2.10.1. TIPOS DE CRM SOFTWARE

Existen varios tipos de CRM:

- ✓ **CRM Operativo:** Se centra en la gestión del marketing, ventas y servicios al cliente. Todos estos procesos son denominados "*Front Office*" porque la empresa tiene contacto con el cliente.
- ✓ **CRM Analítico:** Se corresponde con las diferentes aplicaciones y herramientas que proporcionan información de los clientes, por lo que el CRM analítico está ligado a un depósito de datos o información denominado *Data Warehouse*. Se utiliza con el fin de tomar decisiones relativas a productos y servicios, y evaluar resultados.
- ✓ **CRM Colaborativo:** Su función es centralizar y organizar toda la información y los datos que el cliente proporciona a través de diferentes canales.

2.11. HERRAMIENTAS PARA LA IMPLEMENTACIÓN

2.11.1. PHP

Es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo.

Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre.

Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

2.11.2. JAVASCRIPT

JavaScript, es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo. JavaScript de Netscape es un superconjunto del lenguaje de scripts estándar de la edición de ECMA262 3 (ECMAScript) que presenta sólo leves diferencias respecto a la norma publicada.

Contrariamente a la falsa idea popular, JavaScript no es Java interpretativo. En pocas palabras, JavaScript es un lenguaje de programación dinámico que soporta construcción de objetos basado en prototipos. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje. Las construcciones del lenguaje, tales como sentencias *if*, y bucles *for* y *while*, y bloques *switch* y *try ... catch* funcionan de la misma manera que en estos lenguajes (o casi).

JavaScript puede funcionar como lenguaje procedimental y como lenguaje orientado a objetos. Los objetos se crean programáticamente añadiendo métodos y propiedades a lo que de otra forma serían objetos vacíos en tiempo de ejecución, en contraposición a las definiciones sintácticas de clases comunes en los lenguajes compilados como C++ y Java. Una vez se ha construido un objeto, puede usarse como modelo (o prototipo) para crear objetos similares.

Las capacidades dinámicas de JavaScript incluyen construcción de objetos en tiempo de ejecución, listas variables de parámetros, variables que pueden contener funciones, creación de scripts dinámicos introspección de objetos (mediante *for ... in*), y recuperación de código fuente (los programas de JavaScript pueden descompilar el cuerpo de funciones a su código fuente original).

2.11.3. CSS

CSS (*Cascading Style Sheets*) es lo que se denomina lenguaje de hojas de estilo en cascada y se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS separa el contenido de la representación visual del sitio.

CSS fue desarrollado por W3C (*World Wide Web Consortium*) en 1996 por una razón muy sencilla. HTML no fue diseñado para tener etiquetas que ayuden a formatear la página. Está hecho solo para escribir el marcado para el sitio.

Se incluyeron etiquetas como `` en HTML versión 3.2, y esto les causó muchos problemas a los desarrolladores. Dado que los sitios web tenían diferentes fuentes, fondos de colores y estilos, el proceso de reescribir el código fue largo, doloroso y costoso. Por lo tanto, CSS fue creado por W3C para resolver este problema.

La relación entre HTML y CSS es muy fuerte. Dado que HTML es un lenguaje de marcado (es decir, constituye la base de un sitio) y CSS enfatiza el estilo.

2.11.4. HTML

HTML significa "Lenguaje de Marcado de Hipertexto" por sus siglas en inglés *HyperText Markup Language*, es un lenguaje que pertenece a la familia de los "lenguajes de marcado" y es utilizado para la elaboración de páginas web. El estándar HTML lo define la W3C (*World Wide Web Consortium*) y actualmente HTML se encuentra en su versión HTML5.

Cabe destacar que HTML no es un lenguaje de programación ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de los lenguajes de programación, por lo que HTML genera únicamente páginas web estáticas, sin embargo, HTML se puede usar en conjunto con diversos lenguajes de programación para la creación de páginas web dinámicas.

Básicamente el lenguaje HTML sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces (links) hacia otras páginas o documentos.

HTML es un lenguaje de marcado descriptivo que se escribe en forma de etiquetas para definir la estructura de una página web y su contenido como texto, imágenes, entre otros, de modo que HTML es el encargado de describir (hasta cierto punto) la apariencia que tendrá la página web.

2.11.5. MYSQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por *Oracle Corporation* y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

MySQL fue inicialmente desarrollado por MySQL AB (empresa fundada por David Axmark, Allan Larsson y Michael Widenius). MySQL AB fue adquirida por Sun Microsystems en 2008, y ésta a su vez fue comprada por *Oracle Corporation* en 2010, la cual ya era dueña desde 2005 de Innobase Oy, empresa finlandesa desarrolladora del motor InnoDB para MySQL.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el *copyright* de la mayor parte del código. Esto es lo que posibilita el esquema de doble licenciamiento anteriormente mencionado. La base de datos se distribuye en varias versiones, una *Community*, distribuida bajo la Licencia pública general de GNU, versión 2, y varias versiones Enterprise, para aquellas empresas que quieran incorporarlo en productos privativos. Las versiones

Enterprise incluyen productos o servicios adicionales tales como herramientas de monitorización y asistencia técnica oficial. En 2009 se creó un *fork* denominado MariaDB por algunos desarrolladores (incluido algunos desarrolladores originales de MySQL) descontentos con el modelo de desarrollo y el hecho de que una misma empresa controle a la vez los productos MySQL y Oracle Database.

Está desarrollado en su mayor parte en ANSI C y C++.4 Tradicionalmente se considera uno de los cuatro componentes de la pila de desarrollo LAMP y WAMP.

2.11.6. CODEIGNITER

Codeigniter es un *framework* para el desarrollo de aplicaciones en PHP. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad.

CodeIgniter usa el diseño de Modelo Vista Controlador, que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyectos en los cuales los diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos será mínimo.

El *framework* PHP CodeIgniter funciona con un URL: para dirigirse al controlador, como unidad de control central entre la vista y el modelo, es necesario introducir un URL en la barra de búsqueda del navegador web.

Para ello, los desarrolladores crean las denominadas clases de controlador, archivos PHP con diversas funciones que permiten cargar librerías, extensiones o *helpers*, establecer conexiones a bases de datos, integrar un modelo o seleccionar una vista determinada.

3.1. INTRODUCCIÓN

En el presente capítulo se describe la implementación de la metodología KANBAN aplicando las reglas correspondientes a mencionada metodología. En el proceso de análisis y diseño del Sistema Web se aplicó la metodología de desarrollo UWE.

3.2. APLICACIÓN DE LAS METODOLOGÍAS KANBAN-UWE

La metodología Kanban nos permite visualizar los avances mediante tarjetas y controlar el límite de tareas en cada fase (WIP), gracias a esto es posible detectar las fases donde se generan cuellos de botella o al contrario determinar que personas se encuentran ociosas.

Para la construcción del tablero Kanban se determinan las fases y la cantidad de las mismas adecuando la metodología de ingeniería de software para el desarrollo de aplicaciones web, en este caso particular UWE, de tal manera de que las fases de mencionada metodología de modelado este incluida en el tablero, tomando en cuenta lo mencionado se determinó trabajar con 6 fases o gráficamente columnas en el tablero de la siguiente forma:

- ✓ **Pedido:** En esta fase representa la pila de todos los requerimientos que se necesita para el desarrollo del sistema.
- ✓ **Análisis:** Esta fase consiste en realizar el análisis de los requerimientos, encontrar los usuarios involucrados, aquí hacemos uso del análisis de requerimientos y modelo de contenidos de la metodología UWE.
- ✓ **Diseño:** En esta fase se realiza el diseño del sistema web, todo lo referente a diseño de interfaz y navegación, se hace uso del modelo de navegación y de presentación de la metodología UWE.

- ✓ **Desarrollo:** En esta fase se procede al desarrollo del prototipo del sistema, se hace uso del modelo de flujo de proceso de la metodología UWE.
- ✓ **Implementar y pruebas:** En esta fase se procede a la implementación del prototipo del sistema, utilizando como herramienta principal de desarrollo al *framework Codeigniter*, además de todas las herramientas web propuestas al inicio del proyecto. Posteriormente continuando con las correspondientes pruebas.
- ✓ **Producción:** Esta es una fase donde se reflejará toda la pila de trabajos finalizados, actualizados en el servidor de la empresa.

En el caso de las columnas análisis, diseño, desarrollo e implementar, estas se subdividen en dos columnas, en curso y hecho, tal y como se muestra en la figura 3, la columna en curso está destinada para todas las tareas que se están realizando, la columna hecho es para las tareas terminadas dentro de esa fase y están listas para avanzar a la siguiente.

	REQUERIMIENTOS	ANALISIS 0-2		DISEÑO 0-3		DESARROLLO 0-3		PRUEBAS E IMPLEMEN. 0-3		PRODUCCION
		En curso	Hecho	En curso	Hecho	En curso	Hecho	En curso	Hecho	
Modulo 1	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea
Modulo 2	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea
Modulo 3	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea
Modulo 4	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea

Figura 3: Tabla Kanban-UWE

Fuente: (Elaboración propia, 2019)

3.3. ESPECIFICACIONES

La Especificación de Requisitos de Software es una descripción completa del comportamiento del sistema que se va a desarrollar.

Se efectuó un replanteamiento sobre el uso de las historias de usuario en la pila de productos que son el principal elemento del cual está conformado este, para el presente proyecto y conforme a la metodología de modelado empleada UWE se remplazaron estas historias de usuario por casos de uso diagramas especificados en UWE.

Una vez definidos la mayoría de los casos de uso para la elaboración del sistema, se procedió a la realización de reuniones de planificación con los interesados y posteriormente se seleccionó algunos casos de uso para su desarrollo.

3.4. RECOPIACIÓN DE REQUERIMIENTOS

Mediante una reunión con las partes interesadas logro definirse una lista de tareas a realizar, mediante esta se determinan los requerimientos del sistema a un nivel general.

Luego de diversas reuniones con el cliente, donde éste realiza el listado de las necesidades que se solicitan para la construcción del producto final.

Junto a esto se proponen ideas para lograr consensos y comenzar a detallar en el documento los requerimientos formalmente.

La especificación de estos requisitos se realiza, de conversaciones donde se extraen los detalles que indica el cliente, siguiendo la plantilla, en paralelo con el prototipo y los casos de usos referentes a ese requerimiento. De esta forma se hace un trabajo cíclico con cada requerimiento tratado.

A continuación, se presenta la lista de requerimientos junto a las características del sistema:

Nro.	Requerimientos	Descripción
1	Registrar y administrar datos de usuarios.	Se debe registrar a todos los usuarios que tengan acceso al sistema junto a la información básica del mismo, dando a cada usuario acceso solo a la información requerida.
2	Registrar y controlar los servicios o trabajos en proceso de elaboración hasta su entrega.	Se debe registrar todos los trabajos pendientes junto a la información requerida para su elaboración además de los datos del cliente, de tal manera de tener la información acerca de sus avances y saber cuándo mencionado servicio ya esté terminado.
3	Registrar datos de clientes potenciales y gestionar propuestas.	En necesario tener un entorno donde determinados empleados de la empresa puedan registrar a los posibles clientes y mantener contacto con estos, hasta hacer que se conviertan en clientes fijos.
4	Generar reportes y datos estadísticos de ingresos(ganancias).	Debe haber la posibilidad de que, en base a la información almacenada en el sistema, esta se pueda manipular y poder generar reportes acerca de las ganancias, ingresos, trabajos realizados de manera semanal, mensual y anual, además de listas de clientes con los servicios que se le han brindado, listas de servicios, listas de productos, etc.
5	Guardar el historial de actividades realizadas por los usuarios y de los servicios que se prestó a los clientes.	Tener guardado de las principales actividades que los empleados realizaron en el sistema lista para su consulta a momento de solicitud del jefe, también tener un historial de los clientes junto a los trabajos o servicios que requirió.
6	Controlar y hacer seguimiento a los pagos por servicios prestados.	Tener un control continuo de los pagos que se realizan por los servicios brindados a los clientes hasta su cancelación además de los montos que se fueron pagando con sus respectivas fechas de pago.
7	Registrar la venta de productos o servicios extra.	Tener un registro de productos o servicios extra que el cliente consumió en el proceso de elaboración de un servicio contratado.
8	Registrar fechas de eventos importantes.	Tener un entorno en el que se pueda registrar fechas de eventos importantes, como ser: reuniones, consultas, visitas, etc.
9	Asignar encargados de trabajos a empleados	Tener un registro donde se pueda asignar a empleados a determinados trabajos o servicios para su elaboración y cumplimiento dentro de los requerido por el cliente.

Tabla 3: Lista de requerimientos
Fuente: (Elaboración propia, 2019)

3.5. APLICACIÓN DE REGLAS KANBAN

3.5.1. MOSTRAR EL PROCESO (VISUALIZAR)

Se elaboro el tablero Kanban con las tareas que se fijaron, el cual los primeros días se actualizo con el fin de asegurar la correcta planificación del proyecto y mostrar avances los cuales estén visibles a todo momento. Las columnas son representadas mediante diferentes estados y muestran los pasos del flujo (ver figura 4).

	REQUERIMIENTOS	ANALISIS 0/2		DISEÑO 0/2		DESARROLLO 0/2		PRUEBAS E IMPLEMEN. 0/2		PRODUCCION
		En curso	Hecho	En curso	Hecho	En curso	Hecho	En curso	Hecho	
Modulo 1	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea
Modulo 2	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea
Modulo 3	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea
Modulo 4	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea	+ añadir tarea

Figura 4: Tabla Kanban-UWE

Fuente: (Elaboración Propia, 2019)

3.5.2. LIMITANDO EL WIP (WORK IN PROGRESS)

Para definir los límites del trabajo en proceso (*WIP, Work In Progress*) se restringen las diferentes fases con la cantidad máxima de elementos de trabajo (columnas del tablero Kanban).

La implementación de los límites de WIP ayuda a su equipo a enfocarse solo en las tareas actuales y así le permite terminar más rápido con los elementos de trabajo individuales.

Debemos tener en cuenta que limitar el WIP no aplica solo a la fase de desarrollo, sino que también a las fases previas de definición y análisis de requerimientos, se limitó la fase de análisis a 3 tareas, diseño a 3 días, desarrollo a 3 tareas y en las pruebas e implementación a 3 tareas.

3.5.3. OPTIMIZAR EL FLUJO DE TRABAJO

Se mantiene una supervisión, medición y reportes del flujo de trabajo a través de cada estado o columna.

Al gestionar activamente el flujo, los cambios continuos, graduales y evolutivos del sistema son evaluados para tener efectos positivos y desechar posibilidades de generar efectos negativos.

3.5.4. ROLES KANBAN

En la tabla 4 se determinan los roles para cada fase y el correspondiente limite WIP.

FASE	WIP	RESPONSABLE
Requerimientos	-	Propietario del producto: Lic. José Gutiérrez. Flow Manager: Wilson Mamani Condori
Análisis	3	Analista: Wilson Mamani Condori
Diseño	3	Diseñador: Wilson Mamani Condori
Desarrollo	3	Desarrollador: Wilson Mamani Condori
Pruebas e implementación	3	Desarrollador: Wilson Mamani Condori
Producción	-	Flow Manager: Wilson Mamani Condori

Tabla 4: Tabla de roles
Fuente: (Elaboración Propia, 2019)

3.6. MÓDULO DE ADMINISTRACIÓN DE USUARIOS Y GESTIÓN DE CLIENTES

3.6.1. FASE DE ANÁLISIS

3.6.1.1. ANÁLISIS DE REQUERIMIENTOS

Con el fin de realizar la implementación de la metodología UWE, a continuación, se detalla los usuarios que intervienen, los roles y sus correspondientes funciones.

- a) **Administrador del sistema**, crea, edita, elimina y consulta los datos de los usuarios para que estos puedan ingresar al sistema y cumplir sus tareas.
- b) **Gerente**, crea, edita, elimina y consulta los datos (registro de servicios realizados) de los clientes y posibles clientes.

3.6.1.2. CASOS DE USO

A continuación, en la figura 5, se detalla el caso de uso para la administración de roles de usuarios:

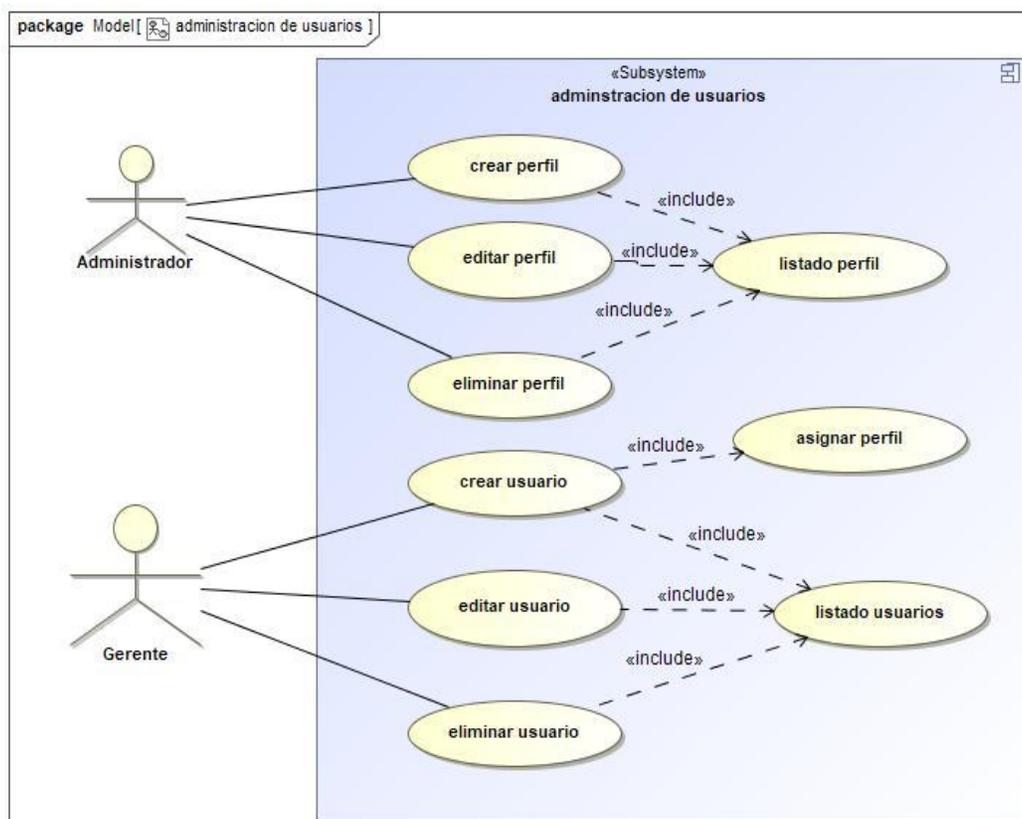


Figura 5: Caso de uso para administración de roles de usuario

Fuente: (Elaboración Propia,2019)

A continuación, en la tabla 5 se desglosa las especificaciones del caso de uso de Administración de Usuarios.

Caso de uso: Administrador de Usuarios
Actores: Administrador de sistema Gerente general de la empresa
Descripción: El administrador y el gerente de la empresa pueden realizar operaciones básicas en cuanto a usuarios del sistema, como ser: crear, editar, eliminar o ver los datos de todos los registros realizados en el sistema, en cuanto a registro de usuarios.
Precondiciones: Tanto como el administrador como el gerente de la empresa deben estar anticipadamente registrados con los privilegios más altos del sistema.
Flujo Normal: El actor ingresa al sistema con una cuenta administrativa. El actor ingresa a la interfaz donde se despliega el listado de usuarios con las diferentes acciones mencionadas. El actor realiza la acción que desee, pudiendo elegir entre crear usuario, eliminar usuario, o simplemente editar el usuario elegido.
Postcondiciones: En base a la opción elegida por el actor se mostrarán en pantalla los cambios realizados.

