

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



PROYECTO DE GRADO

**“MODELADOR DE FORMULARIOS MUNICIPALES GOBIERNO
AUTONOMO MUNICIPAL DE LA PAZ
CASO: UNIDAD DE CATASTRO”**

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

POSTULANTE: EDWIN MARCELO CONDORI CALLA
**TUTORA METODOLÓGICA: M.Sc. FÁTIMA CONSUELO DOLZ DE
MORENO**
ASESORA: LIC. CARMEN ROSA HUANCA QUISBERT

LA PAZ – BOLIVIA

2013



**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

Dedico mi proyecto de Grado a:

A Dios por que ha sido lo más importante en mi vida,
quien debo todos mis logros porque siempre ha estado
a mi lado, guiándome, protegiéndome.

AGRADECIMIENTOS

A Dios por guiarme siempre, a mis padres por darme su apoyo incondicional, a mis hermanos por mi enamorada evita ayudarme siempre, gracias por ser parte de mi vida.

Al lic. Carmen Rosa Huanca Quisbert y M.Sc Fatima Dolz De Moreno, por su comprensión y disposición que me dieron para el desarrollo de este proyecto, gracias a su asesoramiento pude presentar y defender este proyecto.
Al Jefe de unidad Limbert Quispe y Roberto Morales y a todos los que son parte del municipio, por confiar y dar todo su apoyo y comprensión, para que este proyecto se pueda ver la luz y su utilidad.

ÍNDICE

CAPITULO 1.....	1
MARCO REFERENCIAL.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES.....	2
1.3. MARCO INSTITUCIONAL.....	3
1.3.1. ANTECEDENTES DE LA SITUACIÓN ACTUAL DE LA INSTITUCIÓN.....	3
1.4. ANÁLISIS DE INVOLUCRADOS.....	4
1.5. IDENTIFICACIÓN DEL PROBLEMA.....	5
1.5.1. PROBLEMA GENERAL.....	5
1.5.2. PROBLEMAS ESPECÍFICOS.....	6
1.5.3. FORMULACIÓN DEL PROBLEMA.....	7
1.6. OBJETIVOS.....	7
1.6.1. OBJETIVO GENERAL.....	7
1.6.2. OBJETIVOS ESPECÍFICOS.....	7
1.7. MÉTODOS Y TÉCNICAS.....	8
1.7.1. DISEÑO GENERAL.....	8
1.7.2. METODOLOGIA DE DESARROLLO.....	9
1.7.3. FACTIBILIDAD.....	9
1.7.3.1. OPERATIVO.....	9
1.7.3.2. TÉCNICO.....	9
1.8. JUSTIFICACION.....	11
1.8.1. JUSTIFICACIÓN CIENTÍFICA.....	11
1.8.2. JUSTIFICACION ECONOMICA.....	11
1.8.3. JUSTIFICACIÓN SOCIAL.....	11
1.8.4. JUSTIFICACIÓN TÉCNICA.....	12
1.9. LÍMITES Y ALCANCES.....	12
1.9.1. ALCANCES.....	12
1.9.2. LÍMITES.....	13
1.10. APORTES.....	13
CAPITULO 2.....	14
MARCO TEÓRICO.....	14
2.1. INTRODUCCIÓN.....	14

2.2. WORKFLOW.....	14
2.2.1. CONCEPTOS DE WORKFLOW	14
2.2.2. BENEFICIOS DE IMPLEMENTAR WORKFLOW	15
2.2.3. COMPONENTES DE UN WORKFLOW	16
2.2.3.1. PROCESOS DE NEGOCIO.....	16
2.2.3.2. INFORMACION.....	16
2.2.3.3. ORGANIZACIÓN.....	17
2.2.4. CONCEPTOS DE USO PARA MODELAR WORKFLOWS.....	17
2.2.5. PATRONES DE DISEÑO DE WORKFLOWS	19
2.3. NOTACIÓN PARA EL MODELADO DE PROCESOS DE NEGOCIO (BPMN).....	22
2.3.1. ELEMENTOS DE MODELAMIENTO BPMN BASICOS.....	23
2.4. MÉTRICAS DE CALIDAD DE SOFTWARE	25
2.4.1. FUNCIONALIDAD.....	25
2.4.2. MANTENIBILIDAD.....	27
2.4.3. CONFIABILIDAD.....	29
2.5. GESTIÓN DE SEGURIDAD.....	30
2.5.1. COSTO BENEFICIO.....	33
2.5.2. FACTORES DE ESCALA.....	35
2.6. ARQUITECTURA CLIENTE/SERVIDOR.....	36
2.6.1. PROGRAMACIÓN EN TRES CAPAS	37
2.6.2. HERRAMIENTA PARA LA IMPLEMENTACIÓN.....	38
2.6.2.1. NET FRAMEWORK.....	38
2.6.2.2. ASP.NET	38
2.6.2.3. SQL SERVER 2008 R2.....	39
2.7. DISEÑO METODOLÓGICO.....	39
2.7.1. METODOLOGÍA XP	39
2.7.1.1. FASES DE LA METODOLOGÍA XP	40
2.8. LENGUAJE UNIFICADO DE MODELADO (UML).....	48
2.8.1. TIPOS DE DIAGRAMAS EN UML	49
CAPITULO 3.....	53
MARCO APLICATIVO.....	53
3.1. INTRODUCCIÓN.....	53
3.2. ANÁLISIS DEL DOMINIO E IDENTIFICACIÓN DEL PROCESO.....	53
3.2.1. ANALISIS DEL DOMINIO DE LOS PROCESOS.....	53
3.2.2. IDENTIFICACION DEL PROCESO.....	53
3.3. FASES DE LA METODOLOGIA XP	57
3.3.1. FASE DE PLANIFICACION.....	57

3.3.1.1 EQUIPO DEL PROYECTO.....	57
3.3.1.2 HISTORIAS DE USUARIO.....	58
3.3.1.3 ITERACIONES.....	65
3.3.1.3.1 CRONOGRAMA FINAL.....	65
3.3.1.4 PLAN DE ENTREGAS.....	65
3.3.1.5 VELOCIDAD DEL PROYECTO.....	67
3.3.1.6 REUNIONES.....	68
3.3.2 FASE DE DISEÑO.....	68
3.3.2.1 METAFORA DEL SISTEMA.....	68
3.3.2.2 ACTORES DEL NEGOCIO.....	69
3.3.2.3 DIAGRAMA DE CASOS DE USO.....	72
3.3.2.4. MODELO FISICO.....	74
3.3.2.5 DIAGRAMA DE CLASES.....	76
3.3.2.6 TARJETAS CLASE, RESPONSABILIDAD Y COLABORACION (CRC).....	77
3.3.2.7 DIAGRAMA DE PAQUETES.....	81
3.3.3. FASE DE DESARROLLO.....	82
3.3.3.1 ITERACION FINAL DEL SISTEMA.....	83
3.3.3.1.1 SEGURIDAD Y CONTROL DE ACCESO.....	83
3.3.3.1.2 MODELADO DE FORMULARIO.....	85
3.3.3.1.3 SEGUIMIENTO DE FORMULARIO.....	87
3.3.3.1.4 REPORTES.....	89
3.3.4. FASE DE PRUEBAS.....	91
3.3.4.1 PRUEBAS DE ACEPTACION.....	91
3.4 ARQUITECTURA DEL SISTEMA.....	93
3.5. SEGURIDAD Y CONTROL DE ACCESO.....	94
3.5.1 CONTROL DE ACCESO.....	94
3.5.2 SEGURIDAD EN LA BASE DE DATOS.....	94
3.4.3 SEGURIDAD DE LA INFORMACION.....	95
CAPITULO 4.....	97
CALIDAD DE SOFTWARE.....	97
4.1 METRICAS DE CALIDAD.....	97
4.1.1 FUNCIONALIDAD.....	97
4.1.2 PORTABILIDAD.....	100
4.1.3 MANTENIBILIDAD.....	101
4.1.3.1 MANTENIMIENTO CORRECTIVO.....	101
4.1.3.2 MANTENIMIENTO ADOPTIVO.....	102
4.1.3.3 MANTENIMIENTO PERFECTIVO.....	102
4.1.4 FACILIDAD DE USO.....	103
CAPITULO 5.....	105

CONCLUSIONES Y RECOMENDACIONES.....	105
5.1 CONCLUSIONES.....	105
5.2 RECOMENDACIONES.....	106
BIBLIOGRAFÍA.....	107
ANEXO A.....	109
ANEXO B.....	113
ANEXO C.....	126

INDICE DE FIGURAS

Figura 2.1. Patrón de secuencia	30
Figura 2.10 Ejemplo Diagrama de Clases.....	61
Figura 2.11 Representación de un Diagrama de Paquetes	62
Figura 2.12 Ejemplo de un Diagrama de Paquetes.....	62
Figura 2.2. Patrón en paralelo.....	31
Figura 2.3. Proceso de Solicitud de Crédito.....	33
Figura 2.4 Fases de la Metodología XP	51
Figura 2.5 Ciclo de vida de eXtremeProgramming,.....	52
Figura 2.6 Árbol de requisitos de Calidad propuesta de Olsina	56
Figura 2.7 Representación de un Diagrama de Casos de Uso	60
Figura 2.8 Ejemplo de un Diagrama de Casos de Uso	60
Figura 2.9 Representación de un Diagrama de Clases.....	61
Figura 3.1 Proceso – Aprobación de planos (Proceso tipo)	64
Figura 3.10 Pantalla de registro de Usuario Rol.....	96
Figura 3.10 Pantalla de Registro Usuario	96
Figura 3.11 Pantalla de registro de Usuario Rol.....	96
Figura 3.12 Pantalla de Asignación de Usuario – Rol	97
Figura 3.13 Pantalla Adicionar Repositorio	97
Figura 3.14 Pantalla Listar Campos Físicos.....	98
Figura 3.16 Pantalla Listar Campo lógicos.....	99
Figura 3.17. Pantalla Listar Actividades	99
Figura 3.18 Pantalla Listar Vistas.....	100
Figura 3.19 Pantalla Listar Dominios	101
Figura 3.2 Velocidad del proyecto.....	80
Figura 3.20 Pantalla de la hoja de Ruta	101
Figura 3.21 Pantalla de Formulario creado	102
Figura 3.22 Pantalla de Seguimiento de Formularios	102
Figura 3.23 Arquitectura del sistema.....	105

Figura 3.23 Pantalla de Reportes estáticos	103
Figura 3.4 Casos de Uso del Sistema	85
Figura 3.5 Modelo Físico.....	87
Figura 3.6 Diagrama de clases	88
Figura 3.7 Diagrama de paquetes del sistema (MVC3)	93
Figura 3.8 Diagrama de paquetes del sistema.....	94
Figura 3.9 Pantalla de Autenticación.....	95
Figura. 3.15 Pantalla Adicionar proceso	98

