

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



PROYECTO DE GRADO

**“SISTEMA WEB DE SEGUIMIENTO Y CONTROL DE LA
INFORMACIÓN DE CLIENTES”
CASO: LINXS S.R.L.**

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

POSTULANTE: HECTOR ALBERTO CUSSI MERLO
TUTOR METODOLÓGICO: M. Sc. ALDO RAMIRO VALDEZ ALVARADO
ASESOR: LIC. FREDDY MIGUEL TOLEDO PAZ

**LA PAZ – BOLIVIA
2016**



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



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

LICENCIA DE USO

El usuario está autorizado a:

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

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

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

DEDICATORIA

A Dios, por su infinito amor me permitió llegar hasta esta etapa de mi vida, ha sido mi guía y ayudador, siempre ha sido fiel.

A mis padres Roberto y Lucia, quienes con tanto esfuerzo y sacrificio me ayudaron a salir adelante, siempre confiaron en mí, gracias por tanto amor, los quiero mucho.

A mis hermanos Lidia, Abel y Maribel, por todo su cariño y paciencia, son parte fundamental en mi vida.

Agradecimientos

A Dios, por permitirme llegar hasta esta etapa de mi vida, cuidándome y protegiéndome de todo mal, no me hizo faltar salud y bienestar.

A mis padres Roberto Cussi y Lucia Merlo, que siempre estuvieron apoyándome en todo momento, gracias por todo su esfuerzo y sacrificio para ayudarme a salir adelante, sin Uds. no fuese posible todo este logro, gracias por tanto amor.

A mi tutor M. Sc. Aldo Ramiro Valdez Alvarado, quien me brindó todo su conocimiento y tiempo, para poder realizar este proyecto, gracias por su paciencia, sus enseñanzas y su comprensión.

A mi asesor Lic. Freddy Miguel Toledo Paz, por su ayuda y comprensión, quien me ayudo en todo el proceso de este proyecto con su conocimiento y sus consejos.

Al Lic. Roberto Claros Uzqueda, quien me permitió realizar este proyecto en la empresa LINXS S.R.L., gracias por sus ánimos, sus consejos y su ayuda.

A todos mis amigos, hermanos, compañeros de trabajo que me tuvieron paciencia, me animaron y ayudaron en esta etapa de mi vida.

A todos ellos, estaré eternamente agradecido, muchas gracias.

RESUMEN

Tener la información actualizada y en el momento oportuno es una necesidad primordial en todas las empresas para que puedan cumplir con sus objetivos, dicha información puede ser manipulada desde cualquier dispositivo y cualquier explorador que tenga una conexión a internet.

La empresa LINXS S.R.L. está al servicio del país desde julio de 2005, se dedica al desarrollo de páginas web, material de publicidad y venta de espacios publicitarios, cuenta con su portal web Bolivia en tus manos, donde tiene las guías de mas importantes de noticias, páginas amarillas, guía de hoteles, restaurantes, industrias, salud, agenda cultural y cartelera de cine.

El proyecto “Sistema Web de seguimiento y control de la información de clientes Caso: LINXS S.R.L.”, tiene como principal objetivo ayudar al personal de la empresa en la obtención y manipulación de información de sus clientes.

El sistema cuenta con tres módulos, para la administración de usuarios, para la gestión de clientes y servicios y para la obtención de reportes.

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 esta 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.

Para la calidad del sistema web se utiliza la norma ISO 9126 y para la seguridad del sistema se realiza por niveles, tanto del lado del cliente como del lado del servidor.

Palabras clave: Metodología ágil Kanban, UWE, ISO 9126, Sistema Web, COCOMO II.

ABSTRACT

To have the information updated and in the opportune moment is a primordial necessity in all the companies so that they can fulfill their objectives, that information can be manipulated from any device and any explorer that has an Internet connection.

The company LINXS S.R.L. Is in the service of the country since July 2005, is dedicated to the development of web pages, publicity material and sale of advertising space, has its web portal Bolivia in your hands, where you have the most important news guides, yellow pages, Guide of hotels, restaurants, industries, health, cultural agenda and film billboard.

The project "Web System for monitoring and control of customer information Case: LINXS S.R.L.", has as main objective to help the company's staff in obtaining and manipulating information of its customers.

The system has three modules, for the administration of users, for the management of clients and services and for obtaining reports.

In the development of the web system a combination between the agile Kanban methodology and the UWE methodology is made, this combination is a contribution for future projects.

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

UWE details the process of the applications, has five models that are the analysis of requirements, content model, navigational model, presentation model and process model.

For the quality of the web system ISO 9126 is used and for system security it is performed by levels, both on the client side and on the server side.

Keywords: Kanban Agile Methodology, UWE, ISO 9126, Web System, COCOMO II.

ÍNDICE

RESUMEN.....	4
ABSTRACT	5
MARCO INTRODUCTORIO.....	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES.....	2
1.2.1. ANTECEDENTES INSTITUCIONALES.....	2
1.2.1.1. VISIÓN DE LA EMPRESA.....	3
1.2.1.2. MISIÓN DE LA EMPRESA	3
1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES	4
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.3.1. PROBLEMA CENTRAL	5
1.3.2. PROBLEMAS SECUNDARIOS.....	5
1.4. DEFINICIÓN DE OBJETIVOS	6
1.4.1. OBJETIVO GENERAL	6
1.4.2. OBJETIVOS ESPECÍFICOS.....	6
1.5. JUSTIFICACIÓN	7
1.5.1. JUSTIFICACIÓN SOCIAL.....	7
1.5.2. JUSTIFICACIÓN ECONÓMICA	7
1.5.3. JUSTIFICACIÓN TECNOLÓGICA	8
1.6. ALCANCES Y LÍMITES.....	8
1.6.1. ALCANCES	8
1.6.1.1. MÓDULOS.....	8
1.6.1.2. FUNCIONALIDADES	9
1.6.2. LÍMITES.....	9
1.7. APORTES.....	9
1.7.1. APORTE PRÁCTICO.....	9
1.7.2. APORTE TEÓRICO	10
1.8. METODOLOGÍA	10
MARCO TEÓRICO.....	12
2.1. INTRODUCCIÓN	12

2.2.	INGENIERÍA DE SOFTWARE	12
2.3.	METODOLOGÍAS DE DESARROLLO DE SOFTWARE	13
2.3.1.	METODOLOGÍAS DE DESARROLLO TRADICIONALES	13
2.3.2.	METODOLOGÍAS DE DESARROLLO ÁGILES	14
2.4.	METODOLOGÍA ÁGIL KANBAN	15
2.4.1.	ROLES KANBAN	16
2.4.1.1.	SERVICE REQUEST MANAGER	17
2.4.1.2.	SERVICE DELIVERY MANAGER	17
2.4.2.	PRINCIPIOS DE LA METODOLOGÍA KANBAN	17
2.4.2.1.	VISUALIZAR EL WORKFLOW (FLUJO DE TRABAJO)	17
2.4.2.2.	LIMITAR EL TRABAJO EN CURSO (WIP)	18
2.4.2.3.	OPTIMIZAR EL FLUJO DE TRABAJO	19
2.4.3.	CUELLOS DE BOTELLA	19
2.4.4.	MÉTRICAS EN KANBAN	20
2.5.	INGENIERÍA WEB	21
2.6.	INGENIERIA WEB BASADA EN UML (UWE)	22
2.6.1.	FASES DE DESARROLLO	23
2.6.2.	ANÁLISIS DE REQUISITOS	23
2.6.3.	MODELO DE CONTENIDO	24
2.6.4.	MODELO NAVEGACIONAL	25
2.6.5.	MODELO DE PRESENTACIÓN	26
2.6.6.	MODELO DE PROCESOS	27
2.6.6.1.	MODELO DE FLUJO DEL PROCESO	27
2.7.	PATRONES DE DISEÑO	29
2.7.1.	MODELO VISTA CONTROLADOR (MVC)	29
2.8.	TECNOLOGÍAS WEB	31
2.8.1.	PHP	31
2.8.2.	FRAMEWORK CODEIGNITER 3	31
2.8.3.	JAVASCRIPT	33
2.8.4.	JQUERY	33
2.8.5.	BOOTSTRAP 3	33

2.8.6. GESTOR DE BASE DE DATOS MYSQL.....	34
MARCO APLICATIVO	35
3.1. INTRODUCCIÓN	35
3.2. COMBINACIÓN DE METODOLOGÍAS KANBAN – UWE.....	35
3.3. ROLES KANBAN	36
3.4. MÓDULO DE ADMINISTRACIÓN DE USUARIOS	37
3.4.1. FASE DE ANÁLISIS.....	37
3.4.1.1. ANÁLISIS DE REQUERIMIENTOS	37
3.4.1.2. CASOS DE USO.....	38
3.4.1.3. MODELO DE CONTENIDOS.....	40
3.4.2. FASE DE DISEÑO	40
3.4.2.1. MODELO NAVEGACIONAL.....	40
3.4.2.2. MODELO DE PRESENTACIÓN	41
3.4.3. FASE DE DESARROLLO	43
3.4.3.1. MODELO DE FLUJO DE PROCESO	43
3.4.4. FASE DE IMPLEMENTACIÓN.....	44
3.5. MÓDULO DE GESTIÓN DE CLIENTES Y SERVICIOS.....	44
3.5.1. FASE DE ANÁLISIS.....	44
3.5.1.1. ANÁLISIS DE REQUERIMIENTOS	44
3.5.1.2. CASOS DE USO.....	45
3.5.1.3. MODELO DE CONTENIDOS.....	47
3.5.2. FASE DE DISEÑO	47
3.5.2.1. MODELO NAVEGACIONAL.....	47
3.5.2.2. MODELO DE PRESENTACIÓN	48
3.5.3. FASE DE DESARROLLO	50
3.5.3.1. MODELO DE FLUJO DE PROCESO	50
3.5.4. FASE DE IMPLEMENTACIÓN.....	51
3.6. MÓDULO DE REPORTE DE SERVICIOS.....	52
3.6.1. FASE DE ANÁLISIS.....	52
3.6.1.1. CASOS DE USO.....	52
3.6.1.2. MODELO DE CONTENIDOS.....	53

3.6.2.1.	MODELO NAVEGACIONAL.....	54
3.6.2.2.	MODELO DE PRESENTACIÓN	55
3.6.3.	FASE DE DESARROLLO	55
3.6.3.1.	MODELO DE FLUJO DE PROCESO	55
3.6.4.	FASE DE IMPLEMENTACIÓN.....	56
3.7.	FASE DE PRODUCCIÓN	56
3.7.1.	PRUEBA DE ESFUERZO	56
	CALIDAD Y SEGURIDAD	58
4.1.	INTRODUCCIÓN	58
4.2.	CALIDAD DE SOFTWARE.....	58
4.3.	NORMA ISO 9126.....	58
4.3.1.	FUNCIONALIDAD	59
4.3.2.	CONFIABILIDAD.....	63
4.3.3.	USABILIDAD	63
4.3.4.	MANTENIBILIDAD	64
4.3.5.	PORTABILIDAD	65
4.3.6.	CALIDAD GLOBAL.....	66
4.4.	SEGURIDAD	66
4.4.1.	SEGURIDAD DEL LADO DEL CLIENTE.....	67
4.4.2.	SEGURIDAD DEL LADO DEL SERVIDOR.....	67
4.4.2.1.	ENCRIPCIÓN DE CONTRASEÑAS	67
4.4.2.2.	CONTROL DE ACCESOS Y PERMISOS.....	68
4.4.2.3.	FILTROS CONTRA ATAQUES XSS – CROSS SITE SCRIPTING	68
	COSTO Y BENEFICIO.....	69
5.1.	INTRODUCCIÓN	69
5.2.	COCOMO II	69
5.3.	COSTO DEL SISTEMA	70
5.3.1.	COSTO DE DESARROLLO DEL SISTEMA.....	71
5.3.2.	COSTO DE ELABORACIÓN DEL PROYECTO.....	72
5.3.3.	COSTO TOTAL DEL SISTEMA.....	72
5.4.	VALOR ACTUAL NETO.....	73

5.5. COSTO BENEFICIO	74
5.6. TASA INTERNA DE RETORNO.....	75
CONCLUSIONES Y RECOMENDACIONES	76
6.1. CONCLUSIONES	76
6.2. RECOMENDACIONES	77
BIBLIOGRAFÍA	78
ANEXOS.....	81
ANEXO A: ARBOL DE PROBLEMAS	82
ANEXO B: ÁRBOL DE OBJETIVOS.....	83
ANEXO C: MARCO LÓGICO.....	84
ANEXO D: CRONOGRAMA DE ACTIVIDADES	86
ANEXO E: ESTRUCTURA ORGANIZACIONAL DE LA EMPRESA LINXS S.R.L....	87
ANEXO F: CODIGO FUENTE DE LA CLASE ACL	88
ANEXO G: MANUAL DE USUARIO DEL SISTEMA	92

ÍNDICE DE FIGURAS

Figura 2.1: Relación de Kanban frente a otras metodologías.....	16
Figura 2.2: Ejemplo de tablero Kanban.....	18
Figura 2.3: Ejemplo de cuello de botella en la fase de desarrollo.....	19
Figura 2.4: Gráfico fecha de entrada y salida de tareas.....	21
Figura 2.5: Ejemplo de un diagrama de caso de uso.....	24
Figura 2.6: Ejemplo de un modelo conceptual.....	24
Figura 2.7: Ejemplo de un modelo de navegación.....	25
Figura 2.8: Nombre de estereotipos y sus iconos del modelo navegacional.....	26
Figura 2.9: Nombre de estereotipos y sus iconos del modelo de presentación.....	26
Figura 2.10: Ejemplo de un modelo de presentación.....	27
Figura 2.11: Estereotipos y sus iconos del modelo de procesos.....	28
Figura 2.12: Ejemplo de un modelo de flujo de proceso.....	28
Figura 2.13: Modelo Vista Controlador.....	30
Figura 2.14: Diagrama de flujo de la aplicación.....	32
Figura 3.1: Combinación de metodología Kanban y UWE.....	36
Figura 3.2: Caso de uso administrar roles de usuario.....	38
Figura 3.3: Caso de uso administrar usuarios.....	39
Figura 3.4: Diagrama de contenidos de la administración del sistema.....	40
Figura 3.5: Diagrama de navegación para la administración de usuarios.....	41
Figura 3.6: Diseño de la vista administración de usuarios.....	41
Figura 3.7: Diseño de la vista registro de usuario.....	42
Figura 3.8: Diseño de la vista registro de usuario.....	42
Figura 3.9: Diseño de la vista de control de permisos de roles.....	43
Figura 3.10: Diseño del modelo de flujo de procesos de la administración de usuarios.....	43
Figura 3.11: Captura de pantalla del prototipo de Administración de usuarios.....	44
Figura 3.12: Caso de uso de gestión de clientes.....	45
Figura 3.13: Caso de uso para operaciones de un cliente.....	46
Figura 3.14: Diagrama de contenido para el módulo de gestión de clientes y servicios.....	47
Figura 3.15: Diagrama de navegación para la gestión de clientes y servicios.....	48
Figura 3.16: Diagrama de presentación para el módulo de gestión de clientes y servicios.....	49
Figura 3.17: Diagrama de presentación para detalle de cliente.....	49
Figura 3.18: Modelo de flujo de procesos del módulo de gestión de clientes y servicios.....	50
Figura 3.19: Captura de pantalla del prototipo para la gestión de clientes.....	51
Figura 3.20: Captura de pantalla del prototipo para la gestión de servicios.....	51
Figura 3.21: Caso de uso reportes de servicios.....	52
Figura 3.22: Diagrama de contenidos del módulo de reportes.....	53
Figura 3.23: Diagrama de navegación del módulo de reportes.....	54
Figura 3.24: Diagrama de presentación del módulo de reportes.....	55
Figura 3.25: Diseño de modelo de flujo de procesos para el módulo de reportes.....	55
Figura 3.26: Captura de pantalla del prototipo de Administración de usuarios.....	56

Figura 3.27: Resultado de la prueba con 50 usuarios en cada página.....	57
Figura 3.28: Resultado de la prueba con 100 usuarios en cada página.....	58
Figura 4.1: Código fuente de la clase Hash para encriptar contraseñas.....	68

ÍNDICE DE TABLAS

Tabla 2.1: Tabla comparativa entre la metodología tradicional y ágil.....	13
Tabla 3.1: Fases de trabajo, limite WIP y responsables.....	37
Tabla 3.2: Especificación del caso de uso administrar roles de usuario.....	38
Tabla 3.3: Especificación del caso de uso administrar usuarios.....	39
Tabla 3.4: Especificación del caso de uso gestión de clientes.....	45
Tabla 3.5: Especificación del caso de uso operaciones de un cliente.....	46
Tabla 3.6: Especificación del caso de uso reportes de servicios.....	52
Tabla 4.1: Lista de entradas de usuario del sistema.....	59
Tabla 4.2: Lista de salidas de usuario del sistema.....	59
Tabla 4.3: Lista de peticiones del usuario al sistema.....	59
Tabla 4.4: Lista de archivos lógicos del sistema.....	60
Tabla 4.5: Factores de ponderación.....	60
Tabla 4.6: Factor de ajuste de complejidad.....	61
Tabla 4.7: Preguntas para obtener el grado de usabilidad.....	64
Tabla 4.8: Evaluación de mantenibilidad.....	64
Tabla 4.9: Calidad global.....	66
Tabla 5.1: Constantes a,b,c,d COCOMO II.....	70
Tabla 5.2: Conversión de PF a KLDC.....	71
Tabla 5.3: Costos de recursos empleados para la elaboración del sistema.....	72
Tabla 5.4: Costo total del sistema.....	73
Tabla 5.5: Estimación de gastos y ganancias en 5 años.....	74

CAPÍTULO I

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

En la actualidad es muy importante tener información actualizada en todas las organizaciones y empresas. Esto ayuda a tomar mejores decisiones en distintas situaciones que a diario se presentan para así poder alcanzar los objetivos propuestos.

El avance de las tecnologías relacionadas con el manejo de la información hace que más personas y servicios estén conectadas e interactúen entre sí. Las tareas y funciones que tradicionalmente se realizaban manualmente, ahora son hechas por medio de un sistema de información.

El uso de internet es un servicio imprescindible para compartir información, ya que se puede acceder desde cualquier computador o dispositivo móvil en diferentes partes del mundo.

Una de los medios más requeridos para poder acceder a la información son los sistemas web, porque no requieren de un sistema operativo o plataforma, ya que pueden ser alojados en un servidor de internet o intranet (red local) y ser utilizados desde cualquier navegador web, sin importar el dispositivo.

Es así que las empresas no pueden darse el lujo de desaprovechar estas tecnologías, que benefician tanto a los usuarios que operan en el sistema como a la misma empresa.

LINXS S.R.L. es una empresa que ofrece a sus clientes los servicios de desarrollo de páginas web, venta de espacios publicitarios en su portal web Bolivia en tus manos, publicidad en páginas amarillas, compra de dominio y hosting entre otros.

Tiene como actividad diaria el registro de clientes de los nueve departamentos de Bolivia y la cantidad de registros va en aumento, pero este proceso se realiza manualmente en planillas excel, existen datos duplicados u omitidos, por tal motivo es difícil acceder a esta

información y se recurren a constantes llamadas telefónicas generando molestia en los clientes y pérdida económica en la empresa.

Considerando todo lo mencionado, el propósito de este proyecto es implementar un sistema web de seguimiento y control de la información de clientes de la empresa LINXS S.R.L. para que brinde información oportuna y actualizada al personal de la empresa, que ayude en la toma de decisiones al momento de buscar nuevos clientes y ofrecer nuevos servicios y finalmente que evite pérdida económica a la empresa.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES INSTITUCIONALES

LINXS S.R.L. es una empresa dedicada al desarrollo de páginas web, material de publicidad y venta de espacios publicitarios, inicia sus actividades en julio de 2005 y actualmente cuenta con alrededor de 4500 clientes en los nueve departamentos de Bolivia.

Cuenta con su portal web Bolivia en tus manos (boliviaentusmanos.com), donde tiene las guías más importantes del país como noticias, páginas amarillas, cartelera de cine, hoteles, restaurantes, salud, industrias y agenda cultural.

Aprovechando la gran visitación en su portal web por su contenido, ofrece el servicio de venta de espacios publicitarios, en diferentes partes del portal que el cliente vea conveniente.

En la guía de páginas amarillas los clientes pueden publicitar su empresa o negocio, de acuerdo a su rubro o actividad, puede poner su dirección, teléfonos, correo electrónico, mapa de ubicación, galería de fotos, actividades, productos y servicios entre otros.

Para el servicio de diseño web de calidad, utilizan las últimas tendencias de programación y diseño gráfico, las interfaces son responsivas y agradable a la vista de los usuarios. Pueden elegir entre dos tipos de diseño uno con plantillas wordpress o con plantillas HTML5 con CSS3 y javascript También ofrece el servicio de base de datos a los clientes que requieran manejar su información de manera dinámica.

Sus clientes no tienen que preocuparse por buscar otras empresas de proveedores de dominio y hosting ya que la empresa cuenta con estos servicios a precios accesibles, el dominio puede ser comprado para uno, dos o más años según las necesidades, y cuenta con un hosting de confianza que le provee la empresa Mega Link reconocida a nivel nacional.

La empresa cuenta con el personal capacitado para la atención a los clientes, para el desarrollo de páginas web y demás servicios que ofrece. En el Anexo E, se puede observar la estructura organizacional de la empresa, conformado por el gerente general, responsable de área contable, responsable de recursos humanos, jefe de sistemas, vendedores, diseñador, maquetador web, administrador de base de datos, encargado de área de noticias y atención al cliente.

Los vendedores de la empresa son los encargados de conseguir clientes para la empresa, ellos realizan llamadas telefónicas, entrevistas y visitas a diferentes empresas, ofreciendo el servicio de publicidad en su portal web Bolivia en tus manos.

1.2.1.1. VISIÓN DE LA EMPRESA

Ser la mejor alternativa para presentar productos y servicios empresariales en internet, en productos multimedia y materiales promocionales.

1.2.1.2. MISIÓN DE LA EMPRESA

Satisfacer las necesidades de publicidad, presencia en internet e imagen corporativa de nuestros clientes utilizando medios tecnológicos, digitales, e innovadores procesos de impresión.

Superar las exigencias de los clientes y buscar la excelencia en todos los servicios. Desarrollar profesionales para que ellos trabajen de manera continua y progresiva en alcanzar siempre mayores estándares de calidad y eficiencia.

Contribuir al desarrollo de nuestro país, de las empresas que confían en nosotros y de las personas que se relacionan comercialmente con la empresa.

1.2.2. ANTECEDENTES DE PROYECTOS SIMILARES

En la Carrera de Informática de la Universidad Mayor de San Andrés (UMSA) se verificó la existencia de proyectos de grado relacionados con el Sistema Web, dichos proyectos están orientados a la automatización de los procesos manuales.

Se verificó los siguientes proyectos de grado:

- Sistema de Información Vía Web de Seguimiento y Control a los Proyectos de Inversión en el Gobierno Autónomo Municipal de Palos Blancos, realizado por Erick Marquez Mayta, en el año 2013, presenta una propuesta para la alcaldía de Palos Blancos, pretende generar un núcleo de entre la alcaldía y su pueblo, para que en las áreas dispersa puedan acceder de manera inmediata a la información, para el desarrollo de este trabajo se utilizó teorías demostradas y comprobadas como la Ingeniería Web y el Lenguaje de Modelado Unificado (UML).
- Sistema Web de Control y Seguimiento de Documentación Caso: Departamento de Infraestructura de la UMSA, realizado por Wilmer Mijhael Yucra Lecoña, en el año 2016, tiene como objetivo colaborar al personal para mejorar su tiempo y esfuerzo en los procesos de trabajo de documentos mediante el sistema web de información, para el desarrollo utiliza la metodología Scrum combinada con la metodología UWE.
- Sistema de Información Vía Web para el Seguimiento y Control de Trámites Caso: Escuela Superior de Formación de Maestros Simón Bolívar, realizado por María Elena Chavez Vallejos, en el año 2014, agiliza y optimiza los procesos que realizan los maestros o funcionarios en la solicitud de trámites, haciendo un seguimiento y control de las mismas, para el desarrollo del Sistema Web utiliza una combinación de las metodologías RUP y UWE.
- Sistema Web de Registro, Seguimiento y Control de Correspondencia Basado en BPM Caso: Carrera de Informática, realizado por Alan Rodrigo Corini Guarachi, en el año 2014, automatiza los procesos de control, seguimiento y envío de diferentes tramites dentro de la carrera de informática, utilizando un BPM llamando LDAP.

- Sistema Web de Control y Seguimiento de Personal Caso: Universidad Tecnológica Boliviana, realizado por María Juana Aguilar Torrez, en el año 2011, el sistema realiza procesos de registro de personal, cálculo de planilla de sueldo, control de horarios de entrada y salida de los empleados, para el desarrollo utiliza el Proceso Unificado Ágil (AUP) y para el modelado del sistema utiliza UWE.