Tabla 5: Especificaciones del Administrador de usuarios

Fuente: (Elaboración Propia, 2019)

En la figura 6, se muestra el caso de uso para la gestión de clientes, donde se desenvuelve como único actor el Gestor de Clientes.

En el siguiente escenario, aunque el único actor es el Gestor de Clientes, este administra a los servicios y a los clientes, asociando a cada cliente los servicios que actualmente se está brindando, como también todos aquellos servicios que se le hayan brindado.

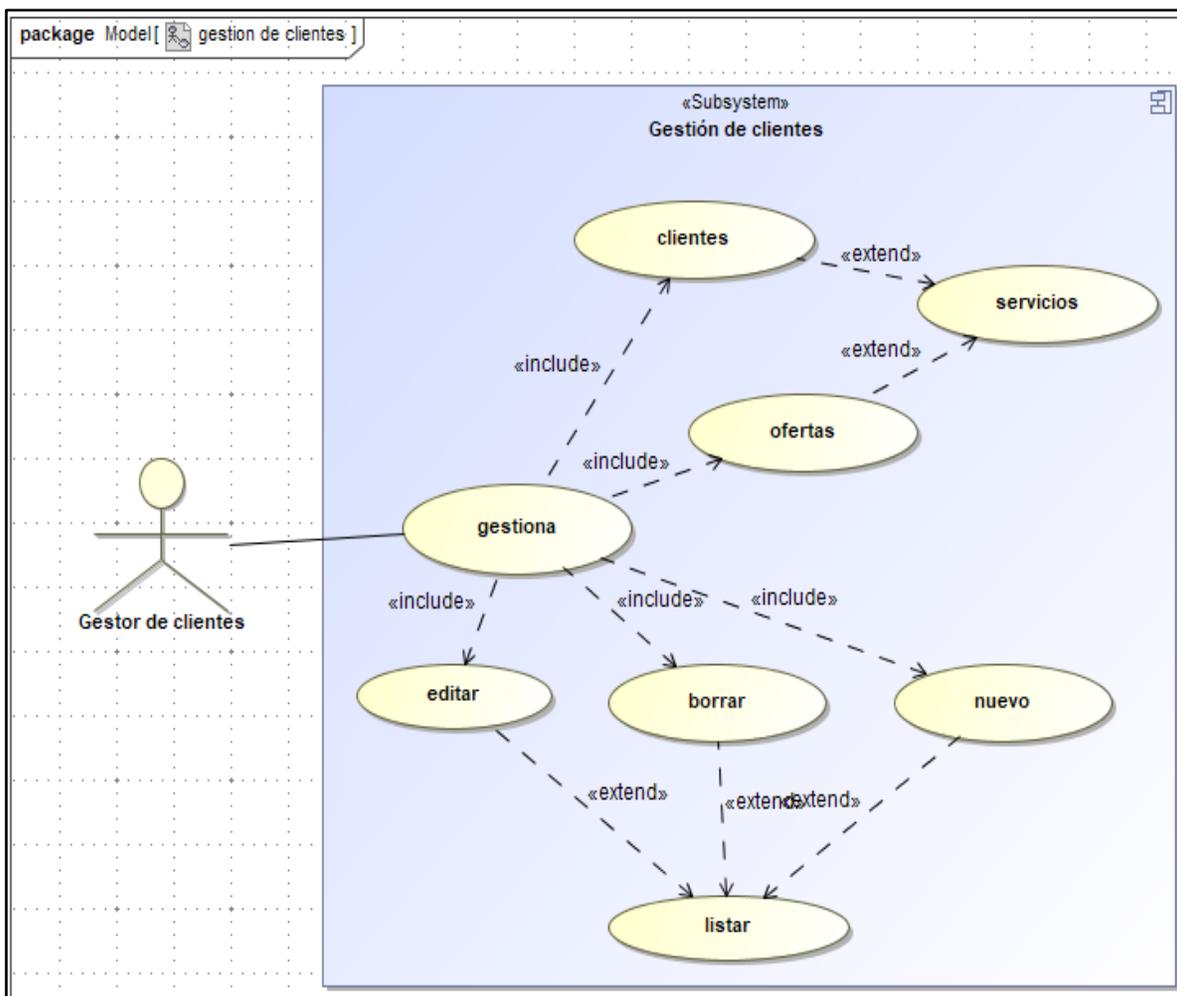


Figura 6 : Casos de uso de la gestión de Clientes

Fuente: (Elaboración Propia, 2019)

En la tabla 6 de desarrolla las especificaciones para el caso de uso de la Gestión de Clientes.

<p>Caso de uso: Gestión de Clientes</p>
<p>Actores: Gestor de Clientes</p>
<p>Descripción: El gestor de clientes tiene el trabajo de captar nuevos clientes a través de una relación mas cercana con el cliente, ofreciendo los servicios de la empresa, ofertas y realizando notificaciones acerca de los servicios ofrecidos por la empresa.</p>
<p>Precondiciones: El gestor de Clientes debe tener una cuenta inferior a la de administrador de sistemas, es decir el administrador asignara un perfil determinado para que el gestor tenga acceso a la información de los clientes de la empresa, como también de los potenciales clientes.</p>
<p>Flujo Normal: El actor ingresa al sistema con una cuenta de Gestor de Clientes El actor ingresa a la interfaz donde se despliega el listado de clientes con las diferentes acciones mencionadas, teniendo acceso a la información almacenada a cerca de los clientes y los servicios que anteriormente contrataron los clientes. El actor utilizara la información almacenada, para comunicarse con clientes, registrarlos en el sistema como nuevos clientes, y mantenerlos informados sobre los servicios que ofrece la empresa. El actor cambia de estado, de prospectos de clientes a clientes fijos.</p>
<p>Postcondiciones: Tener un registro de nuevos clientes con la intención de adquirir los servicios de la empresa.</p>

Tabla 6: Especificaciones de la gestión de clientes

Fuente: (Elaboración Propia, 2019)

3.6.1.3. MODELOS DE CONTENIDOS

Un modelo de contenidos especifica la abstracción de clases que se realizaron en base al contexto en el que se trabaja, para el módulo de administración de usuarios, trabajando en base a los actores de los casos de uso, obteniendo de estos, tal cual se muestra en la figura 7, las clases más importantes como ser:

- ✓ Persona
- ✓ Cliente
- ✓ Usuario
- ✓ Oferta
- ✓ Servicio
- ✓ Permiso
- ✓ Perfil

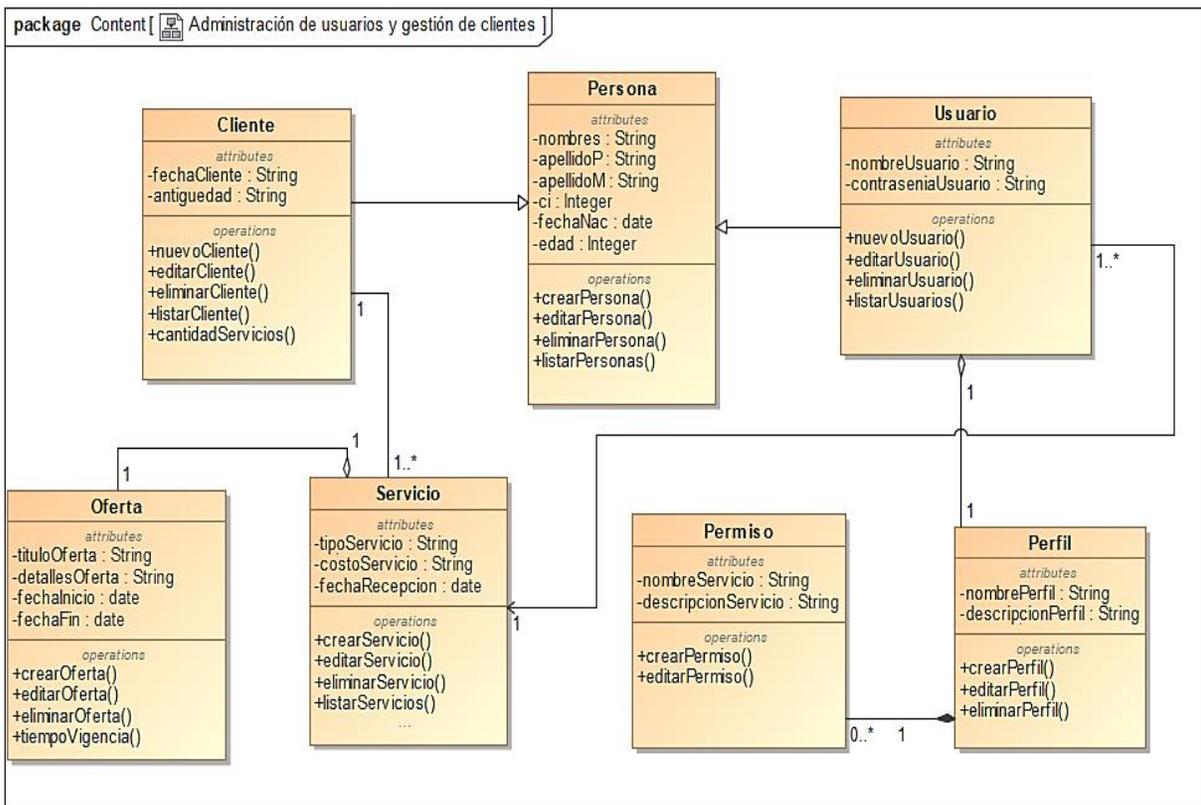


Figura 7: Modelo de Contenidos de administración de usuarios y gestión de clientes

Fuente: (Elaboración Propia, 2019)

3.6.2. FASE DE DISEÑO

3.6.2.1. MODELO NAVEGACIONAL

Mediante el modelo navegacional se determinan las páginas del módulo de Administración de Usuarios, utilizando los nodos y enlaces tal cual se muestra en la Figura 8.

Donde se realiza un flujo de navegación de los entornos en los que el usuario se desplazara para obtener el registro de usuarios.

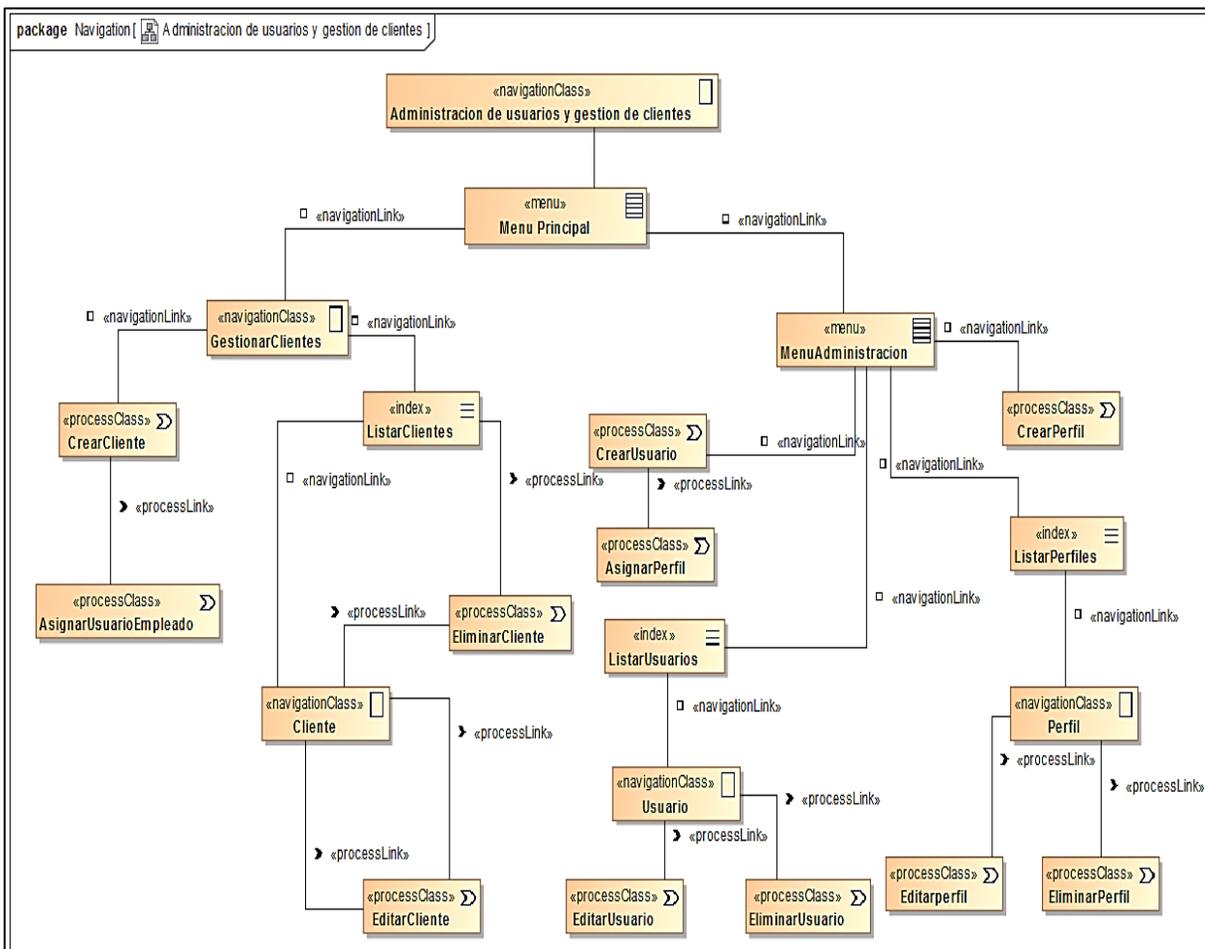


Figura 8: Modelo Navegacional de la Administración de Usuarios y Gestión de Clientes

Fuente: (Elaboración Propia, 2019)

3.6.2.2. MODELO DE PRESENTACIÓN

El modelo de presentación muestra un bosquejo de la interfaz de la vista para la Administración de usuarios y la gestión de Clientes, tal como se observa en la figura 9.

Se despliegan listas tanto de usuarios como de clientes con sus distintos atributos desplegándose de forma simultánea para hacer visual la información más importante, y teniendo en las mismas interfaces las acciones necesarias para cumplir las tareas solicitadas.

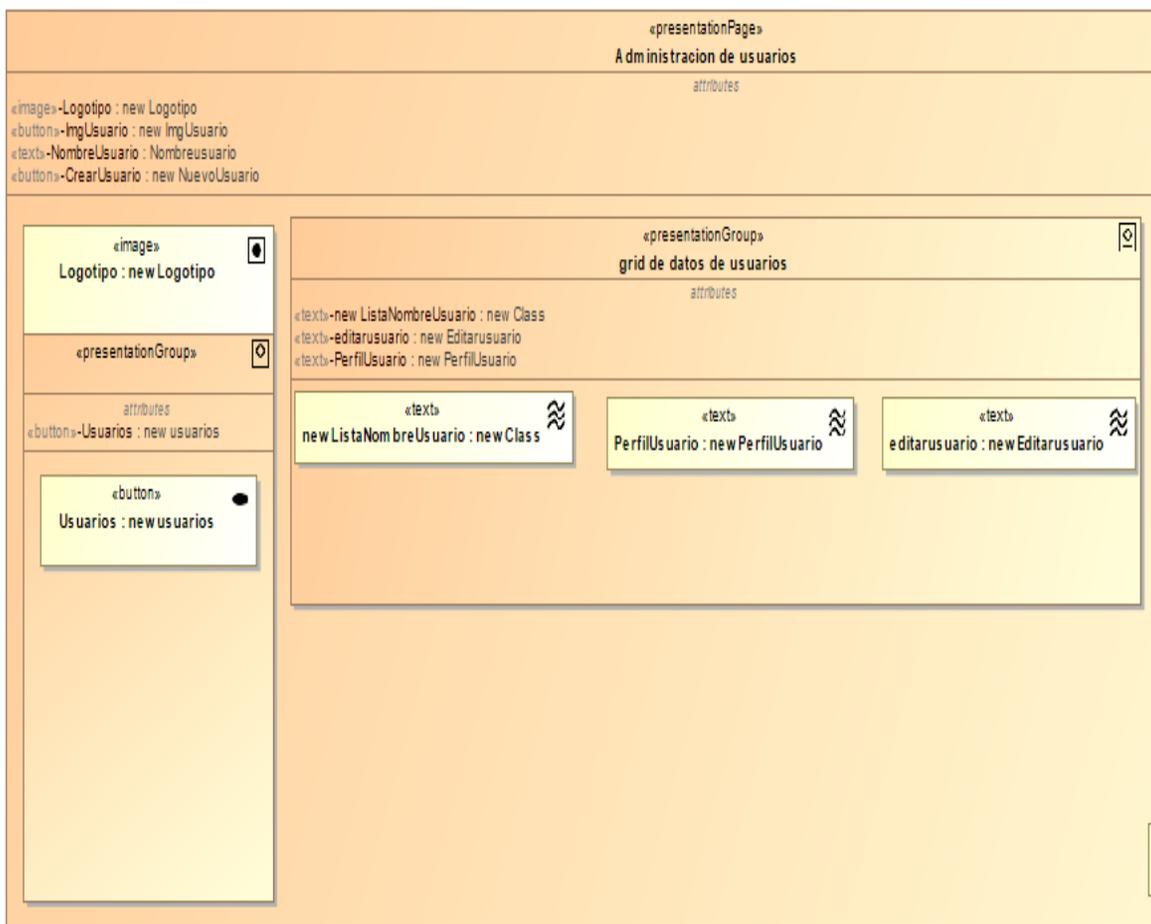


Figura 9: Modelo de Presentación del Administrador de usuarios

Fuente: (Elaboración Propia, 2019)

En la figura 10, se muestra el Modelo de la interfaz para editar y guardar información.

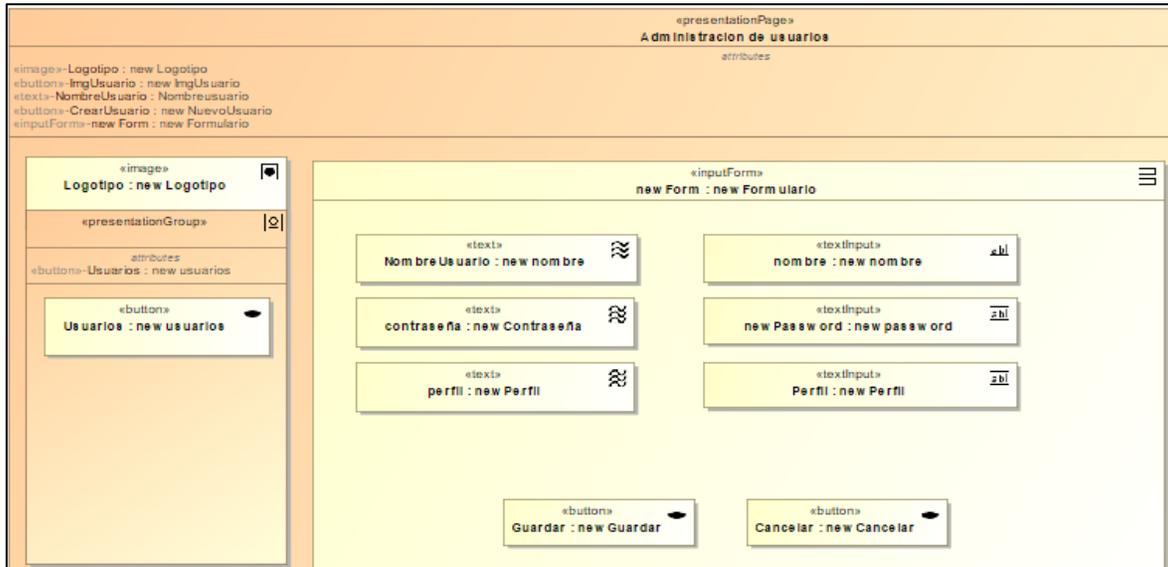


Figura 10: Modelo de presentación para editar y guardar usuarios
Fuente: (Elaboración Propia, 2019)

En la figura 11, se muestra el Modelo de presentación de la interfaz para listar y desplegar la información de clientes registrados en el módulo de Gestión de Clientes.