INDICE DE TABLAS

Tabla 1.1 Descripción de las características esenciales para implementar el sistema.....	20
Tabla 2.1. Elementos de modelamiento BPMN básicos	34
Tabla 2.2 Escala de Punto Función	36
Tabla 2.3 Escala de Mantenibilidad	38
Tabla 2.4 Escala de Portabilidad.....	39
Tabla 2.5 Valores del modelo COCOMO II	45
Tabla 2.6 Valores de calibrado del modelo COCOMO II.....	45
Tabla 3.1 Equipo del Proyecto	66
Tabla 3.10 Historia del usuario: Agregación de Campos Lógicos	73
Tabla 3.11 Historia del usuario: Creación de Actividades	74
Tabla 3.12 Historia del usuario: Vistas.....	75
Tabla 3.13 Historia del usuario: Dominios.....	75
Tabla 3.14 Historia del usuario: Seguimiento de Formularios	76
Tabla 3.15 Historia del usuario: Reportes	77
Tabla 3.16 Estimación de Esfuerzo.....	77
Tabla 3.17 Módulo de Seguridad y Accesos	78
Tabla 3.18 Módulo de Modelado de Formulario	78
Tabla 3.19 Historia del usuario: Modulo de Seguimiento de Formulario	79
Tabla 3.2 Equipo del Proyecto	67
Tabla 3.20 Cronograma por Historias por Historias de Usuario	80
Tabla 3.21 Plan de entrega	80
Tabla 3.22 Reuniones	81
Tabla 3.22 Velocidad del proyecto	80
Tabla 3.23 Reuniones	81
Tabla 3.24 Actores del negocio.....	83
Tabla 3.25 Actividades de los Actores	84
Tabla 3.26 Descripción del modelado del Sistema	86

Tabla 3.27 tarjeta CRC de Personas	89
Tabla 3.28 tarjeta CRC de Usuarios	89
Tabla 3.29 tarjeta CRC de Roles	89
Tabla 3.3 Historia del usuario: Seguridad y Control de Acceso	68
Tabla 3.30 tarjeta CRC de Roles	89
Tabla 3.31 tarjeta CRC de Accesos	90
Tabla 3.32 tarjeta CRC de Opciones	90
Tabla 3.33 tarjeta CRC de Grupos.....	90
Tabla 3.34 tarjeta CRC de Usuarios de las Unidades.....	90
Tabla 3.35 tarjeta CRC de Expedidos.....	91
Tabla 3.36 tarjeta CRC de Repositorio	91
Tabla 3.37 tarjeta CRC de Procesos.....	91
Tabla 3.38 tarjeta CRC de Procesos.....	92
Tabla 3.39 tarjeta CRC de Usuarios	92
Tabla 3.4 Historia del usuario: Seguridad - Personas	69
Tabla 3.40 tarjeta CRC de Usuarios	92
Tabla 3.41 tarjeta CRC de Usuarios	92
Tabla 3.42 tarjeta CRC de Usuarios	92
Tabla 3.43 tarjeta CRC de Usuarios	93
Tabla 3.44 Tabla de valores obtenidos en 8 días.....	112
Tabla 3.45 de valores en porcentajes de Facilidad de usabilidad.....	114
Tabla 3.5 Historia del usuario: Seguridad - Usuarios	70
Tabla 3.6 Historia del usuario: Seguridad - Roles	70
Tabla 3.7 Historia del usuario: Creación de Repositorio	71
Tabla 3.8 Historia del usuario: Creación de Campos Físicos.....	72
Tabla 3.9 Historia del usuario: Adición de Proceso.....	73

CAPITULO 1

MARCO REFERENCIAL

1.1. INTRODUCCIÓN

La tecnología Workflow (Entendido como el flujo de procesos administrativos o de negocio, es el conjunto de actividades realizadas en secuencia o en paralelo por dos o más miembros de un equipo para lograr un objetivo común siguiendo unas reglas de negocio preestablecidas. La automatización de procesos de negocio. Históricamente, la automatización de procesos que hacen uso intensivo de Recursos Humanos, lo que plantea el problema de la colaboración de tipo persona-persona.

Los recursos de una organización se componen tanto de Recursos Humanos como de sus Recursos Tecnológicos. Para lograr el máximo rendimiento, es necesario lograr una alta eficiencia en el uso de la información en la gestión de la organización, reduciendo la discrecionalidad funcionaria y contribuyendo a la transparencia de las decisiones democráticas.

La Dirección de Administración Catastral del GAMLP (Gobierno Autónomo Municipal de La Paz), a través de la Unidad de Catastro para realizar el seguimiento de trámites, referentes a: Registro de Planos, Aprobación de Planos División y Partición, Fraccionamiento, Autorización de Obras Horizontal y Autorización de Obras Menores, es muy importante para el Municipio de La Paz como para el mismo Municipio de la ciudad La Paz, ya que tiene como objetivo principal generar recursos e incrementar sus ingresos, optimizando el control y administración.

El proyecto Modelador de Formularios Municipales Gobierno Autónomo Municipal de La Paz Caso: Unidad de Catastro, propone una solución al problema de la generación de formularios de forma estática. La propuesta se basa en un sistema de información que

sea capaz de generar formularios de forma dinámica y así agilizar el seguimiento de trámites, que apoya el fortalecimiento de la capacidad técnica y gestión de las instancias participantes e involucradas.

El objetivo del proyecto es desarrollar un "Modelador de Formularios Municipales Gobierno Autónomo Municipal de La Paz Caso: Unidad de Catastro", con el objeto de lograr que la administración municipal (funcionarios) logre acercarse al ciudadano, a través de una óptima presentación de servicios, garantizando la transparencia y eficiencia, simplificando la gestión municipal y fortaleciendo la autoridad del municipio. De acuerdo a esto, el desarrollo de aplicaciones informáticas tiende a unirse a los métodos y herramientas, que ayuden a la construcción de sistemas con las expectativas de los funcionarios.

1.2. ANTECEDENTES

El presente proyecto busca reemplazar el sistema que actualmente es manejado por el "Sitr@m" ya "obsoleto" y no responde adecuadamente con los requerimientos del GAML (Gobierno Autónomo Municipal de La Paz), este nuevo sistema permitirá realizar un modelado de formularios de forma dinámica, logrando de esta manera su fácil manipulación.

En los últimos años en nuestro país la carrera de Informática (Universidad Mayor de San Andrés) se han realizado trabajos de investigación relacionados al tema. Para tener una mejor idea, vemos qué relación existe con la investigación.

El Gobierno Municipal de Santa Cruz de la Sierra - Bolivia, que en marzo del 2004, implemento la automatización de trámites de licencias de funcionamiento para actividades económicas. A través de este sistema, la Municipalidad de Santa Cruz

obtiene una mayor eficiencia y transparencias en sus trámites, constituyendo un claro beneficio para la ciudadanía.

En enero, 2005 ENTEL Móvil de Bolivia eligió a Q-flow Enterprise Edition a través de Intersoft Bolivia para llevar a cabo la automatización de procesos críticos de negocios, selecciono a Q-flow.

En enero, 2003 - Abbott Laboratorios de Bolivia incorpora Q-flow para la gestión y automatización de sus notas clínicas, debido por las oportunidades y la necesidad de introducir mejoras en la gestión y gestión en varios de sus procesos internos.

En junio de 2006, en el marco del proyecto de automatización de procesos, se realizó el proyecto: “Workflows Aplicado a los Procesos de Seguimiento de Casos de Procesos Caso: Instituto Nacional de Estadística”

1.3. MARCO INSTITUCIONAL

1.3.1. ANTECEDENTES DE LA INSTITUCIÓN

El Municipio de la Ciudad de La Paz tiene como objetivo de brindar servicio para el beneficio de ciudadanos, el crecimiento de la población obliga a la creación de varias direcciones y unidades (comités de catastro).

El Gobierno Municipal de la Ciudad La Paz y la Dirección de Administración Territorial y Catastral, como institución pública del estado tienen como función brindar servicios para el beneficio de los ciudadanos en los diferentes trámites que requiere la ciudadanía actualmente. El crecimiento acelerado de la población de La Paz hace que se creen problemas como; falta atención a los ciudadanos o contribuyentes retrasando solicitudes de los trámites de Aprobación de Planos, Aprobación de Planos División y Partición,



Fraccionamiento de Propiedad Horizontal y Autorización de Obras Menores, referentes a la Unidad de Catastro del Municipio de La Paz, en el cual no hay un seguimiento completo además de un control sobre el seguimiento o estado de sus trámites, además llevan un lapso de tiempo extendido, y el proporcionar esta información es difícil lo que es un problema.

1.4. ANÁLISIS DE INVOLUCRADOS

¿Quiénes son los involucrados?

- La Dirección de Desarrollo de Tecnologías de Información (DDOTI).
- La Unidad de Catastro y Administración Territorial y Catastral.
- Profesionales funcionarios, topógrafos, geodesta, ingenieros, etc.
- Técnicos auxiliares, funcionarios.

¿Qué actividades efectúan los involucrados?

- Desarrollo y mantenimiento de sistemas de información.
- Planificar y organizar, dirigir y controlar las actividades de Gestión Catastral.
- Participar en la elaboración de proyectos con el equipo técnico. Proponer políticas en materia de Catastro.
- Fiscalizar las labores de cada una de las unidades.
- Coordinar con las instancias de planificación y normatividad municipal y Dirección de Ordenamiento Territorial.
- Revisar, atender y responder la correspondencia recibida de manera oficial.
- Los trámites que lleguen vía Secretaría General u otras instancias, deberán tener visto bueno.
- Presentar informes a instancias superiores cuando le sea requerido.



¿Para que efectúan estas actividades?

- Para establecer un sistema de información catastral, con registros de la propiedad inmueble de todo el territorio municipal.
- Establecer, regular y controlar el cumplimiento de normas de edificación, partición de tierras y uso de suelo a través de una gestión administrativa eficiente y oportuna.
- Realizar trámites de las distintas unidades que son solicitadas para su registro correspondiente.
- Brindar servicio de consultas y asesoramiento a los propietarios de los inmuebles.

1.5. IDENTIFICACIÓN DEL PROBLEMA

El Gobierno Autónomo Municipal de La Paz procesa aproximadamente más de 30.000 trámites de catastro y catastral al año, de a un promedio de 150 personas por día, que incluyen una gran diversidad de trámites que se procesan en las distintas unidades de la Dirección Municipal de Catastro y Catastral del GAMLP.

El problema es que no existe un sistema de construcción de formularios de forma dinámica, el cual aglomera los trámites y el seguimiento de los mismos, en coordinación con los funcionarios de las distintas unidades, con el control y evaluación en las diferentes tramites que se atienden en la Dirección Municipal de Catastro. Ocasionando retrasos que existe en la realización de trámites como: Inscripción de plano de lote para DD.RR., Visado de plano de lote, Registro de plano de lote (registro de nombre, Línea y Nivel, etc.), no existe un seguimiento y control de los trámites de manera eficiente. La elaboración de la documentación se realiza de manera lenta, por lo que el proceso de trámites toma más tiempo de lo estimado y la aprobación de los diferentes trámites e informes tardan más de lo indicado.

1.5.1. PROBLEMA GENERAL

El análisis realizado permite identificar el siguiente problema general:

"No existe una herramienta capaz de modelar formularios de forma dinámica y que estos a su vez sean utilizados para el seguimiento de trámites y control de los procesos municipales que se registran en la dirección de Catastro, lo que ocasiona falta de coordinación entre funcionarios administrativos y auxiliares técnicos además se presentan dificultades en la presentación eficiente y oportuna de servicios a los contribuyentes, respecto al estado y control de trámites que realiza el interesado ”.

1.5.2. PROBLEMAS ESPECÍFICOS

Después de un análisis se determinó el problema general los siguientes problemas específicos de la Dirección de Administración Territorial y Catastral del GAMLP.