1.3. PLANTEAMIENTO DEL PROBLEMA

La información de un cliente es muy importante para la empresa, ya que la misma es necesaria para la toma de decisiones, como: clientes registrados en cada ciudad, fechas de vencimiento de cada servicio, renovación de servicios, etc. Acceder a esta información es muy dificultosa y morosa, ya que el responsable del área contable tiene que llenar los datos manualmente en planillas excel, y luego para poder visualizar se tiene que buscar los registros uno por uno.

Los vendedores al momento de realizar una cobranza, renovación o venta de servicios a un cliente no cuentan con información fidedigna. También no se pueden detectar a clientes con servicios vencidos. Por tal motivo existen pérdidas económicas, información duplicada, insatisfacción y desconfianza en la empresa.

En ese contexto se plantea el siguiente problema:

1.3.1. PROBLEMA CENTRAL

¿Cómo se puede mejorar el seguimiento y control de la información de clientes en la empresa LINXS S.R.L.?

1.3.2. PROBLEMAS SECUNDARIOS

Se mencionan los problemas más destacados que ocurren en la empresa, para ver más claro se realizó un árbol de problemas en el Anexo A.

- Los registros de clientes se guardan de forma manual en planillas excel, esto ocasiona pérdida de tiempo al momento de querer acceder a las mismas.

- La gerencia general no cuenta con información oportuna y actualizada de sus clientes y de los servicios que brinda, esto no le ayuda en la toma de decisiones al momento de buscar nuevos clientes, ofrecer nuevos servicios y cumplir con sus objetivos.
- El encargado del área contable toma mucho tiempo para llenar un formulario de registro, porque primero lo llena en hojas y luego en la planilla excel, esto ocasiona que se omitan y/o dupliquen datos importantes de los clientes.
- Los vendedores no tienen acceso a la información de clientes, muchas veces tienen que llamar constantemente al encargado del área contable para obtener dicha información, lo que genera incertidumbre y malestar en el cliente también pérdida de tiempo y dinero a la empresa.
- No se realiza un control a las fechas de vencimiento de los servicios a los clientes, ocasionando que a largo tiempo sean olvidados y no se pueda cobrar o renovar el servicio, lo que genera pérdida económica a la empresa.
- No se tiene información de los paquetes de servicios que ofrece la empresa a sus clientes, esto ocasiona que los vendedores no puedan vender todos los servicios.

1.4. DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Implementar un Sistema Web de Seguimiento y Control de Información de Clientes en la empresa LINXS S.R.L., que brinde información oportuna y actualizada a gerencia general, área contable y vendedores.

1.4.2. OBJETIVOS ESPECÍFICOS

Se mencionaron los problemas más destacados que pasa en la empresa LINXS S.R.L. para ver con más detalle ver el Anexo B donde está el árbol de objetivos.

- Brindar a gerencia general información oportuna y actualizada de sus clientes y servicios que le ayude en la toma de decisiones.

- Recolectar y procesar los datos de los clientes registrados manualmente en planillas excel, para poder tener información rápida y accesible.
- Generar un registro de clientes en línea, para que el encargado del área contable ahorre tiempo, y no omita y/o duplique los datos de los clientes.
- Brindar información sobre los servicios de clientes a los vendedores que requieran dicha información, que permita a la empresa disminuir los gastos innecesarios en llamadas telefónicas y así brindar una mejor atención a los clientes.
- Facilitar y brindar un control a las fechas de vencimiento de los servicios de clientes, para que se puedan cobrar y renovar servicios con anticipación.

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN SOCIAL

A medida que avanza el tiempo, la cantidad de clientes va en aumento, entonces la empresa debe ser capaz de atender y mejorar sus servicios, para que los mismos se sientan seguros y confíen en la empresa.

Con el sistema, los vendedores de la empresa cuentan con información rápida oportuna y segura de los clientes suscritos, y mejora la atención a los mismos, también es una ayuda para gerencia general y el área contable porque les facilita en el control y seguimiento de los servicios, les ayuda en la toma de decisiones para lograr el objetivo de la empresa.

1.5.2. JUSTIFICACIÓN ECONÓMICA

Con la implementación del Sistema en la empresa disminuirán las pérdidas económicas, a través del seguimiento y control de servicios a los clientes, evitando que existan servicios sin cobrar, clientes y servicios olvidados, permitiendo renovar con anticipación, también evitará las solicitudes de baja por parte de los clientes y disminuirán los gastos en llamadas telefónicas continuas por partes de los vendedores y del mensajero para acceder a la información.

En cuanto a las herramientas para el desarrollo del Sistema se empleará la Licencia Pública General (GPL, por sus siglas en inglés de General Public License) que es totalmente gratuito, por lo que no se tendrá ningún costo por ese motivo. Para el alojamiento del Sistema, la empresa cuenta con un hosting y un dominio que no está siendo utilizada en su totalidad, es ahí donde estará alojado el sistema web, entonces tampoco se tendrá un gasto adicional con relación a lo mencionado.

1.5.3. JUSTIFICACIÓN TECNOLÓGICA

Actualmente la empresa LINXS S.R.L. cuenta con las herramientas tecnológicas adecuadas como recursos de software y hardware para implementar el Sistema Web.

La empresa cuenta con equipos de computación en cada uno de sus áreas, gerencial general, área Contable, vendedores y área de sistemas, mismos que tienen acceso a una red internet e intranet que es fundamental para la implementación del sistema. Así mismo la empresa cuenta con un dominio y hosting, donde estará alojado la aplicación del sistema, ya sea para uso interno como para el acceso externo mediante la web, garantizando un acceso inmediato.

1.6. ALCANCES Y LÍMITES

1.6.1. ALCANCES

Basado en las principales necesidades de la empresa, el Sistema cuenta con los siguientes módulos y funcionalidades:

1.6.1.1. MÓDULOS

- Módulo de administración para la gestión de usuarios, roles, accesos, permisos y configuración del Sistema.
- Módulo de gestión de clientes y servicios, para registrar, modificar, actualizar los datos, para verificar el detalle de servicios y sus formas de pago.
- Módulo de reportes para informes diarios, semanales mensuales y anuales.

1.6.1.2. FUNCIONALIDADES

- Funcionalidad de control de servicios, que permita a los usuarios recordar, alertar y notificar sobre los vencimientos de los mismos.
- Funcionalidad responsiva, para que la interfaz del Sistema sea agradable en los dispositivos móviles de los usuarios.
- Funcionalidad de seguimiento de clientes, que permita ver la información de estos y hacer un seguimiento a los servicios que tienen.
- Funcionalidad de vista e impresión, para poder entregar información detallada de los servicios adquiridos a los clientes que requieran.
- Funcionalidad de control de visitas, permitirá que el vendedor asigne una fecha para poder realizar una entrevista con un cliente, con el fin de no perjudicar el desarrollo de sus actividades.

1.6.2. LÍMITES

El Sistema Web tiene los siguientes límites:

- Solo usuarios habilitados por el administrador del sistema tendrán acceso al sistema dependiendo el rol asignado.
- El Sistema no controla la parte contable de la empresa.
- El Sistema no estará disponible para los clientes, solo tendrán acceso el personal de la empresa.
- El sistema solo realizará control y seguimiento a 4 servicios de la empresa

1.7. APORTES

1.7.1. APORTE PRÁCTICO

El sistema propuesto, es de mucha utilidad para la empresa porque brinda información oportuna y segura a los usuarios, generando una mejor interacción entre el personal y buena atención a los clientes, además de realizar un mejor seguimiento a los servicios de clientes.

1.7.2. APORTE TEÓRICO

Este proyecto pretende ser una referencia para el desarrollo de futuros proyectos, ya que en el desarrollo del sistema se emplea la metodología ágil Kanban, para la parte del modelado y análisis del sistema se emplea UML basado en Ingeniería Web (UWE), además se emplea el patrón de diseño Modelo Vista Controlador (MVC).

1.8. METODOLOGÍA

Utilizamos el Método Científico, que 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 que tomamos en cuenta son:

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 aun determinantes; para luego pasar a la Investigación Descriptiva: que se efectúa cuando se desea describir, en todos sus componentes principales, una realidad.

Para el desarrollo del Sistema se utilizará la metodología ágil Kanban que permite visualizar el flujo de trabajo, dividiendo el trabajo en bloques, cada elemento del trabajo es descrito en una tarjeta para luego ponerlo en un muro. Permite utilizar columnas con nombre para ilustrar dónde está cada elemento en el flujo de trabajo. Limita el trabajo en curso (WIP, por sus siglas en inglés de Work In Progress), asigna límites concretos a cuántos elementos pueden estar en progreso en cada estado del flujo de trabajo. Mide el lead time (tiempo medio para completar un elemento, a veces llamado tiempo de ciclo), optimiza el proceso para que el lead time sea tan pequeño y predecible como sea posible.

Para el modelado y análisis del sistema utilizaremos UWE que sirve para modelar aplicaciones web, presta una especial atención a la sistematización y personalización

(sistemas adaptativos). Provee de perfiles UML, metamodelos, un proceso de desarrollo dirigido por modelos, y herramientas de soporte para el diseño sistemático de aplicaciones web. Utiliza notación basada en UML para aplicaciones web en general y para aplicaciones adaptativas en particular. La metodología consta de seis modelos:

- Modelo de casos de uso para capturar los requisitos del sistema.
- Modelo conceptual para el contenido (modelo del dominio).
- Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.
- Modelo de estructura de presentación, modelo de flujo de presentación.
- Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
- Modelo de adaptación.

Para la implementación de este sistema web se utilizará las siguientes herramientas tecnológicas:

- Plataforma: Windows
- Lenguaje de programación: PHP versión 5.6 o superior
- Sistema de gestor de Base de Datos: MySQL versión 5.6 o superior
- Herramientas de maquetación: HTML5, CSS3, Javascript
- Motor de plantillas: Smarty

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 de seguimiento y control de la información de clientes para la empresa LINXS S.R.L., sin embargo no contempla la teoría completa, pero contiene los conceptos más importantes para que pueda aplicarse y despejar cualquier duda.

2.2. INGENIERÍA DE SOFTWARE

Existe una variedad de definiciones para ingeniería de software, entre tantas se mencionan las más destacadas.

- Esa forma de ingeniería que aplica los principios de la informática y las matemáticas para conseguir soluciones rentables a problemas software (SEI, 1990).
- El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales (Bauer, 1969).
- La Ingeniería del Software incluye la aplicación práctica del conocimiento científico en el diseño y construcción de los programas y la documentación requerida para su desarrollo, operación y mantenimiento (Boehm, 1976).
- El enfoque sistemático para el desarrollo, operación, mantenimiento y eliminación del software, definiendo como software los programas, procedimientos, reglas y documentación, así como los datos de operación de un sistema de cómputo (IEEE, 1983).
- La Ingeniería del software es una disciplina o área de la informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo (Pressman, 2010).

2.3. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Para tener éxito en el proyecto y obtener un producto final para nuestros clientes, depende en gran parte de la metodología que escoge el equipo de desarrollo, estas pueden ser tradicional o ágil, para que los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempo establecido. La metodología elegida tiene que adaptarse a las necesidades definidas al inicio del proyecto (Cabrera, Figueroa, y Solís, 2008).

Tabla 2.1: Tabla comparativa entre la metodología tradicional y ágil.

Metodologías Tradicionales	Metodologías Agiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (menor a 10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

Fuente: Cabrera, Figueroa, y Solís, 2008

En la Tabla 2.2 se puede observar una comparación entre las metodologías tradicionales y ágiles que nos ayuda al momento de elegir una metodología para el desarrollo del proyecto.

2.3.1. METODOLOGÍAS DE DESARROLLO TRADICIONALES

Al inicio el desarrollo de software era artesanal en su totalidad, la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada, tuvieron que importarse la concepción y

fundamentos de metodologías existentes en otras áreas y adaptarlas al desarrollo de software.

Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo del software.

Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP y MSF entre otros, que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto.

Otra de las características importantes dentro de este enfoque tenemos los altos costos al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil.

2.3.2. 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 competitiva 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 auto-organizados.
- 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

El origen de la metodología Kanban está en los procesos de producción JIT ideados por Toyota, en los que se utilizaban tarjetas para identificar necesidades de material en la cadena de producción, donde los trabajadores están organizados de tal forma que cada uno produce su parte siempre y cuando les llegue una tarjeta que les informe de que pueden producir. Si no les llega la tarjeta, tienen prohibido producir su parte de su trabajo.

Actualmente, el término Kanban ha pasado a formar parte de las llamadas metodologías ágiles, cuyo objetivo es gestionar de manera general cómo se van completando las tareas.

Derivado de la combinación de dos palabras japonesas, *kan*, que quiere decir visual, y *ban*, que quiere decir tarjeta, nace la palabra Kanban, con la que se denomina una metodología que se basa en tarjetas visuales para gestionar el esfuerzo y dedicación del equipo (Bermejo, 2015).

Las principales ventajas de esta metodología es que es muy fácil de utilizar, actualizar y asumir por parte del equipo. Además, destaca por ser una técnica de gestión de las tareas muy visual, que permite ver a golpe de vista el estado de los proyectos, así como también pautar el desarrollo del trabajo de manera efectiva.

Kanban comparte con otras metodologías como Scrum la idea de formar un backlog del producto que tenga una serie de ítems priorizados. La principal diferencia es que no existen iteraciones. En su lugar, Kanban se centra en controlar trabajo en curso (WIP, por sus siglas en inglés de Work In Progress). Esto significa que, cuando hay poco WIP, se añade el ítem más prioritario de la lista backlog, y se controla que nunca se supere una cierta cantidad de WIP.

Como se puede ver en la figura 2.1 Kanban es más adaptativo en relación a otras metodologías, deja casi todo abierto. Las únicas normas son visualiza el flujo de trabajo y limita tu WIP. A pocos centímetros de haz lo que sea.

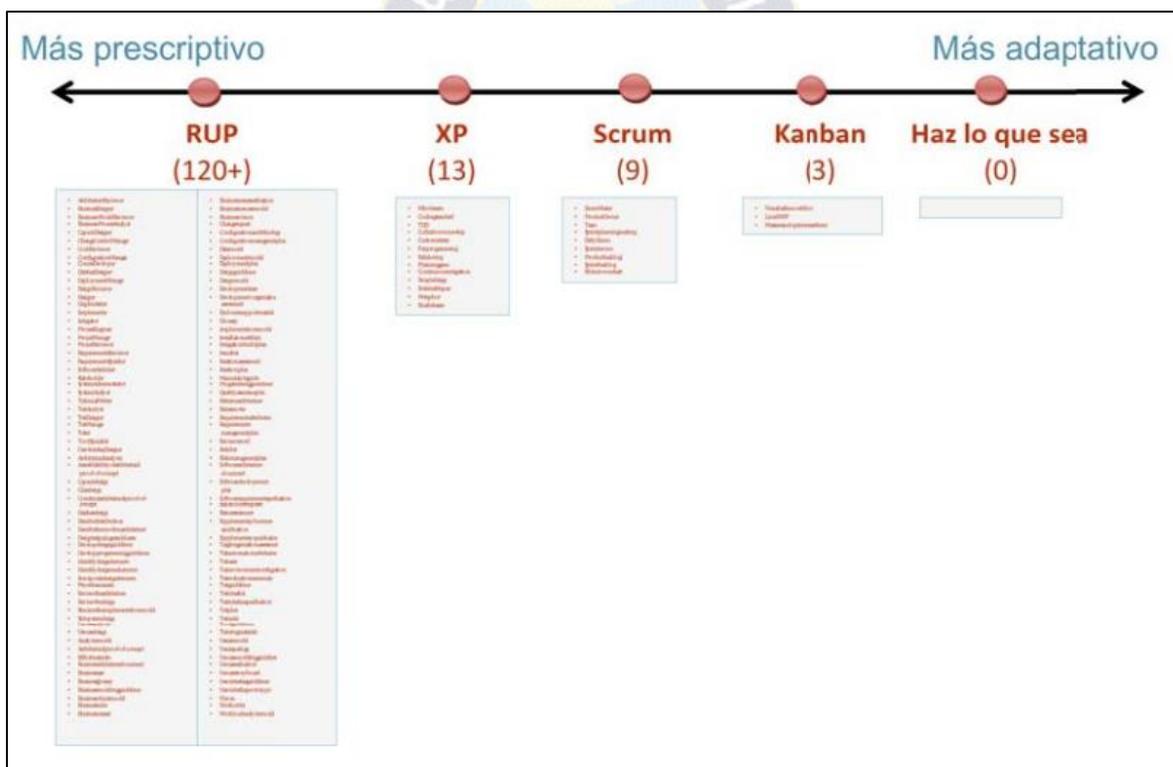


Figura 2.1: Relación de Kanban frente a otras metodologías.

Fuente: Kninberg y Skarin, 2010

2.4.1. ROLES KANBAN

Kanban, a diferencia de otras metodologías ágiles como Scrum, no prescribe roles o reuniones. Pero existen una serie de roles que pueden ser adoptados en Kanban, estos han

sido creados a partir de la observación en organizaciones que han utilizado el método y lo han evolucionado de forma colaborativa (Palacios, 2016).

2.4.1.1. SERVICE REQUEST MANAGER

Se encarga de gestionar la demanda y los requisitos dentro del sistema Kanban, manejando las relaciones con los stakeholders y fomentando la transparencia dentro del sistema en torno a la priorización del trabajo. Alternativamente, este se puede llamar Product Owner.

2.4.1.2. SERVICE DELIVERY MANAGER

Es responsable del flujo de trabajo dentro de un sistema Kanban y/o determinados ítems de trabajo y facilita el Kanban Meeting y el Delivery Planning. Algunos nombres alternativos son Flow Manager, Delivery Manager o incluso Flow Master.

2.4.2. PRINCIPIOS DE LA METODOLOGÍA KANBAN

La metodología Kanban está basado en principios que le diferencian de las demás metodologías ágiles:

2.4.2.1. VISUALIZAR EL WORKFLOW (FLUJO DE TRABAJO)

Consiste en la representación visual de todo el flujo de trabajo mediante un tablero físico, asequible, que refleje la realidad del equipo en cada momento.

El objetivo de mostrar todo el proceso consiste en:

- Entender mejor el proceso actual.
- Conocer los problemas que puedan surgir y tomar decisiones.
- Mejorar la comunicación entre todo el equipo del proyecto.
- Hacer futuros procesos más predecibles.
- Conocer la información visual de qué miembro del equipo está ejecutando cada tarea.

La figura 2.2 muestra un tablero Kanban constituido por 5 columnas, que representan las diferentes fases por las que una tarea tiene que pasar para ser terminada (Cola de espera,

análisis, desarrollo, implementar y producción). Cuatro fases están subdivididas en dos estados que son en curso y lista, para pasar a la siguiente fase. El estado en curso significa que el equipo actualmente está trabajando en esa tarea y el estado lista significa que el equipo ya ha acabado la tarea que tenía que ejecutar en esta fase y puede pasar a la siguiente fase. Esta división nos permite localizar atascos en el proceso de producción del proyecto.

Cola de espera	Análisis 3		Desarrollo 3		Implementar 5		Producción 4	
	En curso	Lista	En curso	Lista	En curso	Lista	En curso	Lista
■ ■	■	■		■	■	■		
■ ■	■			■	■			
■ ■				■	■			
					■			

Figura 2.2: Ejemplo de tablero Kanban

Fuente: Elaboración propia

2.4.2.2. LIMITAR EL TRABAJO EN CURSO (WIP)

Una vez que se conoce la realidad actual del equipo, se tiene que aplicar la primera restricción del Kanban, que es limitar el trabajo en curso (WIP, por sus siglas en inglés work in progress).

Consiste en acordar la cantidad de tareas que se pueden realizar en cada fase, es decir en cada columna del tablero Kanban.

El principal objetivo de establecer estos límites, es detectar los cuellos de botella para que el equipo se dedique a terminar aquellas tareas que están bloqueadas, esto motiva a la colaboración del equipo en diferentes fases. Hay que lograr que solo realice la cantidad de tareas que el tablero acepta.

Según Bermejo (2015), para limitar el WIP la clave es experimentar empíricamente y empezar con un número no demasiado pequeño al principio. La mejor opción es optar por un límite alto e irlo bajando cada dos o tres semanas hasta encontrar un equilibrio.

Kniberg (2010), basado en su experiencia establece un límite $2n-1$ donde n es el número de miembros del equipo y -1 un modo de potenciar la cooperación. Para poder obtener un límite correcto es mejor descubrir sobre la marcha.

2.4.2.3. 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), 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.

Con estos valores, la optimización del flujo de trabajo consistirá en la búsqueda de minimizar el cycletime, maximizar el throughput.

2.4.3. CUELLOS DE BOTELLA

Los cuellos de botella se encuentran cuando se llega la limite WIP de una fase de desarrollo esto provoca un atasco en el desarrollo y es muy importante detectarlos y enfocarse en solucionar, por ejemplo la figura 2.3 muestra un cuello de botella en la fase de desarrollo.

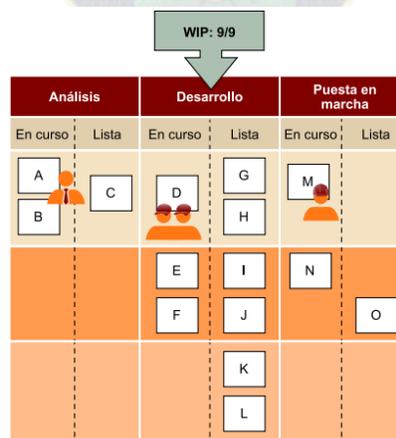


Figura 2.3: Ejemplo de cuello de botella en la fase de desarrollo
Fuente: Bermejo, 2015

En ese momento, se pueden tomar tres decisiones diferentes:

- **La peor.** Aumentar o sacar el límite WIP. Sabemos la teoría y sabemos que el Kanban limita el WIP, pero como no podemos, y es imposible, decidimos sacarlo y ponemos un límite WIP de 10, para más adelante subirlo a 15. Y realmente no ayuda en nada a la empresa y al equipo.
- **La mala.** Contratar a gente para la fase en el cuello de botella. La solución en un equipo sobrecargado no siempre es poner más gente. Se debe tener en cuenta que incorporar personal nuevo necesita tiempo de dedicación, hay un periodo en el que se tiene que invertir tiempo en formar al personal nuevo (nunca nadie entra sabiéndolo todo, por muy experto que sea quien se contrata) y este tiempo se estará quitando de las personas que se encuentran justamente en la fase de la que depende el tiempo de servicio del equipo.
- **La buena.** Poner a uno de los programadores a ayudar en la fase de puesta en marcha. Un buen equipo de alto rendimiento es aquel que está formado por personas expertas en alguna tarea, pero que conoce y se interesa por todas las que le rodean. En el momento en el que al programador se le dice que deje de hacer lo que está haciendo y que haga otra tarea, una respuesta muy habitual es "mi trabajo es programar, a mí me pagan por programar y programar es lo que voy a hacer". En ese momento, hay que tener muy claro que el objetivo de todo el equipo tiene que ser el servicio al cliente (es decir, facturar). Un trabajador que no está dispuesto a asimilar que el trabajo de programar no es lo que el sistema de producción necesita en ese momento (donde el panel Kanban refleja que tenemos nueve tareas programadas y solo una facturada), no le está haciendo un buen favor a la empresa.

2.4.4. MÉTRICAS EN KANBAN

En este punto de madurez del Kanban en el equipo de desarrollo, se empieza a obtener datos estadísticos del equipo.

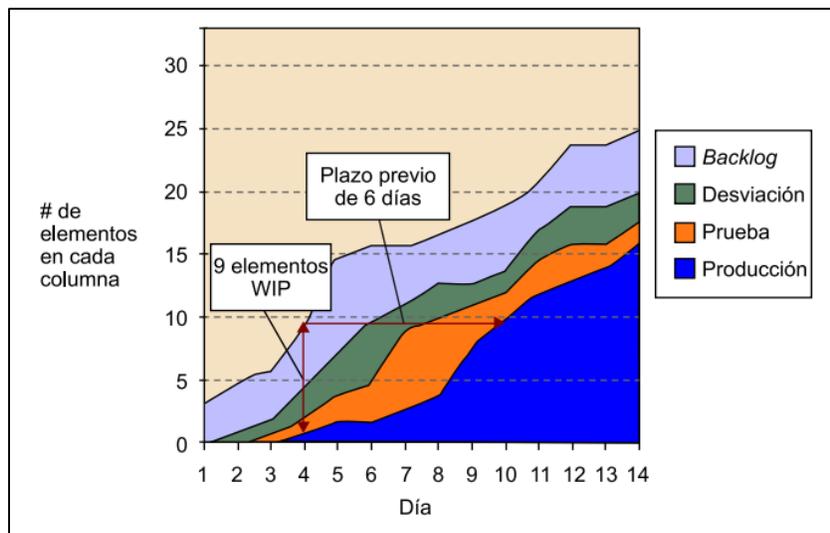


Figura 2.4: Gráfico fecha de entrada y salida de tareas.

Fuente: Bermejo, 2015

Una muy buena opción es anotar las fechas de entrada y salida de la tarea por cada fase. Así podréis ir obteniendo toda una serie de gráficos del tiempo que tardan las tareas en ser servidas y qué fases son las que más tiempo exigen como se muestra en la figura 2.4.