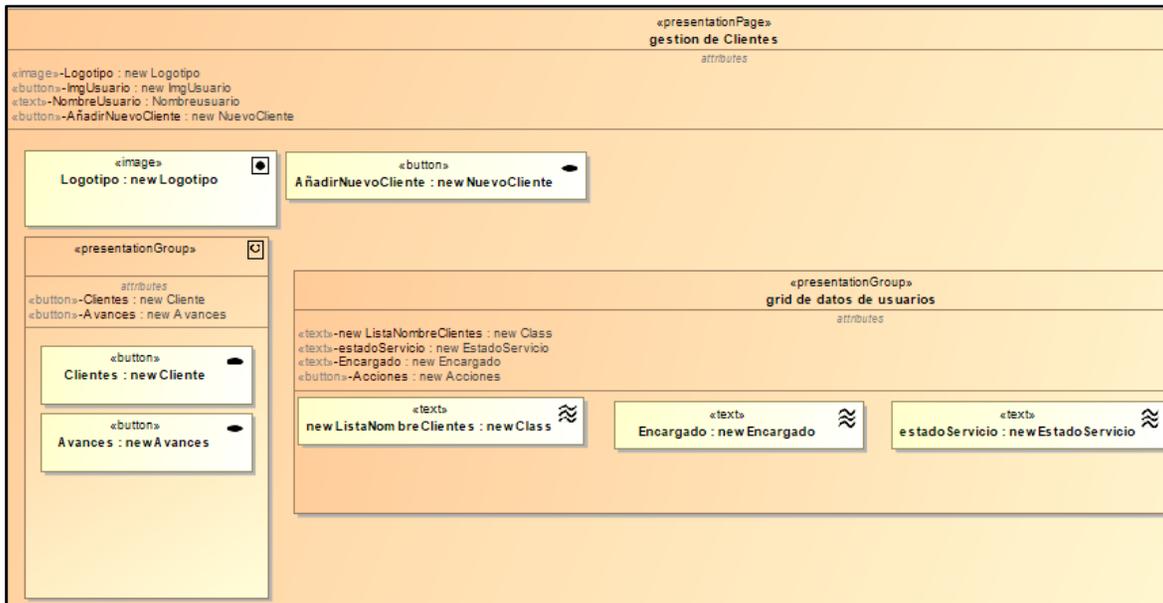


Figura 11: Modelo de presentación para listar y desplegar la información
Fuente: (Elaboración Propia, 2019)

En la figura 12, se muestra el modelo de presentación de la interfaz para registrar la información de nuevos clientes.

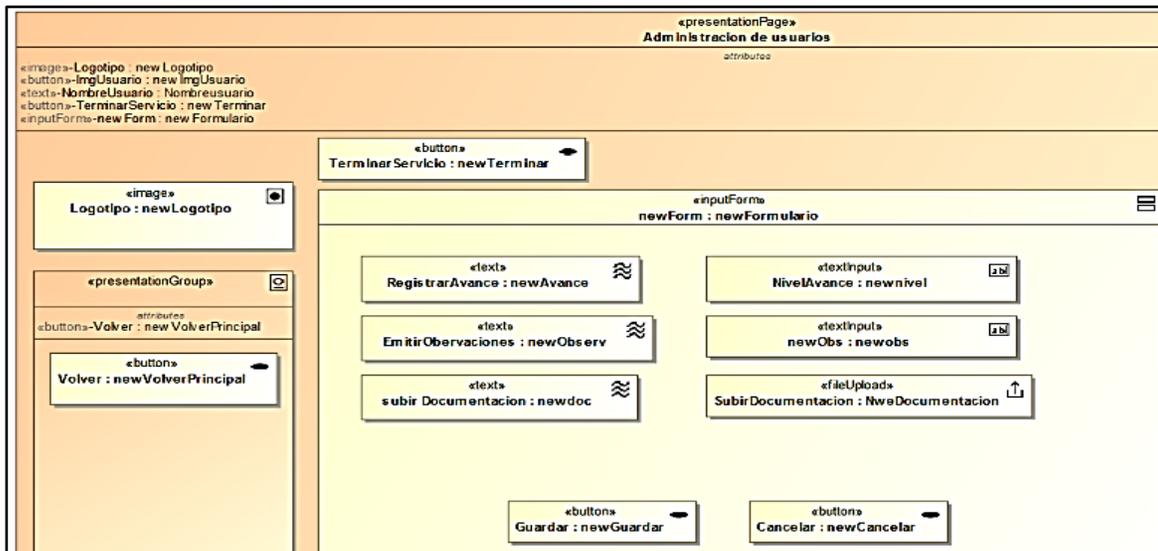


Figura 12: modelo de presentación para registrar y editar clientes
Fuente: (Elaboración Propia, 2019)

3.6.3. FASE DE DESARROLLO

3.6.3.1. MODELO DE FLUJO DE PROCESO

El modelo de flujo de proceso que se encuentra en la figura 13, determina todo el proceso que sigue el usuario a momento de realizar registros.

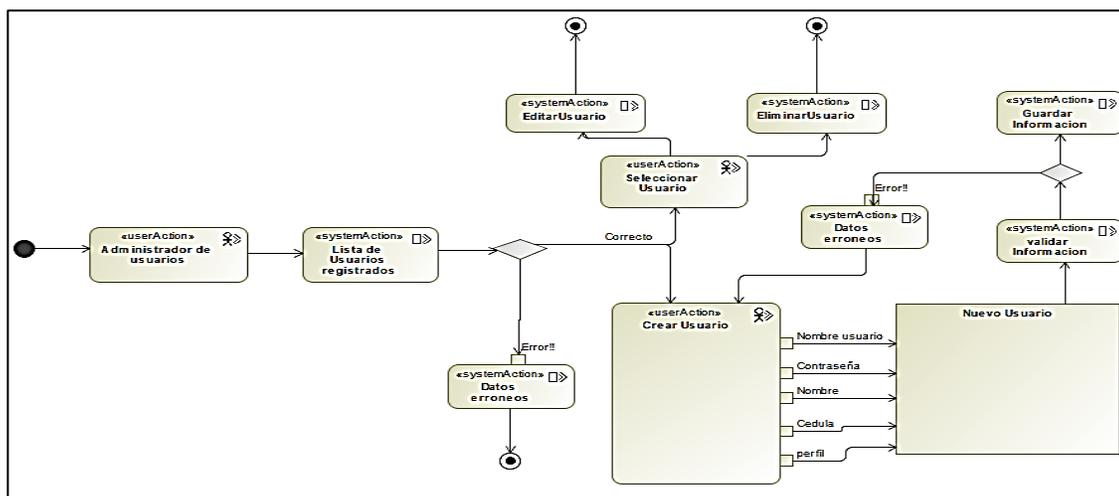


Figura 13: Modelo de flujo de proceso del módulo de usuarios y clientes
Fuente: (Elaboración Propia, 2019)

3.6.4. FASE DE IMPLEMENTACIÓN

Para llegar a esta etapa se desarrolló el prototipo de la interfaz gráfica como se puede observar en la figura 14, siendo una captura con el sistema funcionando correctamente.

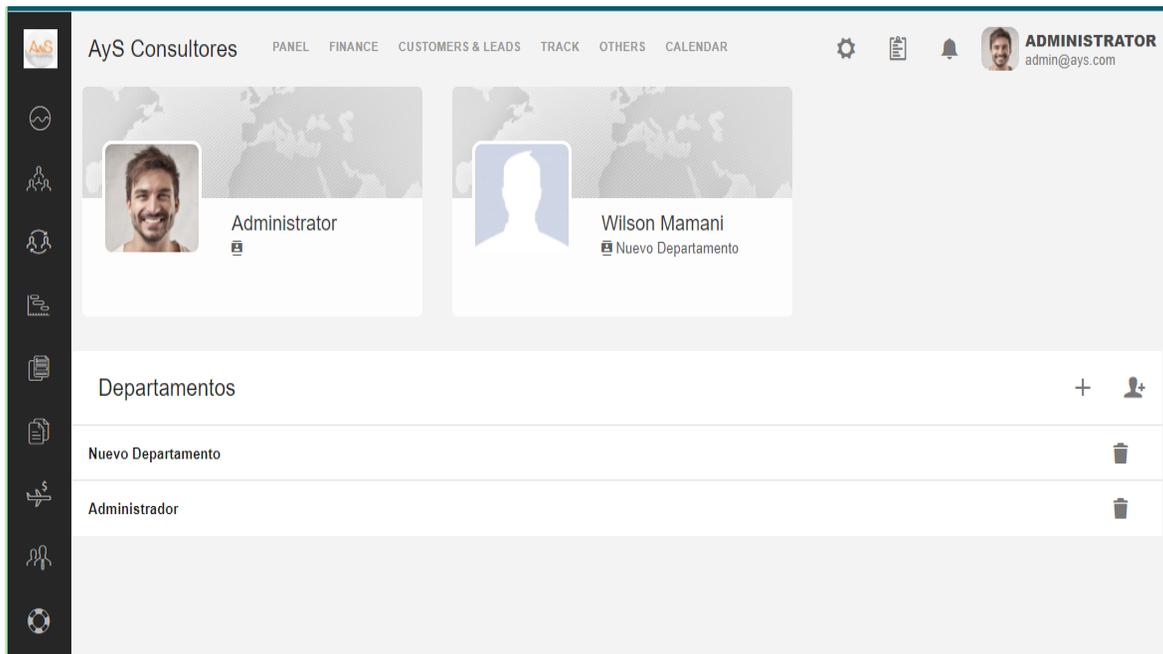


Figura 14: Captura de la interfaz de Administrador de usuarios

Fuente: (Elaboración Propia, 2019)

3.7. MÓDULO DE CONTROL Y SEGUIMIENTO DE SERVICIOS

3.7.1. FASE DE ANÁLISIS

3.7.1.1. ANÁLISIS DE REQUERIMIENTOS

Con el fin de realizar la implementación de la metodología UWE, a continuación, se detalla los usuarios que intervienen, los roles y sus correspondientes funciones.

a) **Recepcionista**

Crea, edita, elimina y consulta los registros de los servicios que se encuentran en una etapa de elaboración, como también designa al encargado de que desarrolle el trabajo hasta su conclusión.

b) **Gestor de servicios**

Es el encargado de realizar el trabajo completo del servicio y registrar todos los avances, observaciones y documentación que se fueron realizando a medida que se va avanzando con el servicio brindado.

3.7.1.2. CASOS DE USO

A continuación, en la figura 15, se detalla el caso de uso para la administración de roles de usuarios:

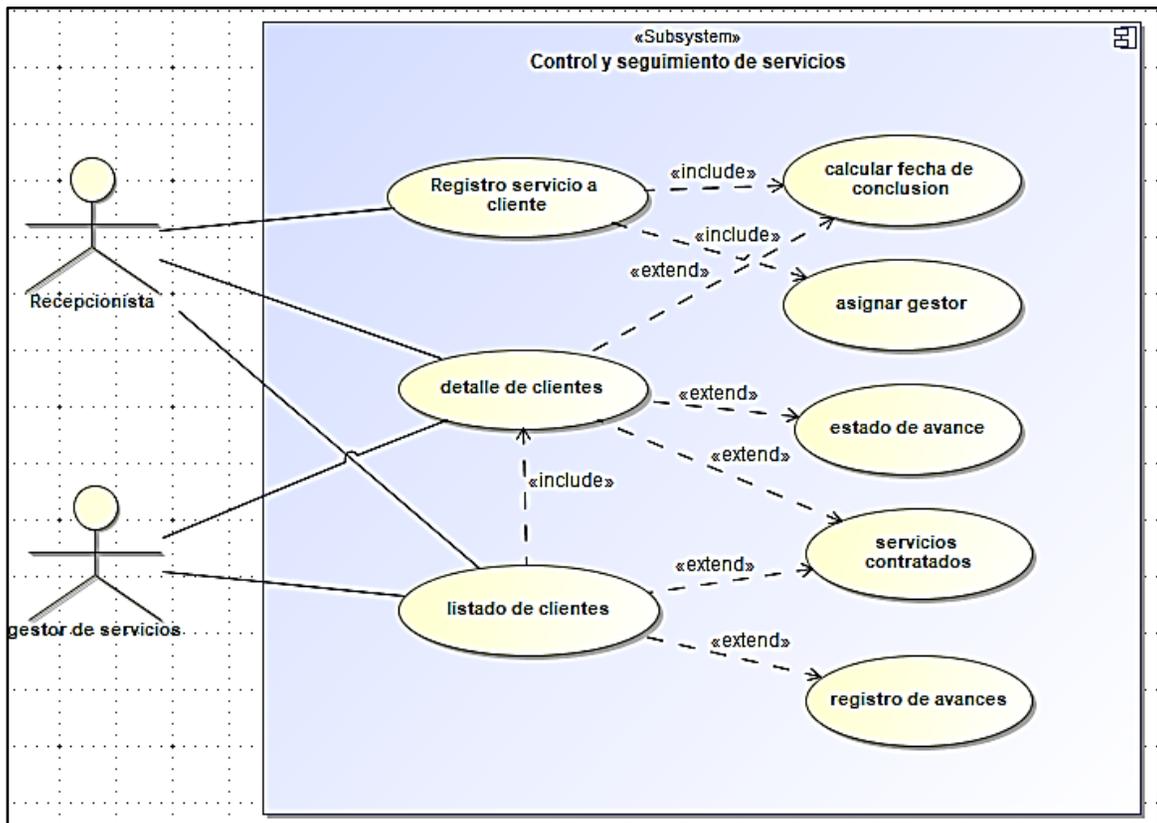


Figura 15: Caso de uso para la gestión de servicios

Fuente: (Elaboración Propia,2019)

A continuación, en la tabla 7 se desglosa las especificaciones del caso de uso de Administración de Usuarios.

Caso de uso: Control y seguimiento de servicios
Actores: Recepcionista, Gestor de servicios
Descripción: En el presente proceso, la recepcionista registrara el servicio que se está brindando a un determinado cliente, o a una determinada empresa donde posterior al registro designara a un gestor de servicio que se encargara de realizar el servicio que se está brindando al cliente.
Precondiciones: Para que la recepcionista registre al cliente y el servicio que está consumiendo, debe estar registrada con el perfil de “recepcionista” el cual le dará los privilegios de asignar a un determinado gestor de servicio.
Flujo Normal: El actor recepcionista ingresa al sistema El actor recepcionista registra un nuevo servicio contratado por un cliente. Se designa a un actor gestor de servicios, quien se encarga de realizar el trabajo y cumplir con el servicio, registrando los avances, pagos, observaciones, documentación recibida y generada. Mediante el sistema se verifica la conclusión del servicio brindado junto a la verificación de pago y documentación correctamente almacenada y se terminara el proceso.
Postcondiciones: El proceso será correctamente culminado cuando el cliente este satisfecho por el servicio contratado y el trabajo se haya realizado en su totalidad, y habiendo registrado el pago total por el servicio en el sistema, y con la documentación generada y almacenada correctamente en el sistema.

Tabla 7: Especificaciones del control y seguimiento de servicios

Fuente: (Elaboración Propia, 2019)

3.7.2. FASE DE DISEÑO

3.7.2.1. MODELO NAVEGACIONAL

Mediante el modelo navegacional se determinan las páginas del módulo de control y seguimiento de servicios, utilizando los nodos y enlaces tal cual se muestra en la Figura 17, realizando un flujo de navegación de los entornos en los que el usuario se desplazara para realizar un correcto seguimiento a los servicios contratados por los clientes.

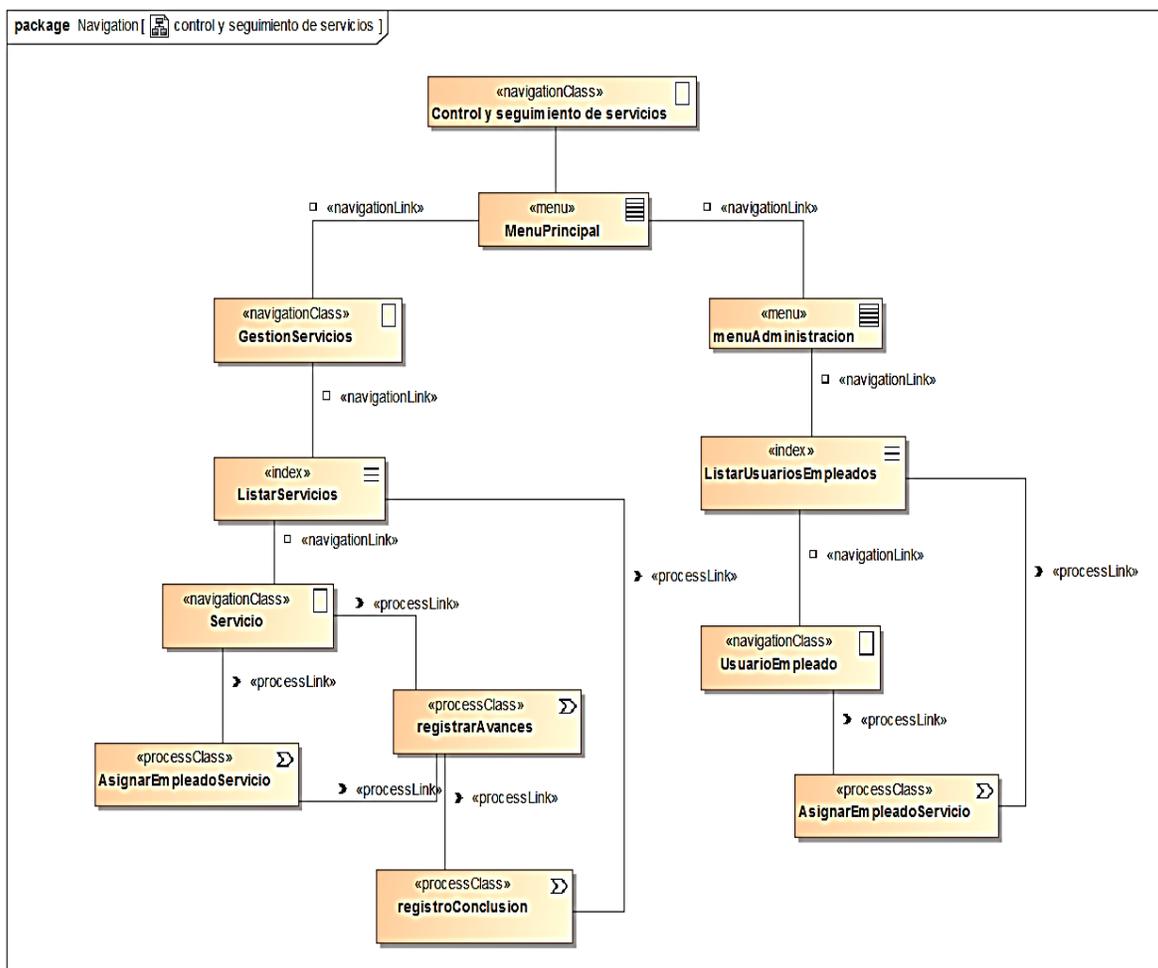


Figura 17: Modelo Navegacional del control y seguimiento de servicios

Fuente: (Elaboración Propia, 2019)

3.7.2.2. MODELO DE PRESENTACIÓN

El modelo de presentación muestra un bosquejo de la interfaz de la vista para el control y seguimiento de servicios, tal como se observa en la figura 18.