- a. No existe una herramienta capaz de modelar formularios de forma dinámica, para posteriormente ser utilizados para el seguimiento de los trámites en las diferentes unidades de la Dirección de Administración Territorial y Catastral del GAMLP. Los encargados en el momento de dar respuesta a los contribuyentes presentan cierta deficiencia y demora en dar respuesta a los contribuyentes).
- b. Falta de coordinación entre funcionarios administrativos respecto al modelado de formularios para el seguimiento de trámites y control de los procesos municipales que se registran en la Dirección de Administración Territorial y Catastral del GAMLP.
- c. Si bien se tiene un determinado proceso de procesamiento para cada trámite distinto, este sigue siendo lento para el solicitante o interesado.
- d. La gran cantidad de información que se maneja ocasiona que corra el peligro de no ser: íntegra, confiable y oportuna para la toma de decisiones.
- e. Falta de un proceso de tramitación normalizado para todos los tipos de trámites en la dirección de catastro.
- f. Falta de control en la administración de los funcionarios en el momento de acceder y manipular la información lo que produce inseguridad en la consistencia de los mismos.



- g. Ausencia de un sistema de elaboración de formularios dinámicos.

1.5.3. FORMULACIÓN DEL PROBLEMA

¿Existe alguna forma de agilizar la creación de formularios, mediante un método y/o sistema, para posteriormente realizar el seguimiento de trámites y brindar un mejor servicio al solicitante y dar respuestas oportunas a consultas respecto al estado de trámites que se realiza, además de estar integrada con el control y administración de trámites de la Unidad de Catastro?

1.6. OBJETIVOS

1.6.1. OBJETIVO GENERAL

Desarrollar e implementar un sistema de Formularios Municipales Gobierno Autónomo Municipal de Cochabamba, Unidad de Catastro, que permita elaborar formularios de forma dinámica, integrados con los sistemas proporcionados por la Unidad de Catastro, además de estar integrada con el control y administración de trámites de la Unidad de Catastro, para la realización de trámites.

1.6.2. OBJETIVOS ESPECÍFICOS

Los objetivos específicos consisten en:

- a. Implementar el proceso de Aprobación de Planos ya definido por la Unidad de Catastro, para la elaboración del Formulario.
- b. Realizar el análisis y diseño del sistema aplicando la metodología XP.
- c. Diseñar la Base de Datos que se adecuó para el Registro de Seguimiento de Trámites.
- d. Disponer de información oportuna, confiable y completa.



- e. Desarrollar un módulo de formularios dinámicos administrado por un técnico en informática.
- f. Realizar un almacenamiento de la información, que se genera en los formularios (para la realización trámites) de la Unidad de Catastro del GAMLP.
- g. Aplicar normas de seguridad (backup) para la Base de datos.

1.7. MÉTODOS Y TÉCNICAS

1.7.1. DISEÑO GENERAL

La valoración de las necesidades de la administración pública en el ámbito de la prestación de los servicios digitales proporcionados por las autoridades locales para los ciudadanos. Por eso se plantea una metodología de desarrollo de un sistema de formularios municipales para aplicarla en el municipio. Es importante que el sistema sea fácil de usar y aceptado para la administración, y que pueda usarse como herramienta de trabajo en el sector público. Para la construcción e implementación del Sistema de Formularios Municipales del Gobierno Autónomo Municipal de Cochabamba, se utilizara:

- a. Método científico: Técnicas de recolección, muestreo, identificación de variables que intervienen en el fenómeno y planteamientos de soluciones y estadísticas.
- b. Métodos de obtención de información: Entrevistas e información obtenida de la institución pública y antecedentes de formularios municipales.
- c. Métodos del marco lógico orientado a la planificación de proyectos.
- d. Técnicas y herramientas de desarrollo de software:

UML: (Lenguaje de Modelado Unificado- UnifiedModelingLanguage) como herramienta para la presentación del modelo del sistema en el desarrollo, utilizando vistas para visualizar proyecciones del sistema relacionadas con aspectos particulares, funcionales y no funcionales.

1.7.2. METODOLOGIA DE DESARROLLO

Como herramienta de modelado para su construcción se utilizara la metodología XP (Programación Extrema) por (Beck, 1997).

1.7.3. FACTIBILIDAD

1.7.3.1. OPERATIVO

El proyecto tendrá un diseño de interfaz de usuario final (funcionarios, administrativos y contribuyentes) que podrá acceder al sistema mediante una cuenta de usuario y contraseña, para los usuarios correspondientes.

1.7.3.2. TÉCNICO

El presente proyecto se basará en la utilización de la tecnología existente para diseño de aplicaciones web de acuerdo a los requerimientos del usuario. ASP.NET MVC3 con Entity Framework. El frontal de la aplicación se implementará con Microsoft SQL Server 2008 R2 y Windows Server 2007 R2. Su configuración de servidor de base de datos multi-usuario permitirá el acceso a la base de datos local de computadoras en la que el sistema será implementado.

SOFTWARE Y HARDWARE

Usuarios (Windows - Intranet Local)

Requerimientos mínimos de hardware:	Requerimientos mínimos de software:
<ul style="list-style-type: none"> ● PC Pentium III de 500Mhz o superior. ● Memoria: 128 MB de RAM ● Tarjeta de Red 10/100 BaseT ● Es imprescindible que los equipos estén conectados en RED 	<ul style="list-style-type: none"> ● Sistema operativo Windows ● Mozilla Firefox versión 12 ● Plataforma de Framework 2.0, 1.0,1.1 ● Servidor Web con soporte para ASP.NET. desde la versión 3. ● Red de un Dominio. ● Base de Datos SQL Server R2
Servidor Web	
Requerimientos mínimos de hardware:	Requerimientos mínimos de software:
<ul style="list-style-type: none"> ● Windows Server 2003 ● Windows XP Internet ● Information Services 7 o superior ● Microsoft .net ● Framework 3.0, 3.5 	<ul style="list-style-type: none"> ● requisitos de Microsoft.net ● Framework 3.0, Framework 3.5 y de Internet ● Information Services 7 o superior ● versión mínima recomendada: o superior ● con 1GB de memoria.

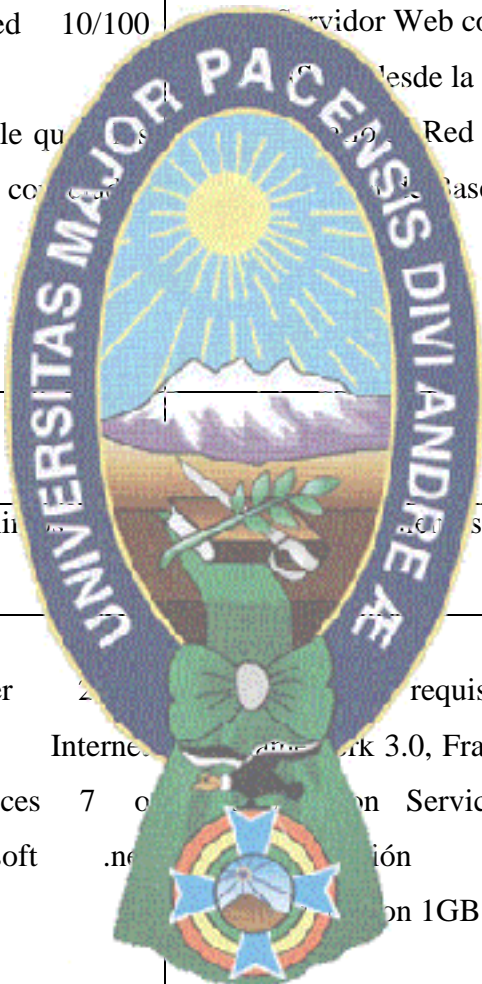


Tabla 1.1 Descripción de las características esenciales para implementar el sistema.

Fuente: Elaboración propia.

1.8. JUSTIFICACION

1.8.1. JUSTIFICACIÓN CIENTÍFICA

Científicamente el trabajo es un aporte en cuanto a otras áreas de investigación, en nuestro ámbito de estudio Intranet Municipal, ya que se toma las áreas de Administración, Normas, Artículos, Decretos, uso de nueva tecnología, lo cual se convierte en una herramienta para la obtención de datos cuantitativos y cualitativos para la actualización y clasificación de los diferentes tipos de datos que pueden ser obtenidos.

1.8.2. JUSTIFICACIÓN ECONÓMICA

Económicamente está justificada porque este tiene un costo bajo desde un punto de vista de inversión, la Unidad de Catastro obtendrá beneficios que obtendrá de la implementación del sistema que serán mayores a los gastos para la implementación de dicho sistema.

En cuanto a la adquisición de equipos se tiene la necesidad de invertir en nuevos equipos puesto que la Unidad de Catastro cuenta con todos los requisitos en cuanto al hardware.

1.8.3. JUSTIFICACIÓN SOCIAL

Con el sistema se beneficiaran los funcionarios, administrativos en sus actividades y funciones que desempeñan cotidianamente en la Unidad de Catastro, permitiéndoles llevar sus funciones con mayor eficiencia y eficacia en el manejo de la información y así brindar un mejor servicio al contribuyente.



1.8.4. JUSTIFICACIÓN TÉCNICA

Para la implementación del sistema, en la Unidad de Catastro, la aplicación tecnológica, métodos y técnicas de última generación, permitirá así un adecuado manejo de la información. La tecnología está dirigida a la aplicación de Base de Datos desarrollado en SQL Server 2008 R2, teniendo como interfaz aplicaciones Web de Microsoft NET, tomando en cuenta que el sistema cumple con las cualidades de integridad, seguridad y disponibilidad de los datos. El sistema es fácil de utilizar por lo que no se requiere personal especializado en informática, pero sí es necesario tener conocimientos de modelado de datos, se realiza cursos de capacitación o la elaboración de manuales de usuario, el personal podrá manejar el sistema sin ningún problema en un tiempo corto.

1.9. LÍMITES Y ALCANCES

1.9.1. ALCANCES

Considerando que el entorno de trabajo es amplio. El campo de investigación que se realizará está delimitado en el Municipio de la Ciudad de La Paz en la Unidad de Catastro, se realiza diferentes tipos de trámites concernientes al proceso de Aprobación del GAMLP, a partir de lo expuesto anteriormente en los problemas de investigación se plantean cuatro módulos o procesos que satisfacen la problemática a nivel catastral, siendo estos módulos correspondientes a continuación:

- Generación de formularios de forma dinámica.
- Seguimiento de Formularios (Tramites).
- Consultas de los Formularios.
- Reportes de los Formularios.

1.9.2. LÍMITES

El proyecto se limitara a la Unidad de Catastro, dependiente de la Dirección de Administración Territorial y Catastral. Se implementara el Modelador de Formularios en el proceso: Aprobación de planos. Teniendo en cuenta que el Modelador de Formularios Dinámicos puede implementarse con otros procesos similares que maneja la Unidad de Catastro.

1.10. APORTES

El presente proyecto tiene como aporte a la Unidad de Catastro, entre los que se pueden destacar:

La agilización de trámites mediante la automatización del proceso (Generación del Formulario Dinámico).

Desarrollo de componentes de sistemas con características similares, gracias a la aplicación de metodologías ágiles y el paradigma orientado a objetos.



CAPITULO 2

MARCO TEÓRICO

2.1. INTRODUCCIÓN

En este capítulo se realizara una descripción de los conceptos básicos de herramientas que se utilizaran en el transcurso del desarrollo del proyecto, entre los puntos se verá: Definición de Workflow, metodologías (como el modelo XP), gestión de seguridad, métricas de calidad, arquitectura a implementar.