Un simple gráfico acumulativo nos dará mucha información en un solo vistazo: ¿cuántas tareas pendientes hay a día de hoy? ¿Cuántas hay aproximadamente en producción? ¿Cuál tiene una pendiente más pronunciada? Si crecen más las tareas pendientes que las servidas al cliente tenemos un problema

2.5. INGENIERÍA WEB

La ingeniería web es la aplicación de sólidos principios científicos, de ingeniería y enfoques disciplinados y cuantificables al desarrollo, despliegue y mantenimiento exitoso de sistemas y aplicaciones basadas en web de alta calidad.

Actualmente existen millones de sitios web que cubren diferentes necesidades de la vida, seguramente la comunicación es la más importante teniendo en cuenta que las distancias pasaron a un segundo plano. El avance de Internet y las nuevas tecnologías de comunicación, marcan una nueva tendencia en el mercado del software.

La exigencia de los usuarios, cada vez más numerosos, hizo que Internet creciera en forma acelerada y desprolija, produciendo un impacto social, económico y político impensado.

Según Pressman (2010), el aumento de páginas web por medio de herramientas de desarrollo permitió a los ingenieros web brindar capacidad de combinar contenido estático con información almacenada en una base de datos, de esta manera nacen los sistemas y aplicaciones basados en la web (webapps). En la actualidad las webapps son una herramienta sofisticada y útil para el usuario final, empresas, negocios y otros.

Las webapps son accesibles y llegan a atender las necesidades de diferentes tipos de usuarios, ya sea de manera global por medio de internet o de manera local a través de intranet. Pueden acceder un gran número de usuarios a la vez.

2.6. INGENIERIA 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.

2.6.1. FASES DE DESARROLLO

Las actividades base de modelado de UWE son el análisis de requerimientos, el modelo conceptual, el modelo de navegación y el modelo de presentación. A estos modelos se pueden sumar otros modelos como son el modelo de usuario, modelo de adaptación y modelo de tareas para representar los aspectos dinámicos de la aplicación mediante la descripción de situaciones. De esta manera se obtiene una colección de modelos y diagramas que describen una aplicación web de manera integral (Pérez, 2010).

En su implementación se deben contemplar las siguientes etapas y modelos:

- **Análisis de requisitos:** Plasma los requisitos funcionales de la webapp mediante un modelo de casos de uso.
- **Modelo de contenido:** Define, mediante un diagrama de clases, los conceptos a detalle involucrados en la aplicación.
- **Modelo navegacional:** Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.
- **Modelo de presentación:** Representa las interfaces de usuario por medio de vistas abstractas.
- **Modelo de proceso:** Representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

2.6.2. ANÁLISIS DE REQUISITOS

El análisis de requisitos en UWE se modela con casos de uso. Está conformado por los elementos actor y caso de uso. En este sentido, los actores se utilizan para modelar los usuarios de la webapp (Nieves, Ucán y Menéndez, 2014).

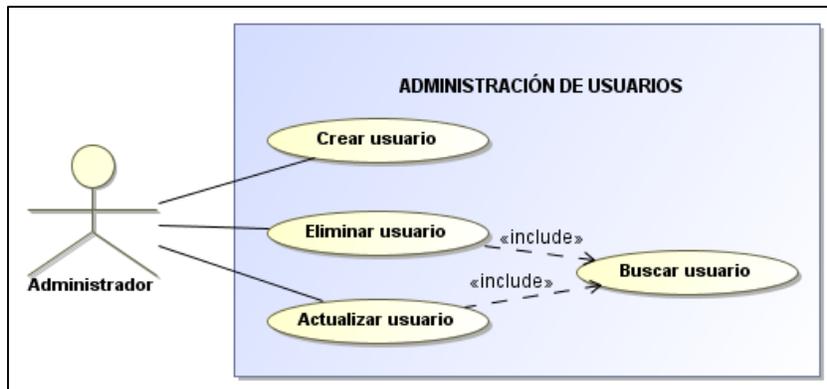


Figura 2.5: Ejemplo de un diagrama de caso de uso.

Fuente: Elaboración propia

Permite diferenciar los procesos de navegación de los procesos del negocio mismo mediante el uso de diagramas de casos de uso para la captura de requisitos, como se muestra en la figura 2.5, que da como resultado un modelo de casos de uso acompañado de documentación que describe las reglas de adaptación, los usuarios y las interfaces. Esta etapa permite determinar cuáles son los actores (usuarios) de la aplicación, representando la funcionalidad de cada uno de ellos en el sistema.

2.6.3. MODELO DE CONTENIDO

El modelo de contenido es el modelo conceptual del dominio de aplicación tomando en cuenta los requerimientos especificados en los casos de uso y se representa con un diagrama de clases.

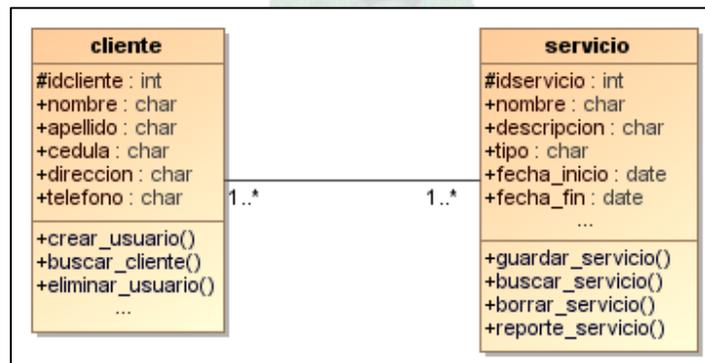


Figura 2.6: Ejemplo de un modelo conceptual.

Fuente: Elaboración propia

Este modelo especifica cómo se encuentran relacionados los contenidos del sistema, es decir, define la estructura de los datos que se encuentran alojados en el sitio web. En la figura 2.6 se muestra un ejemplo de este modelo.

2.6.4. MODELO NAVEGACIONAL

Según Nolivos y Coronel (2010), el modelo de navegación de una aplicación web comprende la especificación de qué objetos pueden ser visitados mediante la navegación, a través del sistema y las asociaciones entre ellos. Mediante estos diagramas se representa el diseño y la estructura de las rutas de navegación al usuario para evitar la desorientación en el proceso de navegación. Los elementos básicos en el modelo de navegación son los nodos y enlaces como se puede apreciar en la Figura 2.7.

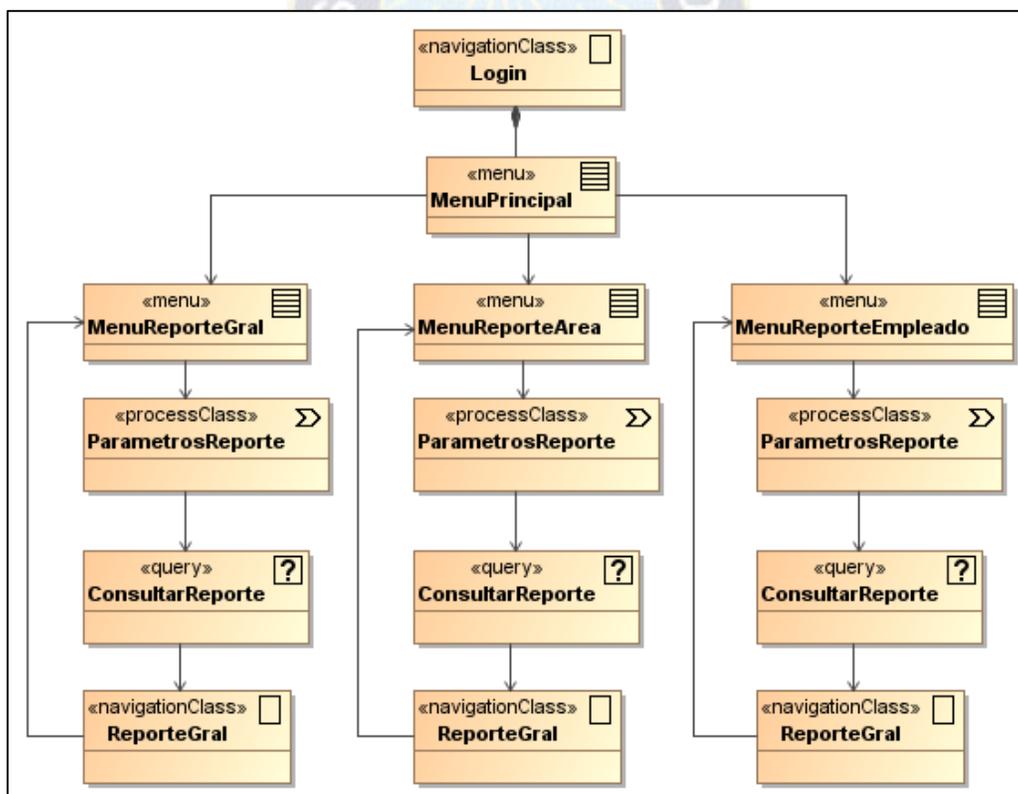


Figura 2.7: Ejemplo de un modelo de navegación.

Fuente: Elaboración propia

En la figura 2.8 se muestran los estereotipos y sus iconos que se pueden usar en este modelo:



Figura 2.8: Nombre de estereotipos y sus iconos del modelo navegacional.
Fuente: UWE, 2016

En un sistema para la web es útil saber cómo están enlazadas las páginas. Esto significa que necesitamos un diagrama con nodos (nodes) y enlaces (links). Los nodos son unidades de navegación y están conectados por medio de enlaces, pueden ser presentados en diferentes páginas o en una misma página.

Para los nodos y enlaces los estereotipos usados «navigationClass» en caso de que el enlace implique ejecución de código y «navigationLink» cuando no implica ejecución de código.

2.6.5. MODELO DE PRESENTACIÓN

El modelo de presentación en UWE está muy relacionado con los elementos de las interfaces definidas en HTML. Estos elementos también están definidos como estereotipos de UML. Los elementos del modelo de presentación son: ventanas, entradas de texto, imágenes, audio y botones (Nolivos y Coronel, 2010).

Los estereotipos y sus iconos que se utilizan en este diseño se muestran en la figura 2.9:

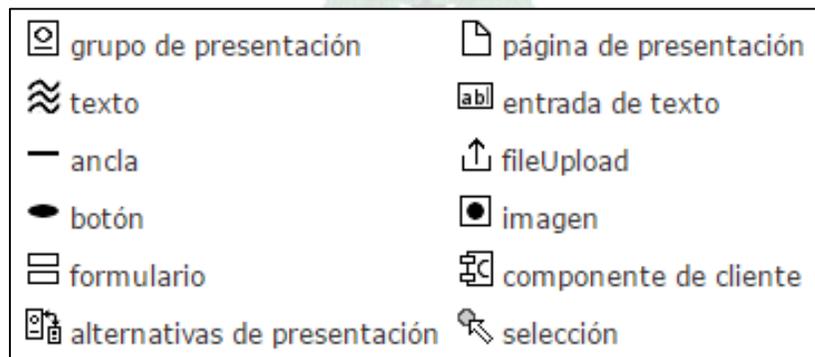


Figura 2.9: Nombre de estereotipos y sus iconos del modelo de presentación.
Fuente: UWE, 2016

Como se puede ver en la figura 2.10, estos diagramas permiten especificar dónde y cómo los objetos de navegación serán presentados al usuario, una representación esquemática visible para el usuario.

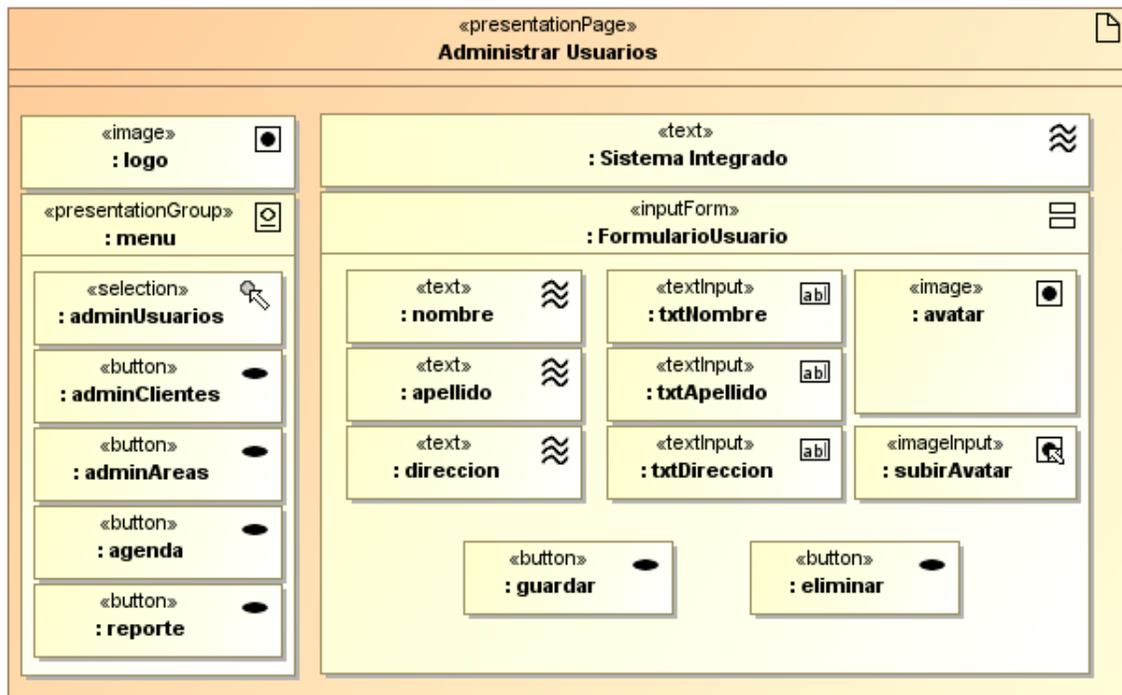


Figura 2.10: Ejemplo de un modelo de presentación.
Fuente: Elaboración propia.

2.6.6. MODELO DE PROCESOS

Este modelo provee los elementos necesarios para representar procesos de negocio en un modelo UWE. Al contrario del modelo de navegación que representa la estructura estática de la aplicación Web, el modelo de proceso representa la parte dinámica de la misma.

2.6.6.1. MODELO DE FLUJO DEL PROCESO

Este tipo de modelo define el comportamiento y los detalles de un proceso de negocio. El process flow, como también se lo llama, describe minuciosamente los pasos dentro de un proceso, en el caso que el usuario navegue por éste. Maneja algunos estereotipos como se observa en la Figura 2.11



Figura 2.11: Estereotipos y sus iconos del modelo de procesos.

Fuente: UWE, 2016

El estereotipo «user Action» es usado para indicar interacciones de usuario con la página web iniciando un proceso o respondiendo a un requerimiento explícito de información. Por el contrario, «system Action» describe acciones, que son ejecutadas por el sistema. Ambos tipos de acciones pueden ser agregadas usando la barra de herramientas (toolbar).

La Figura 2.12 muestra un ejemplo de modelo de flujo de proceso:

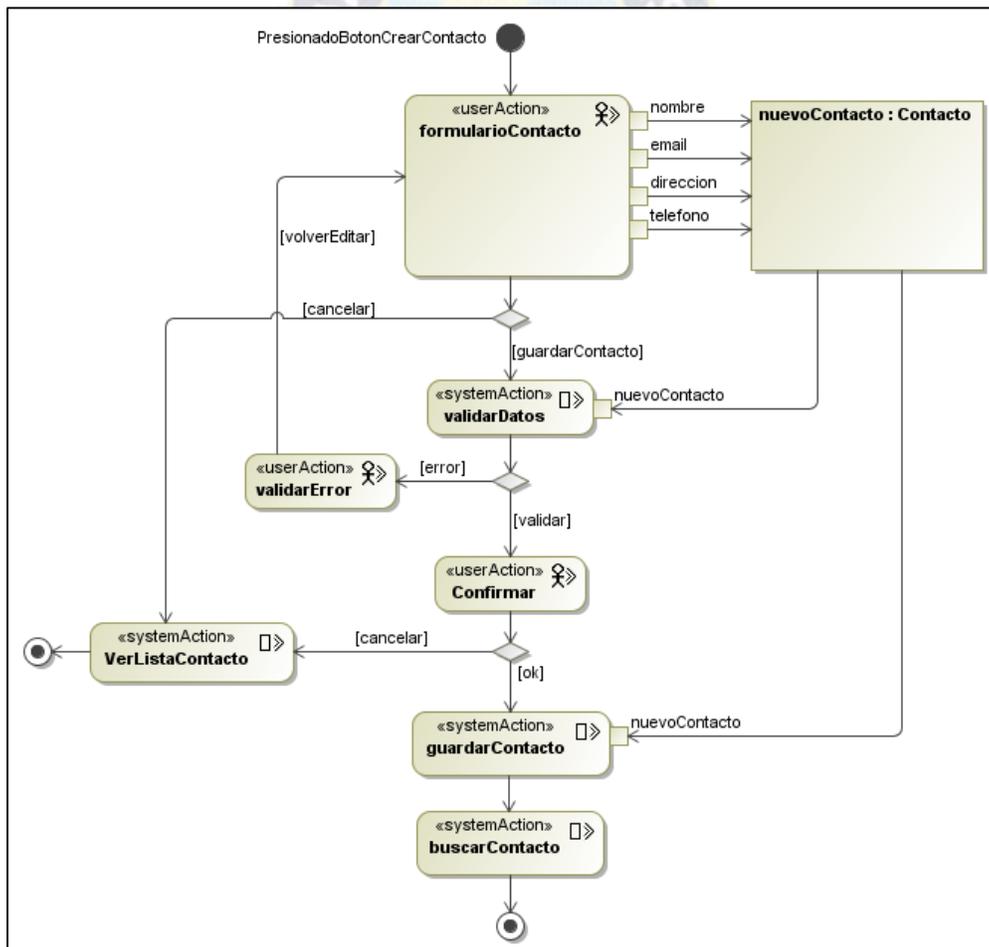


Figura 2.12: Ejemplo de un modelo de flujo de proceso.

Fuente: Elaboración propia.

2.7. PATRONES DE DISEÑO

Los patrones son soluciones simples y elegantes a problemas específicos y específicos en el desarrollo de software. Son soluciones basadas en la experiencia y que se han demostrado que funcionan.

Cada patrón es adecuado para ser adaptado a un cierto tipo de problema y con eso se pretende:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas de software, lo que significa que debe ser aplicable a diferentes problemas de diseño en distintas circunstancias.
- Evitar la reiteración en la búsqueda de soluciones ya conocidos y solucionados anteriormente.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento existente.

2.7.1. MODELO VISTA CONTROLADOR (MVC)

EL Modelo Vista Controlador (MVC), es un patrón de diseño que separa en capas bien definidas el desarrollo de una aplicación, esas partes son tres, el Modelo encargado de la lógica del negocio y la persistencia de los datos, las Vistas son las responsables de mostrar al usuario el resultado que obtienen del modelo a través del controlador, el Controlador encargado es el encargado de gestionar las peticiones del usuario, procesarlas invocando al modelo y mostrarlas al usuario a través de las vistas (Mouse, 2011).

A continuación se describe cada uno de los componentes citados:

- **Modelo:** El modelo es la representación lógica de la información y describe la funcionalidad del sistema, por lo tanto se encarga de gestionar los permisos para ingresar a dicha información, también se encarga de realizar recuperación, actualización, inserción y eliminación de datos; a través de un conjunto de reglas (lógica de negocio).

- **Vista:** La vista es la interfaz de usuario. Muestra al usuario una representación visual del modelo, sus datos y estado, tomándolos directamente del modelo. También contiene los elementos de la interfaz que permiten al usuario interactuar con el programa, tales como botones y menús. Sin embargo, no es tarea de la vista implementar cómo se deben comportar esos elementos.
- **Controlador:** El controlador es el intermediario entre la vista y el controlador, responde a eventos generados por el usuario el cual constituyen llamados al Modelo (en el caso que se solicite alguna información como lista de clientes, consultar un proveedor o editar un producto) o a la Vista (en el caso de mostrar un formulario o un reporte estadístico). En pocas palabras el componente que hace posible tener separada la lógica de negocio con la vista es el Controlador.

En la figura 2.13 se muestra el funcionamiento del patrón MVC, que se describe en los siguientes pasos (Pilar, 2014):

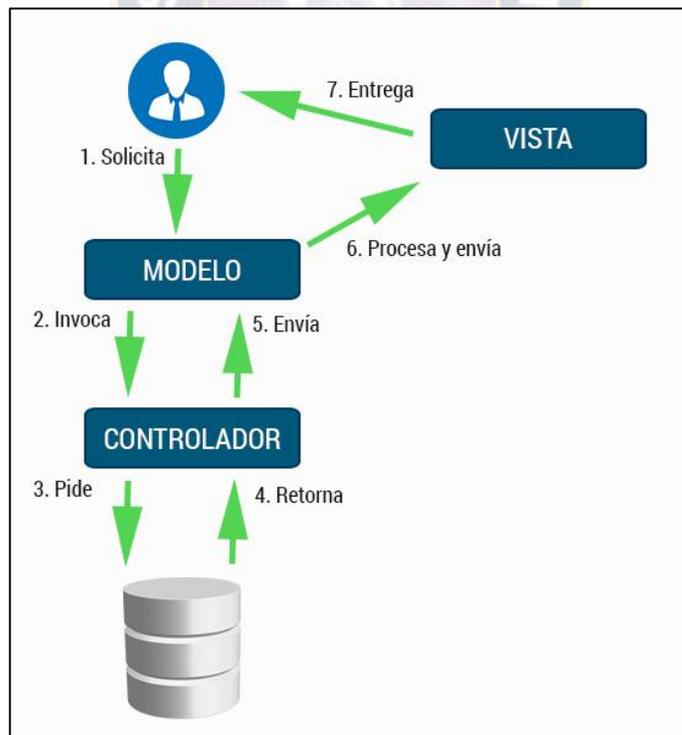


Figura 2.13: Modelo Vista Controlador

Fuente: Elaboración propia

1. El usuario realiza una petición que captura el controlador
2. El controlador llama al modelo correspondiente
3. El modelo solicita la información a la base de datos
4. El modelo recoge la información de la base de datos
5. El controlador recibe la información
6. El controlador procesa y envía la información a la vista
7. La vista entrega al usuario la información de forma clara y precisa

2.8. TECNOLOGÍAS WEB

2.8.1. PHP

PHP acrónimo de Hypertext Preprocessor es un lenguaje de programación del lado del servidor diseñado originalmente para la generación de páginas web dinámicas. Es un lenguaje de programación interpretado o de script que permite insertar fragmentos de código dentro de una página HTML y realizar determinadas acciones de forma fácil y eficiente.

Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en una herramienta ideal al momento de crear una página web dinámica:

- Fácil de mantener y mejorar el código desarrollado.
- Soporte para una gran cantidad de base de datos como MySQL, PostgreSQL, Oracle, MariaDB, MS SQL Server, entre otras.
- Facilidad de integración con varias bibliotecas externas utilizando Composer.
- Amplia comunidad de desarrolladores, por ser de código abierto, goza del aporte de gran número de programadores.
- Permite aplicar técnicas de programación orientada a objetos.

2.8.2. FRAMEWORK CODEIGNITER 3

CodeIgniter es un conjunto de herramientas para desarrollar aplicaciones web usando PHP. Su objetivo es permitir desarrollar proyectos mucho más rápido, al proveer un rico conjunto

de bibliotecas para tareas comúnmente necesarias, tanto como una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. Permite enfocarse en el proyecto al minimizar la cantidad de código necesario, está liberado bajo licencias open source como Apache/BSD, así que puede usarse donde sea necesario.

El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

Usa el patrón MVC, que permite una separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo:

- El Modelo representa sus estructuras de datos. Típicamente sus clases del modelo contendrán funciones que los ayudarán a devolver, insertar y actualizar información de su base de datos.
- La Vista es la información que se presenta al usuario. Una vista será normalmente una página web, pero en CodeIgniter, una vista también puede ser un fragmento de página como el encabezado o pie de página.
- El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

En la figura 2.14 se puede observar el diagrama de flujo de la aplicación.

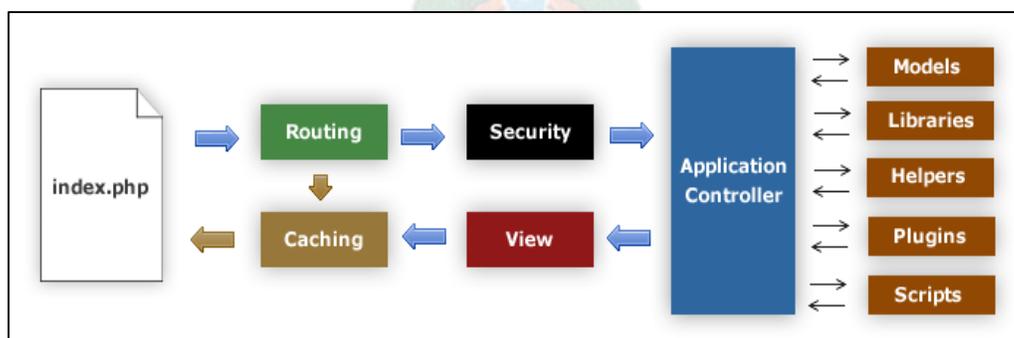


Figura 2.14: Diagrama de flujo de la aplicación.

Fuente: CodeIgniter, 2016.