Donde se despliegan tarjetas donde se encuentran registrados todos los servicios pendientes y sus estados de avance, además de la persona encargada de su elaboración.

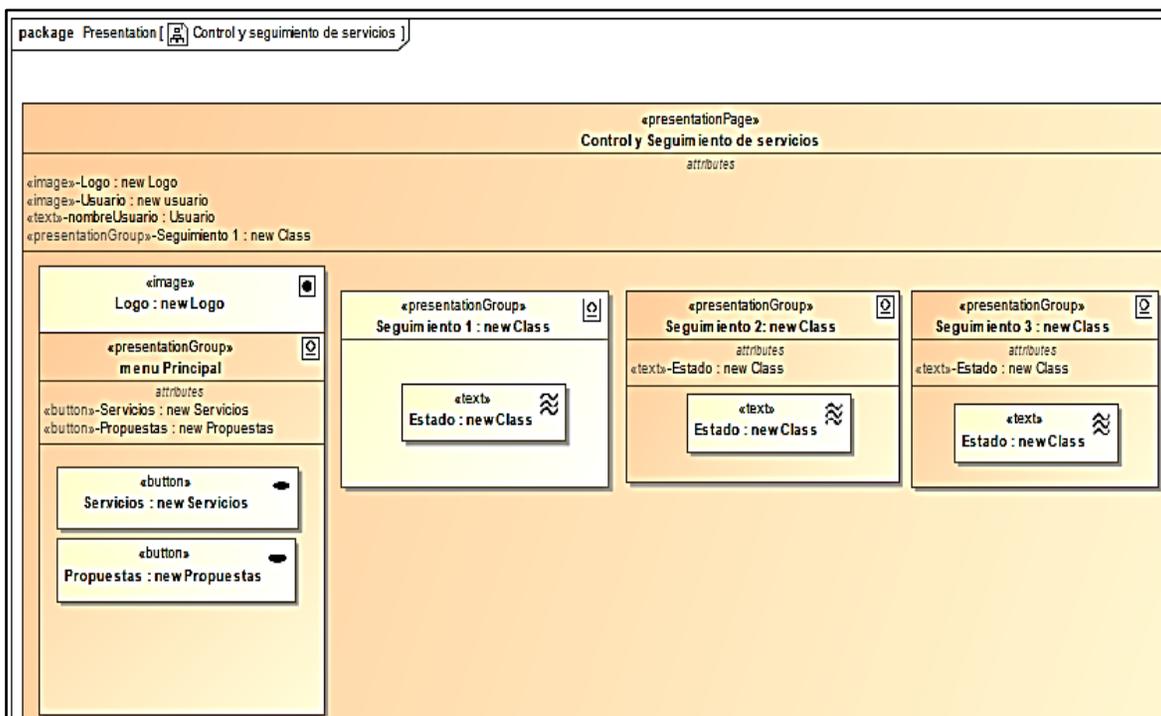


Figura 18: Modelo de Presentación del control y seguimiento de servicios

Fuente: (Elaboración Propia, 2019)

3.7.3. FASE DE DESARROLLO

3.7.3.1. MODELO DE FLUJO DE PROCESO

El modelo de flujo de proceso que se encuentra en la figura 19, determina todo el proceso que sigue tanto la recepcionista como el gestor de servicio para la realización del trabajo y todo lo que incluye el servicio contratado por el cliente.

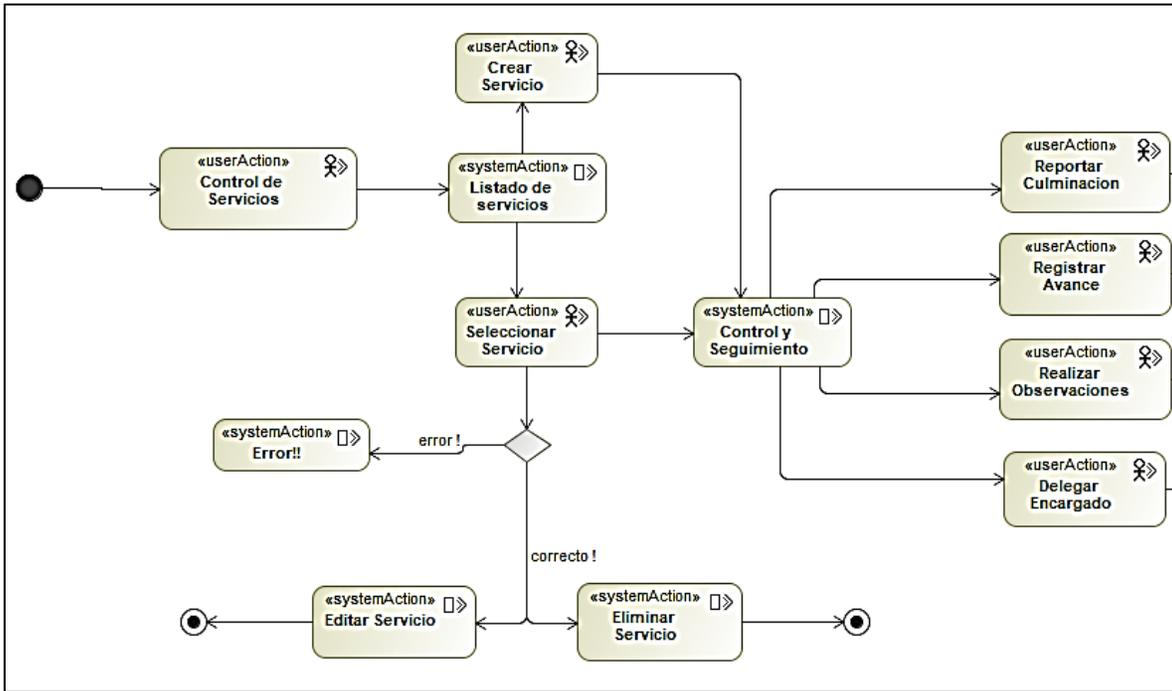


Figura 19: Modelo de flujo del control y seguimiento de servicios
Fuente: (Elaboración Propia, 2019)

3.7.4. FASE DE IMPLEMENTACIÓN

Para llegar a esta etapa se desarrolló el prototipo de la interfaz gráfica como se puede observar en la figura 20, siendo una captura con el sistema funcionando correctamente.

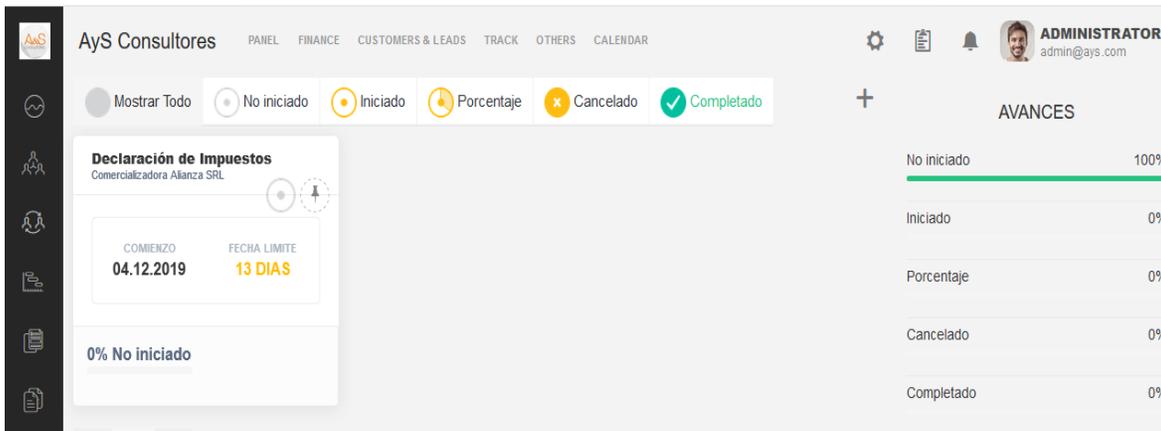


Figura 20: Captura del entorno grafico del control y seguimiento de servicios
Fuente: (Elaboración Propia, 2019)

3.8. MÓDULO DE CONTROL Y SEGUIMIENTO DE PAGOS POR SERVICIOS

3.8.1. FASE DE ANÁLISIS

3.8.1.1. ANÁLISIS DE REQUERIMIENTOS

Con el fin de realizar la implementación de la metodología UWE, a continuación, se detalla los usuarios que intervienen, los roles y sus correspondientes funciones.

- a) **Recepcionista**, registra los pagos totales o parciales, además de los gastos extras en el proceso de elaboración de los trabajos dentro del servicio brindado.
- b) **Gestor de servicios**, es el encargado de registrar gastos extras del servicio, tareas que se realizaron que generen costo.

3.8.1.2. CASOS DE USO

A continuación, en la figura 21, se detalla el caso de uso para control de pagos y reportes:

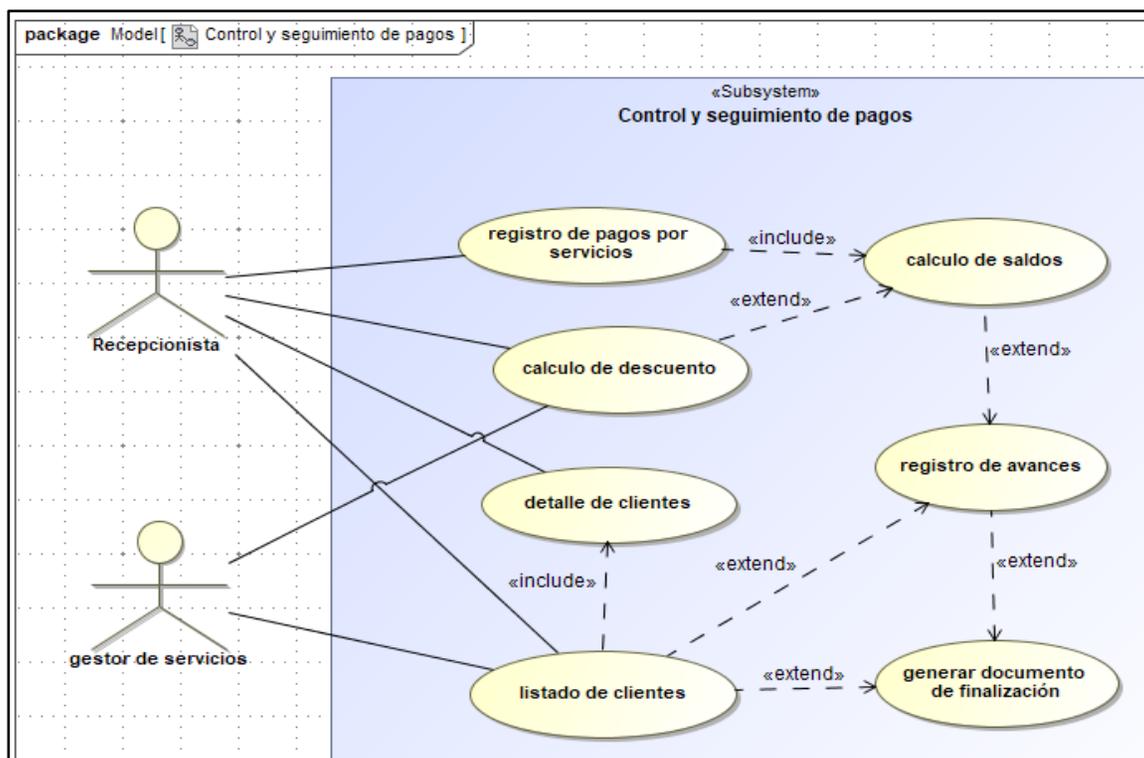


Figura 21: Caso de uso para control de pagos y reportes

Fuente: (Elaboración Propia,2019)

A continuación, en la tabla 8 se desglosa las especificaciones del caso de uso para control de pagos y reportes.

Caso de uso: Control de pagos y reportes	
Actores:	<ul style="list-style-type: none"> ✓ Recepcionista ✓ Gestor de servicios ✓ Gerente ✓ Gestor de Clientes ✓ Administrador de servicios
Descripción:	<p>En el presente proceso, la recepcionista registrara los pagos realizados por el cliente, ya sean adelantos o pagos parciales y la cancelación del mismo cuando el servicio haya terminado, además de gastos extras o costos de servicios complementarios, el pago final se realizará siempre y cuando el Gestor de Clientes haya notificado la culminación de los trabajos realizados dentro del servicio contratado. Además, todos los demás usuarios tienen la posibilidad de generar reportes de acuerdo a los parámetros solicitados en el sistema, con fines informáticos o de estudio de mercado.</p>
Precondiciones:	<p>Para que la recepcionista registre al cliente y el servicio que está consumiendo, debe estar registrada con el perfil de “recepcionista” el cual le dará los privilegios para registrar pagos por servicios, gastos y servicios extras.</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor recepcionista ingresa al sistema 2. El actor recepcionista registra un pago por el servicio, ya sea parcial o total. 3. El gestor de servicio encargado, deberá notificar mediante sistema cualquier avance que genere costo o servicio extra y también cuando termine el servicio para que en el sistema se encuentre registrado el costo total por los trabajos realizado dentro del servicio. 4. Los usuarios podrán generar reportes del propio servicio brindado como también de los trabajos realizados con anterioridad a distintos clientes, con el fin de informarse o de hacer estudios de mercado.
Postcondiciones:	<ul style="list-style-type: none"> ✓ El proceso será correctamente culminado cuando el cliente este satisfecho por el servicio contratado y el trabajo se haya realizado en su totalidad, y habiendo registrado el pago total por el servicio en el sistema, y con la documentación generada y almacenada correctamente en el sistema.

Tabla 8: Especificaciones del control de pagos y reportes

Fuente: (Elaboración Propia, 2019)

3.8.1.3. MODELOS DE CONTENIDOS

Un modelo de contenidos especifica la abstracción de clases que se realizaron en base al contexto en el que se trabaja, para el módulo de control de pagos y reportes, trabajando en base a los actores de los casos de uso, obteniendo de estos, tal cual se muestra en la figura 22:

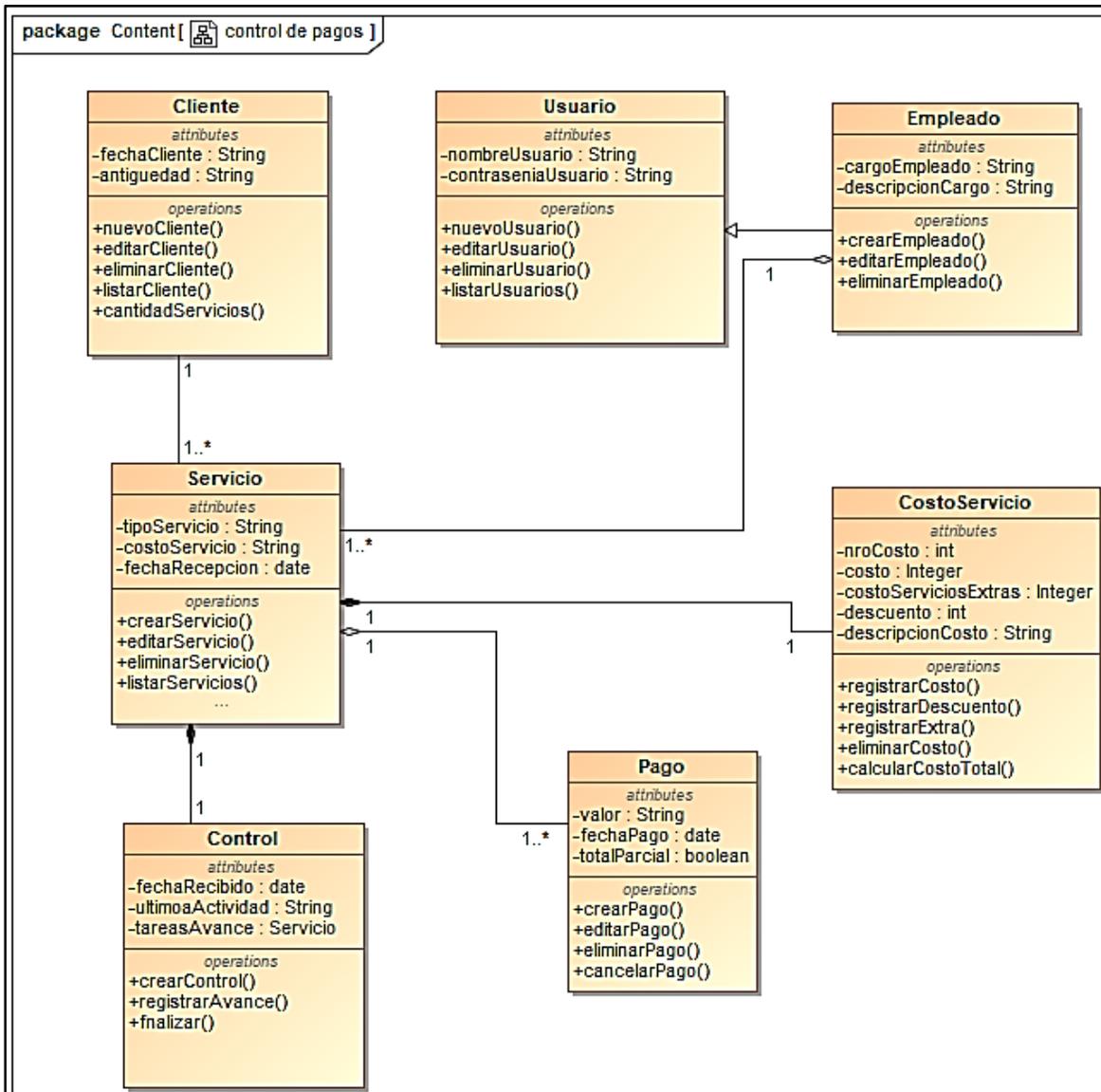


Figura 22: Modelo de Contenidos del control de pagos y reportes

Fuente: (Elaboración Propia, 2019)

3.8.2. FASE DE DISEÑO

3.8.2.1. MODELO NAVEGACIONAL

Mediante el modelo navegacional se determinan las páginas del módulo de control de pagos y reportes, utilizando los nodos y enlaces tal cual se muestra en la Figura 23.

Realizando un flujo de navegación de los entornos en los que el usuario se desplazara para realizar un correcto control de pagos y generación de reportes.