2.2. WORKFLOW

El workflow, es un software que regula el flujo de procesos administrativos o de negocios para definir todas las tareas, actividades o procesos que deban realizarse en secuencia de manera eficiente en una organización. Esta forma de trabajo ayuda a definir todos los procedimientos inherentes al negocio; es decir, para lograr un objetivo común, es decir una meta que alcanzar y poder medirlo, también debe existir una regla definida que los integrantes de la organización sigan y respeten. Esta herramienta de gestión, permite la automatización de las actividades que componen un negocio, mediante un flujo de información que define áreas y sus encargados, el tiempo dado para la ejecución, las reglas predefinidas y el control del proceso en su conjunto (Dellen, 2006)

2.2.1. CONCEPTOS DE WORKFLOW

En 1993, con la misión de promover y desarrollar el uso de workflow a través del establecimiento de estándares para la terminología de los sistemas, interoperabilidad y conectividad, se fundó la Coalición de Administración de Workflow (Workflow Management Coalition). Actualmente está integrada por más de doscientos ochenta y

cinco miembros, distribuidos en todo el mundo. La Coalición se ha convertido rápidamente en un cuerpo primario para estándares del mercado creciente de sistemas de workflow. Comercialmente, es conocida por el acrónimo WfMC. Sus investigaciones y estándares son publicados periódicamente en un manual de trabajo conocido como “WorkflowHandbook” (Manual de Workflow). La WfMC define workflow como “la automatización de un proceso de negocios, durante el cual se trasladan documentos, información o tareas de un participante a la espera que se realice una actividad, de acuerdo a un conjunto de reglas”. Los participantes pueden ser personas, que interactúan a través de estos sistemas, a través de redes de comunicación (Dellen, 2000).

2.2.2. BENEFICIOS DE LA TECNOLOGÍA WORKFLOW

Para determinar los principales beneficios de implementar la tecnología workflow, la WfMC lleva a cabo investigaciones que involucran a usuarios y proveedores de soluciones de workflow. Los resultados de una investigación en el portal de workflow patrocinado por la coalición (WfMC, 2005):

- Incremento de la eficiencia. La automatización de varios procesos de negocios resulta en la eliminación de actividades redundantes.
- Mejor control de los procesos. La automatización de los procesos de negocio se facilita a través de la implementación de los métodos de trabajo y la disponibilidad de pistas de auditoría.
- Mejora en el servicio al cliente. La consistencia en los procesos conlleva una mayor capacidad de predicción en los niveles de respuesta.
- Flexibilidad. Se posibilita el rediseño de los procesos sobre la marcha, a medida que cambian las necesidades del negocio.

- Mejores procesos de negocio. Se fomenta la agilidad y simplicidad de estos.

2.2.3. COMPONENTES DE UN WORKFLOW

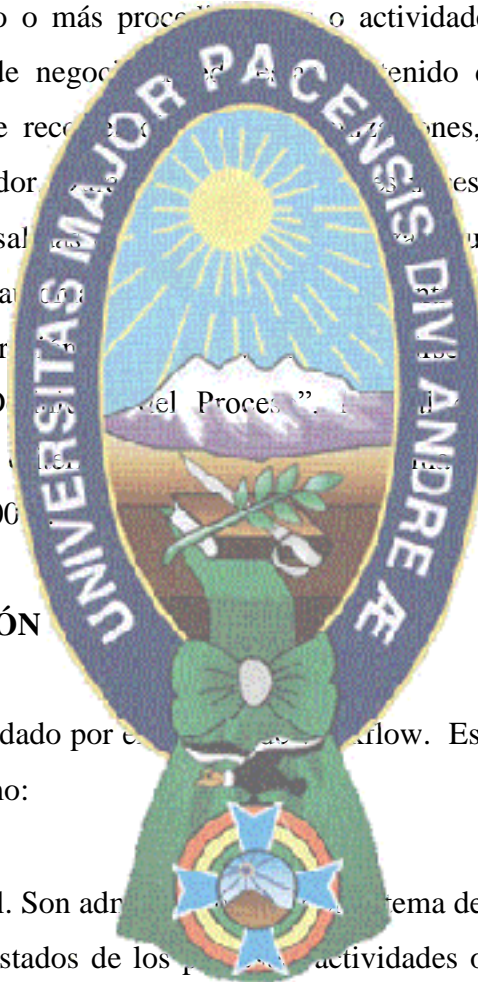
2.2.3.1. PROCESOS DE NEGOCIO

Es un conjunto de uno o más procesos o actividades enlazadas, que permiten alcanzar un objetivo de negocio. Este puede ser contenido dentro de una sola unidad organizacional o puede requerir de varias organizaciones, como sería el caso de la relación cliente-proveedor. Es necesario definir las condiciones que lo iniciaran y las salidas que se obtendrán. Su duración es variable y las actividades pueden ser administradas, entendiéndose estas últimas fuera del alcance de la administración. Para la modelación, la WfMC sugiere el término “Diagrama del Proceso”. Este consiste en un conjunto de actividades, relaciones, y condiciones de transición propia de cada actividad (Borbon & Villareal, 2006).

2.2.3.2. INFORMACIÓN

Contenido que es trasladado por el sistema de workflow. Esta se compone de datos que pueden clasificarse como:

- Datos de control. Son administrados por el sistema de workflow y utilizados para identificar los estados de los procesos, actividades o cualquier otro recurso del sistema.
- Datos relevantes al flujo. Son utilizados para determinar las condiciones de transición entre las distintas actividades. Estos se encuentran accesibles para el sistema de administración del flujo de trabajo.
- Datos aplicativos. Estos no se encuentran accesibles para el sistema de administración de flujo de trabajo, pues son específicos de la aplicación y son



relevantes únicamente para el desarrollo de las actividades del usuario o de los procesos aplicativos.

2.2.3.3. ORGANIZACIÓN

Como parte de la definición del flujo de trabajo, los procesos establecen qué personas llevan a cabo las tareas individuales, definiendo sus relaciones, las funciones que desempeñan y los roles que desempeñan. Esta información incluye aspectos normativos, tales como políticas, procedimientos y calendarios de trabajo. Un modelo organizativo representa una empresa en términos de empleados, puestos, unidades, funciones y relaciones. Un modelo organizativo es un conjunto de puestos con tareas comunes o relacionadas que constituyen una relación jerárquica (Borbon & Villareal, 2006).

2.2.4. CONCEPTOS DE UN MODELO DE FLUJO DE TRABAJO

Cuando se realiza el modelado de un flujo de trabajo, generalmente se identifican definiciones de los elementos que conforman dicho sistema. A continuación se describe cada uno de ellos (Borbon & Villareal, 2006):

- **Tareas:** Es un conjunto de actividades manejadas como una sola unidad. Son desempeñados por una persona dentro de los roles que pueden realizar dicha tarea. Las tareas se definen en el análisis del flujo de trabajo, donde se define por quienes deben ser ejecutadas.
- **Usuario:** Las tareas son realizados en un orden definido por determinadas personas (o agentes automatizados tomando el rol de las personas) basados sobre las condiciones o reglas de negocio.
- **Roles:** Cada rol se define las distintas competencias potenciales que existen en el sistema. Se definen independientemente de las personas físicas a las cuales se les

van a asignar dichos roles. Una persona puede tener más de un rol identificado.

- Rutas: Define la secuencia de pasos a seguir por los documentos (información) dentro de un sistema de Workflow. La capacidad de rutear las tareas a usuarios remoto u ocasionales es vital en una aplicación de Workflow. Se identifican dos tipos de rutas que son:

- Rutas fijas: En este tipo de documentos siguen siempre el mismo camino. Se define la ruta de la próxima etapa a seguir.
- Rutas condicionales: Este tipo de rutas depende de la evaluación de condiciones. En este momento que se pasa por el punto de decisión.
- Rutas Adicionales: Este tipo de rutas se define explícitamente cual es la siguiente etapa a seguir.

- Reglas de transición: Estas reglas determinan la navegación del documento dentro del sistema. Se va a tomar dependiendo del valor de expresiones. Las reglas de las reglas puede ser muy complicada, con múltiples reglas y excepciones.

- Datos: Son documentos, archivos, registros de la base de datos, y otros utilizados como la información para hacer a cabo el trabajo. Entre los datos manejados son:

- Datos de control: Son datos internos manejados por la lógica del sistema de Workflow.
- Datos relevantes: Son aquellos datos utilizados para determinar el ruteo de las distintas tareas del sistema.
- Datos de Aplicación: Estos datos son específicos de la aplicación, no son accedidos por la lógica de Workflow.



- **Eventos:** Un evento es una interrupción que contiene información, el mismo tiene origen y uno o más destinatarios. La información contenida en el mensaje que se produjo por el evento puede ser implícita o dada por el usuario. Los eventos pueden ser disparados voluntariamente por el usuario, o en forma implícita durante un proceso según el estado de los datos o de decisiones tomadas por el usuario, en forma implícita.

- **Plazos:** Se pueden ver ciertos tipos de plazos que se le asignan a ciertos elementos.

2.2.5. PATRONES DE DISEÑO

El departamento de "Tecnología de Manejamiento" de Technische Universiteit Eindhoven ha identificado y publicado los patrones de diseño que se encuentran en forma recurrente dentro de la industria de Workflows. Estos patrones se han convertido en un estándar, las herramientas de encargar la solución de las situaciones que estos patrones representan como forma de evaluar la capacidad de las herramientas para su capacidad para implementar en forma natural dichos patrones (Caldwell, 2007).

CLASIFICACIÓN DE PATRONES DE DISEÑO (Q-flow)

Patrones de secuencia: Son los patrones más comunes que se aplican para diseñar un Workflow. Utilizan todos los conceptos manejados por la tecnología Workflows. Se ven de manera general y son muy similares a los diagramas de flujos conocidos (Q-Flow, 2011).

- **Secuencia:** Una actividad en un proceso de Workflow es habilitada después de ser completada otra actividad en el mismo proceso.



- Separación en paralelo AND/SPLIT: Dos o más actividades de un proceso se ejecutan en paralelo, en un punto del proceso de Workflow, el hilo de control se divide en múltiples hilos de control, habilitando la ejecución de las tareas en paralelo y sin restricciones de orden entre ellas.
- Sincronización: Una actividad se ejecuta cuando dos o más hilos completan la ejecución de sus actividades. En un punto del proceso, dos o más hilos de control convergen en un solo hilo de control.
- Este paso de unión de caminos es el patrón más común de sincronización. Al ser este el patrón más común de sincronización, se porta por defecto de acuerdo al mismo, sincronizando los caminos antes de activar la siguiente actividad.
- Opción Exclusiva (XOR): En un punto del proceso, una o más de sus ramas son seleccionadas en exclusiva. El patrón de selección exclusiva de camino es el más común de selección. El patrón de selección exclusiva de camino es el más común de selección, donde es evaluada una condición y, de acuerdo al valor de la evaluación se selecciona uno u otro camino de ejecución.
- Fusión simple: Se mezclan varios hilos de control en uno solo, en determinados puntos del proceso. La fusión simple de caminos no utiliza ningún constructor específico sino que la simple unión de caminos o hilos de ejecución sobre una tarea determina que cualquiera de ellos llegue a la tarea y la active.



Patrones avanzados en paralelismo y sincronización: Tiene una mayor complejidad en su concepción, puesto que se pueden elegir múltiples rutas de manera paralela, ejecutando las tareas con diferentes roles asignados, y sincronizando la conclusión de una o más de estas rutas, para continuar con las siguientes tareas (Q-Flow, 2011).