1. El index.php sirve como el controlador frontal, inicializando los recursos básicos que necesita CodeIgniter para ejecutar.
2. El Ruteador examina la solicitud HTTP para determinar que debería hacer con ella.
3. Si existe el archivo de caché, se lo envía directamente al navegador, sin pasar por la ejecución normal del sistema.
4. Seguridad. Antes que se cargue el controlador de la aplicación, por razones de seguridad se filtran la solicitud HTTP y cualquier otro dato enviado por el usuarios.
5. El controlador carga el modelo, las bibliotecas del núcleo, helpers, y cualquier otro recurso requerido para procesar una solicitud específica.
6. La Vista terminada se procesa y se envía al navegador para que se pueda ver. Si el caché está habilitado, la vista se cachea primero para que las siguientes solicitudes que la necesiten puedan ser servidas.

2.8.3. JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden ser interpretados por los navegadores.

2.8.4. JQUERY

Jquery es un framework de JavaScript, permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejo de eventos, desarrollar animaciones y agregar interacción con la técnica AJAX en páginas web (Comesaña, 2012).

2.8.5. BOOTSTRAP 3

Bootstrap es un framework de CSS, de código abierto, en otras palabras es un conjunto de archivos CSS que se incluye en una página y se puede empezar a maquetar el sitio web en minutos, sin tocar una sola línea de CSS, esto agiliza el desarrollo de aplicaciones web

quitándonos de encima todo la carga del diseño y dándonos un diseño elegante y bueno gracias a sus clases ya predefinidas.

2.8.6. GESTOR DE BASE DE DATOS MYSQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

Son muchas las razones para escoger MySQL como solución para la administración de datos (Gilfillan, 2003):

- Coste: El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- Asistencia: Existe una nutrida y activa comunidad MySQL.
- Facilidad de uso: MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- Velocidad: MySQL es mucho más rápido que la mayor parte de sus rivales.
- Funcionalidad: MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplication, funciones SSL e integración con la mayor parte de los entornos de programación.
- Portabilidad: MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.

3.1. INTRODUCCIÓN

En el desarrollo de este capítulo se analiza la combinación de la metodología ágil Kanban con la metodología UWE, también se describe la planificación y desarrollo del sistema web de seguimiento y control de la información de clientes para la empresa LINXS S.R.L.

3.2. COMBINACIÓN DE METODOLOGÍAS KANBAN – UWE

Kanban nos permite visualizar el trabajo que realizamos, esto es una ayuda porque nos podemos saber en qué fases tenemos cuellos de botella, o qué fases y personas están libres.

La construcción del tablero Kanban es de vital importancia, se deben elegir las fases de acuerdo a las necesidades del sistema a desarrollar, en este caso como se ve en la figura 3.1 el tablero Kanban que se construye consta de 6 columnas descritas a continuación:

- **Pedido:** En esta fase representa la pila de todos los requerimientos que se necesita para el desarrollo del sistema.
- **Análisis:** Esta fase se encarga de hacer el análisis del sistema, 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 realiza el desarrollo del prototipo del sistema, se hace uso del modelo de flujo de proceso de la metodología UWE.
- **Implementar:** En esta fase se dedica a implementar el prototipo del sistema, utilizando el framework Codeigniter y todas las herramientas web propuestas al inicio del proyecto.

- **Producción:** Esta es una columna donde estará la pila de trabajos finalizados, subidos en el servidor de la empresa.

Las columnas análisis, diseño, desarrollo e implementar se subdividen en dos columnas, en curso y hecho, la columna en curso es para todos las tareas que se están realizando, la columna hecho es para las tareas terminadas en esa fase y están listas para avanzar a la siguiente.

Pedido	Análisis 3		Diseño 3		Desarrollo 2		Implementar 2		Producción
	En curso	Hecho	En curso	Hecho	En curso	Hecho	En curso	Hecho	
Módulo de administración de usuarios									
Módulo de gestión de clientes y servicios									
Módulo de reportes									

Figura 3.1: Combinación de metodología Kanban y UWE

Fuente: Elaboración propia

En la columna pedido están las tres tareas que vamos a realizar para poder terminar el sistema, estas tareas son el módulo de administración de usuarios, el módulo de gestión de clientes y servicios y el módulo de reportes, estas tareas pasarán por las cuatro fases hasta llegar a la fase de producción.

3.3. ROLES KANBAN

La tabla 3.1 muestra las fases de trabajo detectados en el tablero Kanban, el límite WIP y los responsables que componen del equipo Kanban.

Tabla 3.1: Fases de trabajo, limite WIP y responsables

FASE	WIP	RESPONSABLE
Pedido	-	Propietario del producto: Lic. Roberto Claros Uzqueda Flow Manager: Hector Alberto Cussi Merlo
Análisis	3	Analista: Hector Alberto Cussi Merlo
Diseño	3	Diseñador: Hector Alberto Cussi Merlo
Desarrollo	2	Desarrollador: Hector Alberto Cussi Merlo
Implementar	2	Desarrollador: Hector Alberto Cussi Merlo
Producción	-	Flow Manager: Hector Alberto Cussi Merlo

3.4. MÓDULO DE ADMINISTRACIÓN DE USUARIOS

3.4.1. FASE DE ANÁLISIS

3.4.1.1. ANÁLISIS DE REQUERIMIENTOS

Implementado la metodología UWE, a continuación se detalla la lista de los usuarios involucrados y los roles con sus funciones respectivas.

1. El administrador del sistema y encargado del área contable crean, editan, eliminan y consultan datos de los usuarios.
2. El administrador del sistema crea, edita, elimina y consulta datos de los roles de usuario.
3. El administrador del sistema habilita las acciones que cada rol y usuario de tener para acceder a ciertas partes del sistema.

De acuerdo con la estructura jerárquica de la empresa se identificaron los siguientes roles de usuario para el sistema web con lo que se controlará el acceso a ciertas partes del sistema.

- Administrador del sistema
- Encargado de área contable
- Jefe de sistemas
- Vendedores
- Mensajero

3.4.1.2. CASOS DE USO

La figura 3.2 muestra el caso de uso para administrar roles de usuarios:

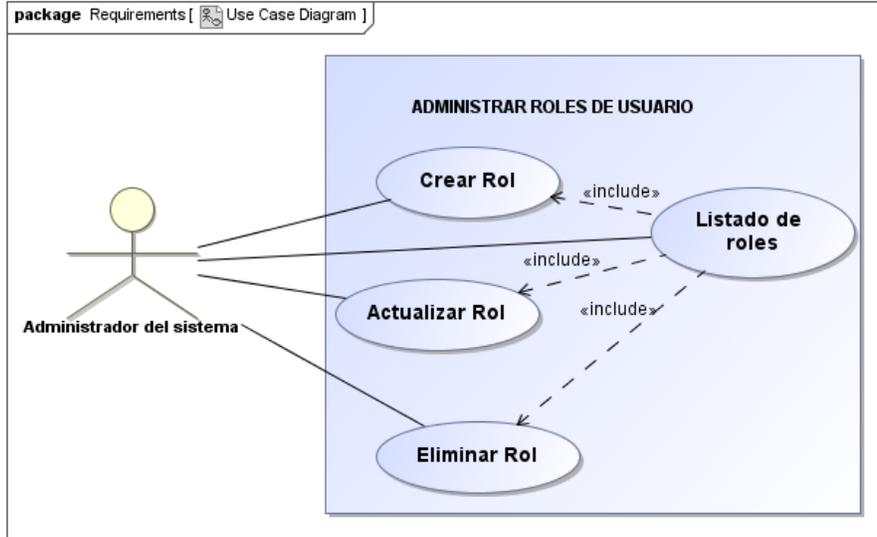


Figura 3.2: Caso de uso administrar roles de usuario.

Fuente: Elaboración propia

La tabla 3.2 detalla las especificaciones del caso de uso administrar roles de usuario:

Tabla 3.2: Especificación del caso de uso administrar roles de usuario

Caso de uso: Administrar roles de usuario
Actores: Administrador del sistema
Descripción: Permite al administrador del sistema crear, editar, elimina o ver el listado de roles para ser asignados a los usuarios registrados.
Precondiciones: El actor debe ser un usuario registrado y con el rol asignado de administrador del sistema.
Flujo Normal: <ol style="list-style-type: none"> 1. El actor ingresa a la opción de roles de usuario. 2. El sistema despliega el listado de roles registrados, cada uno con las opciones de crear, editar y eliminar. 3. El actor escoge una de las opciones.
Postcondiciones: El sistema ejecuta la opción elegida por el actor y muestra los cambios realizados.

La figura 3.3 muestra el caso de uso para administrar usuarios:

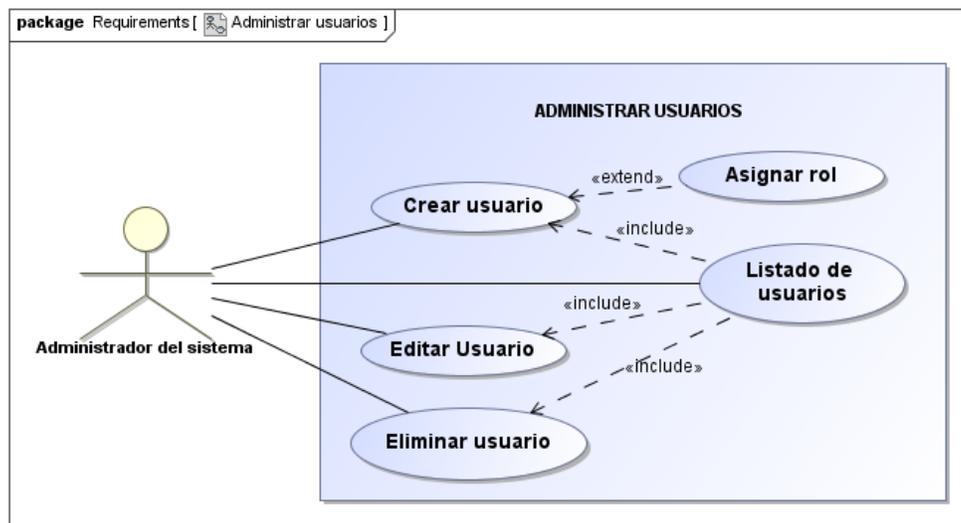


Figura 3.3: Caso de uso administrar usuarios.

Fuente: Elaboración propia.

El detalle de las especificaciones del caso de uso administrar usuarios se muestra en la tabla 3.3:

Tabla 3.3: Especificación del caso de uso administrar usuarios.

Caso de uso: Administrar usuarios
Actores: Administrador del sistema
Descripción: Permite al administrador del sistema crear, editar, elimina o ver el listado de usuarios
Precondiciones: El actor debe ser un usuario registrado y con el rol asignado de administrador del sistema. El rol que se asigne tiene que estar creado y habilitado.
Flujo Normal: 1. El actor ingresa a la opción de administración de usuario. 2. El sistema despliega el listado de usuarios registrados, cada uno con las opciones de crear, editar, eliminar y rol. 3. El actor escoge una de las opciones.
Postcondiciones: El sistema ejecuta la opción elegida por el actor y muestra los cambios realizados.

3.4.1.3. MODELO DE CONTENIDOS

El modelo de contenidos especifica las clases que se ha realizado en la etapa de desarrollo del módulo de administración de usuarios, teniendo en cuenta clase Usuario que contiene datos necesario, la clase Roles que se asignarán a los usuario registrados y la clase Permiso que controla a los roles de cada usuario, la figura 3.4 muestra el diagrama de contenidos de la administración del sistema.

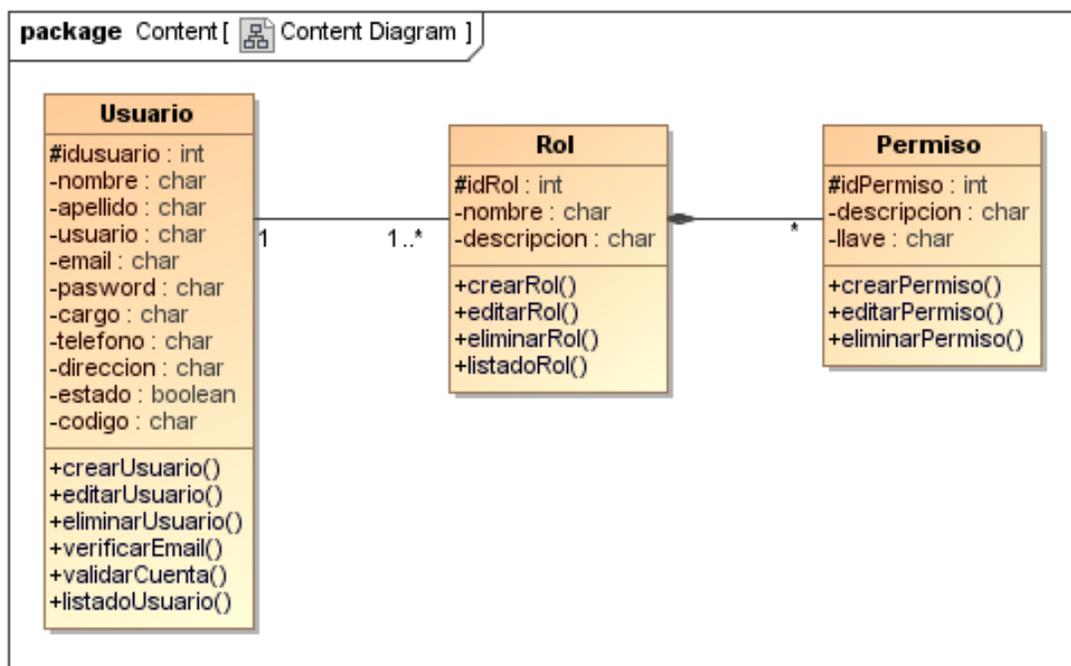


Figura 3.4: Diagrama de contenidos de la administración del sistema.

Fuente: Elaboración propia

3.4.2. FASE DE DISEÑO

3.4.2.1. MODELO NAVEGACIONAL

El modelo navegacional determina las páginas de Administración de Usuarios, utilizando los nodos (nodes) y enlaces (links), en la Figura 3.5 muestra el comportamiento del usuario en el momento de ingresar a la página Administración de Usuario, donde puede elegir entre ver el listado de roles, listado de usuarios y crear roles y usuarios..

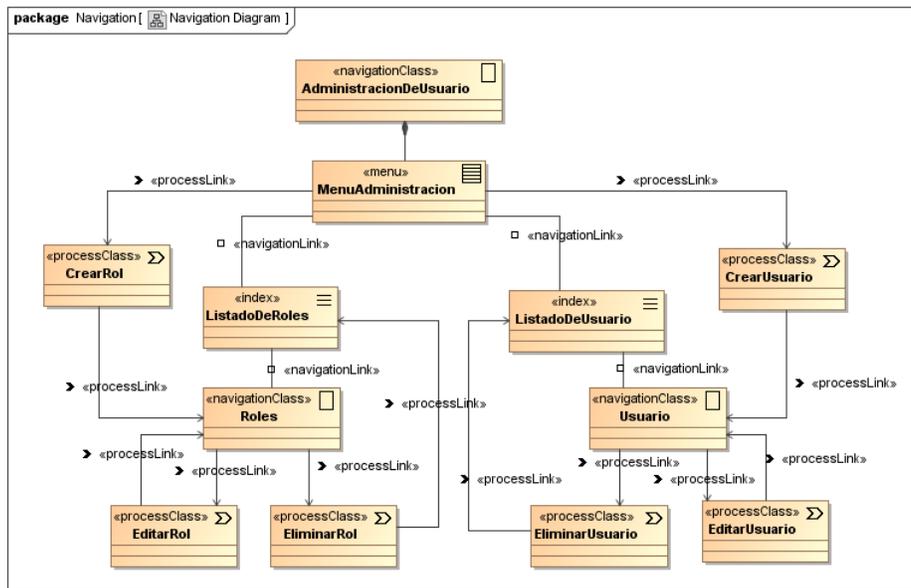


Figura 3.5: Diagrama de navegación para la administración de usuarios.

Fuente: Elaboración propia

3.4.2.2. MODELO DE PRESENTACIÓN

Para el modelo de presentación se realizó el diseño de interfaz de la vista de administración de usuarios, como se puede observar en la figura 3.6, consta del listado de usuarios con datos importantes para ser encontrado de manera rápida, como su nombre de usuario, cargo que ocupa en la empresa, rol asignado por el administrador del sistema y sus botones de acciones para editar, eliminar y ver sus permisos

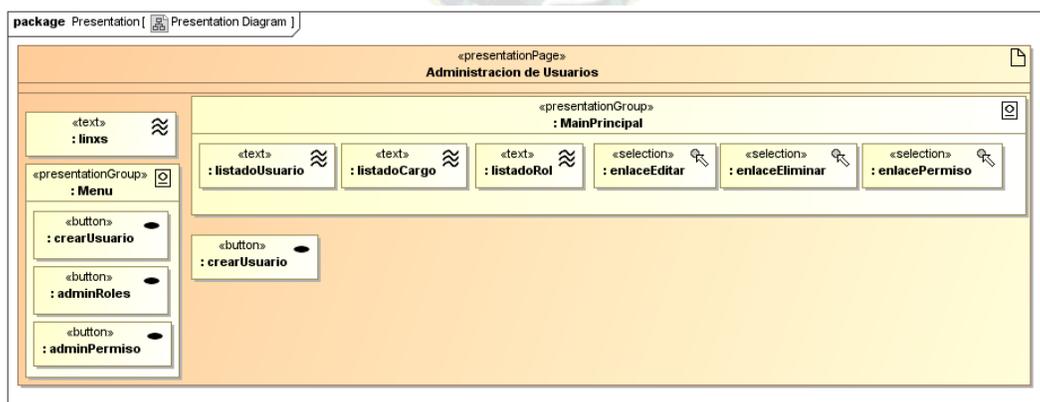


Figura 3.6: Diseño de la vista administración de usuarios

Fuente: Elaboración propia

En la figura 3.7 se muestra el diseño de la vista de registro de usuarios:

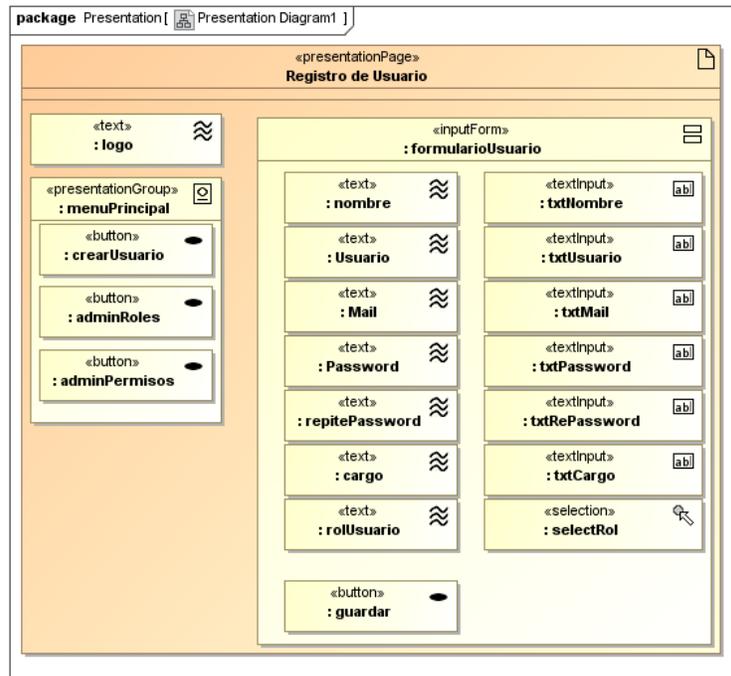


Figura 3.7: Diseño de la vista registro de usuario
Fuente: Elaboración propia

La figura 3.8 muestra el diseño de la vista de registro de roles.

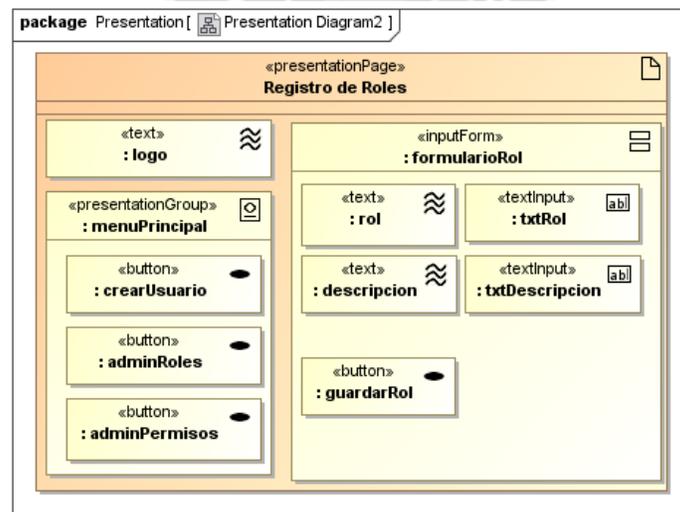


Figura 3.8: Diseño de la vista registro de usuario
Fuente: Elaboración propia

La figura 3.9 muestra la interfaz de usuario para controlar los permisos de los roles.

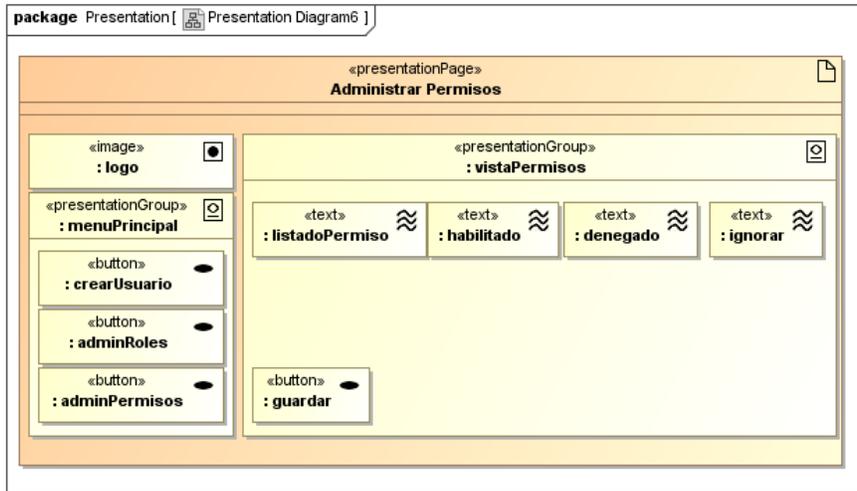


Figura 3.9: Diseño de la vista de control de permisos de roles.

Fuente: Elaboración propia

3.4.3. FASE DE DESARROLLO

3.4.3.1. MODELO DE FLUJO DE PROCESO

En este modelo se encuentra el flujo de proceso del comportamiento de la administración de usuarios como muestra la figura 3.10.

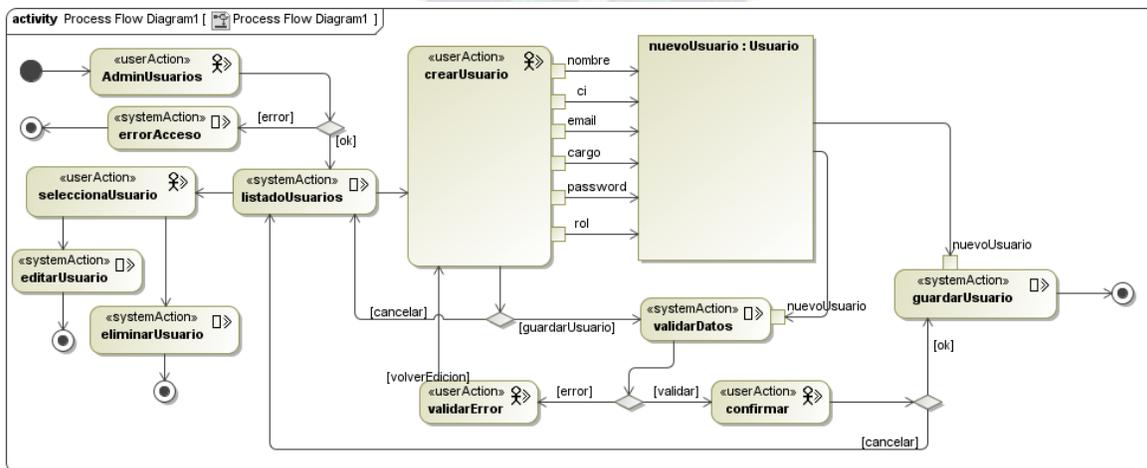


Figura 3.10: Diseño del modelo de flujo de procesos de la administración de usuarios.

Fuente: Elaboración propia

3.4.4. FASE DE IMPLEMENTACIÓN

Para esta parte se desarrolló el prototipo de la página de Administración de usuarios como se puede observar la figura 3.11 que muestra una captura de pantalla.