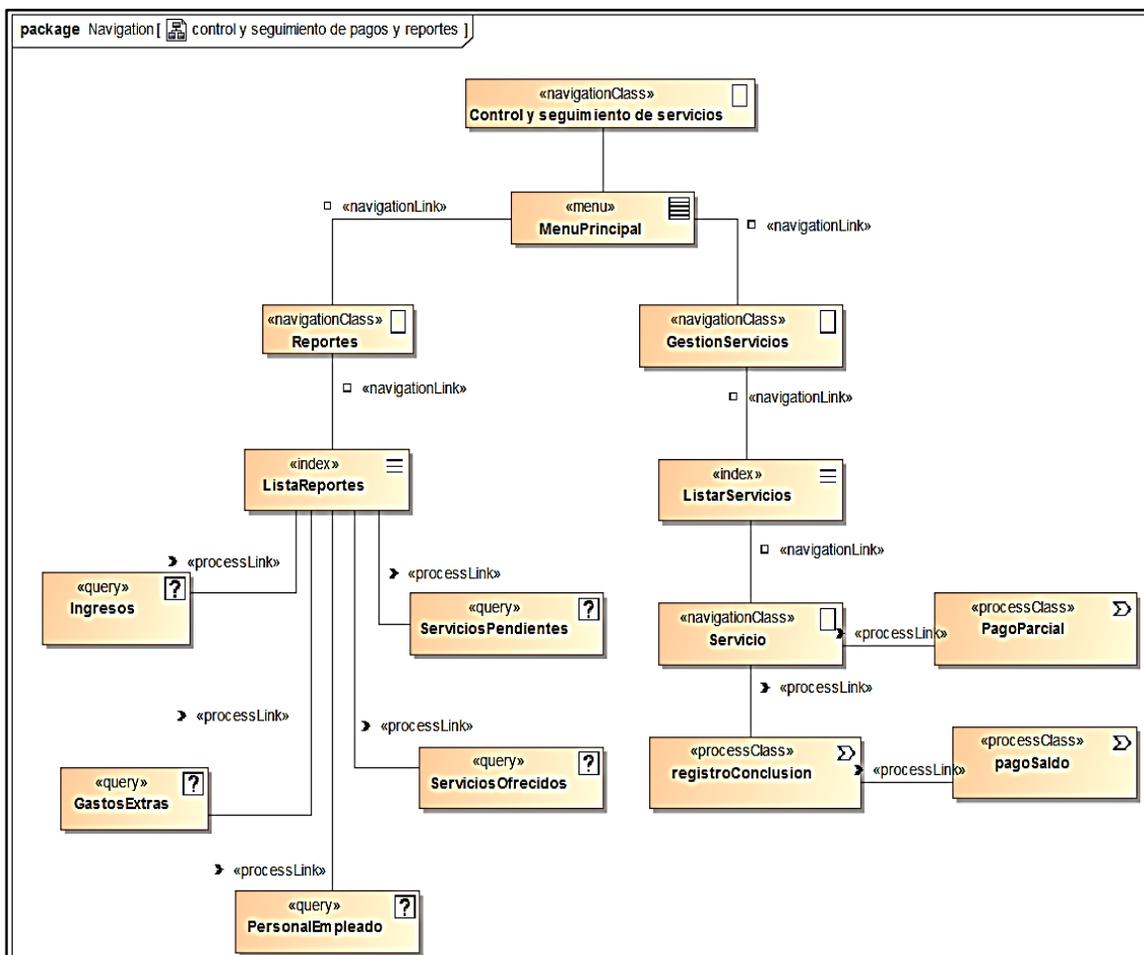


Figura 23: Modelo Navegacional del control de pagos y generación de reportes

Fuente: (Elaboración Propia, 2019)

3.8.2.2. MODELO DE PRESENTACIÓN

En el siguiente modelo de presentación muestra un bosquejo de la interfaz de la vista para el control de pagos tal como se observa en la figura 24:

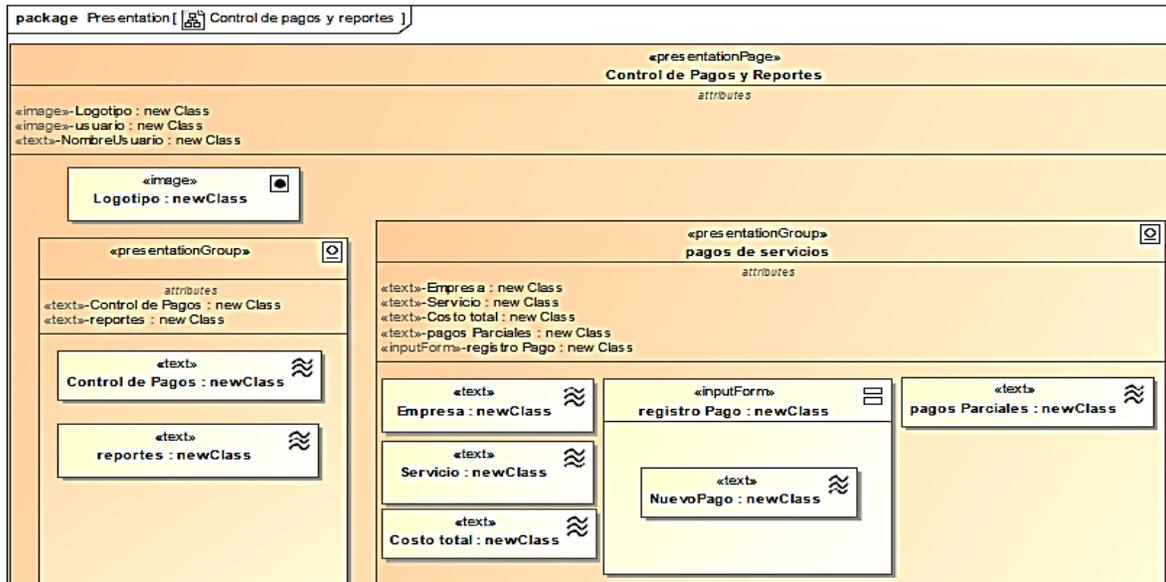


Figura 24: Modelo de Presentación del control de pagos

Fuente: (Elaboración Propia, 2019)

En la figura 25, con el fin de que el usuario pueda manipular la información almacenada y con esta pueda generar información útil en base a criterios ya asignados en el sistema.

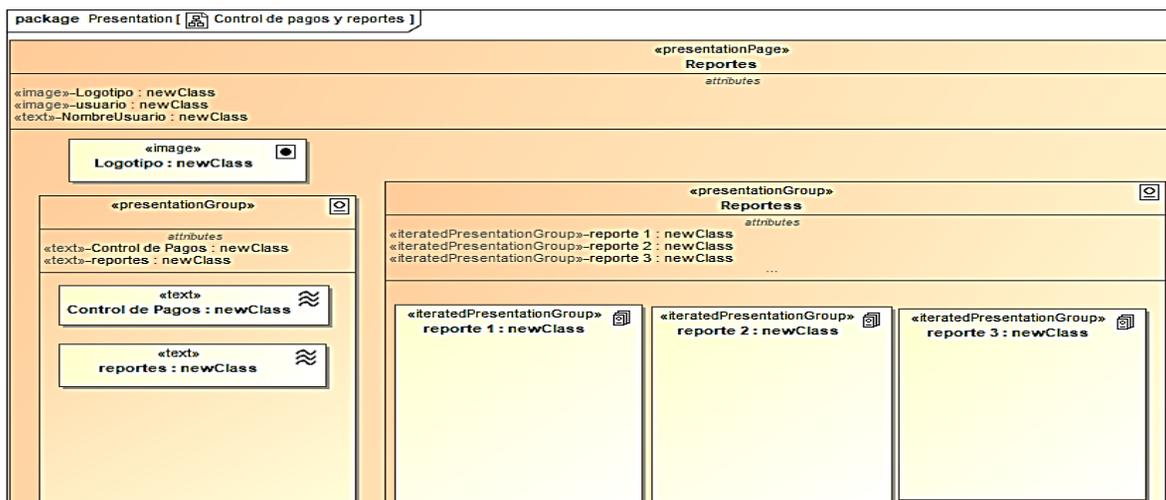


Figura 25: Modelo de Presentación del entorno de generación de reportes

Fuente: (Elaboración Propia, 2019)

3.8.3. FASE DE DESARROLLO

3.8.3.1. MODELO DE FLUJO DE PROCESO

El modelo de flujo de proceso que se encuentra en la figura 26, determina todo el proceso que sigue el usuario con el fin de controlar los pagos y generar reportes.

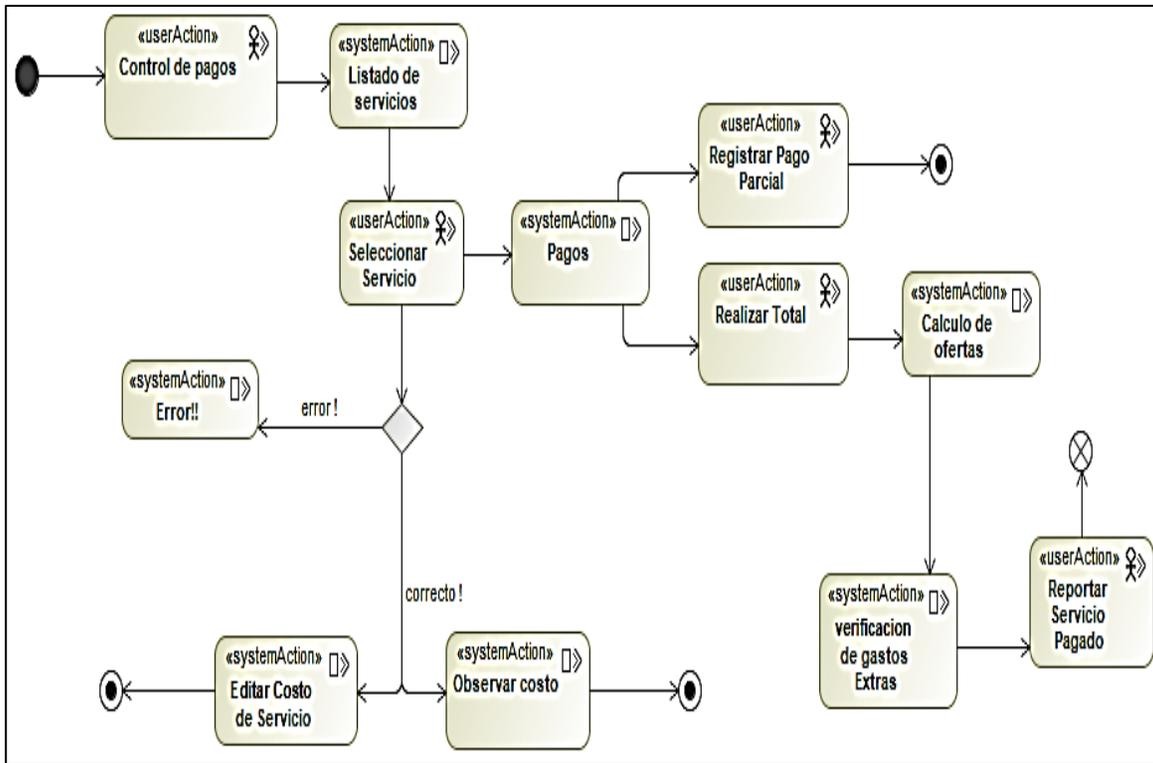


Figura 26: Modelo de flujo del control de pagos y reportes

Fuente: (Elaboración Propia, 2019)

3.8.4. FASE DE IMPLEMENTACIÓN

Para llegar a esta etapa se desarrolló el prototipo de la interfaz gráfica como se puede observar en la figura 27.

Siendo una captura con el sistema funcionando correctamente, de la siguiente manera:

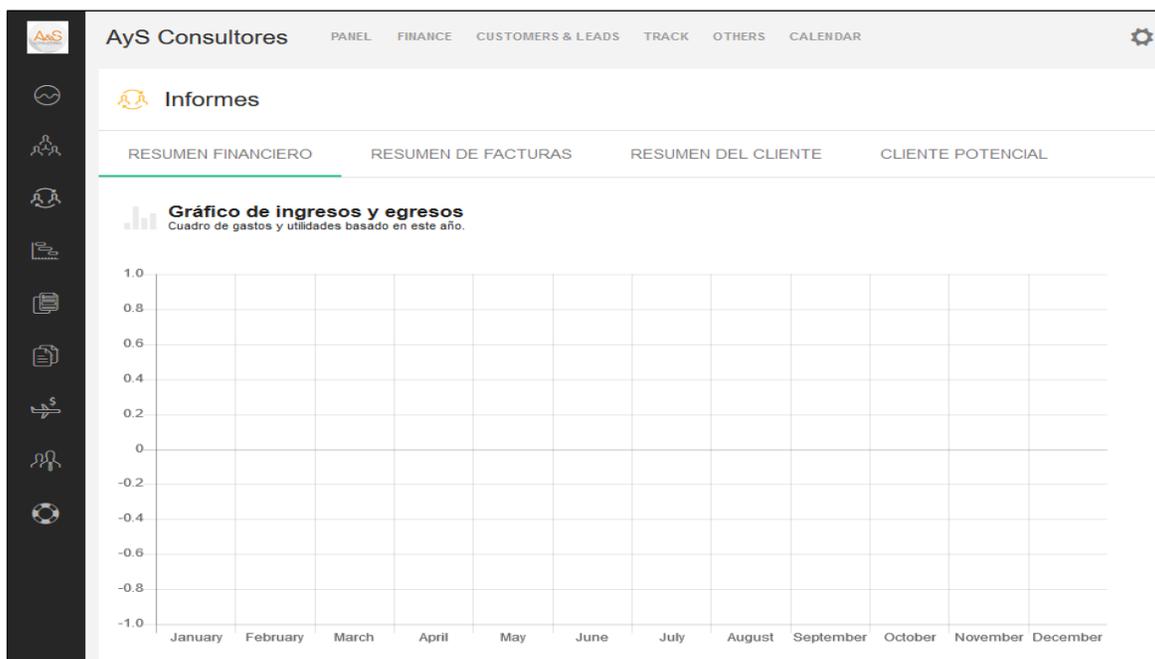


Figura 27: Modelo de flujo del control de pagos y reportes
Fuente: (Elaboración Propia, 2019)

3.9. PRUEBA DE ESFUERZO

Habiendo puesto en producción el sistema web, mediante un hosting para pruebas se realizaron las correspondientes pruebas de esfuerzo con el fin de evaluar el sistema, con la siguiente dirección web: http://www.aaps.gob.bo/consultora_ays/. Se procedió a la prueba de esfuerzo o prueba de estrés con la ayuda de la herramienta de software Apache JMETER.

En la figura 28 se muestra el resultado de la prueba con 20 usuarios ingresados al mismo tiempo en la página principal, administración de usuarios, la página de gestión de clientes, la página de control de pagos, y la página de reportes, haciendo un total de 100 usuarios ingresando y saliendo del sistema donde no causo ningún problema, generando un error aceptable de un 0% en todas las páginas anteriormente mencionadas.

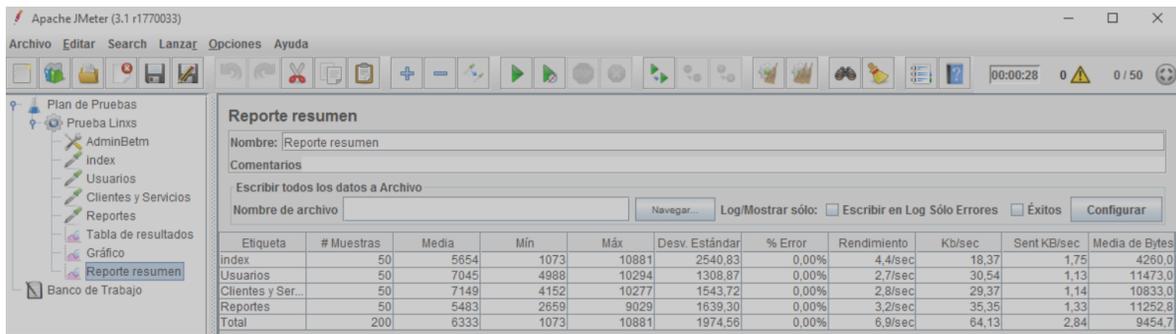


Figura 28: Resultado las pruebas de esfuerzo

Fuente: (Elaboración propia,2019)

Sin embargo, incrementando la cantidad de usuarios a 200 en cada página ya mencionada anteriormente, el sistema tiene problemas de lentitud inclusive en momentos se colapsa, tal como de detalla en la figura 29, pues con la cantidad de páginas visitadas, existirían alrededor de 1000 usuarios navegando a través del sistema, generando un error de entre 30% y 70%.

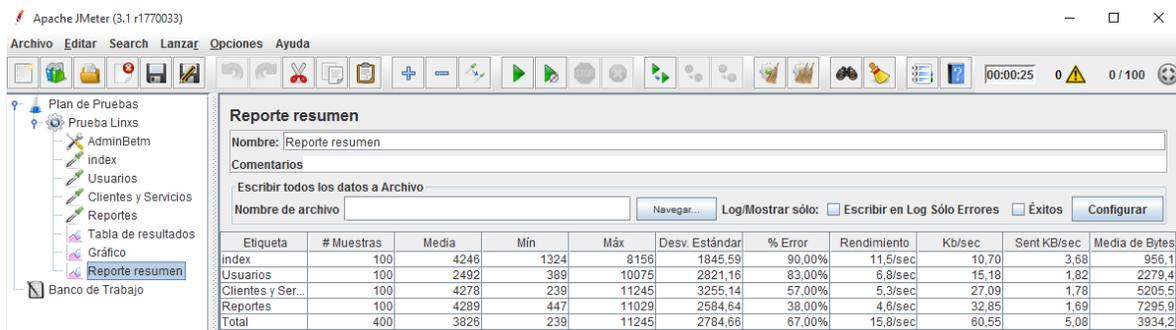


Figura 29: Resultado de la prueba con 100 usuarios en cada página.

Fuente: (Elaboración propia,2019)

Por todo lo anterior se concluye que el sistema soporta hasta 100 usuarios navegando en cada página, y si navegaran alrededor de 150 el sistema respondería con importante lentitud.

CAPÍTULO IV

CALIDAD Y SEGURIDAD

4.1. INTRODUCCIÓN

En el desarrollo de este capítulo se determina la calidad del sistema web en base a los parámetros de medición de la norma ISO 9126 y aplicando Web QEM el cual es una metodología cuantitativa que permite determinar un factor como calidad (o costo) de sitios y aplicaciones Web, donde hallaremos el punto función que nos servirá para el capítulo de costo beneficio, y las medidas de seguridad que se deben adoptar del lado del cliente y lado del servidor.

4.2. CALIDAD DE SOFTWARE

La ingeniería de software se diferencia de otras áreas, al no estar basada en leyes cuantitativas básicas, en su lugar se realiza un conjunto de medidas conocidas como métricas, las cuales proporcionan una referencia de la calidad algún producto de software (Pressman, 2002).

Para valorar la calidad de los productos de software o sistemas que se desarrollan se proporcionan información adecuada sobre los datos referentes de la misma a la calidad del producto, permitiendo una visión más profunda sobre el cumplimiento de los objetivos del proyecto (Pressman, 2002).

Medir la calidad de un software determina una de las tareas más complicadas que se presenta en el desarrollo de un sistema. Pero gracias a esta necesidad se fueron creando diferentes formas de medición de las mismas. Para el presente proyecto implementado utilizaremos el modelo de calidad de la ISO 9126, donde se medirá aspectos como la funcionalidad, fiabilidad, usabilidad y mantenibilidad (Pressman, 2002).

4.3. NORMA ISO 9126

La norma ISO 9126 (*International Standard Organization – Organización Internacional de Normalización*) es un estándar internacional para la evaluación de software, que nos ayudará a medir la calidad del sistema siguiendo los correspondientes criterios.

4.4. METODOLOGÍA WEB QEM

El principal objetivo de esta metodología cuantitativa consiste en evaluar y determinar el nivel de cumplimiento de las características especificadas para lo cual se analizan las preferencias elementales, parciales y globales. El resultado del proceso de evaluación (y eventualmente de comparación) puede ser interpretado como el grado de requerimientos de calidad satisfechos.