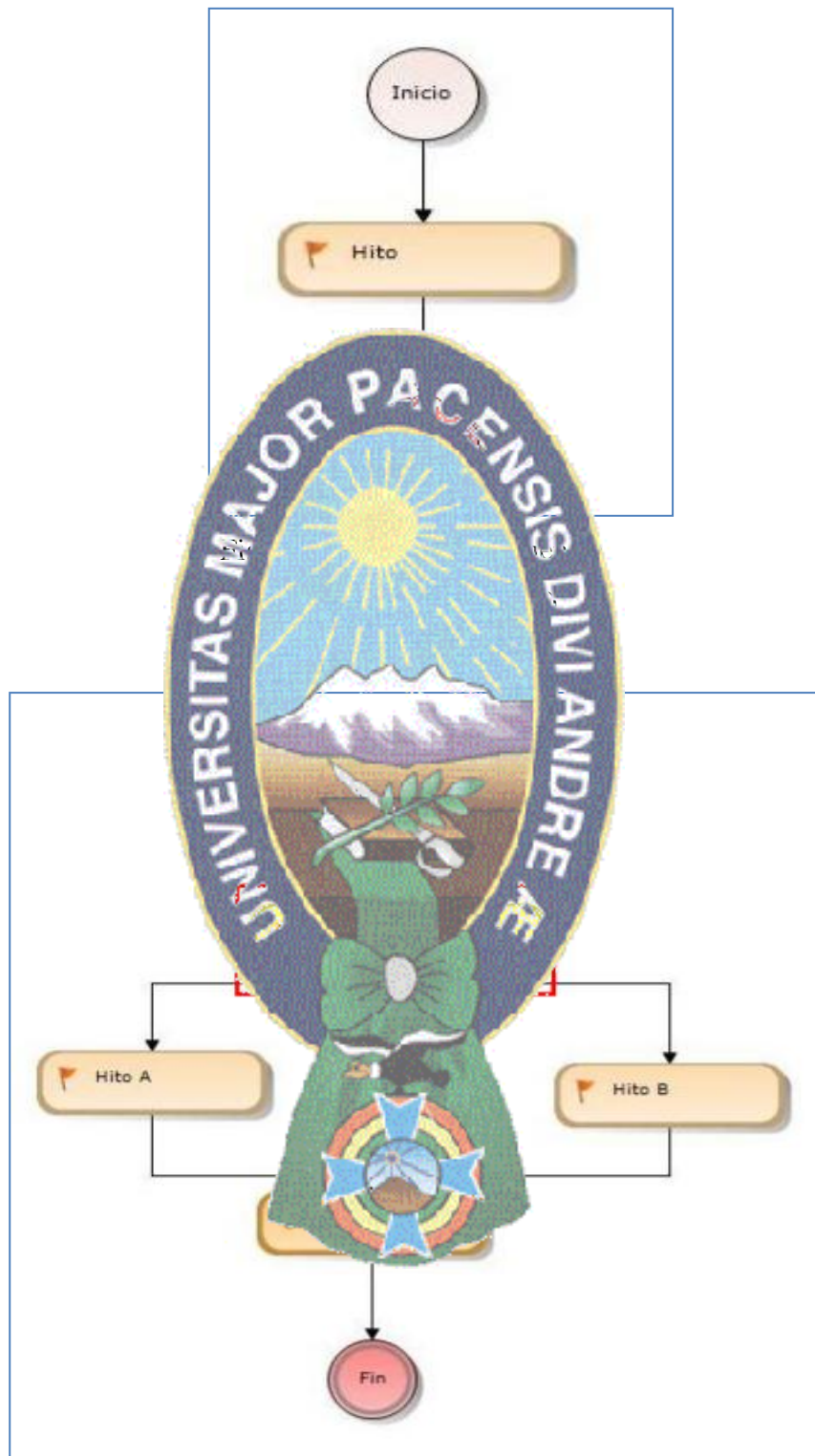


Figura 2.2. Patrón en paralelo

Fuente: (Q-Flow, 2011)

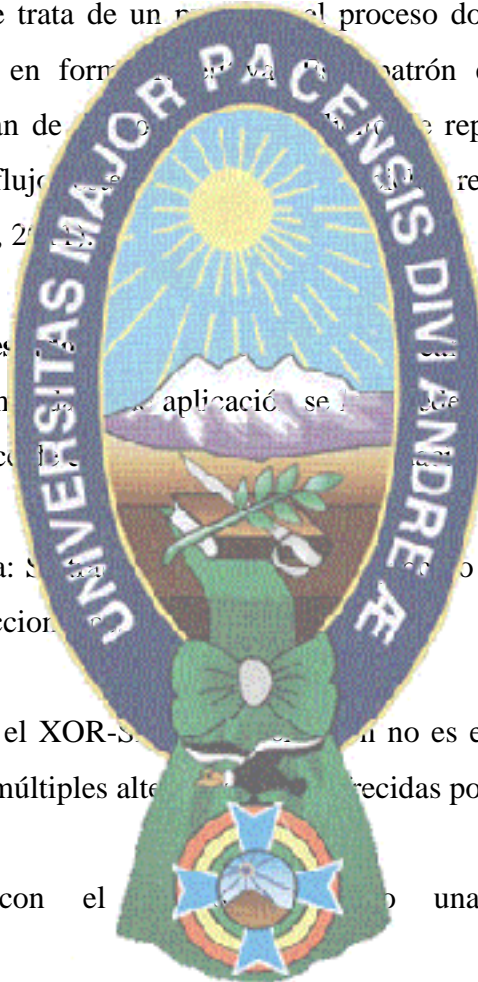
Múltiple opción (OR-SPLIT): En un punto del proceso, basado en los datos de control del proceso, uno o más caminos o ramas son relacionados (Q-Flow, 2011).

Patrones estructurales: Delimitan un orden establecido para la ejecución de las tareas o actividades a realizar por los usuarios implicados (Q-Flow, 2011).

Ciclos Arbitrarios: Se trata de un patrón del proceso donde una o más actividades pueden ser realizadas en forma repetitiva. Este patrón existe para diferenciar los Workflow que necesitan de actividades que se repitan. Haciendo analogía con los diagramas de flujo, se repiten actividades repetitivas mientras cumplan una condición (Q-Flow, 2011).

Patrones basados en estado: Se trata de un patrón del proceso que no o ruta según el estado o valor que tome uno o más de los datos de la aplicación. Se puede asignar un valor al iniciar el flujo, o durante el avance del flujo (Q-Flow, 2011).

- a) Selección diferida: Se trata de un patrón del proceso donde uno o más múltiples caminos son seleccionados.
- b) En contraste con el XOR-SPLIT, la selección no es explícita (basada en datos o decisiones) pero múltiples alternativas son ofrecidas por el entorno.
- c) En contraste con el XOR-SPLIT, una de las opciones es seleccionada.



2.3. NOTACIÓN PARA EL MODELADO DE PROCESOS DE NEGOCIO (BPMN)

Business Process Model and Notation (BPMN) es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente

diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades. BPMN proporciona un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente. De esta forma BPMN define la notación y semántica de un Diagrama de Procesos de Negocio (Modeling, 2000).

BPD es un diagrama diseñado para representar gráficamente la secuencia de todas las actividades que ocurren durante un proceso. Basado en la técnica de “Flow Chart”, incluye además toda la información necesaria para el análisis. BPD es un diagrama diseñado para ser utilizado por quienes diseñan, controlan y gestionan procesos. Dentro de un Diagrama de Negocio BPD se utiliza un conjunto de elementos gráficos que permite el fácil desarrollo de diagramas simples y complejos que manejan la complejidad inherente a los procesos (Modeling, 2000).

Esta sección introduce los conceptos básicos de procesos con BPMN. Se pretende mostrar cómo es posible modelar funciones de negocio utilizando BPMN. Los diagramas no se muestran en este paso, pero deberían ser una referencia para entender el uso de los elementos de la notación en un contexto de procesos de negocio reales (Modeling, 2000).

2.3.1 ELEMENTOS DE MODELADO EN BPMN BASICOS

Cabe recalcar que una de las direcciones del desarrollo de BPMN es crear un mecanismo simple para diagramar flujos de proceso y que a su vez maneje la complejidad inherente a los procesos del negocio. El acercamiento tomado para manejar estos dos requisitos que estaban en conflicto fue el organizar los aspectos gráficos de la notación en categorías específicas. Esto proporciona un sistema de categorías que ayuda al lector de un diagrama de BPMN a reconocer fácilmente los tipos básicos de elementos y entender el diagrama (Modeling, 2000).



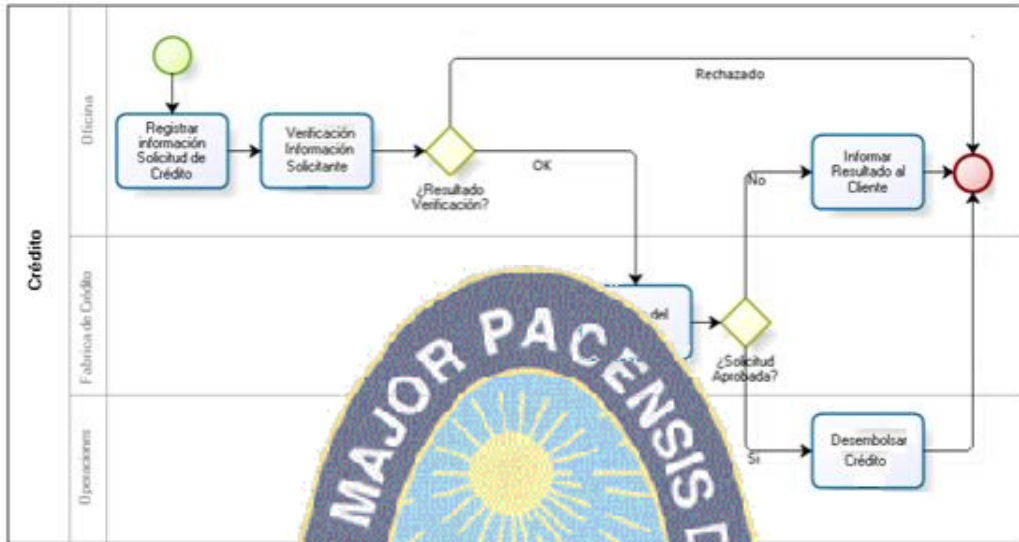


Figura 2.1. Diagrama de flujo de un proceso de crédito

fuente: (birogi, 2017)

Las cuatro categorías básicas de los elementos de un BPMN se muestran en la Tabla 2.1:

ELEMENTO	DEFINICIÓN	TRANSICIONAN	NOMBRE
		ANTERIOR	BPMN
Elementos de Flujo (FlowObjects)	Los elementos de flujo son los principales elementos que definen el flujo de los procesos de negocio.	Eventos	Events
		Actividades	Activities
		Decisión	Gateways
Conectores (ConnectingObjects)	Los objetos de conexión se conectan entre ellos a través de los conectores para crear el esqueleto básico de la estructura del proceso de negocio.	Transición	SequenceFlow
		Flujo de mensaje	MessageFlow
		Asociación	Association

Canales (Swimlane)	Los canales son mecanismos de organización de las actividades en categorías visuales separadas para ilustrar las diferentes áreas funcionales o responsables.	Área Funcional	Pools
		Fase	Lanes
Artefactos (Artifacts)	Los artefactos sirven para proveer información sobre flexibilidad de expresión en función	Objeto de Datos	Data Object
		Grupo	Group
		Asociación	Annotation

Tabla 2. Símbolos básicos

2.4. MÉTRICAS DE CALIDAD

El objetivo no es necesariamente la perfección, sino la necesaria y suficiente para cada contexto de entrega y del uso por parte de los usuarios. Es necesario comprender las necesidades reales de los usuarios con tanto detalle como sea posible (requerimientos). En el presente proyecto se escogió la funcionalidad, mantenibilidad, costo y confiabilidad para medir la calidad de software por que tanto el problema como el objetivo general del proyecto giran alrededor de estos criterios (Pressman, 2002).