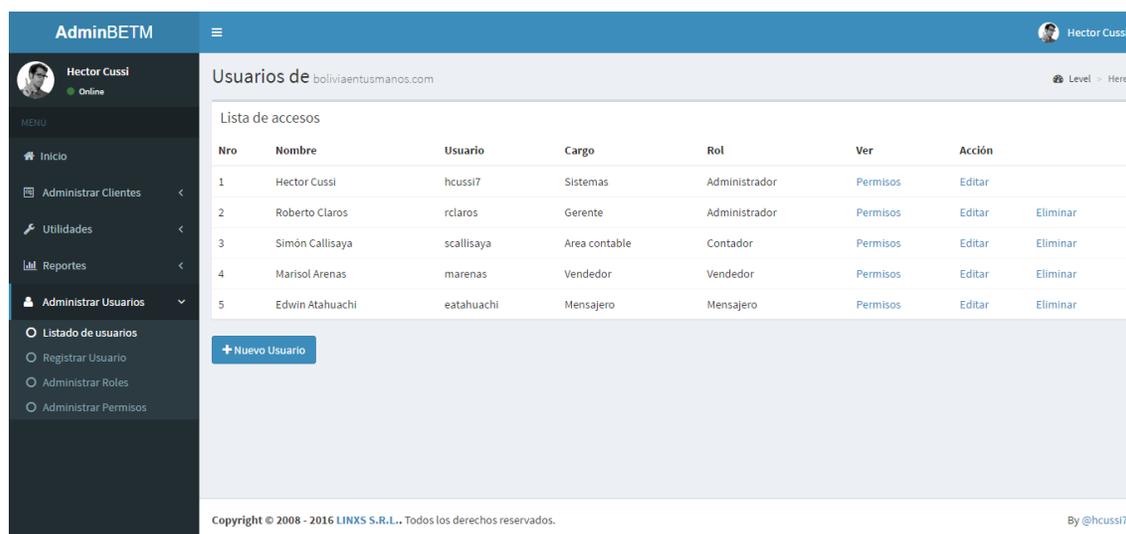


Figura 3.11: Captura de pantalla del prototipo de Administración de usuarios.

Fuente: Elaboración propia

3.5. MÓDULO DE GESTIÓN DE CLIENTES Y SERVICIOS

3.5.1. FASE DE ANÁLISIS

3.5.1.1. ANÁLISIS DE REQUERIMIENTOS

A continuación se detalla la lista de los actores involucrados para esta sección.

1. El gerente y el encargado de área contable son las únicas personas que pueden llenar datos de clientes y sus servicios según el formulario de ventas que le entreguen los vendedores.
2. Los vendedores y el mensajero pueden acceder al detalle de datos de cada cliente, según su rol asignado no pueden crear, editar o eliminar datos.
3. El llenado del formulario es según los servicios que el cliente solicita, puede ser entre los servicios betm, páginas web, sponsor y publicidad.

3.5.1.2. CASOS DE USO

La figura 3.12 muestra el caso de uso de Gestión de clientes, con los actores involucrados gerente, contador, vendedor y mensajero. En el detalle de cada cliente se puede observar los servicios y los pagos realizados.

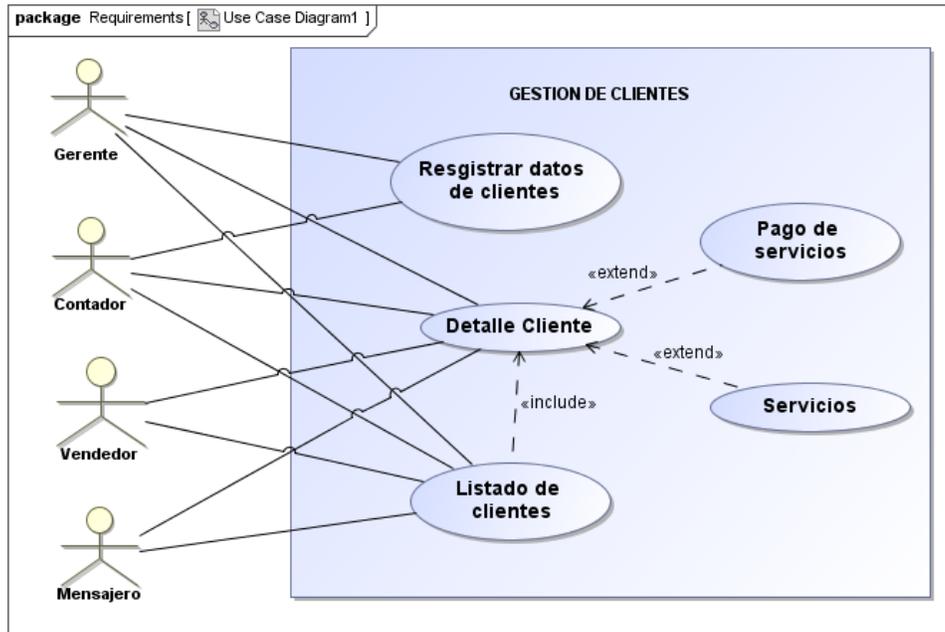


Figura 3.12: Caso de uso de gestión de clientes.

Fuente: Elaboración propia.

La tabla 3.4 detalla las especificaciones del caso de uso gestión de clientes:

Tabla 3.4: Especificación del caso de uso gestión de clientes

Caso de uso: Gestión de clientes
Actores: Gerente, contador, vendedores y mensajero
Descripción: Permite a los actores gerente y contador crear un nuevo cliente, ver listado y ver detalle de cada cliente. A los actores vendedor y mensajero solo les permite acceder al listado y ver detalle de cada cliente.
Precondiciones: Los actores deben ser usuarios registrados y con un rol asignado.
Flujo Normal: 1. Los actores ingresan a la opción de clientes.

<ol style="list-style-type: none"> 2. El sistema despliega el listado de clientes registrados, cada uno con opciones habilitadas según el rol de cada usuario. 3. El actor escoge una de las opciones.
<p>Postcondiciones: El sistema ejecuta la opción elegida por el actor y muestra el listado o el detalle de clientes.</p>

La figura 3.13 muestra el caso de uso para realizar operaciones de un cliente.

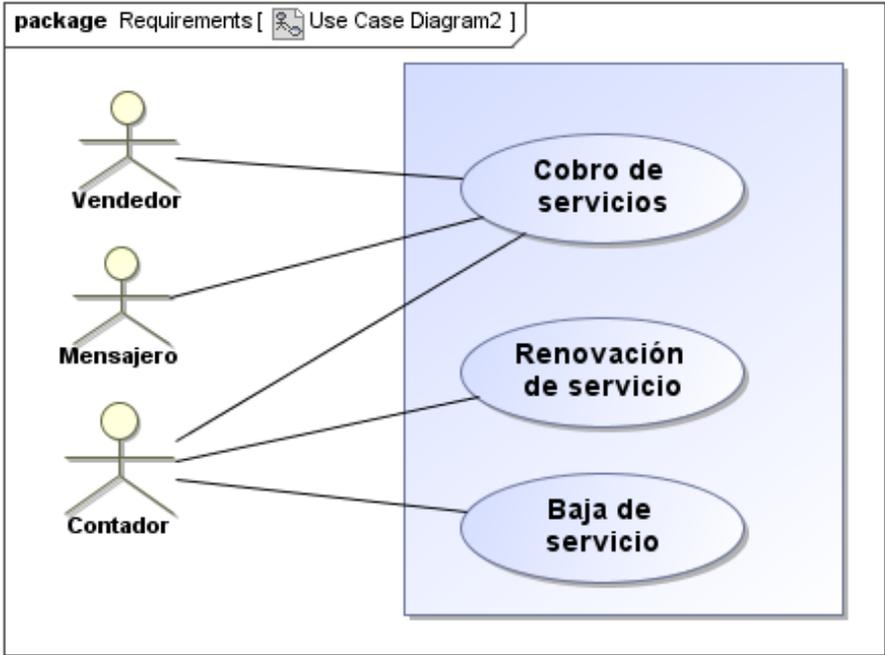


Figura 3.13: Caso de uso para operaciones de un cliente
Fuente: Elaboración propia

La tabla 3.5 detalla las especificaciones del caso de uso operaciones de un cliente:

Tabla 3.5: Especificación del caso de uso operaciones de un cliente.

<p>Caso de uso: Operaciones de un cliente</p>
<p>Actores: Contador, vendedores y mensajero</p>
<p>Descripción: Permite a los actores realizar operaciones que los clientes requieren como renovación, cobro de servicios y baja de servicios.</p>
<p>Precondiciones: Los actores deben ser usuarios registrados y con un rol asignado.</p>

<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El contador escoge una de las opciones para renovar o dar de baja. 2. El sistema despliega una ventana emergente con las acciones que solicitó 3. El contador llena los datos.
<p>Postcondiciones:</p> <p>El sistema ejecuta la opción elegida por el actor y muestra el listado de clientes.</p>

3.5.1.3. MODELO DE CONTENIDOS

La figura 3.14 muestra el diagrama de contenido para el módulo de gestión de clientes y servicios.

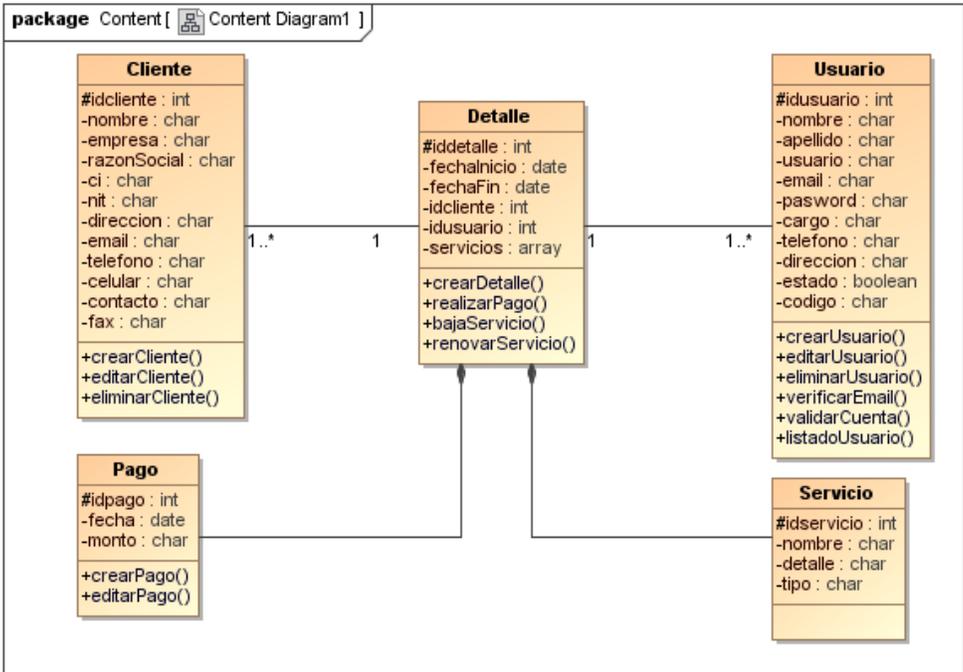


Figura 3.14: Diagrama de contenido para el módulo de gestión de clientes y servicios.
Fuente: Elaboración propia

3.5.2. FASE DE DISEÑO

3.5.2.1. MODELO NAVEGACIONAL

La Figura 3.15 muestra el comportamiento del usuario en el momento de ingresar a la página de gestión de clientes y servicios.

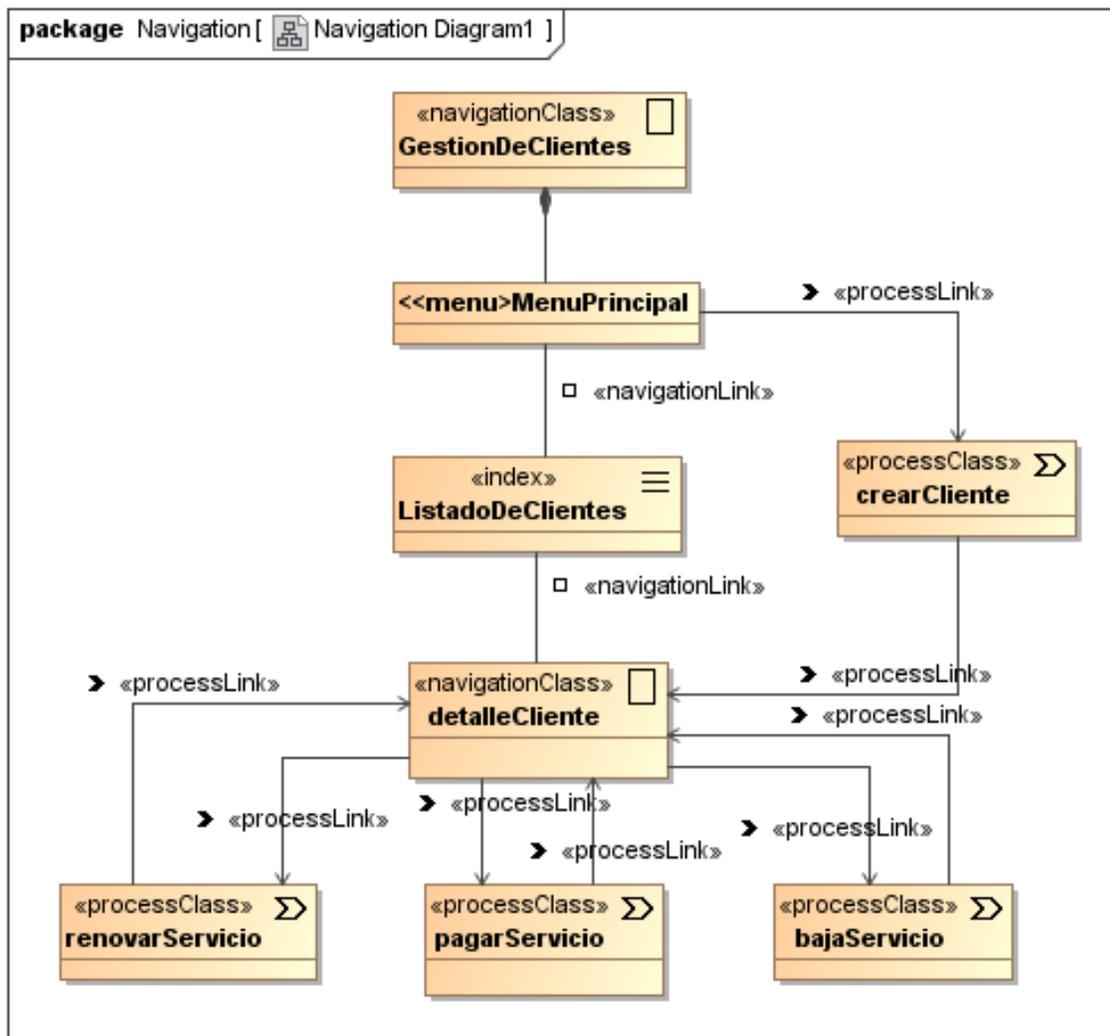


Figura 3.15: Diagrama de navegación para la gestión de clientes y servicios.
Fuente: Elaboración propia

3.5.2.2. MODELO DE PRESENTACIÓN

La figura 3.16 muestra el diagrama de presentación para el módulo de gestión de clientes y servicios, que representa la interfaz de usuario. Muestra un listado de nombres de clientes, nombre de la empresa a la que pertenece y los servicios que se le brinda, en cada servicio que tiene estará activado con un check, para ver las fechas de vencimiento de las empresas estarán con colores distintos.

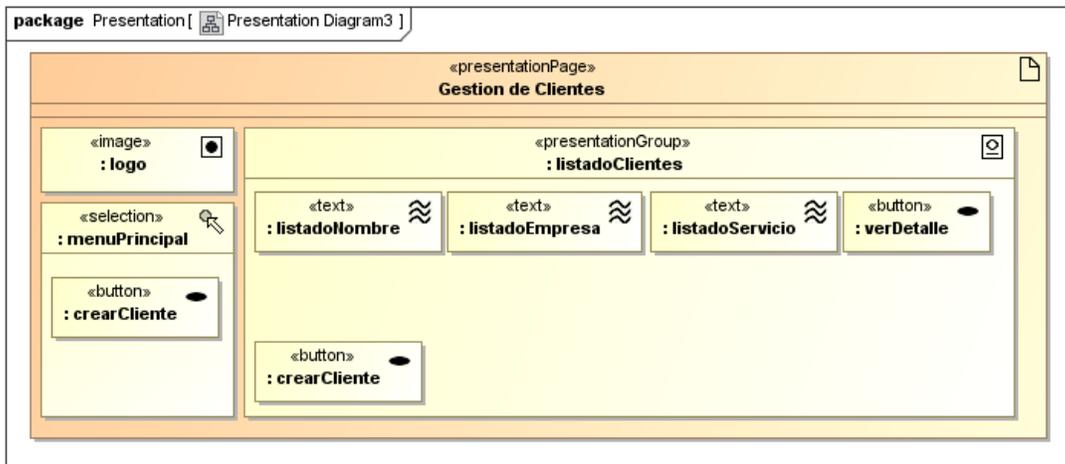


Figura 3.16: Diagrama de presentación para el módulo de gestión de clientes y servicios.
Fuente: Elaboración propia

La figura 3.17 muestra la interfaz de usuario del detalle de clientes.

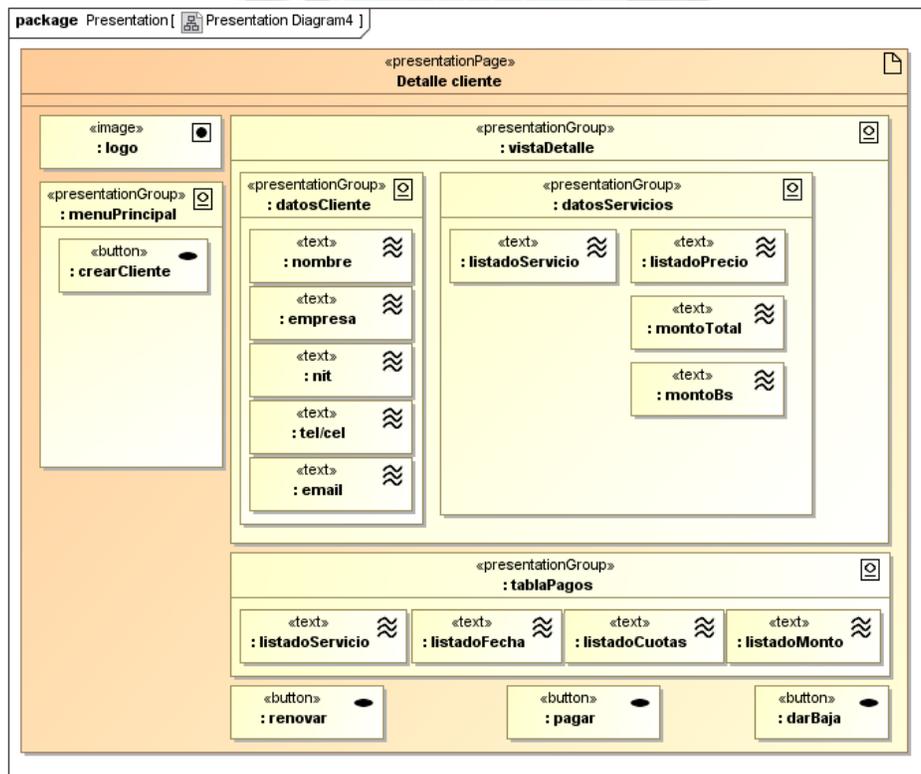


Figura 3.17: Diagrama de presentación para detalle de cliente.
Fuente: Elaboración propia

3.5.3. FASE DE DESARROLLO

3.5.3.1. MODELO DE FLUJO DE PROCESO

La figura 3.18 muestra el modelo de flujo de proceso del módulo de gestión de clientes y servicios, que nos muestra el comportamiento del usuario.

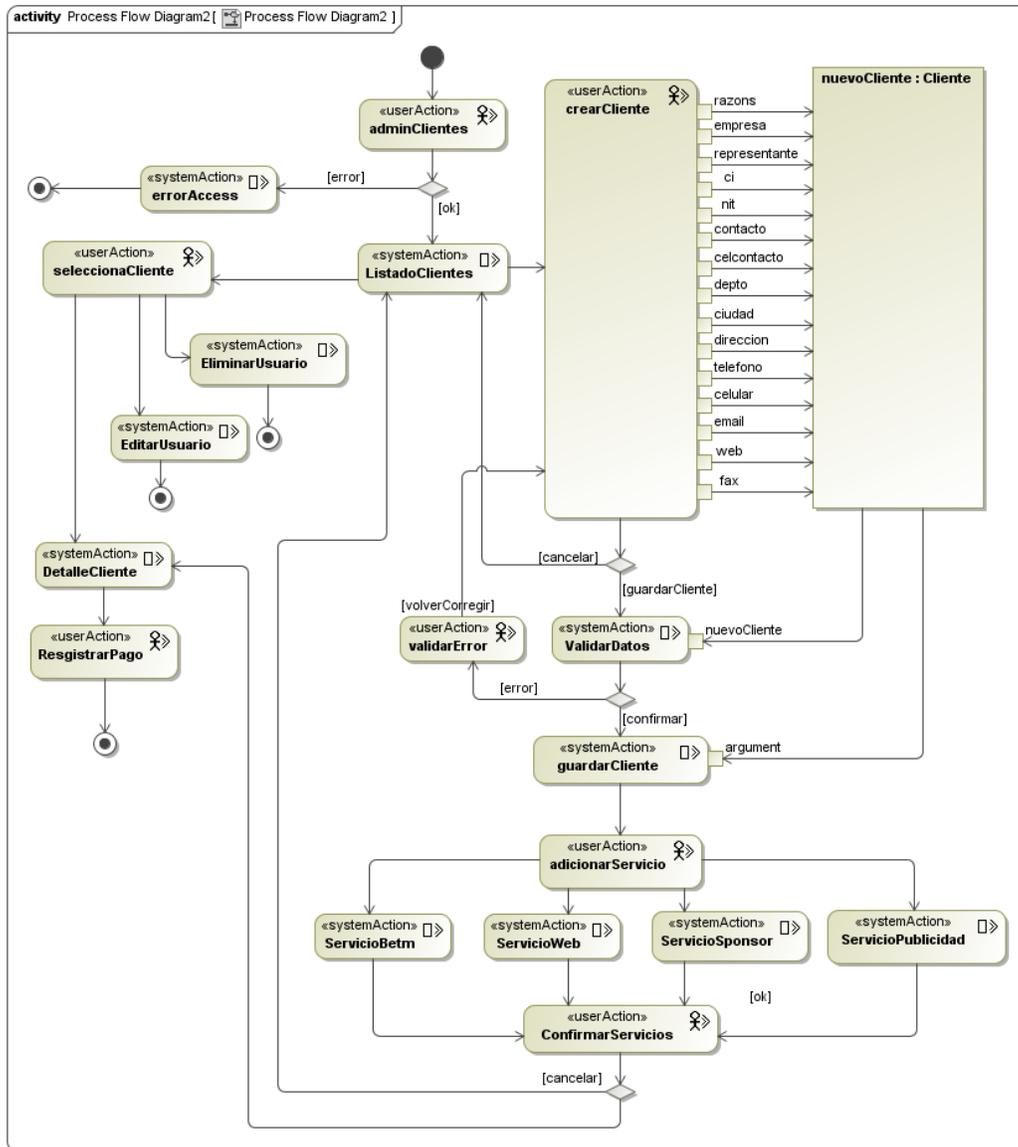
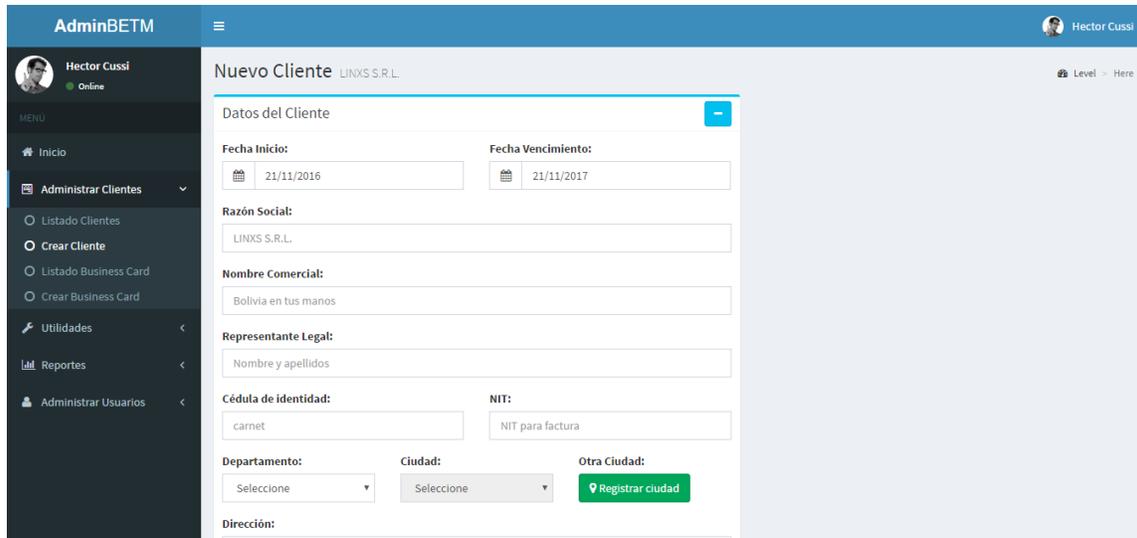


Figura 3.18: Modelo de flujo de procesos del módulo de gestión de clientes y servicios.
Fuente: Elaboración propia

3.5.4. FASE DE IMPLEMENTACIÓN

La figura 3.19 muestra la captura de pantalla del prototipo para la gestión de clientes.