La metodología comprende una serie de fases y actividades que los evaluadores deben llevar a cabo en el proceso; entre las que podemos citar las siguientes actividades técnicas:

4.4.1. FASES DE WEB QEM

4.4.1.1. DEFINICIÓN DE LAS METAS DE EVALUACIÓN Y SELECCIÓN DEL PERFIL DE USUARIO

Los evaluadores deben definir metas y establecer alcances del proyecto de evaluación web. La evaluación se lleva a cabo tanto en la fase de desarrollo como en la fase operativa de un proyecto web, y así se puede valorar la calidad de un producto completo o bien se puede valorar la calidad de un conjunto de características y atributos de un componente. Los resultados podrán ser utilizados para comprender, mejorar, controlar o predecir la calidad de los productos.

Para propósitos de evaluación de dominios web, hemos considerados tres perfiles de usuario a un alto nivel de abstracción, a saber: visitantes, desarrolladores, y gerenciados.

4.4.1.2. DEFINICIÓN DE LOS REQUERIMIENTOS DE CALIDAD Y/O COSTO

Los evaluadores deben especificar atributos y características de calidad que estarán presentes en el proceso, agrupándolos en un árbol de requerimientos. Toma en cuenta las características de ISO.

- ✓ Metas de evaluación
- ✓ Perfil de usuario (gerente, desarrollador, visitante)
- ✓ Definición de requisitos no-funcionales

4.4.1.3. DEFINICIÓN DE CRITERIOS BÁSICOS Y PROCEDIMIENTOS DE MEDICIÓN

Los evaluadores deben definir una base una base de criterios para evaluación elemental, y realizar el proceso de medición y puntaje elemental.

Un criterio de evaluación elemental clara y específica de cómo medir atributos cuantificables. El resultado final es una preferencia o indicador elemental, el cual puede ser interpretado como el grado o resultado del requerimiento.

Por lo tanto, para cada métrica de un atributo necesitamos establecer un rango de valores aceptables y definir la función de criterio elemental, que producirá una correspondencia entre el valor de la métrica con el nuevo valor que representa la preferencia elemental.

4.4.1.4. DEFINICIÓN DE ESTRUCTURAS DE AGREGACIÓN E IMPLEMENTACIÓN DE LA EVALUACIÓN GLOBAL

En los pasos anteriores se definieron preferencias de calidad elemental realizados en el árbol de requerimientos. Por lo tanto, aplicando un mecanismo de agregación paso a paso, las preferencias elementales se pueden agrupar convenientemente para producir al final un esquema de agregación.

4.4.1.5. ANÁLISIS DE RESULTADOS Y RECOMENDACIONES

Una vez diseñado e implementado el proyecto de evaluación, el proceso culmina con la documentación de las conclusiones y recomendaciones. Los evaluadores analizan los resultados considerando las metas y el perfil de usuario establecidos.

A continuación de muestra las fases de WebQEM :

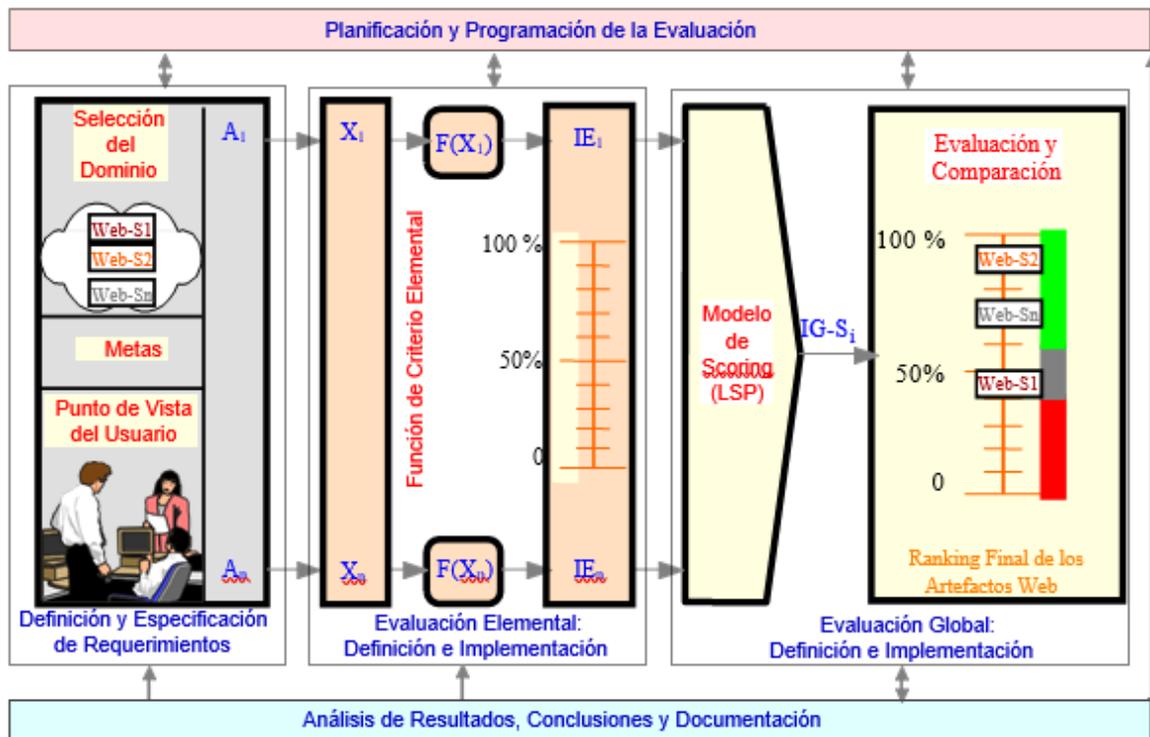


Figura 30: Fases de WebQEM

Fuente: (OLSINA, 2004)

4.4.2. DESARROLLO DE CARACTERÍSTICAS DE WEB QEM

WebQem toma las métricas del modelo de calidad ISO 9126-1 la cual da referencia a las siguientes características:

4.4.2.1. FUNCIONALIDAD

Métrica para obtener una valoración mediante el cálculo del punto función en base a la evaluación de un conjunto de características y capacidades que debe cumplir el sistema:

FACTORES DE COMPLEJIDAD	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y de recuperaciones fiables?				x			3
¿Se requiere comunicación de datos?					x		4
¿Existen funciones de procesamiento distribuido?					x		4
¿Es crítico el rendimiento?				x			3
¿Se ejecutará el sistema con un entorno operativo existente y fuertemente utilizado?					x		4
¿Requiere el sistema entrada de datos interactiva?						x	5
Facilidad Operativa					x		4
¿Se actualiza los archivos maestros de forma interactiva?					x		4
¿Son complejas las entradas, las salidas, los archivos o las peticiones?						x	5
Procesamiento interno complejo				x			3
Diseño de código reutilizable						x	5
Facilidad de Instalación					x		4
¿Soporta múltiples instalaciones en diferentes sitios?						x	5
Facilidad de cambios						x	5
Factor de ajuste de complejidad							58

Figura 31: Factor de ajuste de complejidad.

Fuente: (Elaboración propia, 2020)

La funcionalidad es medida a través del punto función (PF), que proporciona una medida objetiva, cuantitativa y auditable del tamaño de la aplicación, basada en la visión del usuario de la aplicación (Pressman, 2002).

Para calcular el punto función se utiliza la siguiente relación:

$$PF = Cuenta\ Total * (X + Min\ (Y) * \sum Fi)$$

Dónde:

PF: Medida de funcionalidad

Cuenta Total: Es la suma de los siguientes datos (Nro. de entradas, Nro. de salidas, Nro. de peticiones, Nro. de archivos, Nro. de interfaces externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%.

Min (Y): Error mínimo aceptable al de la complejidad.

$\sum Fi$: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Para calcular el PF se usa la siguiente ecuación:

$$PF = cuenta\ total * (X + Min\ (Y) * \sum Fi)$$

$$PF = cuenta\ total * [0.65 + (0.01 * \sum Fi)]$$

Reemplazando los valores obtenidos en la tabla 31, se obtiene el siguiente resultado:

$$PF = 154.98$$

A continuación, calculamos el PF ideal:

$$PF_{ideal} = 126 * [0.65 + (0.01 * 70)]$$

$$PF_{ideal} = 170.1$$

Entonces la funcionalidad del sistema es:

$$Funcionalidad = 91.1\%$$

Con el resultado obtenido se puede interpretar que **9 de cada 10 personas**, consideran que el sistema responde de manera óptima a las funcionalidades requeridas por la empresa.

4.4.2.2. CONFIABILIDAD

Para determinar la confiabilidad de un sistema, se toma en cuenta las fallas que puedan ocurrir en el sistema en un tiempo determinado. En el desarrollo de software las fallas son más que todo por diseño e implementación. Para medir el tiempo medio entre fallos (TMEF) se usará la siguiente formula:

$$TMEF = TMDF + TMDR$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Se estima que un fallo puede ocurrir cada 20 días hábiles y su reparación en promedio pueda tomar 1 hora después de haber entregado una nueva funcionalidad del sistema, entonces:

$$TMEF = (20 \cdot 8) + 1$$

$$TMEF = 161 \text{ horas}$$

Por lo que la disponibilidad es un buen indicador de fiabilidad, en base de la siguiente fórmula se tiene:

$$\text{Disponibilidad} = 99,4\%$$

Con lo que se llega a la conclusión de que el sistema tiene un **99.4% de confiabilidad**.

4.4.2.3. USABILIDAD

La usabilidad representa facilidad de uso que el usuario final percibirá del sistema. Esta métrica nos muestra el esfuerzo necesario para aprender a manipular el sistema (Pressman, 2002).

La tabla 9, muestra los resultados obtenidos en base a preguntas propuestas a los usuarios del sistema.

Nro.	Pregunta	Valor 0 - 100
1	¿Es entendible?	91
2	¿Las pantallas son agradables a la vista del usuario?	95
3	¿Es fácil de aprender?	92
4	¿Contiene información necesaria?	90
5	¿Facilita su trabajo?	86
6	¿La navegabilidad es fluida?	94
Promedio		91.3

Tabla 9: Preguntas para obtener el grado de usabilidad

Fuente: (Elaboración propia, 2020)

Entonces la **usabilidad del sistema seria del 91.3%**, lo que indica que 9 de cada 10 usuarios pueden utilizar el sistema con facilidad.

4.4.2.4. MANTENIBILIDAD

Para la evaluación de la mantenibilidad, se desarrolló algunas preguntas, estas preguntas son valoradas en porcentaje por el desarrollador del sistema al momento de la culminación del proyecto. Este valor tiene consideración por la experiencia y la forma de trabajo de cada programador, el mismo puede ser relativo respecto a otros desarrolladores (Largo y Marín, 2005).

La tabla 10 muestra las preguntas y los resultados obtenidos en la evaluación de mantenibilidad, estas preguntas se las hizo a todo el equipo de desarrollo del sistema.

FACTOR DE AJUSTE	VALOR
¿Se puede modificar el sistema?	97%
¿Deja identificar las partes que deben ser modificadas?	95%
¿Permite implementar una modificación específica?	93%
¿Presenta efectos inesperados como posibles errores?	98%
Total	95.75%

Tabla 10: Evaluación de mantenibilidad

Fuente: (Elaboración propia, 2020)

El resultado de la **mantenibilidad es de 95.75%**, lo que significa que el esfuerzo necesario para realizar mantenimiento al sistema es mínimo.

4.4.2.5. PORTABILIDAD

La portabilidad es la capacidad con que un software puede ser llevado de un entorno a otro, considera la facilidad de instalación, ajuste y adaptación al cambio. Para medir la portabilidad del sistema usaremos la siguiente relación:

Si:

- $GP > 0$, la portabilidad es más rentable que el re-desarrollo.
- $GP < 0$, el re-desarrollo es más rentable que la portabilidad.
- $GP = 0$, la portabilidad es perfecta.

Los recursos necesarios para llevar el sistema a otro entorno son: servicio de hosting para alojar el código fuente, la base de datos, dominio para la URL, conexión FTP, conexión a internet, conexión intranet, responsivo, espacio almacenamiento, y los recursos necesarios para crear el sistema son: IDE de desarrollo, html5 y frameworks (codeigniter v3, jquery, bootstrap 3, css3,).

Por lo que los valores obtenidos son:

$$ET = 9 \text{ y } ER = 10$$

$$GP = 0.90$$

Con el resultado obtenido sabemos que el grado de portabilidad es del 90%, entonces la portabilidad del sistema es más rentable que en el caso se desee volver a desarrollar.

4.4.2.6. CALIDAD GLOBAL

Para poder obtener la calidad global del sistema, se saca la media de todas las medidas expresadas en porcentaje hasta el momento, funcionalidad, confiabilidad, usabilidad, mantenibilidad y portabilidad (Pressman, 2002).

CRITERIOS	RESULTADO
Funcionalidad	91.1%
Confiabilidad	99.4%
Usabilidad	91.3%
Mantenibilidad	95.75%
Portabilidad	90%
Calidad Global	93.51%

Tabla 11: Calidad global

Fuente: (Elaboración propia, 2020)

El resultado de la **calidad global es de 93.51%**, con ese resultado concluimos que 9 de cada 10 usuarios consideran al sistema web de calidad.

4.5. SEGURIDAD

En este apartado se explica el uso del algoritmo de encriptación (o cifrado) sha-2 “secure Hash Algorithm” como medio para cifrar la información del cliente, como característica de seguridad en la base de datos, además se explica el uso de los SSL “Secure Sockets Layer” como certificado de seguridad, como característica de seguridad en el servidor web.

4.5.1. ALGORITMO DE ENCRIPCIÓN SHA-2

SHA-2 es un conjunto de funciones hash criptográficas (SHA-224, SHA-256, SHA-384, SHA-512) diseñadas por la Agencia de Seguridad Nacional (NSA) y publicada en 2001 por el Instituto Nacional de Estándares y Tecnología (NIST) como un Estándar Federal de Procesamiento de la Información (FIPS).

Una función hash es un algoritmo que transforma ("digiere") un conjunto arbitrario de elementos de datos, como puede ser un fichero de texto, en un único valor de longitud fija (el "hash"). El valor hash calculado puede ser utilizado para la verificación de la integridad de copias de un dato original sin la necesidad de proveer el dato original. Esta

irreversibilidad significa que un valor hash puede ser libremente distribuido o almacenado, ya que sólo se utiliza para fines de comparación. SHA significa algoritmo de hash seguro. SHA-2 incluye un significativo número de cambios respecto a su predecesor, SHA-1; y consiste en un conjunto de cuatro funciones hash de 224, 256, 384 o 512 bits.

4.5.2. VARIANTES DEL SHA-2

Valores hash para una cadena de caracteres vacía.

```
SHA224("")
```

```
0x d14a028c2a3a2bc9476102bb288234c415a2b01f828ea62ac5b3e42f
```

```
SHA256("")
```

```
0x e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

Incluso un pequeño cambio en el mensaje (con una apabullante probabilidad) dará como resultado un hash completamente distinto, debido al efecto avalancha. Por ejemplo, añadiendo un punto al final de la frase:

```
SHA224("The quick brown fox jumps over the lazy dog")
```

```
0x 730e109bd7a8a32b1cb9d9a09aa2325d2430587ddbc0c38bad911525
```

```
SHA224("The quick brown fox jumps over the lazy dog.")
```

```
0x 619cba8e8e05826e9b8c519c0a5c68f4fb653e8a3d8aa04bb2c8cd4c
```

4.5.3. CAPA DE CONEXIÓN SEGURA SSL

SSL "Secure Sockets Layer" es un protocolo diseñado para permitir que las aplicaciones transmitan información de manera segura. Las aplicaciones que utilizan el protocolo Secure Sockets Layer conocen el flujo de envío y recepción de claves cifradas con otras aplicaciones,

así como la manera de cifrar y descifrar los datos enviados entre las mismas.

4.5.3.1. IMPLEMENTACIÓN DE CERTIFICADO SSL CON LET'S ENCRYPT Y CERBOT

Let's Encrypt es una autoridad de certificación (CA) automatizada y abierta operada por Internet Security Research Group (ISRG) y fundada por la Electronic Frontier Foundation (EFF), la Mozilla Foundation y otros.

Proporciona certificados SSL / TLS gratuitos que se utilizan comúnmente para cifrar las comunicaciones con fines de seguridad y privacidad, el caso de uso más notable es HTTPS.

Let's Encrypt se basa en el protocolo ACME (Entorno automático de gestión de certificados) para emitir, revocar y renovar certificados. Certbot es una utilidad gratuita y de código abierto que se utiliza principalmente para administrar certificados SSL / TLS de la autoridad de certificación Let's Encrypt. Está disponible para la mayoría de los sistemas operativos UNIX y similares a UNIX, incluidos GNU / Linux, FreeBSD, OpenBSD y OS X.

La mejor forma de utilizar Let's Encrypt sin acceso shell es usando el soporte incorporado del proveedor de hospedaje. Si el proveedor de hospedaje ofrece soporte para Let's Encrypt, es posible solicitar un certificado gratis en su nombre, instalarlo, y mantenerlo actualizado automáticamente.

Para algunos proveedores de hospedaje, esto es un ajuste de configuración que tienes que prender. Otros proveedores automáticamente solicitan e instalan certificados para todos sus clientes.

A continuación, se muestra un ejemplo de certificado activo.

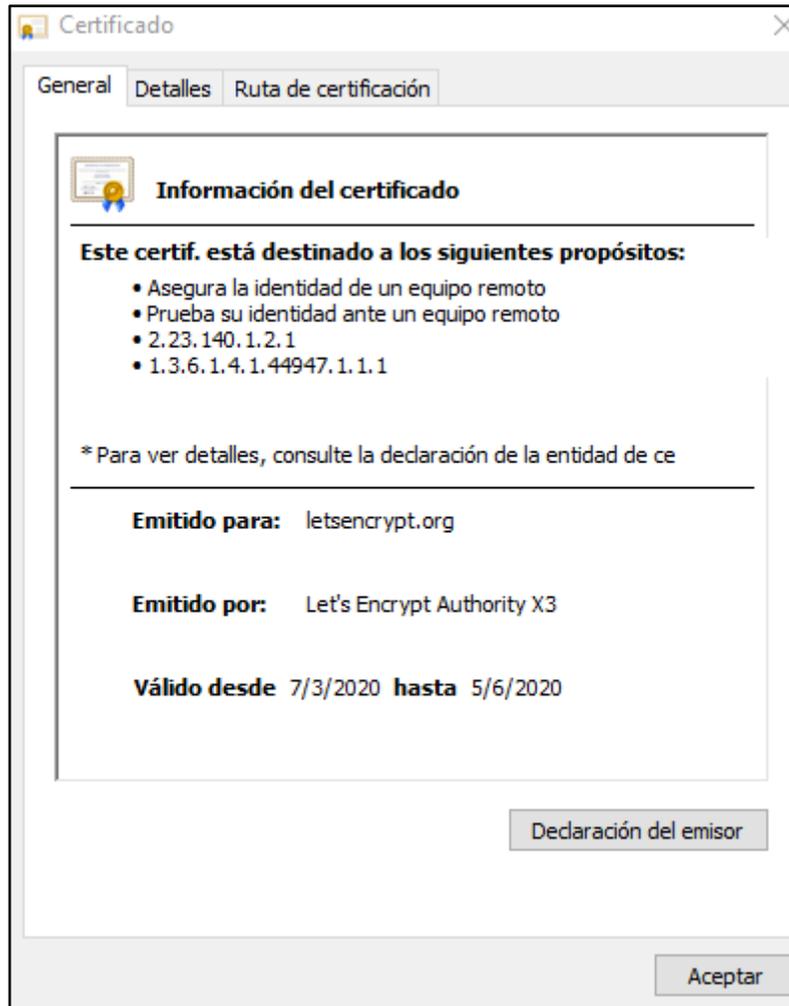


Figura 32: Certificación *Let's Encrypt* activa

Fuente: (Elaboración propia, 2020)

CAPÍTULO V COSTO BENEFICIO

5.1. INTRODUCCIÓN

En el desarrollo del presente capítulo, se dará a conocer a la empresa Consultora Contable AyS S.R.L. que el desarrollo y la posterior implementación del sistema es sustentable y genera beneficios mediante la evaluación efectiva de la relación costo-beneficio, con el fin de tomar la mejor decisión.

Para tal efecto se realiza el análisis de costo del sistema usando el método COCOMO II y en base al costo del sistema y otros gastos se determina la rentabilidad del sistema con el cálculo del valor neto actual (VAN) y la tasa interna de retorno (TIR).

Posterior a los análisis necesarios para la obtención de los resultados esperados estaremos en la capacidad de afirmar si el proyecto es viable, redituable y comprobar que es buena opción invertir en el proyecto.

5.2. COCOMO II

El Modelo Constructivo de Costes (COCOMO) es un modelo matemático de base empírica, utilizando para la estimación de costes de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software:

a) **Básico**

estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (*costdrivers*): el tamaño del software y el modo de desarrollo.

b) Intermedio

Comparado con el modelo anterior, éste provee un nivel de detalle y precisión superior, el cual es más apropiado para la estimación de costos en etapas de mayor especificación. COCOMO Intermedio incorpora un conjunto de quince variables de predicción que toman en cuenta las variaciones de costos no consideradas por COCOMO Básico.

c) Detallado:

El Modelo Detallado provee los medios para generar estimaciones con mayor grado de precisión y detalle. Difiere del Modelo Intermedio en dos aspectos principales que ayudan a superarlas limitaciones: jerarquía de niveles del producto y multiplicadores de esfuerzo.

COCOMO II es aquel que permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y beneficio. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. Cada uno de ellos orientados a sectores específicos del mercado de desarrollo de software y las distintas etapas del desarrollo de software.

Por otro lado, también se define tres modos de desarrollo o tipo de proyectos:

Modos de Desarrollo	a	b	c	d
Orgánico	3,2	1,05	2,5	0,38
Semi- acoplado	3	1,12	2,5	0,35
Empotrado	2,8	1,2	2,5	0,32

Tabla 12: Tabla de modos de desarrollo

Fuente: (COCOMO II, 2000)

a) **Orgánico**

En esta clasificación se encuentran proyectos desarrollados en un ambiente familiar y estable. El producto a elaborar es relativamente pequeño y requiere pocas innovaciones tecnológicas en lo que refiere a algoritmos, estructuras de datos e integración de hardware.

b) **Semi-acoplado**

Es un modelo para productos de software de tamaño y complejidad media. Las características de los proyectos se consideran intermedias a las de los modos Orgánico y Empotrado.

c) **Empotrado**

En esta clasificación están incluidos proyectos de gran envergadura que operan en un ambiente complejo con altas restricciones de hardware, software y procedimientos operacionales, tales como los sistemas de tráfico aéreo.

Estos métodos de estimación de costos se engloban en el grupo de modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Para nuestro caso particular usaremos el modelo intermedio, pues se adecua a nuestro proyecto para lograr una buena precisión, además que se aplicara el modo de desarrollo semi-acoplado para proyecto de complejidad intermedia es decir con líneas de código menores a los 300 KDLC.

5.3. COSTO DEL SISTEMA

En el proceso de estimación se emplearán las siguientes formulas:

$$E = (a)(KLDC)^b * (FAE)$$

$$T = (c) * E^d$$

$$P = E/T$$

Siendo:

- ✓ E: Esfuerzo
- ✓ T: Tiempo
- ✓ P: Personal
- ✓ KLDC: Kilo líneas de código (Miles de líneas de código)
- ✓ FAE: Factor de Ajuste de Esfuerzo
- ✓ a, b, c, d: Dependientes del modo de desarrollo, según tabla 5.1
- ✓ PF: puntos de función

Hallando la variable KLDC:

Con PF = 261,36 y las líneas por cada PF equivalen a 29 para el lenguaje PHP, según la tabla 13

Lenguaje	LDC/PF
Ensamblador	320
C	150
COBOL	105
Pascal	91
PHP	29
C++	64
Visual Basic	32
SQL	12

Tabla 13: Puntos fusión COCOMO II

Fuente: (COCOMO II, 2000)

Por tanto:

$$KLDC = (PF * \text{l\u00edneas de c\u00f3digo por cada PF}) / 1000$$

$$KLDC = (261.36 * 29) / 1000$$

$$KLDC = 7.58$$

Y por otro lado tambi\u00e9n hemos de hallar la variable FAE, la cual se obtiene mediante la multiplicaci\u00f3n de los valores evaluados en los diferentes 15 conductores de costos que se observan en la tabla 5.3.

CONDUCTORES DE COSTO	VALORACI\u00d3N					
	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra alto
Fiabilidad requerida del software	0,7 5	0,8 8	1.00	1,1 5	1,4	-
Tama\u00f1o de la base de datos	-	0,9 4	1.00	1,0 8	1,1 6	-
Complejidad del producto	0,7	0,8 5	1.00	1,1 5	1,3	1,65
Restricciones del tiempo de ejecuci\u00f3n	-	-	1.00	1,1 1	1,3	1,66
Restricciones del almacenamiento principal	-	-	1.00	1,0 6	1,2 1	1,56
Volatilidad de la m\u00e1quina virtual		0,8 7	1.00	1,1 5	1,3	-
Tiempo de respuesta del ordenador	-	0,8 7	1.00	1,0 7	1,1 5	-
Capacidad del analista	1,4 6	1,1 9	1.00	0,8 6	0,7 1	-
Experiencia en la aplicaci\u00f3n	1,2 9	1,1 3	1.00	0,9 1	0,8 2	-
Capacidad de los programadores	1,4 2	1,1 7	1.00	0,8 6	0,7	-
Experiencia en S.O. utilizado	1,2 1	1,1	1.00	0,9	-	-
Experiencia en el lenguaje de programaci\u00f3n	1,1 4	1,0 7	1.00	0,9 5	-	-

Prácticas de programación modernas	1,2 4	1,1	1.00	0,9 1	0,8 2	-
Utilización de herramientas software	1,2 4	1,1	1.00	0,9 1	0,8 3	-
Limitaciones de planificación del proyecto	1,2 3	1,0 8	1.00	1,0 4	1,1	-

Tabla 14: Tabla modo de conductores de costo

Fuente: (COCOMO II, 2000)

$$FAE = 1,15 * 1,00 * 0,85 * 1,11 * 1,00 * 1,07 * 0,86 * 0,82 * 0,70$$

$$* 1,00 * 0,91 * 1,08$$

$$FAE = 0,54$$

Cálculos del esfuerzo de desarrollo:

$$E = 3 * 7,58^{1,12} * 0,54$$

$$E = 14,74 \text{ personas/mes}$$

En este caso las personas se toman como a los desarrolladores que hicieron los esfuerzos por mes.

A continuación, se realiza el cálculo del tiempo de desarrollo:

$$T = 2,5 * (14,74)^{0,35}$$

$$T = 6,41 \text{ meses}$$

A continuación, se realiza el cálculo del personal promedio de desarrollo:

$$P = 14,74 / 6,41$$

$$P = 2,3 \text{ personas}$$

A continuación, se realiza el cálculo de productividad que desarrolla las líneas de código con el esfuerzo en este caso también personas se toman como desarrolladores.

$$PR = LDC/E, \text{ donde PR: productividad}$$

$$PR = 2314/14,74$$

$$PR = 156,99 \text{ LDC/personas mes}$$

La interpretación de este resultado da a conocer que es necesario un equipo de tres personas alrededor de seis a siete meses.

5.4. ESTIMACIÓN DEL COSTO

Se realiza la estimación de COCOMO II mediante su software. A continuación, se muestra la siguiente figura sobre COCOMO II:

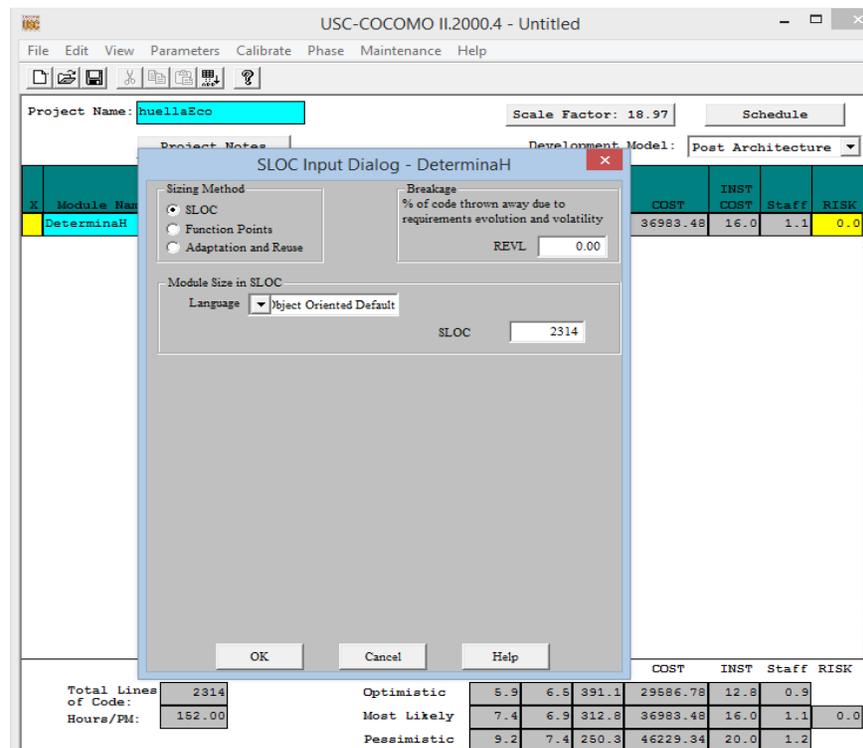


Figura 33: Líneas de código software COCOMO

Fuente: (COCOMO II, 2000)

Una vez puesta las líneas de código nos sale unas tablas las cuales podemos modificar: module name (nombre del proyecto), module size (líneas de código), labor rate (Costo del producto), lo cual nos ayudará a definir la tabla mostrada en la parte superior de la figura 5.2. Que calcula: effort (esfuerzo), sched (tiempo), prod (productividad), cost (costo), Inst (costo intermedio), staff (cosas). Para lo otro nos define el estimado para ello existen tres estimaciones las cuales son: optimistic (óptimo), most likely (lo más probable) y pessimistic (pésimo).

The screenshot shows the USC-COCOMO II.2000.4 software interface. The window title is "USC-COCOMO II.2000.4 - Untitled". The menu bar includes File, Edit, View, Parameters, Calibrate, Phase, Maintenance, and Help. The interface includes a toolbar, a "Project Name" field with the value "huellaEco", a "Scale Factor" of 18.97, a "Schedule" button, and a "Project Notes" button. The "Development Model" is set to "Post Architecture".

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	Language	NCM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	DeterminaH	S:2314	5000.00	1.00	Non-Specified	7.4	7.4	312.8	36983.48	16.0	1.1	0.0

	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
Optimistic	5.9	6.5	391.1	29586.78	12.8	0.9		
Most Likely	7.4	6.9	312.8	36983.48	16.0	1.1	0.0	
Pessimistic	9.2	7.4	250.3	46229.34	20.0	1.2		

Summary statistics at the bottom left:

Total Lines of Code:	2314
Hours/PM:	152.00

Figura 34: Estimación de costo software COCOMO
Fuente: (COCOMO II, 2000)

Una vez sacado esto se determina el punto fusión en este proyecto se manejó el lenguaje PHP el cual en el software se define como Object Oriented Default y tenemos que llenar los demás datos según al proyecto.

A continuación, se muestra en la siguiente figura la parte de punto de fusión.

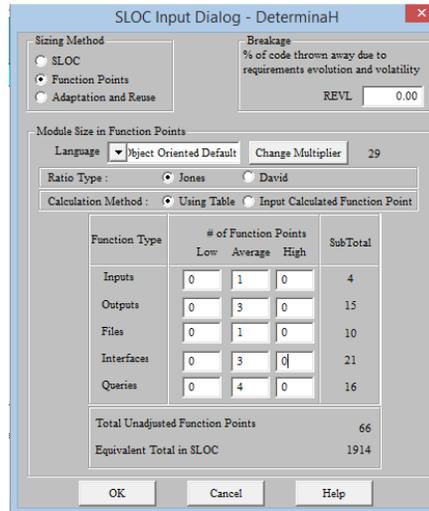


Figura 35: Punto fusión software COCOMO
Fuente: (COCOMO II, 2000)

Una vez puesto punto fusión volvemos al proyecto y vemos que algunos datos cambiaron en cuanto al costo. A continuación, se muestra la figura sobre estimación de costos.

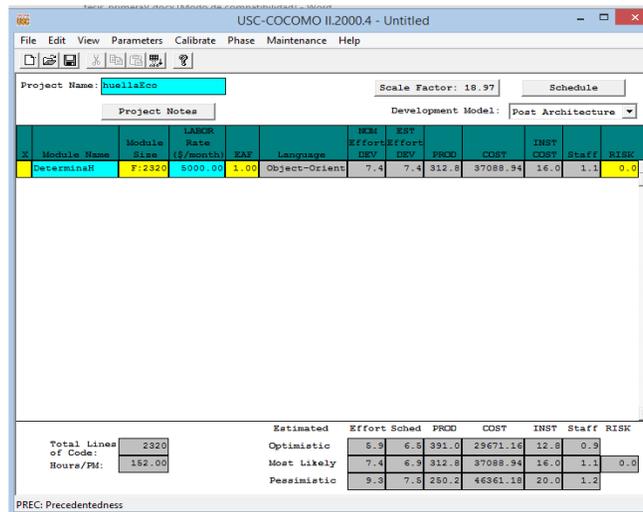


Figura 36: Estimación de costo software COCOMO
Fuente: (COCOMO II, 2000)

Ya teniendo la estimación de costos se puede modificar la parte de *Scale Factor* (factores de escala). Esto depende del desarrollo del modelo en este caso se utiliza *post architecture* en los factores de escala se realiza los ajustes finos.

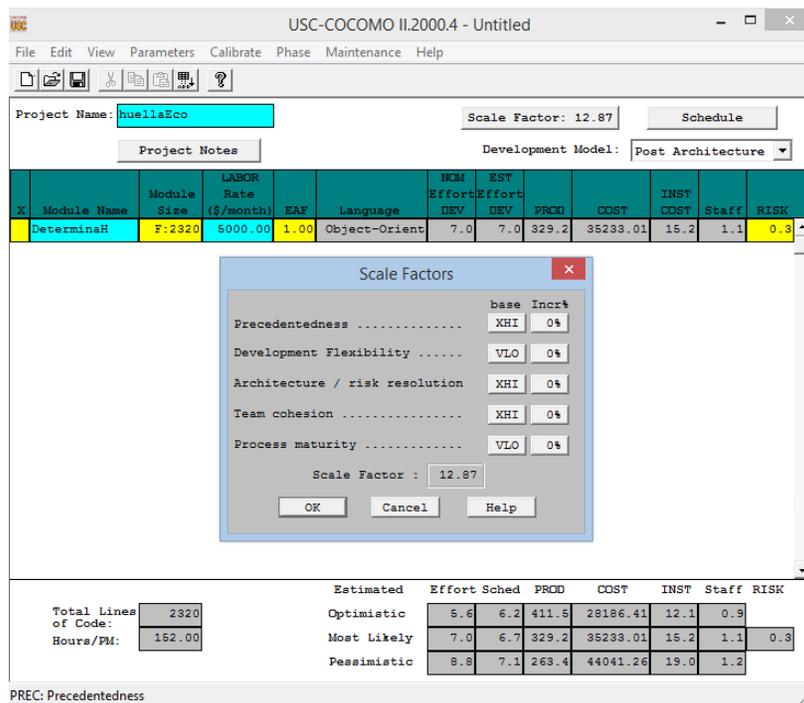


Figura 37: Factores de escala software COCOMO
Fuente: (COCOMO II, 2000)

A continuación, se muestra una tabla que hace referencia al monto total.

MATERIAL	COSTO (USD)	COSTO (BOLIVIANOS)
Material de escritorio	30	208,8
Computadora	800	5568
Desarrollo del software	5000	34800
Internet	30	208,8
Servidor web	120	835,2
Mantenimiento	1000	6960

Tabla 15: Costo total desarrollo de software
Fuente: (Elaboración propia,2020)

5.5. CÁLCULO BENEFICIO VAN Y TIR

El VAN y el TIR son dos herramientas financieras procedentes de las matemáticas financieras que nos permiten evaluar la rentabilidad de un proyecto de inversión, entendiéndose por proyecto de inversión no solo como la creación de un nuevo negocio, sino también, como inversiones que podemos hacer en un negocio en marcha, tales como el desarrollo de un nuevo producto, la adquisición de nueva maquinaria, el ingreso en un nuevo rubro de negocio, etc.

5.5.1. VALOR ACTUAL NETO (VAN)

Es el valor actual de los beneficios netos que genera el proyecto durante toda su vida. El valor actual neto (VAN) es un método de valoración de inversiones que puede definirse como la diferencia entre el valor actualizado de los cobros y de los pagos generados por una inversión. Proporciona una medida de la rentabilidad del proyecto analizado en valor absoluto, es decir expresa la diferencia entre el valor actualizado de las unidades monetarias cobradas y pagadas.

A continuación, se muestra la tabla de la caja de flujo en años con una inversión de 48580.

AÑO	FLUJO DE CAJA (BS)
2016	6000
2017	9000
2018	10000
2019	15000
2020	30000

Tabla 16: Flujo de caja por años
Fuente: (Elaboración propia,2020)

formula:

$$VAN = -A + \sum_{i=1}^n \frac{Q_i}{(1 + K)^i}$$

Donde:

Q_1, Q_2, \dots, Q_n : Flujos de caja

k: Tasa de descuento seleccionada

A: desembolso inicial o inversión inicial n: vida útil del proyecto

i: período

Desarrollando:

$$\begin{aligned}VAN &= -A + \frac{Q_1}{(1+K)^1} + \frac{Q_2}{(1+K)^2} + \dots + \frac{Q_n}{(1+K)^n} \\VAN &= -48580 + \frac{6000}{(1+0,1)^1} + \frac{9000}{(1+0,1)^2} + \frac{10000}{(1+0,1)^3} + \frac{15000}{(1+0,1)^4} \\&\quad + \frac{30000}{(1+0,1)^5} \\VAN &= -48580 + 5454,54 + 7438,02 + 7513,15 + 10245,20 \\&\quad + 18627,64 \\VAN &= 698,55\end{aligned}$$

Interpretación VAN:

VAN > 0; se recomienda pasar a la siguiente etapa del proyecto

VAN = 0; es indiferente realizar la inversión

VAN < 0; se recomienda desecharlo o postergarlo

✓ Como el VAN tiene un resultado mayor a cero, entonces el proyecto es aceptable.

5.5.2. TASA INTERNA DE RETORNO (TIR)

Es una tasa porcentual que indica la rentabilidad promedio anual que genera el capital que permanece invertido en el proyecto. También se define como la tasa de descuento que hace que el VAN = 0, su valor no depende del tiempo y representa el máximo costo que el inversionista podría pagar por el capital prestado.

Desarrollando:

con $VAN = 0$

$$0 = -48580 + \frac{6000}{(1 + 0,1)^1} + \frac{9000}{(1 + 0,1)^2} + \frac{10000}{(1 + 0,1)^3} + \frac{15000}{(1 + 0,1)^4} + \frac{30000}{(1 + 0,1)^5}$$

Se logra calcular el TIR con el siguiente valor TIR (V1:V6)

$K=0,1$, esto equivale al 10%

Interpretación TIR:

TIR > k se recomienda pasar a la siguiente etapa

TIR = k es indiferente invertir

TIR < k se recomienda su rechazo o postergación

- ✓ Notamos que el resultado es igual a cero por lo tanto según los parámetros, el proyecto es indiferente invertir.

5.6. COSTO BENEFICIO

Tomando en cuenta los resultados anteriores (VAN y TIR) se toma en cuenta los valores de $k = 0,1$ y $VAN = 698,55$.

Formula:

$$\text{Costo Beneficio} = \frac{C}{B}$$

Donde:

C: Costo

B: Beneficio

Desarrollando:

$$\text{Costo Beneficio} = \frac{698,55}{0,1}$$

$$\text{Costo Beneficio} = 6,98 \text{ Bolivianos}$$

Interpretación:

Como el resultado obtenido (6,98 Bs.-) redondeando 6, llegamos a la siguiente conclusión:

- ✓ Por cada 6 bolivianos invertidos se obtiene una ganancia de 0,98 centavos, por tanto, el **proyecto es rentable**.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Habiendo culminado el desarrollo y la implementación del Sistema de para la empresa Consultora AyS S.R.L., donde se aplicó todo el conocimiento de metodologías de análisis y diseño de software, todo en base a los requerimientos planteados por el beneficiado se puede concluir que se cumplió con los objetivos planteados de la siguiente manera:

- ✓ Se realizó el proceso para realizar el control y seguimiento de servicios contratados a la empresa consultora AyS S.R.L.
- ✓ Se implementó el control y permisos sobre la administración del sistema, de acuerdo al rol asignado a cada usuario.
- ✓ Se redujo costos o gastos comunes rutinarios que generaban pérdidas
- ✓ Se implementó el módulo de clientes y servicios donde los usuarios registrados según los roles asignados pueden acceder al detalle de información de clientes, donde pueden conocer sus formas de pago, estado de fechas de vencimiento y renovación de servicios.
- ✓ Se implementó el módulo de reportes, en base a diferentes criterios para la obtención de información donde los usuarios pueden obtener información semanal, mensual y anual de los servicios a los clientes.

6.2. RECOMENDACIONES

Se recomiendan varios aspectos para potenciar el uso y aprovechamiento del sistema web, bajo lo siguiente:

- ✓ Se recomienda utilizar la misma versión del *framework Codeigniter* a momento de mejorar el sistema pues mencionado *framework* es sumamente escalable y se adecua a diferentes contextos para su aplicación.
- ✓ La empresa AyS S.R.L. aún tiene necesidades en cuanto a registros contables y facturación electrónica, lo cual se recomienda no dejar pendiente y continuar automatizando los procesos de la empresa.
- ✓ Se recomienda implementar y adaptar al sistema el módulo de recursos humanos para tener control del personal de la empresa.
- ✓ Es recomendable realizar un correcto mantenimiento al sistema en lo que respecta a usuarios que dejaron de trabajar n la empresa y dar de baja en un momento correcto y siguiendo el procedimiento correcto.

BIBLIOGRAFÍA

- Bahit, E. (2011). Desarrollo Ágil con Kanban [en línea]. <<https://desarrolloweb.com/articulos/desarrollo-agil-kanban.html>> [consulta: 03 de junio de 2019]
- Condori, J. G. (2015). Sistema web de inscripciones y control de pagos caso: Kumon. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Corini, A. R. (2014). Sistema Web de Registro, Seguimiento y Control de Correspondencia Basado en BPM Caso: Carrera de Informática. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Cortez, L. I. (2014). Sistema web para el registro y administración de documentos de niños patrocinados Caso: CDI BO – 132. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Cueto, L. (2009). Software boliviano de exportación, a un BIT del negocio. Revista Emprendedores: 23-24.
- Huanca, S. P. (2015). Sistema web de control de pagos, citas e historiales clínicos caso: Clínica Dental Lavadent. Ingeniería de sistemas informáticos. La Paz, Universidad Mayor de San Andrés, Carrera de Informática.
- Mateo, A. (2018). ¿Qué es una aplicación Web? [en línea]. <<https://www.neosoft.es/blog/que-es-una-aplicacion-web>> [consulta: 04 de junio de 2019].

Pressman, F. A. (2002). Ingeniería del software un enfoque práctico. Quinta edición. México. McGraw-Hill Interamericana Editores S.A de C.V.

Suaznabar, R. (2017). Empresas de desarrollo de software en Cochabamba. Primera edición. Cochabamba. Talleres Gráficos “Kipus”

Wolf, D. (2011). Metodología UWE [en línea]. <<https://metodologiauwe.wordpress.com/author/danielthewolf1993>> [consulta: 22 de mayo de 2019]

Yucra, W. (2016). Sistema Web de Control y Seguimiento de Documentación Caso: Departamento de Infraestructura de la UMSA. Universidad Mayor de San Andrés. Carrera de Informática.

Sommerville, Ian. (2005). Ingeniería del software. Séptima edición. Madrid. Pearson Educación S.A.

Nolivos, G., & Coronel, F. (2010). Análisis, diseño, desarrollo e implementación de un sistema web para el control de un taller técnico automotriz en plataforma PHP MYSQL utilizando metodología web UWE para la empresa METROAUTOCENFRANCIA LTDA. Ecuador. Universidad de las Fuerzas Armadas. Carrera de Ingeniería de Sistemas e Informática.

Kniberg, Henrik y Skarin, Mattias. (2010). Kanban y Scrum – Obteniendo lo mejor de ambos. Estados Unidos. C4Media Inc.

Cabrera, A., Figueroa, R., y Solís, C. (2008). Metodologías Tradicionales Vs. Metodologías Ágiles. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.

Largo, C. & Marín, E. (2005). Guía Técnica para la Evaluación del Software. <http://www.puntoexe.com.co>.

Bermejo, M. (2010). El Kanban. España: Creative Commons.

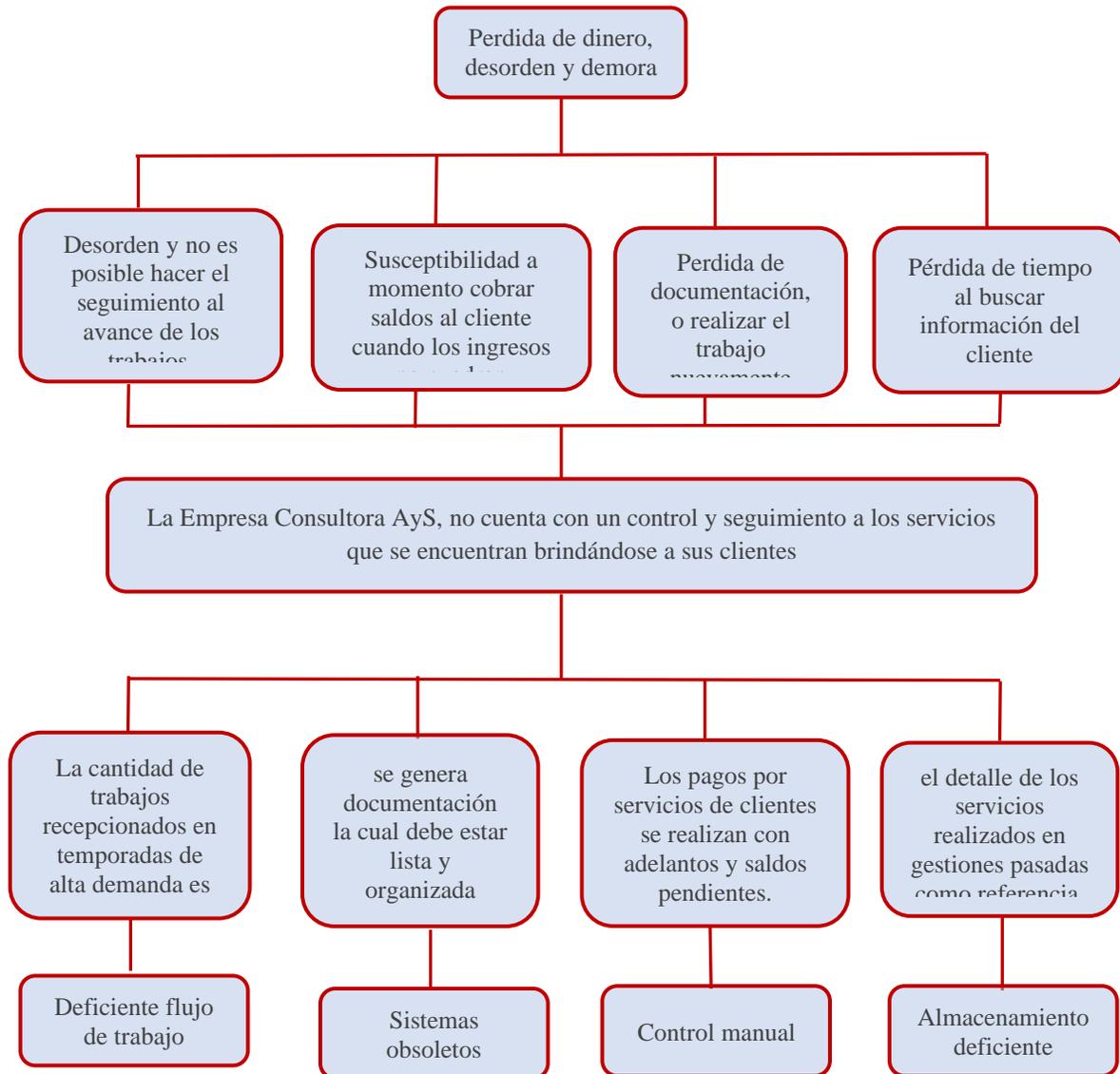
Gallardo, L. (3 de noviembre de 2012). Metodología UWE aplicada a mi solución informática de mi proyecto. Obtenido de Blogger.com: <http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicadami-solucion.html>.

Kniberg & Skrin. (2010). Kanban y Scrum - Obteniendo lo mejor de ambos. Estados Unidos de América: C4Media, editores de InfoQ.com.

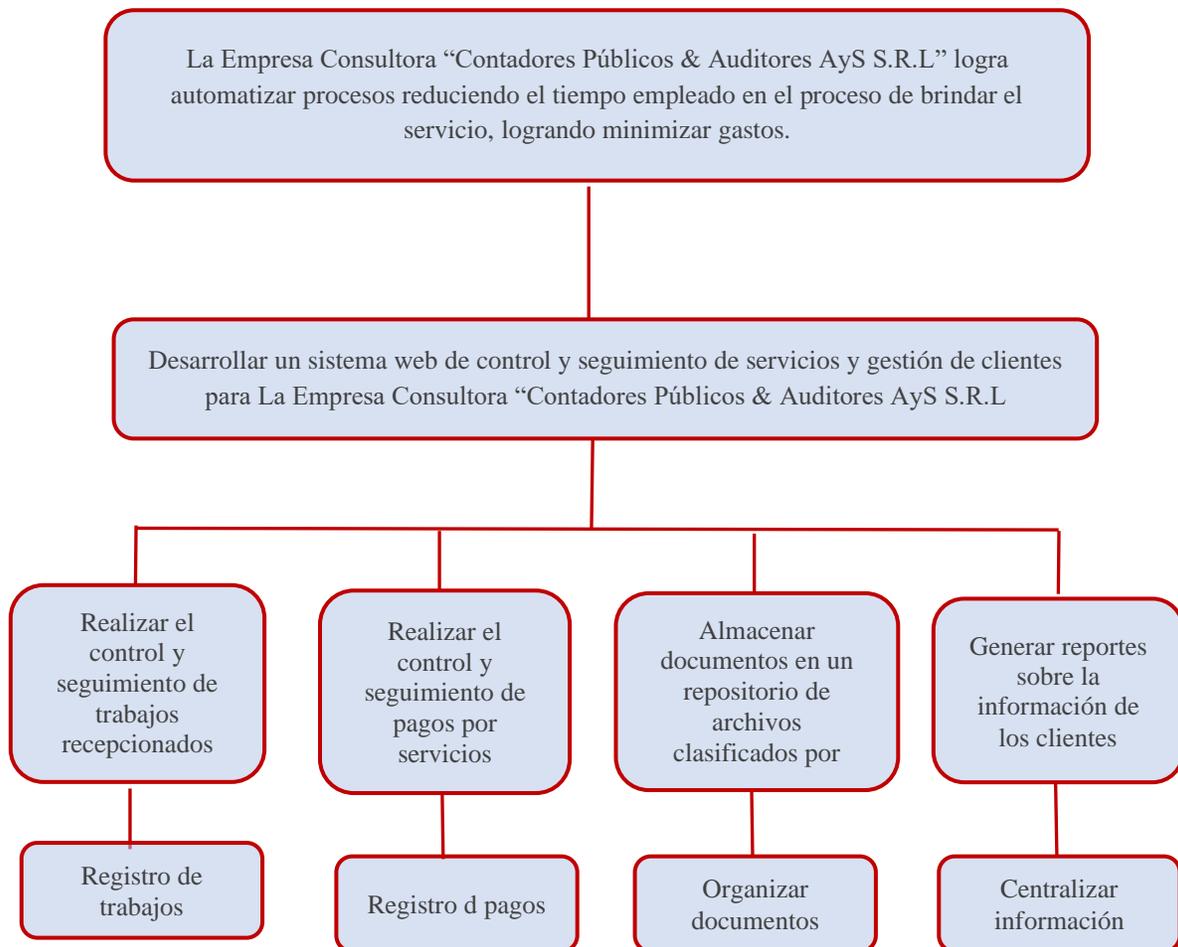
Madachy, R. (s.f.). COCOMO II - Constructive Cost Model. Obtenido de <http://csse.usc.edu/tools/cocomoii.php>.

ANEXOS

ANEXO A – ÁRBOL DE PROBLEMAS



ANEXO B – ÁRBOL DE OBJETIVOS



ANEXO C – MARCO LÓGICO

RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p>FIN: La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L” logra automatizar procesos reduciendo el tiempo empleado en el proceso de brindar el servicio, logrando minimizar gastos.</p>	<p>Comparación de estados de resultados de la empresa, demostrando la diferencia en las ganancias.</p>	<p>Informes y reportes (estados financieros) a partir de la implementación del sistema web dentro de la empresa.</p>	<p>Los usuarios de sistema logran asimilar el flujo de trabajo con el sistema y lo aplican a la rutina de la empresa.</p>
<p>PROPOSITO: Desarrollar un sistema web de control y seguimiento de servicios y gestión de clientes para La Empresa Consultora “Contadores Públicos & Auditores AyS S.R.L.”</p>	<p>Cumplimiento de los requisitos solicitados. Sistema web completamente funcional</p>	<p>Aval de la empresa “Contadores Públicos & Auditores AyS S.R.L”, una vez que el sistema esté terminado.</p>	<p>Se cuenta con la aprobación del gerente general, Sr José Gutiérrez, de la empresa para la implementación del sistema.</p>
<p>PRODUCTOS: Realizar el control y seguimiento de trabajos recepcionados. Realizar el control y seguimiento de pagos por servicios. Almacenar documentos en un repositorio de archivos clasificados por clientes. Facilitar la administración de documentación la cual se genera cuando el servicio al usuario culmino. Generar reportes sobre la información de los clientes.</p>	<p>Registro de todos los trabajos recibidos. Registro de todos los pagos realizados por los clientes. Archivos digitales almacenados por cada servicio. Información centralizada</p>	<p>Los usuarios pueden dar de crear, modificar y borrar clientes. Los usuarios obtienen reportes Para obtener información centralizada. El sistema registra cada trabajo recibido (servicio) y permite al usuario realizar su correspondiente seguimiento.</p>	<p>Se cuenta con toda la información de los clientes registrados manualmente y se los importe a la base de datos del sistema web desarrollado. El personal de la empresa conoce de su rol asignado en el sistema. Se cuenta con todas las herramientas necesarias para el proceso de desarrollo del sistema web.</p>
<p>ACTIVIDADES: Coordinar requerimientos con el personal de la empresa. Implementar los controladores del sistema. Implementar los modelos del sistema. Diseñar e implementar las vistas del sistema. Clasificar y asignar roles de usuario del sistema. Ordenar los servicios de la empresa. Generar reportes de servicios de clientes. Implementar el sistema web. Realizar pruebas al sistema</p>	<p>1 semana 2 semana 2 semana 2 semana 1 semana 3 semana 2 semana 1 semana 1 semana</p>	<p>Revisión de la documentación donde se encuentra todas las fases realizadas en el proceso de desarrollo del sistema web.</p>	<p>Se cuenta con el tiempo planeado dentro del cronograma de actividades. Se cuenta con reuniones de capacitación para los usuarios que manipularan el sistema. El servidor web se encuentre disponible para su implementación.</p>

ANEXO D – CRONOGRAMA DE ACTIVIDADES

ACTIVIDADES	DURACIÓN EN DÍAS	DEL 1 DE JULIO AL 30 DE NOVIEMBRE																					
		JULIO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DIC	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Redacción del Capítulo II - Marco Teórico	7	█																					
Desarrollo del Capítulo III - Marco Aplicativo	91	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	
Modulo de control y seguimiento de servicios	21																						
Fase de analisis	7																						
Análisis de requerimientos	1																						
Casos de uso	1																						
Modelo de contenidos	5																						
Fase de diseño	7																						
Modelo navegacional	3																						
Modelo de presentación	4																						
Fase de desarrollo	6																						
Modelo de flujo procesos	6																						
Fase de implementación	1																						
Modulo de control y seguimiento de pagos por servicios	21																						
Fase de analisis	7																						
Análisis de requerimientos	1																						
Casos de uso	1																						
Modelo de contenidos	5																						
Fase de diseño	7																						
Modelo navegacional	3																						
Modelo de presentación	4																						
Fase de desarrollo	6																						
Modelo de flujo procesos	6																						
Fase de implementación	1																						
Modulo de almacenamiento y administracion de documentacion de clientes	14																						
Fase de analisis	3																						
Análisis de requerimientos	1																						
Casos de uso	1																						
Modelo de contenidos	1																						
Fase de diseño	4																						
Modelo navegacional	2																						
Modelo de presentación	2																						
Fase de desarrollo	6																						
Modelo de flujo procesos	6																						
Fase de implementación	1																						
Modulo de gestion de clientes y empleados	14																						
Fase de analisis	3																						
Análisis de requerimientos	1																						
Casos de uso	1																						
Modelo de contenidos	1																						
Fase de diseño	4																						
Modelo navegacional	2																						
Modelo de presentación	2																						
Fase de desarrollo	6																						
Modelo de flujo procesos	6																						
Fase de implementación	1																						
Modulo de informacion centralizada y reportes	21																						
Fase de analisis	7																						
Análisis de requerimientos	1																						
Casos de uso	1																						
Modelo de contenidos	5																						
Fase de diseño	7																						
Modelo navegacional	3																						
Modelo de presentación	4																						
Fase de desarrollo	6																						
Modelo de flujo procesos	6																						
Fase de implementación	1																						
Implementacion en Servidor Web	7																						
Pruebas	7																						
Redacción del Capítulo III - Marco Aplicativo	7																						
Redacción del Capítulo IV - Calidad y Seguridad	7																						
Redacción del Capítulo V - Análisis Costo Beneficio	7																						
Redacción del Capítulo VI - Conclusiones y Recomendaciones	7																						