2.4.1. FUNCIONALIDAD

La técnica de Punto de Función fue introducida por Albrecht y su propósito es medir al software cualificado la funcionalidad que proporciona externamente, basándose en el diseño lógico del sistema sus objetivos son los siguientes (Pressman, 2002):

- a) Medir lo que el usuario pide y lo que el usuario recibe.
- b) Medir independientemente de la tecnología utilizada en la implantación.
- c) Proporcionar una métrica de tamaño que de soporte al análisis de la calidad y la productividad.
- d) Proporcionar un medio para la estimación del software.
- e) Proporcionar un factor de medición para la comparación de distintos software.

El análisis de Punto de Fusión se basa en considerando cinco parámetros básicos externos del sistema:

- a) Número de entradas de usuario, las cuales se diferencian en las acciones.
- b) Número de salidas que proporciona al usuario informático, se refiere a informes ó planillas.
- c) Número de peticiones que se refiere como una entrada interactiva que produce una respuesta del software.
- d) Ficheros Lógicos Internos, archivo maestro lógico, los cuales son parte de una base de datos independientes.
- e) Ficheros Lógicos Externos, todas las interfaces legibles por la máquina. Como ser archivos de cinta ó disco.

Con estos parámetros, se determina el Punto de Fusión de función sin ajustar (PFsA). A este valor, se le aplica un Factor de Ajuste obtenido en base a unas valoraciones subjetivas sobre la aplicación y su entorno; es decir, las características generales del sistema. La fórmula matemática del Punto Fusión es (Pressman, 2002):

$$PF = Cuenta_total * [0.65 + (0.01 * \Sigma Fi)] \text{ Ecu No. 2.1.}$$

Donde:

Cuenta _ total: Es la suma de todas las entradas PF

ΣFi : son los valores de ajuste de complejidad.

Posteriormente de calcular el Punto Función, para comparar el nivel de funcionalidad que posee el sistema se debe utilizar la Tabla 2.2 escala de Punto Función.

ESCALA	FUNCIÓN
PF > 300	
200 > PF > 300	
100 > PF > 200	
PF < 100	

2.4.2. MANTENIBILIDAD

La mantenibilidad es el atributo que más directamente influye en los costes y necesidades de mantenimiento. Una mayor mantenibilidad menor coste de mantenimiento y viceversa. Los factores que afectan directamente a la mantenibilidad, de forma que si alguno de ellos no se satisface adecuadamente, ésta se ve afectada. Los tres más significativos (Kleinman, 2002):

- Procesos de desarrollo: El proceso de desarrollo debe formar parte del proceso de desarrollo del software. Las herramientas utilizadas deben ser lo menos intuitivas posible con el software existente.
- Documentación: En múltiples ocasiones, ni la documentación ni las especificaciones de diseño están disponibles, por lo tanto, los costes del mantenimiento se incrementa, porque el mantenimiento requerirá más tiempo para entender el diseño del software.
- Compresión de programas: La causa básica de la mayor parte de los altos costes

del mantenimiento del software es la presencia de obstáculos a la comprensión humana de los programas y sistemas existentes.

El estándar IEEE 982.1-1988 sugiere un índice de Madurez del Software (I.M.S.) que proporciona una indicación de la estabilidad de un producto, software basada en los cambios que ocurren con cada versión del producto. Se determinan las siguientes variables (Pressman, 2002):

- Mt = Número de módulos que se han agregado.
- Fe = Número de módulos que se han cambiado.
- Fa = Número de módulos que se han añadido.
- Fd = Número de módulos que se han borrado en la versión actual.

El índice de Madurez del Software se calcula de la siguiente manera:



A medida que el índice de Madurez del Software se aproxima a 1.0 el sistema es estable. Para poder interpretar el índice de Madurez del Software se utiliza la Tabla 2.3.

ESCALA	ESTADO
75% <= IMS <= 100%	Alto
50% <= IMS <= 75%	Buena
25% <= IMS <= 50%	Suficiente
0% <= IMS <= 25%	Deficiente

Tabla 2.3 Escala de Mantenibilidad

Fuente: (Pressman, 2002)

2.4.3. CONFIABILIDAD

Se pretende que el software funcione de forma consistente y correcta durante largos períodos de tiempo, que esté disponible cuando se lo necesita, y que sea reparado rápida y fácilmente si acaso falla. Formalmente se dice que la confiabilidad del software es la probabilidad de que un sistema opere sin fallas bajo determinadas condiciones para un intervalo de tiempo dado. Para estudiar una variable aleatoria se utiliza una distribución exponencial encausada en pruebas de confiabilidad que deriva de una fusión de cifras de falla, el número de fallas en el tiempo (Pressman, 2002):

Este modelo se utiliza para probar un sistema a fin de satisfacer una meta de confiabilidad. El modelo requiere tres entradas: el número de pruebas promedio de fallas, el número total de fallas observada en las pruebas (fallas por hora de ejecución de pruebas hasta la última falla (hora de ejecución de pruebas hasta la última falla) y el número de las horas necesarias de prueba para cero fallas es (Pressman, 2002):

$$\frac{\ln[(fallas) / (0,5 + fallas)]}{\ln[(0,5 + fallas) / (0,5 + fallas + fallas)]}$$

La portabilidad es la facilidad con la que es llevado de un entorno a otro. Este criterio se subdivide en: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio. La portabilidad viene dada por la medida de la sub característica de facilidad de instalación, que se responde con la siguiente pregunta: ¿El usuario o quien mantiene el software fácilmente instala el software en un ambiente operacional?, teniendo la siguiente relación (Pressman, 2002). Ver Ecuación 2.5.

$$X = A / B$$

Donde: A = Número de casos de éxito de la operación por parte del usuario. B = Número total de operaciones de instalación que realizó el usuario. Para saber la portabilidad de un producto software se verifica en la Tabla 2.4 (Pressman, 2002).

ESCALA	PORTABILIDAD
75% <= X <= 100%	
50% <= X <= 75 %	
25% <= X <= 50 %	
0% <= X <= 25 %	

2.5. GESTIÓN DE SEGURIDAD

La información es un recurso más activo, tiene valor para una organización y por lo tanto debe estar protegida. El estándar internacional de alto nivel para la seguridad de la información ISO 17799, fue publicado por la International Organization for Standardization (ISO) en diciembre de 2000 con el objetivo de proporcionar un marco de seguridad sobre el cual trabajen las organizaciones. El estándar ISO 17799, se define como una guía en la implementación del sistema de administración de la seguridad de la información, se orienta a preservar los siguientes aspectos de la seguridad informática (Nuñez, 2005):

- a) Confidencialidad: Asegurar que únicamente personal autorizado tenga acceso a la información.
- b) Integridad: Garantizar que la información no sea alterada, eliminada o destruida por entidades no autorizadas.
- c) Disponibilidad: Asegurar que los usuarios autorizados tendrán acceso a la información cuando la requieran.

El éxito de la implementación del estándar de seguridad ISO 17799 requiere de una serie de procedimientos donde, inicialmente, el análisis de riesgo del software identifica los activos de la información y las amenazas a las cuales se encuentra expuesta. El análisis de riesgo del software guía en la correcta selección de los controles que se apliquen a la organización; este proceso se conoce como la jerga del estándar, que es la definición de los controles que se aplican a la organización con objeto de proporcionar niveles prácticos de seguridad de la información y el cumplimiento de los mismos. A continuación, se describe cada una de las áreas de seguridad con el objeto de esclarecer los objetivos de estos controles.

- Políticas de seguridad: El apoyo gerencial para brindar seguridad de la información requiere establecer una política clara y demostrar apoyo y compromiso con respecto a la seguridad de la información, mediante la formulación de una política de seguridad.
- Seguridad organizacional: La seguridad de la información dentro de la organización. Se refiere a la infraestructura de seguridad de la información, que incluye a los proveedores de servicios por parte de terceros (Se refiere cuando una empresa o institución contrata servicios que no forman parte de la empresa es decir terceras personas).
- Clasificación y control de activos: El análisis de riesgos generará el inventario de activos que deberá ser administrado con base en ciertos criterios de clasificación y etiquetado de los activos. Es decir, los activos deben ser etiquetados de acuerdo a un nivel de confidencialidad.
- Seguridad del personal: Reducir los riesgos de error humano, robo, fraude o uso inadecuado de instalaciones. Las responsabilidades en materia de seguridad deben ser explicitadas en la etapa de reclutamiento, incluidas en los contratos y monitoreadas durante el desempeño del individuo como empleado. Los candidatos a ocupar los puestos de trabajo deben ser adecuadamente seleccionados, especialmente si se trata de tareas críticas. Todos los empleados y

usuarios externos de las instalaciones de procesamiento de información deben firmar un acuerdo de confidencialidad (no revelación).

- Seguridad física y de entorno: Impedir accesos no autorizados, daños e interferencia a las sedes e información de la empresa. Las instalaciones de procesamiento de información crítica o sensible de la empresa deben estar ubicadas en áreas protegidas y resguardadas. Deben estar físicamente protegidas contra accesos no autorizados e intrusiones. La protección provista debe ser proporcional a los riesgos.
- Comunicación y administración de configuración: Integrar los procedimientos de operación de la organización con los procedimientos de controles de seguridad documentados, que se aplican en la configuración de los equipos, manejo de configuración de sistemas, hasta el control de códigos fuente. En esta sub sección se definen los siguientes criterios: protección contra software malicioso, mantenimiento, administración y seguridad de los medios de almacenamiento de información y software.
- Control de acceso: Los procesos de negocio deben ser controlados sobre la información. Se deben tener en cuenta las políticas de difusión y conservación de la información. En esta sub sección se definen los siguientes criterios: procedimientos de negocio para el control de accesos, administración de usuarios, responsabilidades del usuario, control de acceso a la red, acceso al sistema operativo, control de acceso a las aplicaciones, acceso y uso de los sistemas y computación móvil y trabajo remoto.
- Desarrollo de sistemas y mantenimiento: La organización debe disponer de procedimientos que garanticen la calidad y seguridad de los sistemas desarrollados para tareas específicas de la organización.
- Continuidad de las operaciones de la organización: El sistema de administración de la seguridad debe integrar los procedimientos de recuperación en caso de contingencias, los cuales debe ser revisados de manera constante y puestos a



prueba con la finalidad de determinar las limitaciones de los mismos.

- Requerimientos legales: Impedir infracciones y violaciones de las leyes del derecho civil y penal; de las obligaciones establecidas por leyes, estatutos, normas, reglamentos ó contratos; y de los requisitos de seguridad.

2.5.1. COSTO BENEFICIO

Se debe utilizar el análisis de costo beneficio para comparar los costos y beneficios de las diferentes decisiones. Un análisis de costo beneficio por sí solo, no es una guía clara para tomar una buena decisión. Los factores que deben tomar en cuenta.

El modelo original COCOMO (Software Cost Model) se publicó por primera vez en 1981 por Barry Boehm. Este modelo se desarrolló para el desarrollo de software de aquel momento. Es decir, el modelo COCOMO se basaba en sistemas basados en el paradigma estructurado y en un lenguaje de programación como el Pascal.

En el modelo COCOMO el esfuerzo de software es expresado en meses-personas, donde un mes-persona es el tiempo que tarda una persona en un mes dentro de un proyecto de desarrollo de software. El número de meses-persona es diferente que la duración del proyecto. El esfuerzo del proyecto se obtiene de la diferencia entre la fecha de inicio y fin del proyecto, multiplicado por el número de personas que participan en un proyecto. Por ejemplo, si un proyecto de cuatro meses donde participan un analista programador y un programador en los dos últimos meses, se obtendría los siguientes datos: Duración del proyecto = 4 meses - Número de personas = 2 personas - Esfuerzo del proyecto = 6 meses-persona.