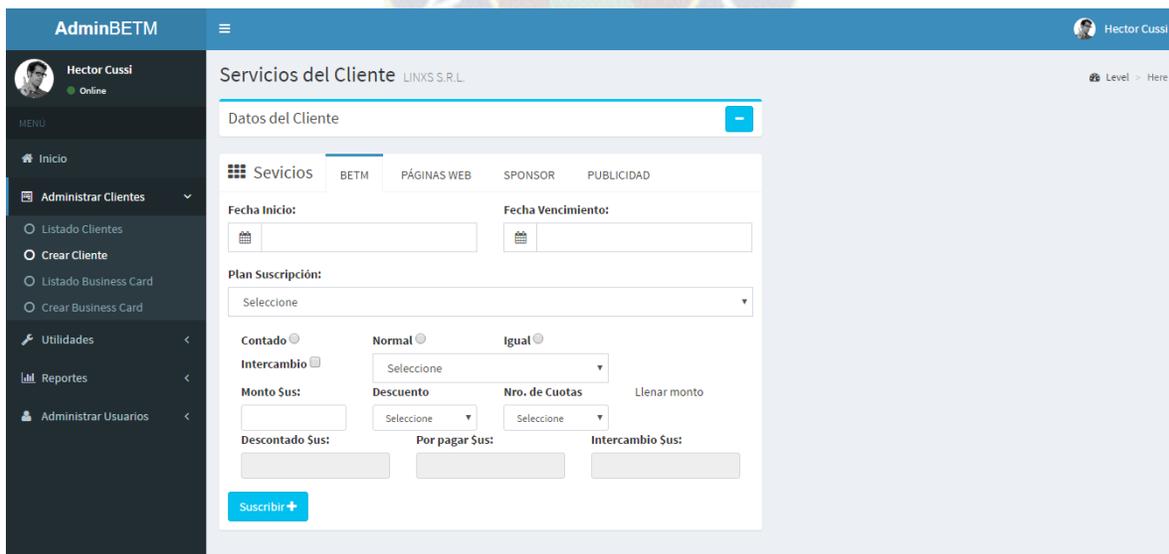


The screenshot displays the 'AdminBETM' interface. On the left is a dark sidebar menu with options like 'Inicio', 'Administrar Clientes', 'Listado Clientes', 'Crear Cliente', 'Listado Business Card', 'Crear Business Card', 'Utilidades', 'Reportes', and 'Administrar Usuarios'. The main content area is titled 'Nuevo Cliente LINXS S.R.L.' and contains a form with the following fields: 'Fecha Inicio' (21/11/2016), 'Fecha Vencimiento' (21/11/2017), 'Razón Social' (LINXS S.R.L.), 'Nombre Comercial' (Bolivia en tus manos), 'Representante Legal' (Nombre y apellidos), 'Cédula de identidad' (carnet), 'NIT' (NIT para factura), 'Departamento' (dropdown), 'Ciudad' (dropdown), 'Otra Ciudad' (with a 'Registrar ciudad' button), and 'Dirección'.

Figura 3.19: Captura de pantalla del prototipo para la gestión de clientes.

Fuente: Elaboración propia

En la figura 3.20 muestra la captura de pantalla del prototipo para la gestión de servicios de cada cliente.



The screenshot displays the 'AdminBETM' interface for 'Servicios del Cliente LINXS S.R.L.'. The sidebar menu is the same as in Figure 3.19. The main content area has a title 'Servicios del Cliente LINXS S.R.L.' and a sub-section 'Datos del Cliente'. Below this is a tabbed interface with 'Servicios' selected, and other tabs for 'BETM', 'PÁGINAS WEB', 'SPONSOR', and 'PUBLICIDAD'. The form includes: 'Fecha Inicio' and 'Fecha Vencimiento' (empty), 'Plan Suscripción' (dropdown), 'Contado' (radio), 'Normal' (radio), and 'Igual' (radio), 'Intercambio' (dropdown), 'Monto \$us' (input), 'Descuento' (dropdown), 'Nro. de Cuotas' (dropdown), 'Llenar monto' (checkbox), 'Descontado \$us' (input), 'Por pagar \$us' (input), and 'Intercambio \$us' (input). A blue 'Suscribir +' button is at the bottom.

Figura 3.20: Captura de pantalla del prototipo para la gestión de servicios.

Fuente: Elaboración propia

3.6. MÓDULO DE REPORTE DE SERVICIOS

3.6.1. FASE DE ANÁLISIS

3.6.1.1. CASOS DE USO

La figura 3.21 muestra el caso de uso reportes de servicios, los únicos actores involucrados son el gerente y contador ya que solo a ellos les importa la información para la toma de decisiones con relación a las ventas.

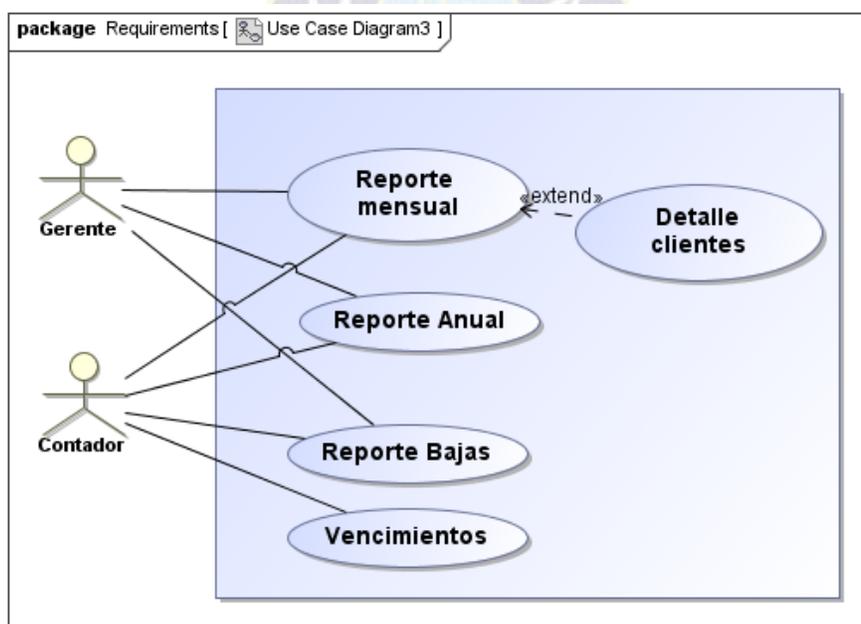


Figura 3.21: Caso de uso reportes de servicios.

Fuente: Elaboración propia

La tabla 3.6 detalla las especificaciones del caso de uso reportes de servicios:

Tabla 3.6: Especificación del caso de uso reportes de servicios.

Caso de uso: Reportes de servicios
Actores: Gerente y Contador.
Descripción: Permite a los actores realizar operaciones que le permita ver reportes en gráficos.

<p>Precondiciones: Los actores deben ser usuarios registrados y con un rol asignado de administrador de sistema y contador.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El contador escoge una de las opciones para poder observar reportes. 2. El sistema despliega los gráficos solicitados por el usuario.
<p>Postcondiciones: El sistema genera el grafico solicitado por el actor.</p>

3.6.1.2. MODELO DE CONTENIDOS

La figura 3.22 muestra el diagrama de contenidos del módulo de reportes.

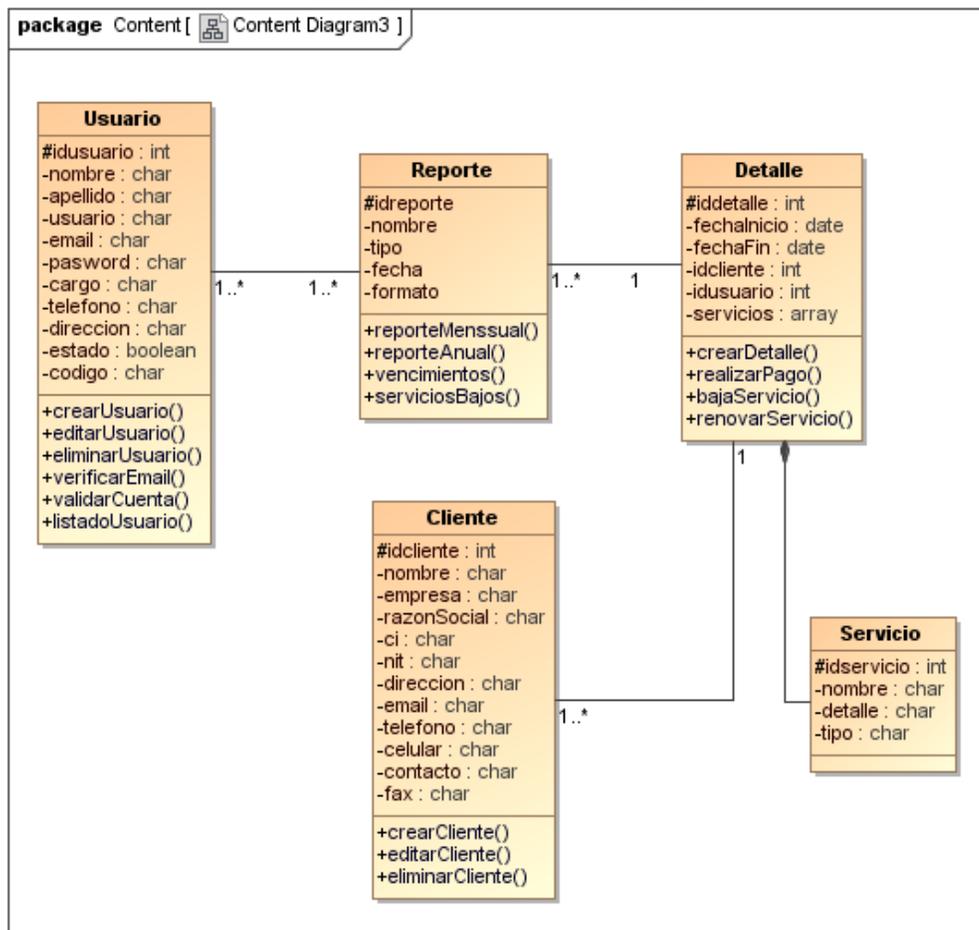


Figura 3.22: Diagrama de contenidos del módulo de reportes.

Fuente: Elaboración propia

3.6.2. FASE DE DISEÑO

3.6.2.1. MODELO NAVEGACIONAL

La figura 3.23 muestra el diagrama de navegación del módulo de reportes, de esa forma navega el usuario, para obtener los reportes primero deberá elegir a un cliente del listado y luego elegir una opción.

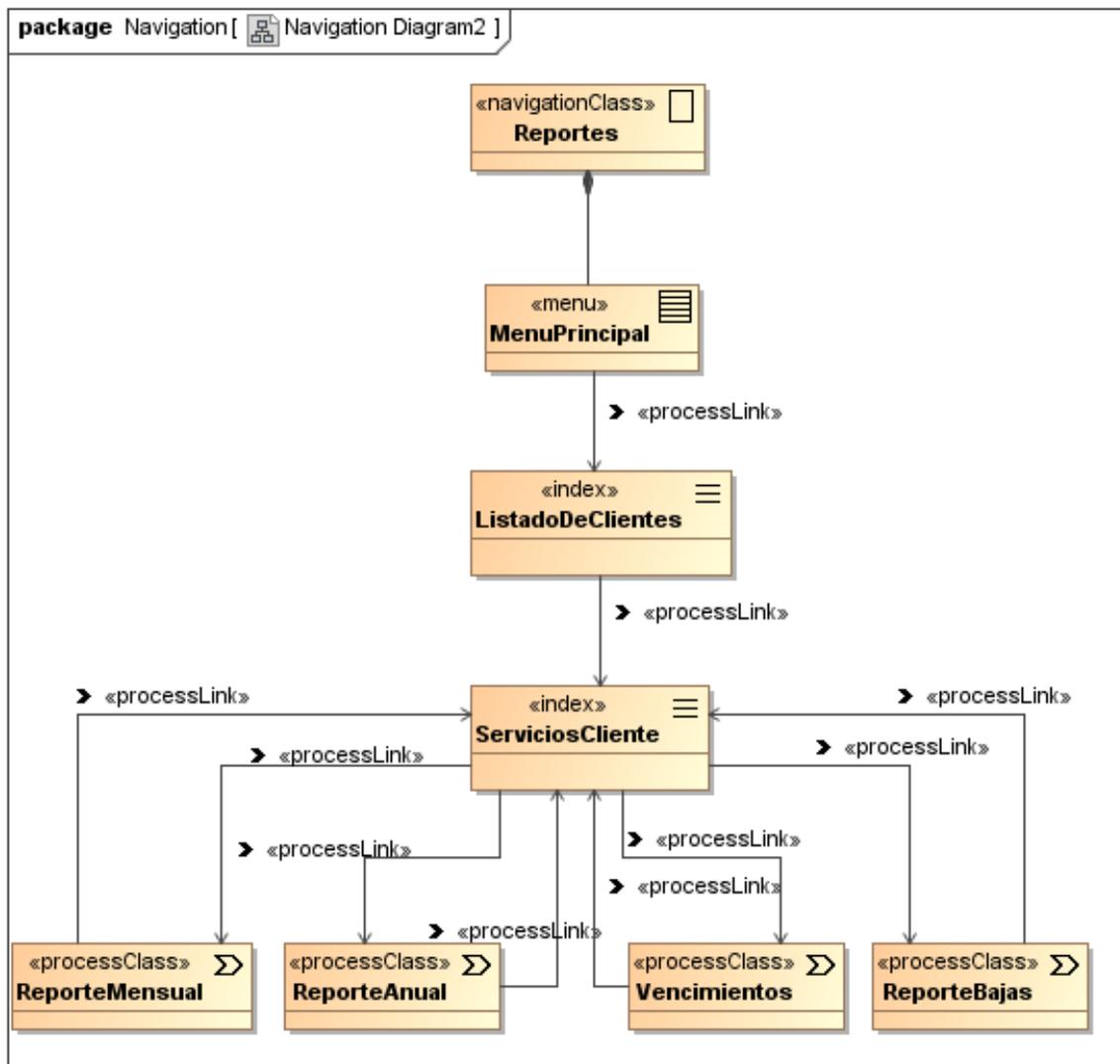


Figura 3.23: Diagrama de navegación del módulo de reportes.
Fuente: Elaboración propia

3.6.2.2. MODELO DE PRESENTACIÓN

La figura 3.24 muestra el diagrama de presentación del módulo de reportes.

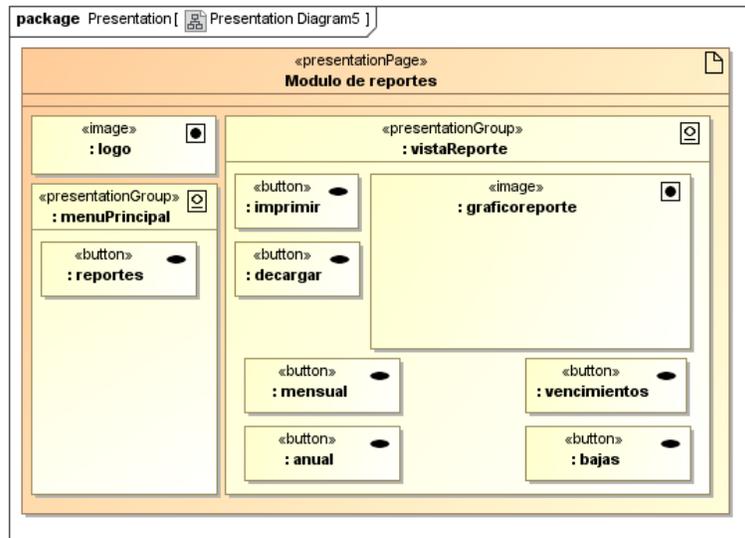


Figura 3.24: Diagrama de presentación del módulo de reportes.

Fuente: Elaboración propia

3.6.3. FASE DE DESARROLLO

3.6.3.1. MODELO DE FLUJO DE PROCESO

En la figura 3.25 se muestra el modelo de flujo de proceso para el módulo de reportes.

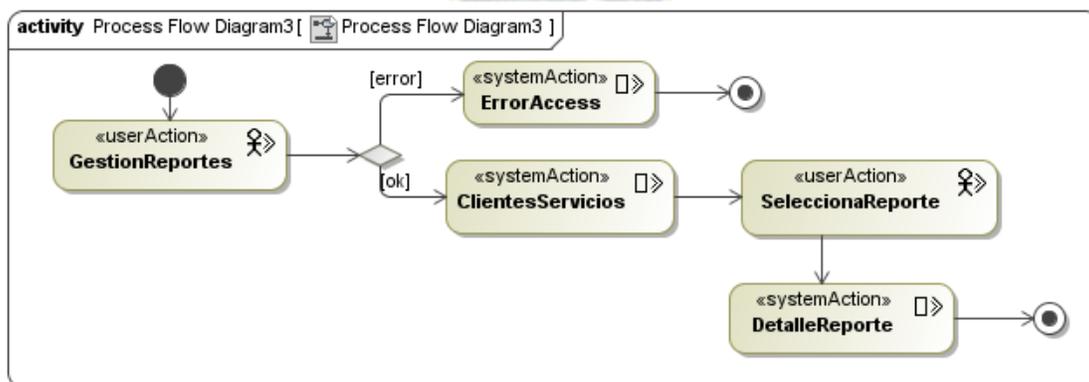
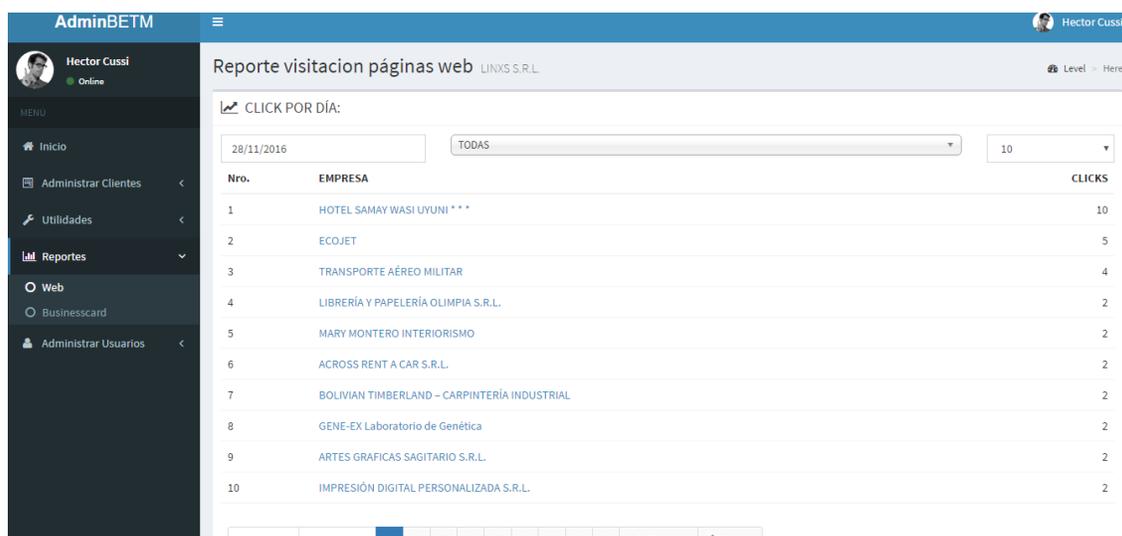


Figura 3.25: Diseño de modelo de flujo de procesos para el módulo de reportes.

Fuente: Elaboración propia

3.6.4. FASE DE IMPLEMENTACIÓN

Para esta parte se desarrolló el prototipo de la página de reportes como se puede observar la figura 3.26 que muestra una captura de pantalla.



Reporte visitación páginas web LINXS S.R.L.		
CLICK POR DÍA:		
28/11/2016	TODAS	10
Nro.	EMPRESA	CLICKS
1	HOTEL SAMAY WASI UYUNI * * *	10
2	ECOJET	5
3	TRANSPORTE AÉREO MILITAR	4
4	LIBRERÍA Y PAPELERÍA OLIMPIA S.R.L.	2
5	MARY MONTERO INTERIORISMO	2
6	ACROSS RENT A CAR S.R.L.	2
7	BOLIVIAN TIMBERLAND - CARPINTERÍA INDUSTRIAL	2
8	GENE-EX Laboratorio de Genética	2
9	ARTES GRAFICAS SAGITARIO S.R.L.	2
10	IMPRESIÓN DIGITAL PERSONALIZADA S.R.L.	2

Figura 3.26: Captura de pantalla del prototipo de Administración de usuarios.

Fuente: Elaboración propia

3.7. FASE DE PRODUCCIÓN

Durante esta última etapa se realizó la prueba de software del producto cumpliendo las tareas trazadas de cada fase.

3.7.1. PRUEBA DE ESFUERZO

Teniendo el sistema web en modo de producción y ya subido en un servidor de la empresa con la siguiente dirección web: <http://boliviaentusmanos.net/adminbetm/>. Se realizó la prueba de esfuerzo o prueba de estrés con la ayuda de la herramienta de software Apache JMETER v3.1.

Como se puede observar en la figura 3.27 se muestra el resultado de la prueba con 50 usuarios ingresados al mismo tiempo en la página de inicio, la página de administración de usuarios y roles, la página de administración de clientes y servicios, la página de reportes, haciendo un total de 200 usuarios navegando en el sistema sin ningún problema, obteniendo un error

del 0% en todas las páginas elegidas para la prueba.

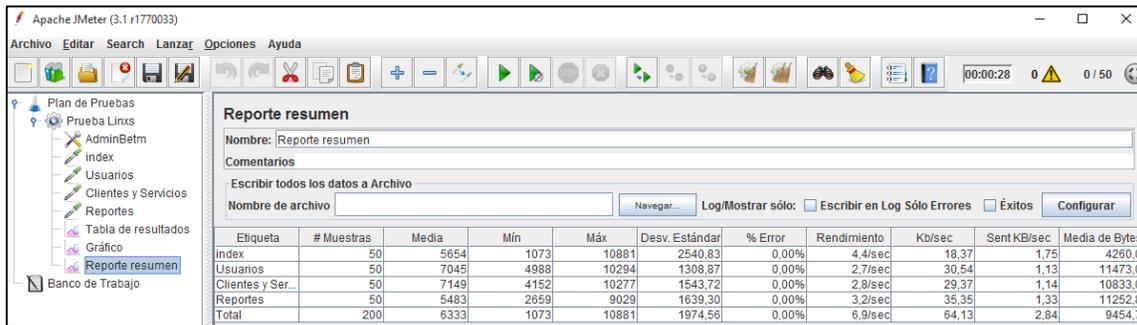


Figura 3.27: Resultado de la prueba con 50 usuarios en cada página.

Fuente: Elaboración propia

Por otra parte, si aumentamos el número de usuarios a 100 en cada página, como muestra la figura 3.28, veremos como el sistema web colapsa ya que están en ese momento existen 400 usuarios navegando en ese instante, obteniendo un margen de error entre 30% y 90%, lo que nos lleva a la conclusión que el sistema soporta hasta 50 usuarios por página, si aumenta un usuario más el sistema colapsa, en algunas páginas tardará en responder, en otras no se obtendrá ninguna respuesta.

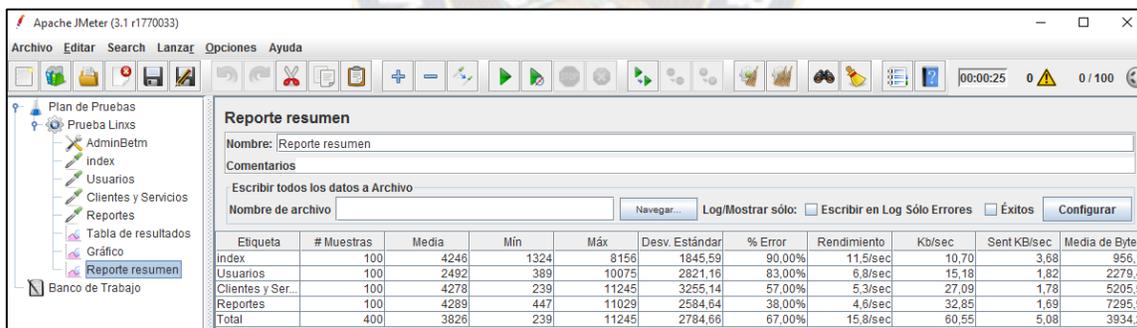


Figura 3.28: Resultado de la prueba con 100 usuarios en cada página.

Fuente: Elaboración propia

La finalidad de estas pruebas se realiza para conocer la cantidad de usuarios que puede soportar el sistema y el servidor web de la empresa.

CAPITULO 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, 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 siguientes criterios.

4.3.1. 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:

- **Número de entradas de usuario:** Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación. La tabla 4.1 muestra la lista de entradas de usuario que tiene el sistema.

Tabla 4.1: Lista de entradas de usuario del sistema

Nro.	Entradas de Usuario
1	Ingreso al sistema
2	Registro de usuarios
3	Registro de roles
4	Registro de permisos
5	Registro de clientes
6	Registro de servicios de clientes
7	Registro de servicio BETM

- **Número de salidas de usuario:** Se refiere a información elaborada por el sistema para ser mostrada al usuario. En la tabla 4.2 se muestra las salidas de usuarios que tiene el sistema.

Tabla 4.2: Lista de salidas de usuario del sistema

Nro.	Salidas de Usuario
1	Listado de clientes
2	Listado de servicios de clientes
3	Listado de roles y permisos
4	Listado de empresas

- **Número de peticiones de usuarios:** Son entradas interactivas entre el usuario y el sistema, donde la salida es inmediata. La tabla 4.3 muestra las peticiones de usuario al sistema.

Tabla 4.3: Lista de peticiones del usuario al sistema

Nro.	Peticiones de Usuario
1	Autenticación de usuario
2	Listar Clientes

3	Listar Servicio de Clientes
4	Mostrar detalle de servicio
5	Mostrar grafico de reportes
6	Listar vencimientos de servicios
7	Listar Roles y permisos

- **Número de archivos:** Se cuenta cada archivo maestro lógico. En otras palabras las tablas existentes en la base de datos. La tabla 4.4 muestra los archivos lógicos del sistema.

Tabla 4.4: Lista de archivos lógicos del sistema

Nro.	Archivos Lógicos
1	Usuario
2	Roles
3	Permisos
4	Cliente
5	Servicios
6	Detalle
7	Directorio
8	Pago

- **Número de interfaces externas:** Se cuenta todas las interfaces legibles por el ordenador que son utilizados para transmitir la información. En este caso es solo internet.

La tabla 4.5 muestra los factores de ponderación con los la lista obtenidas anteriormente, para el caso se utilizó el factor medio.

Tabla 4.5: Factores de ponderación

Parámetros de medición	Cuenta	Factores de ponderación			Valor Obtenido
		Simple	Media	Complejo	
Nro. de entradas de usuario	7	-	4	-	28
Nro. de salidas de usuario	4	-	5	-	20
Nro. de peticiones de usuario	7	-	4	-	28
Nro. de archivos	8	-	6	-	48
Nro. de interfaces externas	1	-	2	-	2
Cuenta total					126

La tabla 4.6 muestra el factor de ajuste de complejidad en base a las respuestas de las siguientes preguntas evaluadas entre 0 y 5.

Tabla 4.6: Factor de ajuste de complejidad.

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 recuperación 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

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:

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

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

Reemplazando los valores obtenidos en las tablas 4.5 y 4.6 se obtiene el siguiente resultado:

$$PF = 126 * [0.65 + (0.01 * 58)]$$

$$PF = 154.98$$

A continuación calculamos el PF ideal:

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

$$PF_{ideal} = 170.1$$

Entonces la funcionalidad del sistema es:

$$\text{Funcionalidad} = \frac{PF}{PF_{ideal}} * 100$$

$$\text{Funcionalidad} = \frac{154.98}{170.1} * 100$$

$$\text{Funcionalidad} = 91.1\%$$

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.

4.3.2. 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:

$$TMEF = (20*8) + 1$$

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

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

$$Disponibilidad = \frac{TMDF}{TMEF} * 100$$

$$Disponibilidad = \frac{160}{161} * 100 = 99,4\%$$

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

4.3.3. 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 4.7 muestra los resultados obtenidos en base a preguntas propuestas a los usuarios del sistema.

Tabla 4.7: Preguntas para obtener el grado de usabilidad

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

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

4.3.4. 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 Marin, 2005).

La tabla 4.8 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.

Tabla 4.8: Evaluación de mantenibilidad

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%

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

4.3.5. 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:

$$GP = 1 - \frac{ET}{ER}$$

Donde:

GP: Grado de portabilidad

ET: Recursos necesarios para llevar el sistema a otro entorno.

ER: Recursos necesarios para crear el sistema en el entorno residente.

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, smarty, css3, api google, json y composer).

Por lo que los valores obtenidos son: $ET = 9$ y $ER = 10$

$$GP = 1 - \frac{9}{10}$$

$$GP = 0.90$$

Con el resultado obtenido sabemos que el **grado de portabilidad es del 90%**, entonces la

portabilidad del sistema es más rentable que su re-desarrollo.

4.3.6. 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).

La tabla 4.9 muestra la calidad global del sistema expresado en porcentajes.

Tabla 4.9: Calidad global

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

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.

4.4. SEGURIDAD

Los problemas de seguridad en sistemas web, pueden venir de las herramientas que se utilizan en el momento del desarrollo o producto de un diseño lógico que no se contempló de las posibles amenazas que pueda surgir como ser (Meucci, 2008):

- Entradas no validas
- Control de accesos rotos
- Sesiones y Autenticaciones no controladas
- Ataques Cross Site Scripting
- Inyección de códigos
- Manejo Inadecuado de Errores

Para las medidas de seguridad para el sistema desarrollado se contemplan dos aspectos

importantes y vulnerables que están en el lado del cliente y lado del servidor.

4.4.1. SEGURIDAD DEL LADO DEL CLIENTE

Uno de los mecanismos de seguridad que se implementan son las validaciones por el lado del cliente. Existen mecanismos de validación provistos por las herramientas que utilizamos para hacer la aplicación, HTML cuenta con atributos para validar datos requeridos, numéricos, formato de correos, etc. estas validaciones son realizadas antes de que la información introducida llegue al servidor, esto evita que se envíen datos incorrectos al servidor, además se ahorra tiempo, ya que si la información es incorrecta simplemente no se envía al servidor (Meucci, 2008).

Las medidas que se implementó en el lado del servidor del sistema es la autenticación de usuarios, los únicos que tienen acceso al sistema son el personal de la empresa, estos están registrados como usuarios y con su contraseña respectiva.

4.4.2. SEGURIDAD DEL LADO DEL SERVIDOR

El desarrollo de una aplicación web requiere de una serie de herramientas: servidores web, servidores de base de datos, servidores de aplicaciones, lenguajes de programación del lado del servidor, etc. Las vulnerabilidades mencionadas que pueden comprometer la seguridad de un sistema web son las siguientes (Meucci, 2008):

- Versiones no actualizadas
- Configuraciones por defecto inadecuadas
- Cuentas por defecto no modificadas

Las medidas tomadas para el lado del servidor del sistema son las siguientes:

4.4.2.1. ENCRIPCIÓN DE CONTRASEÑAS

Es recomendable que las contraseñas de los usuarios estén encriptados, de esta forma se tiene asegurado que las acciones solamente sean únicamente responsables de los usuarios correctamente identificados. Para el sistema desarrollado se utilizó una función de PHP para

encriptar cadenas de texto que hace uso de algoritmo de encriptación hash_init que inicializa un contexto incremental para cifrar.

La figura 4.1 muestra el código fuente de la clase Hash que utiliza hash_init para encriptar las contraseñas.

```
1 <?php
2
3 class Hash{
4     public static function getHash($algoritmo, $data, $key){
5         $hash = hash_init($algoritmo, HASH_HMAC, $key);
6         hash_update( $hash, $data);
7         return hash_final($hash);
8     }
9 }
```

Figura 4.1: Código fuente de la clase Hash para encriptar contraseñas.

Fuente: Elaboración propia

4.4.2.2. CONTROL DE ACCESOS Y PERMISOS

Se desarrolló la clase Acl (ver Anexo F) que controla los accesos a ciertas partes del sistema y permisos de acuerdo al rol asignado a cada usuario. Además los controles de accesos son configurables solamente por el usuario con rol de administrador del sistema.

Además, la clase Acl controla el tiempo de sesión del usuario, en caso de inactividad por el periodo de 20 minutos se cierran todas las sesiones, y evita que se ejecuten peticiones, pidiendo al usuario que vuelva a iniciar su sesión.

4.4.2.3. FILTROS CONTRA ATAQUES XSS – CROSS SITE SCRIPTING

El uso del framework CodeIgniter 3 para el desarrollo del sistema, facilita esta tarea ya que las funciones de obtención de los datos de los formularios vía GET y POST cuentan con filtros contra ataques XSS, evitando que se ejecuten código javascript y consultas SQL introducidas por el usuario de forma fortuitas o intencionales.

5.1. INTRODUCCIÓN

En el desarrollo de este capítulo, se dará a conocer a la empresa LINXS S.R.L. que en la implementación y utilización del sistema se obtendrán muchos beneficios.

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 y la tasa interna de retorno.

Después de realizar los cálculos 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: básico, intermedio y detallado.

En su libro clásico acerca de economía de la ingeniería de software; Barry Boehm introdujo una jerarquía de modelos de estimación de software que llevan el nombre de COCOMO. El modelo original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evoluciono hacia un modelo de estimación más exhaustivo llamado COCOMO II (Pressman, 2010).

COCOMO II consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones:

$$E = a(KLDC)^b ; \text{Persona} - \text{mes}$$

$$D = c(E)^d ; \text{Meses}$$

$$P = \frac{E}{D} ; \text{Personas}$$

Dónde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

a, b, c y d: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código distribuidas en miles

A la vez cada modelo se subdivide en tres modos:

- **Modo orgánico:** Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar.
- **Modo semilibre:** Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo rígido:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas.

La tabla 5.1 muestra las constantes de acuerdo a los modos descritos anteriormente.

Tabla 5.1: Constantes a,b,c,d COCOMO II

Modo	a	b	c	d
Orgánico	2.4	1.05	2.5	0.38
Semilibre	3.0	1.12	2.5	0.35
Rígido	3.6	1.20	2.5	0.32

5.3. COSTO DEL SISTEMA

El costo del sistema se lo planteará en dos partes:

5.3.1. COSTO DE DESARROLLO DEL SISTEMA

Para el cálculo del desarrollo del software se tendrá como partida el punto función no ajustada que se encontró en el capítulo anterior, cuyo valor encontrado es:

$$PF = 154.98$$

La tabla 5.2 muestra la relación para convertir el valor de PF a KLDC (Kilos de líneas de código).

Tabla 5.2: Conversión de PF a KLDC

Lenguaje	Nivel	Factor LDC / PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
PL / I	4	80
Ansi Cobol 74	3	107
Visual Basic	7.00	46
ASP	9.00	36
PHP	11.00	29
Visual C++	9.50	34

Entonces, realizando los cálculos y escogiendo el valor del lenguaje de programación PHP de la tabla 5.2, tenemos:

$$LDC = PF * Factor\ LDC/PF$$

$$LDC = 154.98 * 29$$

$$LDC = 4494,42$$

$$KLDC = \frac{4494,42}{1000}$$

$$KLDC = 4.49$$

Ahora, para hallar el esfuerzo reemplazamos los valores obtenidos hasta ahora:

$$E = 2.4 * (4.49)^{1.05} = 11.62$$

$$D = 2.5 * (11.62)^{0.38} = 6,35$$

Para el cálculo del número de programadores para el desarrollo:

$$P = \frac{11.62}{6.35} = 1,83 \cong 2$$

Estimando que el salario medio de un programador es de Bs. 2500, esta cifra será tomada en cuenta para la siguiente estimación:

Costo del software desarrollado = Numero de programadores * Salario de un programador (Pressman, 2002).

Costo del Software Desarrollado por Persona = 2 * 2.500 = 5.000 Bs.

Costo total del Software Desarrollado = 5.000 * 7 = 35.000 Bs.

Lo que significa que el costo del sistema desarrollado es de Bs. 35.000 por los 7 meses.

5.3.2. COSTO DE ELABORACIÓN DEL PROYECTO

La tabla 5.3 muestra los costos de inversión de los recursos que se usaron para la elaboración del sistema.

Tabla 5.3: Costos de recursos empleados para la elaboración del sistema.

Recursos	Costo (Bs)
Material de escritorio	250
Investigación del proyecto	300
Internet	350
Otros	250
Total	1150

Por tanto el costo de la elaboración del proyecto es de Bs. 1150.

5.3.3. COSTO TOTAL DEL SISTEMA

El costo total del sistema se obtiene de la sumatoria del costo de desarrollo y el costo de elaboración del proyecto, en la tabla 5.4 se puede observar los resultados, todos los costos están expresados en bolivianos.

Tabla 5.4: Costo total del sistema.

Detalle	Costo (Bs)
Costo de desarrollo	35.000
Costo de elaboración del proyecto	1.150
Costo Total	36.150

Entonces, el costo total del sistema es de Bs. 36150.

5.4. VALOR ACTUAL NETO

El valor actual neto (VAN), es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto (Bu, 2009). La fórmula que utilizaremos para hallar el valor actual neto será:

$$VAN = \sum \frac{ganancias}{(1+k)^n} - \sum \frac{costos}{(1+k)^n}$$

Dónde:

VAN: Valor actual neto.

ganancias: ingreso de flujo anual.

costos: salidas de flujo anual.

n: número de periodo.

k: tasa de descuento o tasa de interés de préstamo.

Además, tenemos que, si:

- VAN > 0, el proyecto es rentable
- VAN = 0, no hay pérdidas ni ganancias
- VAN < 0, el proyecto no es rentable

En la tabla 5.5 mostramos los gastos y las ganancias que se estiman en un lapso de 5 años, tomando en cuenta la tasa de descuento del 10%.

Tabla 5.5: Estimación de gastos y ganancias en 5 años.

Año	Costos	Ganancias	costos/(1+k) ⁿ	ganancias/(1+k) ⁿ
1	36150	0	32863,64	0,00
2	20000	10000	16528,93	8264,46
3	10000	20000	7513,15	15026,30
4	5000	30000	3415,07	20490,40
5	0	36150	0,00	22446,31
TOTAL			60320,78	66227,47

Entonces, reemplazando en la ecuación del VAN, tenemos:

$$VAN = \sum \frac{ganancias}{(1 + 0,10)^n} - \sum \frac{costos}{(1 + 0,10)^n}$$

$$VAN = 5906,69$$

Con este resultado concluimos que nuestro proyecto es rentable ya que 5906,69 es mayor a cero.

5.5. COSTO BENEFICIO

Para encontrar la relación costo/beneficio de un proyecto se aplica la siguiente ecuación:

$$costo/beneficio = \frac{\sum ganacias}{\sum costos}$$

Entonces, reemplazando los valores obtenidos en la tabla 5.5 en la ecuación anterior:

$$costo/beneficio = \frac{66227,47}{60320,78}$$

$$costo/beneficio = 1,10$$

Entonces de aquí por cada boliviano invertido en el sistema, la empresa genera una ganancia de Bs. 0,10.

5.6. TASA INTERNA DE RETORNO

La tasa interna de retorno (TIR) es una tasa porcentual que indica la rentabilidad promedio anual, es decir es la máxima tasa de descuento que puede tener un proyecto para que sea rentable. La fórmula para el cálculo del TIR es:

$$VAN = 0$$

$$0 = -C_0 + \sum_{i=1}^n \frac{C_i}{(1+k)^i}$$

Donde de esta ecuación se tiene que despejar k y ese valor sería nuestra TIR:

Si:

- TIR > k, el proyecto es rentable
- TIR = k, no hay perdidas ni ganancias
- TIR < k, el proyecto no es rentable

Se debe aclarar que el valor k que se compara con TIR, es el k usado en la ecuación de VAN.

Entonces, para para hallar nuestra TIR:

$$0 = -36150 + \frac{0}{(1+k)^1} + \frac{10000}{(1+k)^2} + \frac{20000}{(1+k)^3} + \frac{30000}{(1+k)^4} + \frac{36150}{(1+k)^5}$$

$$TIR = 0,29$$

$$TIR = 29\%$$

Como $0,29 > 0,10$ concluimos que el proyecto es rentable.

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Una vez de haber terminado con el desarrollo y la implementación del sistema web para la empresa LINXS S.R.L., aplicando todas las metodologías de análisis y diseño de software en base a los objetivos propuestos, se puede afirmar que se han cumplido con todos los objetivos.

Entonces, se concluye que:

- Se realizó el control y seguimiento de información de clientes, que permite a gerencia general tener dicha información de manera actualizada y oportuna.
- Se implementó el control y permisos sobre la administración del sistema, de acuerdo al rol asignado a cada usuario.
- Se recolectó y procesó los datos de los clientes registrados en planillas excel, que permite al responsable de área contable contar con mayor información.
- Se redujo el costo en llamadas telefónicas por parte de los vendedores para obtener información de servicios de clientes.
- 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, donde los usuarios pueden obtener información semanal, mensual y anual de los servicios a los clientes, en algunos casos se puede saber reportes de clientes individualmente, también se puede imprimir el detalle de cada cliente.
- Se implementó la funcionalidad responsiva al sistema, se puede ver sin ningún problema en varios dispositivos móviles, es agradable a la vista del usuario.
- Se implementó la funcionalidad de control de servicios, lo que permite al encargado de área contable notificar a los clientes sobre el vencimiento de servicios.

6.2. RECOMENDACIONES

Las recomendaciones para mejorar el sistema web son las siguientes:

- Se recomienda utilizar codeigniter 3 para adaptar otras funcionalidades al sistema, ya que las versiones futuras como codeigniter 4 vendrán con otra estructura distinta a la actual.
- La empresa LINXS S.R.L. actualmente no cuenta con una planilla contable, se recomienda adaptarlo al sistema web.
- Se recomienda implementar y adaptar al sistema el módulo de recursos humanos para tener control del personal de la empresa.
- Se recomienda dar de baja a los usuarios que por algún motivo ya no pertenecen a la empresa.



BIBLIOGRAFÍA

- Aguilar, M. J. (2011). *Sistema Web de Control y Seguimiento de Personal Caso: Universidad Tecnológica Boliviana*. Universidad Mayor de San Andrés. Carrera de Informática.
- Bermejo, M. (2015). *El Kanban*. Universitat Oberta de Catalunya.
- Bu, C. (1995). *Análisis y Evaluación de Proyectos de Inversión*. Decima reimpresión. México. Editorial LIMUSA SA.
- Cabrera, A., Figueroa, R., y Solís, C. (2008). *Metodologías Tradicionales Vs. Metodologías Ágiles*. Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación.
- Chavez, M. E. (2014). *Proyecto de Grado: Sistema de Información Vía Web para el Seguimiento y Control de Tramites, Caso: Escuela Superior de Formación de Maestros Simon Bolívar*. 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*. Universidad Mayor de San Andrés. Carrera de Informática.
- Fernández, Y., & Díaz, Y. (2012). *Patrón Modelo-Vista-Controlador*. Obtenido de <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/15/10>
- Hanckes, H. (2012). *Tutorial 10 - Modelo Vista Controlador*. Pontificia Universidad Católica de Chile. Escuela de Ingeniería. Departamento de Ciencia de la Computación. Obtenido de <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/15/10>
- Kniberg, Henrik y Skarin, Mattias. (2010). *Kanban y Scrum – Obteniendo lo mejor de ambos*. Estados Unidos. C4Media Inc.
- Largo, C. & Marin, E. (2005). *Guía Técnica para la Evaluación del Software*. <http://www.puntoexe.com.co>

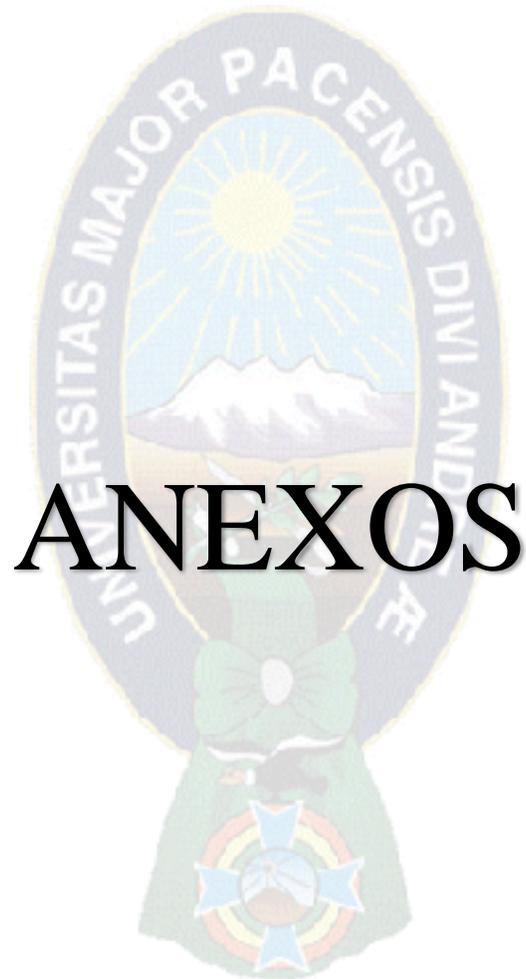
- Marquez, E. (2013). *Sistema de Información Vía Web de Seguimiento y Control a los Proyectos de Inversión en el Gobierno Autónomo Municipal de Palos Blancos*. Universidad Mayor de San Andrés. Carrera de Informática.
- Meucci, M. (2008). *Guía de pruebas OWASP*. OWASP Foundation. Obtenido de https://www.owasp.org/images/8/80/Gu%C3%ADa_de_pruebas_de_OWASP_ver_3.0.pdf
- Mouse, J. (2011). *MVC: Modelo, Vista y Controlador en PHP*. Obtenido de <http://www.jc-mouse.net/proyectos/mvc-modelo-vista-y-controlador-en-php>
- Nieves, C., Ucán J. & Menéndez V. (2014). *UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio*. Universidad Autónoma de Yucatán. Facultad de Matemáticas.
- 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.
- Palacios, J. (2016). *KANBAN (III): Reuniones y roles Kanban*. Obtenido de <https://jeronimopalacios.com/2016/09/kanban-iii-reuniones-roles-kanban/>
- Pérez, H. F. (2010). *Propuesta de Análisis y Diseño Basada en UML y UWE para la Migración de Arquitectura de Software Centralizada hacia Internet*. Guatemala. Universidad de San Carlos de Guatemala. Facultad de Ingeniería.
- Pilar, M. (2015). *Tutorial Patron MVC*. Obtenido de <http://www.codigonexo.com/wp-content/uploads/2014/06/Curso-completo-MVC.pdf>
- Pressman, R. (2002). *Ingeniería del software un enfoque práctico*. Quinta edición. México. McGraw-Hill Interamericana Editores S.A de C.V.

Pressman, R. (2010). *Ingeniería del software: un enfoque práctico*. Séptima edición. México. McGraw-Hill Interamericana Editores S.A de C.V.

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

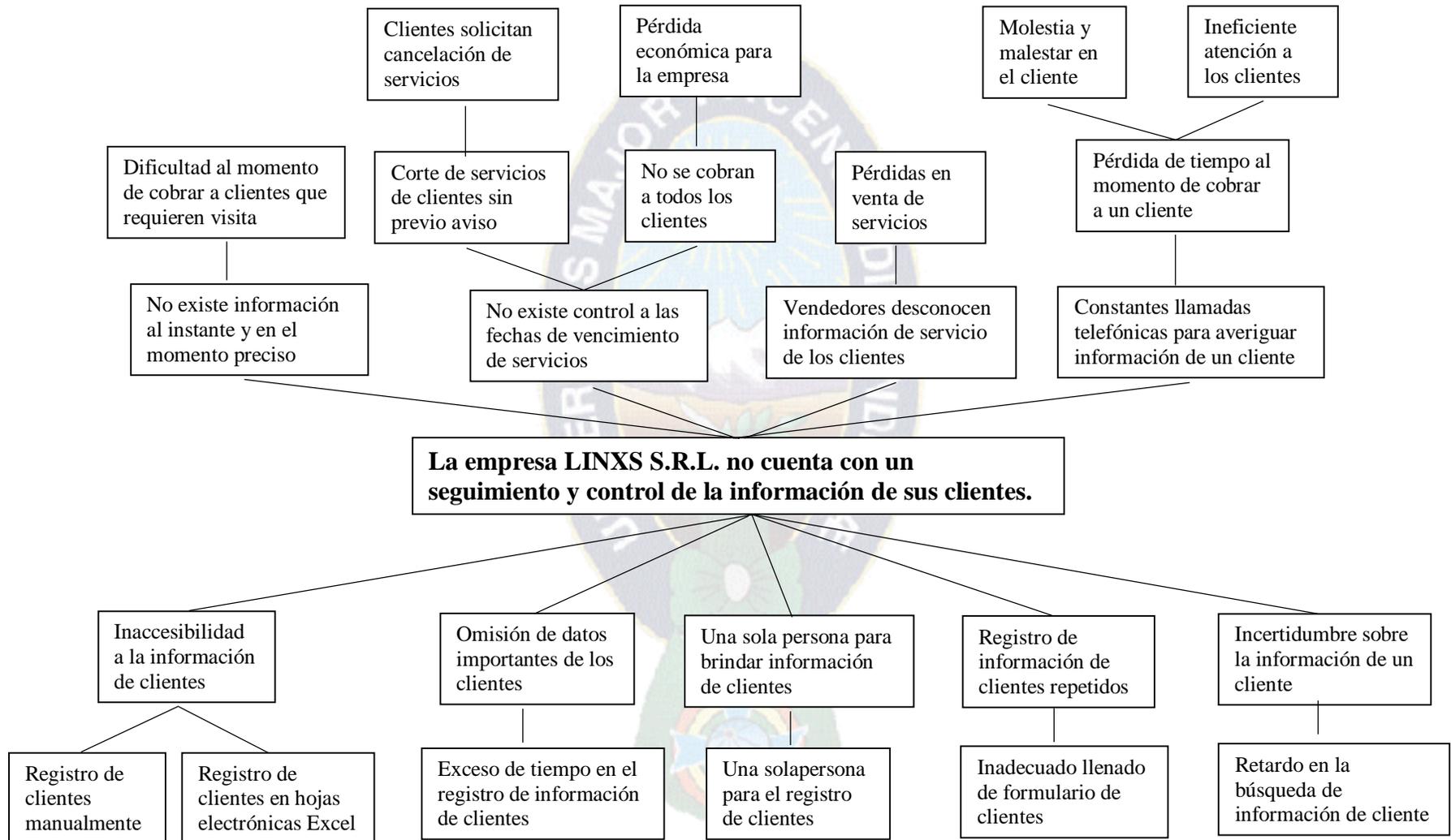
Yucra, W. (2016). *Sistema Web de Control y Seguimiento de Documentación Caso: Departamento de Infraestructura de la UMSA*. Universidad Mayor de San Andrés. Carrera de Informática.