El modelo COCOMO II es usado para:

- a) Hacer inversiones ó tomar otras decisiones financieras que involucren esfuerzo de desarrollo de software.



- b) Establecer presupuestos y calendarios de proyectos como base para la planificación y el control.
- c) Decidir ó negociar entre los costos del software, calendario, funcionalidad, rendimiento ó factores de calidad.
- d) Tomar decisiones sobre el costo de software y el manejo de riesgo del calendario.
- e) Decidir qué partes del sistema se desarrollarán, re-utilizarán, subcontratarán ó comprarán.
- f) Tomar decisiones sobre qué partes se van a modificar ó incluso dejar fuera.

Los objetivos del modelo son:

- a) Promover estimaciones de costo y calendario tanto para los proyectos actuales como futuros.
- b) Proporcionar definiciones para comprender de las entradas, salidas del modelo.
- c) Proporcionar un mecanismo que permite distribuir los recursos necesarios para el desarrollo de un proyecto de software.

El esfuerzo se calcula con la siguiente fórmula:

$$\text{Esfuerzo} = A \times \text{Tamaño del Proyecto} \quad \text{Ecu. No. 2.6.}$$

$$\text{Donde } E = B + 0.01 \times (X \text{ Factor}) \quad \text{Ecu. No. 2.7.}$$

Los valores se muestran en la Tabla 2.5.

SÍMBOLO	DESCRIPCIÓN
A	Coeficiente de esfuerzo, calibrado en 2.94

B	Base del exponente de escalamiento, calibrado en 0.91
Tamaño	Medida del software en líneas de código fuente
E	Exponente de escalamiento de tipo de proyecto
Multiplicador	17 multiplicadores de esfuerzo
Factor	5 factores de escala del proyecto
Esfuerzo	Esfuerzo del proyecto en meses-persona

Tabla 2.5 Valores de calibrado del modelo COCOMO II

Fuente: Modelo COCOMO II

La duración del proyecto se calcula de la siguiente manera:

$$TDEV = \frac{E}{B} \cdot \left(\sum_{i=1}^n F_i \right)^D \cdot SCDE \quad \text{Ecuación No. 2.8}$$

Donde F es el esfuerzo del proyecto en meses-persona. Ecuación No. 2.9

Los valores de calibrado del modelo COCOMO II se detallan en la siguiente tabla. Tabla 2.6.

SÍMBOLO	DESCRIPCIÓN
B	Base del exponente de escalamiento, calibrado en 0.91
C	Coefficiente de planificación, calibrado en 3.67
D	Exponente de base de planificación para planificación calibrado en 0.28
E	Exponente de escalamiento de tipo de proyecto, utilizado en la ecuación de esfuerzo
SCDE %	Porcentaje de planificación con tiempos comprendidos
F	Exponente de escalamiento para planificación

Tabla 2.6 Valores de calibrado del modelo COCOMO II

Fuente: Modelo COCOMO II

2.5.2. FACTORES DE ESCALA

Los factores de escala se utilizan para el cálculo del exponente en la Ecuación No.2.6.

- Precedentes: Refleja la experiencia previa de la organización desarrolladora con respecto al tipo de proyectos. Muy bajo significa sin experiencia previa. Extra alto significa que la organización está completamente familiarizada con el dominio de la aplicación a desarrollarse.
- Flexibilidad de desarrollo: Refleja el grado de flexibilidad del proceso de desarrollo. Muy bajo significa que se sigue un proceso prescrito por ejemplo: uso de determinadas técnicas, procesos. Extra alto significa que el cliente establece sólo metas y los desarrolladores toman decisiones sobre cómo llevar a cabo el proceso en cada momento.
- Arquitectura/Solución: Refleja el nivel de detalle del análisis de riesgo que se debe llevar a cabo. Muy bajo significa poco análisis. Extra alto significa que es independiente, completo y muy detallado.
- Cohesión del equipo: Refleja el conocimiento de los miembros del equipo de desarrollo. Muy bajo significa que las interacciones podrían ser problemáticas. Extra alto significa un equipo integrado y efectivo sin problemas de comunicación.
- Madurez del proceso: Refleja el nivel de madurez de la organización.

2.6. ARQUITECTURA CLIENTE-SERVIDOR

Este es un modelo de proceso en el que los programas se reparten entre programas que se ejecutan en el servidor y otros que se ejecutan en el trabajo del usuario. En una red, cualquier equipo puede ser el servidor. El cliente es la entidad que solicita la realización de una tarea, el servidor es quien la ejecuta en nombre del cliente. Este es el caso de aplicaciones de acceso a bases de datos, en las cuales las estaciones ejecutan las tareas del interfaz de usuario (pantallas de entrada de datos de consultas, listados, etc.) y el servidor realiza las actualizaciones y recuperaciones de datos en la base de datos. En este tipo de redes, las actualizaciones y recuperaciones no se comunican entre sí. Las ventajas de este modelo incluyen (Gomez Perez, 2010):

Incremento en la productividad - Control o reducción de costos al compartir recursos - Facilidad de administración, al concentrarse el trabajo en los servidores - Facilidad de adaptación (Gomez Perez, 2010).

2.6.1. PROGRAMACIÓN EN TRES CAPAS

La programación en tres capas es un tipo de programación en la que el objetivo primordial es la separación de las capas de programación, la lógica de diseño, un ejemplo básico de esto es separar la capa de presentación al usuario. La ventaja principal de este tipo de programación es que puede llevar a cabo en varios niveles y en caso de algún cambio se puede hacer sin tener que revisar todo el código mezclado (Arenas, 2011).

En el diseño de sistemas se utilizan tres tipos de arquitecturas multinivel o programación por capas. En la arquitectura de tres niveles se le confía una misión simple, lo que permite el desarrollo de aplicaciones (que puedan ampliarse con facilidad en caso de que las necesidades cambien) (Arenas, 2011).

El diseño actualmente es el diseño de tres capas (Arenas, 2011).

- Capa de presentación: es la capa que se comunica con el usuario ("capa de usuario") presenta el sistema al usuario, comunica con el usuario, captura la información del usuario dando un mínimo de proceso (normalmente un chequeo previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.
- Capa de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben de cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de

datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

- Capa de datos: es donde residen los datos. Está formada por uno o más gestores de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.6.2. HERRAMIENTA PARA LA IMPLEMENTACIÓN

2.6.2.1. .NET FRAMEWORK

.NET framework constituye la plataforma principal sobre el que se asienta Microsoft .NET. Es la pieza fundamental de todo modelo de trabajo, ya que proporciona las herramientas necesarias para facilitar en su labor habitual de desarrollo. .NET framework facilita el desarrollo de aplicaciones a través del uso de un conjunto de la herramienta de desarrollo que pueden agruparse en tres bloques principales, el Common Language Runtime (CLR) la jerarquía de clases de .NET framework, y el motor de generación de interfaz de usuario. .NET framework ofrece interfaces para la web o para el tradicional entorno Windows, así como para ambos entornos (Gallegos, 2005).

2.6.2.2. ASP.NET

Es un conjunto de tecnologías de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es tecnología sucesora de la tecnología Active Server Pages (ASP) (Gallegos, 2005).

Cualquier persona que esté familiarizado con el desarrollo de aplicaciones Web sabrá que el desarrollo Web no es una tarea simple, ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por

un gran número de lenguajes, herramientas de desarrollo, la aplicación Web es una mezcla de varios lenguajes de etiquetas, un gran uso de scripts y plataformas de servidor (Gallegos, 2005).

Microsoft introdujo esta tecnología llamada Active Server Pages en diciembre de 1996, por lo que no es nada nueva. Esta parte del internet Information Server (US) desde la versión 3.0 es una tecnología de páginas que permiten el uso de diferentes scripts y componentes en un conjunto de páginas HTML para mostrar páginas generadas dinámicamente. La definición de las páginas es que las "Active Server Pages son un ambiente de aplicación que puede combinar código HTML, scripts y componentes Active Server Pages para crear aplicaciones dinámicas y poderosas para la Web (Gallegos, 2005).

2.6.2.3 SQL SERVER

SQL Server 2008 R2 es un sistema de gestión de base de datos. Incluye componentes múltiples y una plataforma comprensiva para programas de la empresa. SQL Server es un sistema de base de datos relacionales (SGBD) basada en el lenguaje de consulta estructurado. Está disponible en una disposición de, muchos usuarios grandes cantidades de datos de una sola máquina. El motor de base de datos es el componente principal de SQL Server que proporciona almacenamiento de datos, recuperación, y servicios de modificación que pueden ser utilizados en soluciones personales hasta el nivel empresa (Puramagia, 2011).



2.7. DISEÑO METODOLÓGICO

2.7.1. METODOLOGÍA XP

En cuanto a la obtención y desarrollo del sistema de información, se basará en una de las metodologías ágiles como es la Programación Extrema denotado por el acrónimo de XP

que se adecúa mejor en la consecución del producto final. Esta metodología tiene los siguientes manifiestos (Joskowicz, 2008):

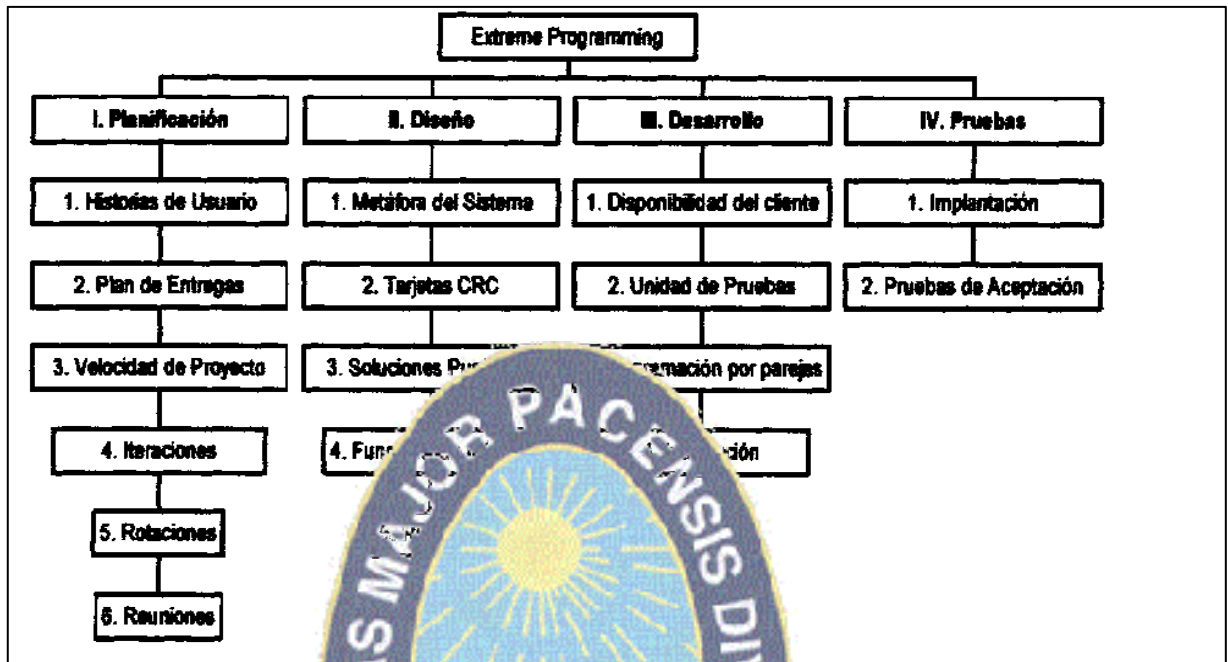
- En el manifiesto ágil se valora al individuo y a las interacciones del equipo de desarrollo sobre el proceso y las herramientas, es decir no necesariamente el equipo debe estar agrupado por desarrolladores brillantes, sino por desarrolladores que se adapten al trabajo en equipo.
- Desarrollar software que funcione. Evitar seguir una amplia documentación.
- La colaboración con el cliente debe ser una condición de un contrato.
- Responder a los cambios. Evitar tener un plan.