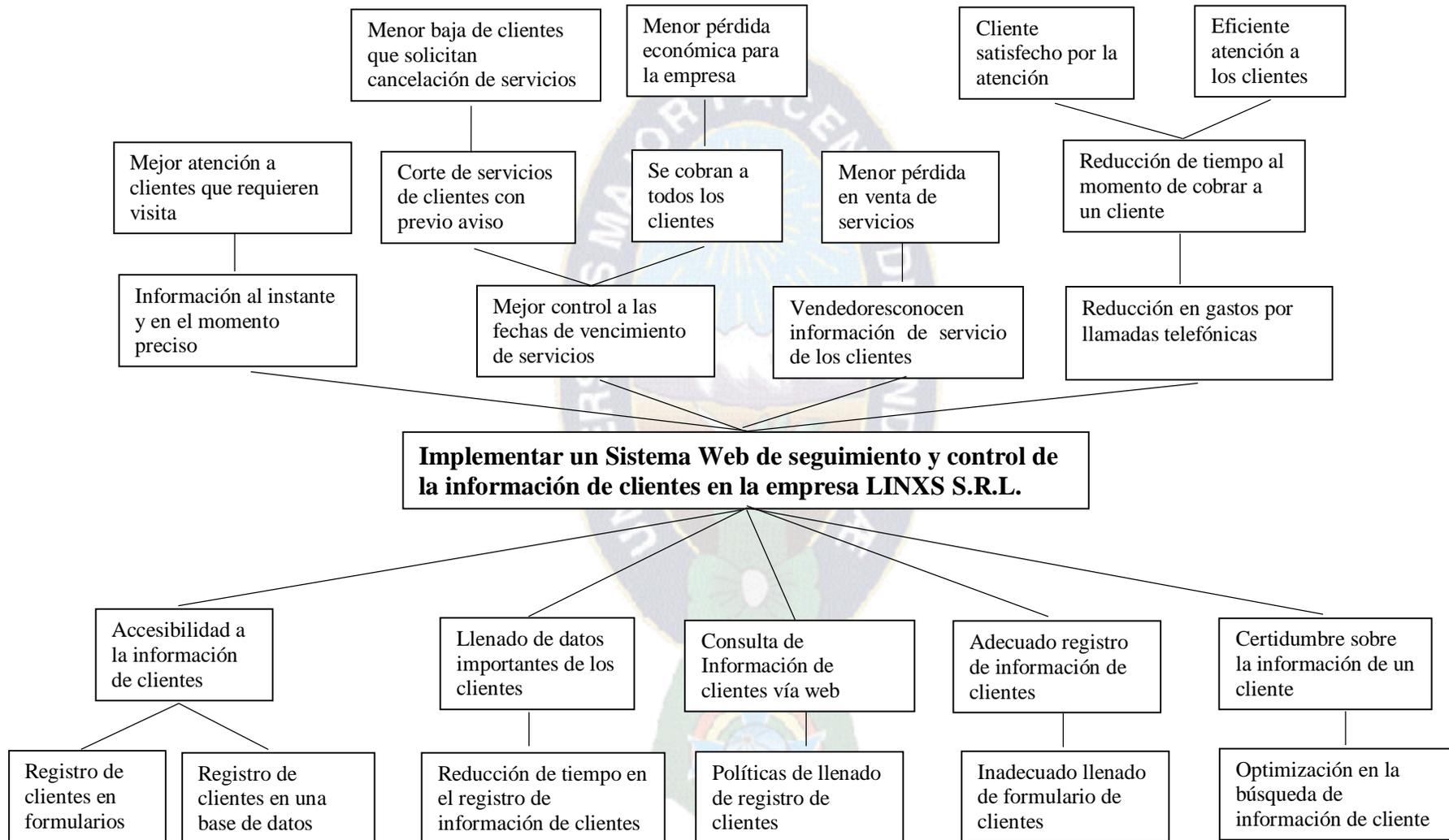


ANEXOS

ANEXO A: ARBOL DE PROBLEMAS



ANEXO B: ÁRBOL DE OBJETIVOS



ANEXO C: MARCO LÓGICO

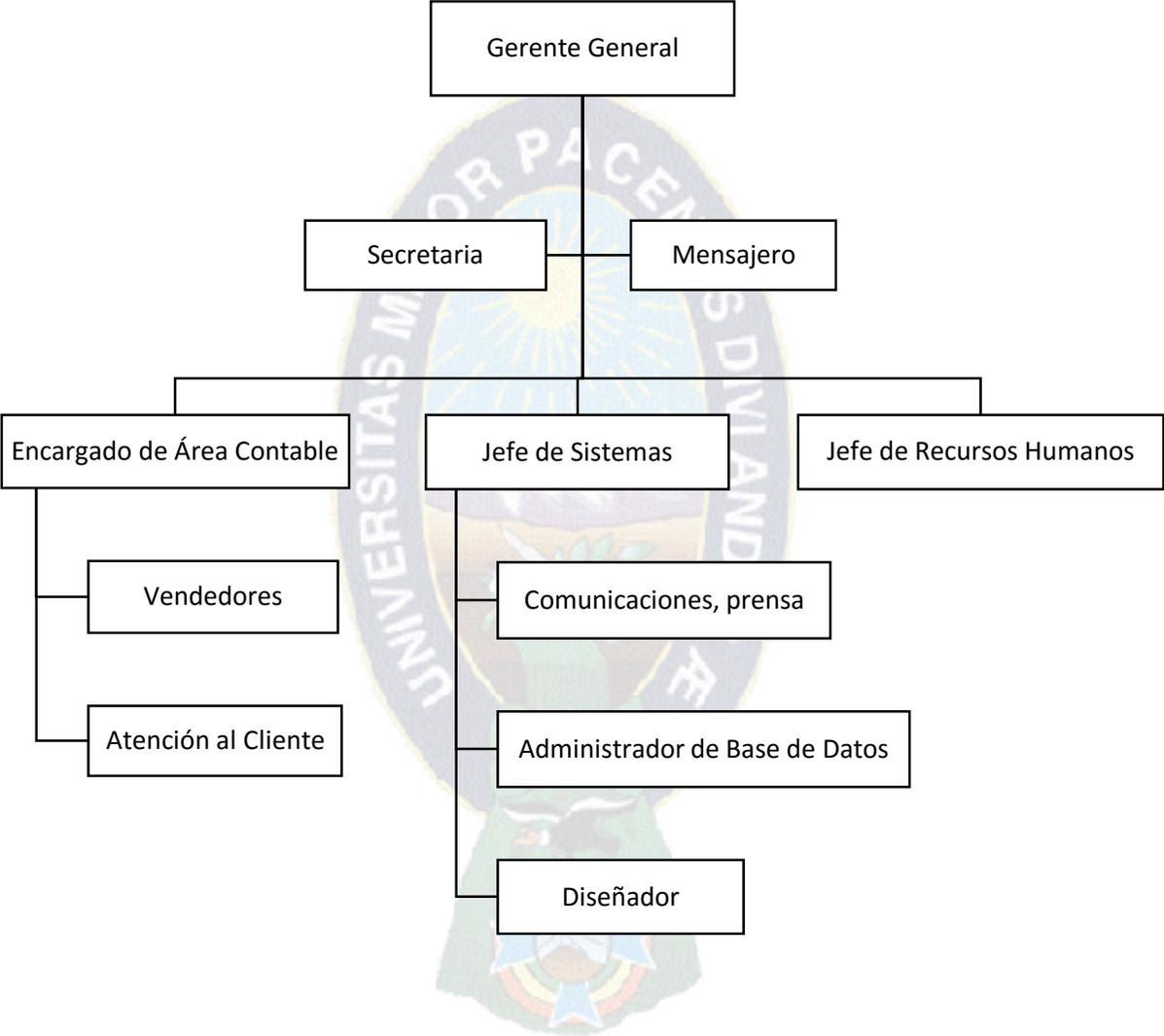
RESUMEN NARRATIVO	INDICADORES OBJETIVAMENTE VERIFICABLES	MEDIOS DE VERIFICACIÓN	SUPUESTOS
<p>FIN: La empresa LINXS S.R.L. tiene a sus clientes satisfechos y menor pérdida económica.</p>	<p>Estadísticas y comparación con gestiones pasadas en pérdidas económicas</p>	<p>Informes y reportes del departamento de contabilidad de la empresa LINXS S.R.L. a partir de la implementación del Sistema.</p>	<p>Los usuarios del sistema conocen y aplican las políticas de uso, además de pertenecer a la empresa.</p>
<p>PROPÓSITO: Desarrollar e implementar un Sistema Web de control y seguimiento de información de clientes en la empresa LINXS S.R.L.</p>	<ul style="list-style-type: none"> - 100% del sistema vía web de seguimiento y control culminado a la finalización de Taller II de año 2016. - 100% de cumplimiento de la especificación de requisitos. 	<ul style="list-style-type: none"> - Aval de la empresa LINXS S.R.L. una vez finalizada el Sistema Web de Seguimiento y Control de Información de Clientes. - Puesta en marcha del Sistema en un hosting de la empresa. 	<ul style="list-style-type: none"> - Se cuenta con la aprobación del Lic. Roberto Claros Uzqueda Gerente General de la empresa LINXS S.R.L. para la implementación del Sistema Web de Seguimiento y Control de Información de Clientes.
<p>PRODUCTOS:</p> <ul style="list-style-type: none"> - Diseñar una base de datos para el registro de clientes. - Desarrollar el módulo de roles y accesos de usuarios del sistema. - Desarrollar el módulo de registro de clientes. - Desarrollar el módulo de registro de servicios de la empresa. - Desarrollar el módulo de reportes. - Desarrollar el módulo de búsqueda y resultados de información de clientes. - Desarrollar el módulo de control de servicios de clientes. 	<ul style="list-style-type: none"> - 90% de los clientes registrados en la base de datos. - 100% de los usuarios del sistema registrados y con roles. - 100% de registro de clientes nuevos. - 100% de servicios de la empresa registrados. - Se puede generar reportes diarios, semanales, mensuales y anuales. - El tiempo de búsqueda es mínimo. - Control de servicios con mensajes de alerta. 	<ul style="list-style-type: none"> - Los usuarios pueden dar de crear, modificar y borrar clientes y servicios según el rol asignado por el administrador de sistema. - Los usuarios obtienen reportes para tomar decisiones. - El sistema avisa con mensajes de alerta sobre la finalización de servicios de clientes, para que pueda avisar al mismo y puedan tomar una decisión. 	<ul style="list-style-type: none"> - Se cuenta con toda la información de los clientes registrados manualmente en papeles y en hojas electrónicas Excel. - El personal de la empresa conoce de su rol asignado en el sistema. - Se cuenta con todas la herramientas necesarias para el desarrollo del sistema web.

<p>ACTIVIDADES:</p> <ul style="list-style-type: none"> - Coordinar requerimientos con el personal de la empresa. - Centralizar los datos existentes en la base de datos. - 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 - Evaluar resultados 	<ul style="list-style-type: none"> - 1 semana - 3 semanas - 2 semanas - 3 semanas - 3 semanas - 1 semanas - 2 semanas - 3 semanas - 3 semanas - 2 semanas - 1 semana 	<ul style="list-style-type: none"> - Material de escritorio, citas coordinadas con el personal - PC, material de escritorio, internet - PC, IDE, servidor local - PC, IDE, internet, servidor local - PC, IDE, internet, servidor local - Lista de usuarios. - Lista de servicios. - Reportes con gráficos e impresos. - PC, internet, servidor - PC, material de escritorio. - PC, material de escritorio. 	<ul style="list-style-type: none"> - Se cuenta con todo los materiales. - El personal asiste a las citas coordinadas para obtener los requerimientos. - Se cuenta con un servidor local. - Se cuenta con acceso a internet. - Se cuenta con un IDE para el desarrollo. - Se tiene acceso a la lista de usuario y servicios de la empresa. - Se cuenta con una impresora. - Se cuenta con el acceso al dominio del servidor compartido
--	---	--	---

ANEXO D: CRONOGRAMA DE ACTIVIDADES

ACTIVIDADES	DURACIÓN EN DÍAS	DEL 1 DE AGOSTO AL 09 DE DICIEMBRE																			
		AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE				DIC			
		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	65		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■				
Modelo de casos de uso	3			■																	
Modelo conceptual	4			■	■																
Modelo de usuario	3			■	■																
Modelo de estructura de presentación	5			■	■	■															
Modelo abstracto de interfaz de usuario	3			■	■																
Modelo de adaptación	3			■	■	■															
Coordinar requerimientos	7			■	■	■															
Centralizar datos existentes en base de datos	14			■	■	■	■	■													
Implementar controladores del Sistema	14			■	■	■	■	■	■												
Diseñar e implementar las vistas del Sistema	21			■	■	■	■	■	■	■											
Clasificar y asignar roles a usuarios del Sistema	7			■	■	■															
Ordenar los servicios de la empresa	14			■	■	■	■	■	■												
Generar reportes de servicios de clientes	21			■	■	■	■	■	■	■											
Implementar el Sistema Web en un hosting	21			■	■	■	■	■	■	■	■										
Realizar pruebas al sistema	14			■	■	■	■	■	■	■	■										
Evaluar resultados	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	3			■	■	■															

ANEXO E: ESTRUCTURA ORGANIZACIONAL DE LA EMPRESA LINXS S.R.L.



ANEXO F: CODIGO FUENTE DE LA CLASE ACL

```
<?php
class Acl{
    private $_registry;
    private $_db;
    private $_id;
    private $_role;
    private $_permisos;

    public function __construct($_id = false){
        if($_id){
            $this->_id = (int) $_id;
        }else{
            if (Session::get('id_usuario')){
                $this->_id = Session::get('id_usuario');
            }else{
                $this->_id = 0;
            }
        }

        $this->_registry = Registry::getIntancia();
        $this->_db = $this->_registry->_db;
        $this->_role = $this->getRole();
        $this->_permisos = $this->getPermisosRole();
        $this->compilarAcl();
    }

    public function compilarAcl(){
        $this->_permisos = array_merge(
            $this->_permisos,
            $this->getPermisosUsuario()
        );
    }

    public function getRole(){
        $role = $this->_db->query(
            "SELECT role FROM usuarios WHERE id = $this->_id"
        );
        $role = $role->fetch();
        return $role['role'];
    }
}
```

```

public function getPermisosRoleId(){
    $sids = $this->_db->query(
        "SELECT permiso FROM permisos_role WHERE role = '$this->_role'"
    );
    $sids = $sids->fetchAll(PDO::FETCH_ASSOC);
    $sid = array();
    for ($i=0; $i< count($sids); $i++){
        $sid[] = $sids[$i]["permiso"];
    }
    return $sid;
}

```

```

public function getPermisosRole(){
    $permisos = $this->_db->query(
        "SELECT * FROM permisos_role WHERE role = '$this->_role'"
    );
    $permisos = $permisos->fetchAll(PDO::FETCH_ASSOC);
    $data = array();
    for ($i = 0; $i<count($permisos); $i++){
        $key = $this->getPermisoKey($permisos[$i]["permiso"]);
        if($key == ""){continue;}
        if($permisos[$i]['valor'] == 1){
            $v = true;
        }else{
            $v = false;
        }

        $data[$key] = array(
            'key' => $key,
            'permiso' => $this->getPermisoNombre($permisos[$i]["permiso"]),
            'valor' => $v,
            'heredado' => true,
            'id' => $permisos[$i]["permiso"]
        );
    }

    return $data;
}

```

```

public function getPermisoKey($permisoId){
    $permisoId = (int) $permisoId;
    $key = $this->_db->query(
        "SELECT llave FROM permisos WHERE idpermiso = $permisoId"
    );
}

```

```

);
$key = $key->fetch();
return $key['llave'];
}

public function getPermisoNombre($permisoId){
    $permisoId = (int) $permisoId;
    $key = $this->_db->query(
        "SELECT permiso FROM permisos WHERE idpermiso = $permisoId"
    );
    $key = $key->fetch();
    return $key['permiso'];
}

public function getPermisosUsuario(){
    $sids = $this->getPermisosRoleId();

    if (count($sids)){
        $permisos = $this->_db->query(
            "SELECT * FROM permisos_usuario WHERE usuario = $this->_id AND permiso
IN (".implode(", ", $sids).")"
        );
        $permisos = $permisos->fetchAll(PDO::FETCH_ASSOC);
    }else{
        $permisos = array();
    }

    $data = array();
    for ($i = 0; $i < count($permisos); $i++){
        $key = $this->getPermisoKey($permisos[$i]["permiso"]);
        if($key == ""){continue;}
        if($permisos[$i]['valor'] == 1){
            $v = true;
        }else{
            $v = false;
        }

        $data[$key] = array(
            'key' => $key,
            'permiso' => $this->getPermisoNombre($permisos[$i]["permiso"]),
            'valor' => $v,
            'heredado' => false,
            'id' => $permisos[$i]["permiso"]
        );
    }
}

```

```

    );
}

return $data;
}

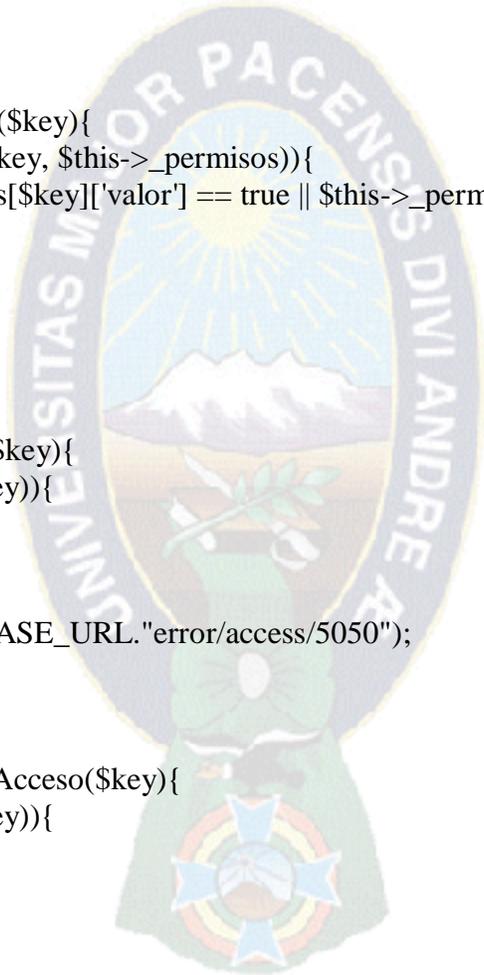
public function getPermisos(){
    if (isset($this->_permisos) && count($this->_permisos)){
        return $this->_permisos;
    }
}

public function permiso($key){
    if(array_key_exists($key, $this->_permisos)){
        if($this->_permisos[$key]['valor'] == true || $this->_permisos[$key]['valor'] == 1){
            return true;
        }
    }
    return false;
}

public function acceso($key){
    if($this->permiso($key)){
        Session::tiempo();
        return;
    }
    header("location: ".BASE_URL."error/access/5050");
    exit;
}

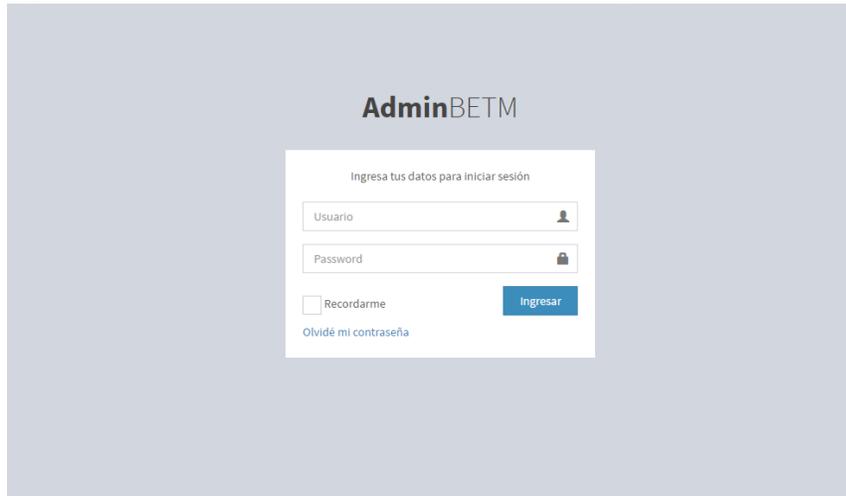
public function verificaAcceso($key){
    if($this->permiso($key)){
        return;
    }
}
}
}

```



ANEXO G: MANUAL DE USUARIO DEL SISTEMA

1. Acceso al sistema web.
 - Ingresar a la dirección: <http://boliviaentusmanos.net/adminbetm/>
2. Luego de ingresar deberá introducir su nombre de usuario y contraseña.



3. A continuación ingresará a la página principal.

Nº	Descripción	Departamento	Ciudad	Dirección
1	HOTEL GLORIA ***	La Paz	La Paz	c. Potr
2	REAL PLAZA HOTEL & CONVENTION CENTER	La Paz	La Paz	Av. Arc
3	ROASTER BOUTIQUE	La Paz	La Paz	c. Gab
4	HOTEL ESPAÑA **	La Paz	La Paz	Av. 6 d
5	OBERLAND S.R.L.	La Paz	La Paz	c. 2 y i
6	HOSTAL REPÚBLICA	La Paz	La Paz	c. Con
7	HOTEL PRESIDENTE *****	La Paz	La Paz	c. Potr
8	RITZ APART HOTEL *****	La Paz	La Paz	Plaza
9	EL REY PALACE HOTEL *****	La Paz	La Paz	Av. 20

4. Crear usuarios y asignar roles
 - Solo tiene acceso al menú y la vista los usuarios con el rol administrador de sistema.
 - Despliegue el menú de administrador de usuario y podrá ver el listado de usuario, registrar usuarios y administrar los roles y permisos.

Registrar Usuario boliviaentusmanos.com

Información del usuario para el Sistema

llenado de datos de usuario

Nombre Ej. Juan Perez

Usuario Ej. jperez

Email Ej. jperez@betm.com

Password 8 caracteres Ej: AbC123fG **Generar**

Confirmar 8 caracteres Ej: AbC123fG

Cargo Ej. Gerente

Rol Seleccione ->

- Seleccione ->
- Administrador
- Contador
- Sistemas
- Vendedor
- Mensajero

Menú administrar usuarios

Asignar Roles de usuario

5. Administrar Roles y permisos de usuarios.

Hector Cussi Online

Administración de permisos de Roles Role: Administrador

Level > Here

Permisos

Permiso	Habilitado	Denegado	Ignorar
Tareas de administracion	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Contabilidad	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Crear	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Editar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eliminar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lectura	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Guardar

Administrar roles y permisos de usuario

6. Llenado de datos de clientes y sus servicios.

7. Llenado de datos para el servicio de páginas amarillas.

8. Despliegue de reportes.

AdminBETM

Hector Cussi

Reporte businesscard

CLICK POR DÍA:

16/11/2016

TODAS

Nro.	EMPRESA	CLICKS
1	ECOJET	275
2	HOTEL ESPAÑA **	154
3	CHOCOLATES EL CEIBO	133
4	MADEPA S.A.	120
5	EMPRESA FERROVIARIA ANDINA S.A.	103
6	BANJO TOURS	101
7	TRANSPORTE AÉREO MILITAR	97
8	TURISMO BOLIVIA PERÚ - AGENCIA DE VIAJES	94
9	PURIFIKA BOLIVIA	93
10	ELI'S PIZZA EXPRESS & FAST FOOD	90

« Primero » « Anterior » 1 2 3 4 5 6 7 ... 53 54 Siguiente » « Último »

9. Cerrar sesión.

- Click en el nombre de usuario ubicado en la parte superior derecha del sistema.
- Se despliega información del usuario.
- En la parte inferior derecha del despliegue de información se encuentra el botón salir que cerrara todas las sesiones.

AdminBETM

Hector Cussi

Reporte businesscard

CLICK POR DÍA:

16/11/2016

TODAS

Hector Cussi - Web Developer
Miembro desde 01/06/2015

Salir

The logo of Universidad Mayor Pacensis Divi Andreae is an oval emblem. It features a sun with rays at the top, a mountain range in the middle, and a green ribbon at the bottom. The text "UNIVERSITAS MAJOR PACENSIS DIVI ANDREAE" is written around the perimeter of the oval. Below the oval is a blue cross with a white center, set against a green background.

DOCUMENTACIÓN