"XP se basa en realimentación constante entre el cliente y el equipo de desarrollo, comunicación fluida e iteraciones cortas que permitan la simplicidad en las soluciones implementadas y con el fin de enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos muy precisos y muy cambiantes, y donde existe un alto riesgo de fracaso".

2.7.1.1 FASES DE LA METODOLOGÍA XP

"Las fases de la metodología XP son: Planificación, diseño, Desarrollo y Pruebas. En esta sección se realizara una breve explicación de cada una de estas fases" (Newkirk & Martin, 2001).





1ra. FASE PLANIFICACIÓN

Historias del usuario.- El primer paso de un proyecto que siga la metodología X.P es definir las historias de usuario. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje sencillo. No se debe hacer mucho hincapié en los detalles; no se debe hablar ni de posibles estructuras de implementación ni de diseños de base de datos adecuados, etc. Se deben especificar los tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario (Newkirk & Martin, 2001).

Plan de entregas.- Después de tener ya definidas las historias de usuario es necesario crear un plan de publicaciones, en inglés "Reléase plan", es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las

historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa (Newkirk & Martin, 2001).

Velocidad del proyecto.- La velocidad del proyecto es una medida que representa la rapidez con la que se desarrolla el proyecto. Usando la velocidad del proyecto controlaremos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración (Newkirk & Martin, 2001).

Iteraciones.- Todo proyecto de desarrollo de software se ha de dividir en iteraciones de aproximadamente 3 semanas. Al inicio de cada iteración los clientes deben seleccionar las historias de usuario que serán implementadas en esa iteración. Este proceso se llama "Reléase planning" que serán implementadas divididas en tareas que se asignarán a los programadores (Newkirk & Martin, 2001).



Figura 2.5 Ciclo de vida de eXtremeProgramming

Fuente: (Newkirk & Martin, 2001)

Rotaciones.- La metodología X.P. aconseja la programación en parejas pues incrementa la productividad y la calidad del software desarrollado. El trabajo en pareja involucra a

dos programadores trabajando en el mismo equipo; mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. De esta forma se consigue un código y diseño con gran calidad (Newkirk & Martin, 2001).

Reuniones.- Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma clara. Las reuniones tienen que ser fluidas y todo el mundo tiene que tener voz y voto (Newkirk & Martin, 2001).

2da. FASE DISEÑO

Metáforas del sistema.- Una metáfora es una comparación que ayuda a la correcta especificación de los nombres de métodos y clases. Una metáfora ayuda a visualizar el sistema y facilitará sus posteriores ampliaciones y la reutilización del código (Newkirk & Martin, 2001).

Tarjetas C.R.C. El uso de tarjetas C.R.C. (Context, Responsibilities and Collaboration) permiten al programador desarrollar un sistema de desarrollo orientado a objetos olvidándose de los malos hábitos de programación procedimental clásica. Las tarjetas C.R.C representan objetos; la parte superior de la tarjeta es el objeto se puede escribir en la parte de arriba de la tarjeta, en la parte inferior izquierda se pueden escribir las responsabilidades u objetivos que debe tener el objeto y a la derecha, las clases que colaboran con cada responsabilidad (Newkirk & Martin, 2001).

Diseños puntuales.- La metodología de diseño puntual sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar (Newkirk & Martin, 2001).

Funcionalidad mínima.- Si surgen problemas potenciales durante el diseño, X.P sugiere utilizar una pareja de desarrolladores para que investiguen y reduzcan al máximo el



riesgo que supone ese problema. Funcionalidad extra: Nunca se debe añadir funcionalidad extra al programa aunque se piense que en un futuro será utilizada. Sólo el 10% de la misma es utilizada, lo que implica que el desarrollo de funcionalidad extra es un desperdicio de tiempo y recursos (Newkirk & Martin, 2001).

Reciclaje.- Es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Reciclar supone eliminar de nuevo estos códigos para procurar optimizar su funcionamiento. Es un error eliminar los códigos ya creados que contienen funcionalidades que no serán utilizadas. Esto es un error porque puede generar código completamente nuevo y no diseñado; por este motivo, es necesario reciclar cuando sea necesario (Newkirk & Martin, 2001).

3ra. FASE DE DESARROLLO

Disponibilidad del cliente.- En esta fase de desarrollo, el cliente es una parte más del equipo de desarrollo. El cliente debe estar presente en las distintas fases de X.P. A la hora de codificar un módulo de usuario, la presencia del cliente es aún más necesaria. No olvidemos que los clientes son los que definen las necesidades de usuario y negocian los tiempos en los que serán implementados. El cliente debe estar presente en el desarrollo de cada historia de usuario. El cliente debe especificar detalladamente lo que esta hará y también tendrá que estar presente cuando se realicen los tests para asegurarse de que la historia implementada cumple la funcionalidad especificada (Newkirk & Martin, 2001).

Unidad de Pruebas.- La codificación debe ser realizada atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Se puede dividir la funcionalidad que debe cumplir una tarea a programar en pequeñas unidades, de esta forma se crearán primero los test para cada unidad y a continuación se desarrollará dicha



unidad, así poco a poco conseguiremos un desarrollo que cumpla todos los requisitos especificados (Newkirk & Martin, 2001).

Programación por parejas.- Como ya se comentó anteriormente, X.P opta por la programación en pareja ya que permite un código más eficiente y con una gran calidad. X.P sugiere un modelo de trabajo usando repositorios de código dónde las parejas de programadores publican cada pocas horas los códigos implementados y corregidos junto a los test que deben pasar. De esta manera los programadores que necesiten códigos ajenos trabajarán siempre con las últimas versiones. X.P también propone permitir al resto de los programadores que no son suyos no supone ningún riesgo ya que para que un código publicado en el repositorio tiene que pasar los test de funcionamiento (Newkirk & Martin, 2001).

Integración.- La optimización de código siempre debe ir mejor para el final. Hay que hacer que funcione y que se pueda optimizar. A la hora de codificar no seguimos la especificación de requisitos hasta el test de funcionamiento con entornos de desarrollo a medida que vamos obteniendo los test los obtendremos de la especificación de requisitos. Las pruebas que deben pasar las distintas funcionalidades del programa se modifican pensando en las pruebas que debe pasar cada funcionalidad (Newkirk & Martin, 2001).

4ta FASE PRUEBAS

Implantación.- Uno de los pilares de la metodología X.P es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando (Newkirk & Martin, 2001).

Pruebas de aceptación

El uso de los test en X.P es el siguiente:

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

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear test que no dependan del futuro código, de esta forma aseguraremos la independencia del código que evalúa. Como se comentó anteriormente los distintos test se ejecutan en un entorno de código acompañados del código que verifican. Ninguno de los test se ejecuta en el repositorio sin que haya pasado su test de funcionamiento. Esto permite el uso colectivo del código (explicado en el apartado 2.1.2).



Figura 2.6 Árbol de requisitos de Calidad propuesta de Olsina

Fuente: (Newkirk & Martin, 2001)

Los pasos para evaluar un sitio Web son las siguientes (Newkirk & Martin, 2001):

- Definiendo el Dominio y Ente para la Evaluación de la Calidad
- Definiendo Metas de Evaluación y Seleccionando el Perfil de Usuario
- Especificando Requerimientos de Calidad para artefactos Web
- Definiendo Criterios Elementales y Complementando Procedimientos de Medición (también llamado Determinando la Estructura de Calidad Elemental)
- Definiendo las Estructuras de Evaluación y Complementando la Evaluación de Calidad Global
- Analizando y comparando los resultados de la Evaluación de Calidad Global.
- En el proceso "Definiendo Metas de Evaluación y Seleccionando el Perfil de Usuario", los tomadores de decisiones deben definir y seleccionar el dominio de la aplicación a evaluar y definir los "Requisitos de Calidad" que pueden intervenir en procesos de evaluación de artefactos Web, métodos, y recursos.

En el paso, "Definiendo Metas de Evaluación y Seleccionando el Perfil de Usuario", los evaluadores deben definir y seleccionar el dominio de la aplicación a evaluar y definir los "Requisitos de Calidad" que pueden intervenir en procesos de evaluación de artefactos Web, métodos, y recursos. Ellos pueden evaluar un proyecto de sistema de información Web en la fase operativa. Pueden evaluar la calidad de un conjunto de características de parte de un sistema, o de un sistema completo. Los resultados de la evaluación de las características y sus preferencias de calidad global de sistemas completos podrían ser utilizados para comprender, mejorar, controlar y evaluar la calidad de artefactos Web. En la tarea "Especificando Requerimientos de Calidad para artefactos Web", los evaluadores deben acordar y especificar las características, subcaracterísticas y atributos de calidad agrupándolas en un árbol de requerimientos. Respecto de las características de calidad de más alto nivel, se sigue la misma clasificación conceptual que la prescrita en el estándar ISO/IEC 9126 (Abud Figueroa, 2000), y la casi idéntica clasificación dada por IEEE Std 1061 (Abud Figueroa, 2000). Estas características de alto nivel son: usabilidad, funcionalidad, confiabilidad, eficiencia, portabilidad, y mantenibilidad

En el proceso "Definiendo Criterios Elementales e Implementando Procedimientos de Medición", los evaluadores deben definir una base de criterios para la evaluación elemental; realizar el proceso de medición, y puntaje elemental. Un criterio de evaluación elemental declara y especifica cómo medir atributos cuantificables. Para cada variable medida $X_i, i = 1 \dots n$ se define una función que representa al criterio elemental. Esta función es una correspondencia (mapeo) de los valores computados a partir del dominio empírico a un dominio numérico, y la denominamos preferencia de calidad elemental $f_i(x_i)$. Cada criterio elemental cae en uno de los tres niveles de aceptabilidad, es decir, excelente (desde 0 hasta 40%), marginal (desde 40 a 60%), y satisfactorio (desde 60 hasta 100%). El análisis de los puntajes cobra más importancia, cuando se realiza el análisis de la evaluación. (Pressman, 2002)

En el proceso "Definiendo Estructuras de Agregación e Implementando la Evaluación de Calidad Global" se debe establecer estructuras de agregación de preferencias elementales para obtener la preferencia de calidad global. Posteriormente, se debe indicar el método de agregación de modo de obtener un indicador de calidad global. En el paso "Analizando y comparando los Resultados Parciales" los evaluadores analizan, evalúan y comparan resultados parciales y globales cuando las metas y la vista de usuario establecidas. (Pressman, 2002)

2.8. LENGUAJE UNIFICADO DE MODELADO (UML)

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos

concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. (Hernández Orallo, 2011)

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se aplica en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software. (Hernández Orallo, 2011)

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado. En programación estructurada se diagrama la realidad de una utilización en un requerimiento de programación estructurada, es una forma de programar con una orientación a procedimientos. Sin embargo, la programación orientada a objetos viene siendo el paradigma de programación de UML, pero no por eso se toma UML sólo para lenguaje de programación. (Hernández Orallo, 2011)

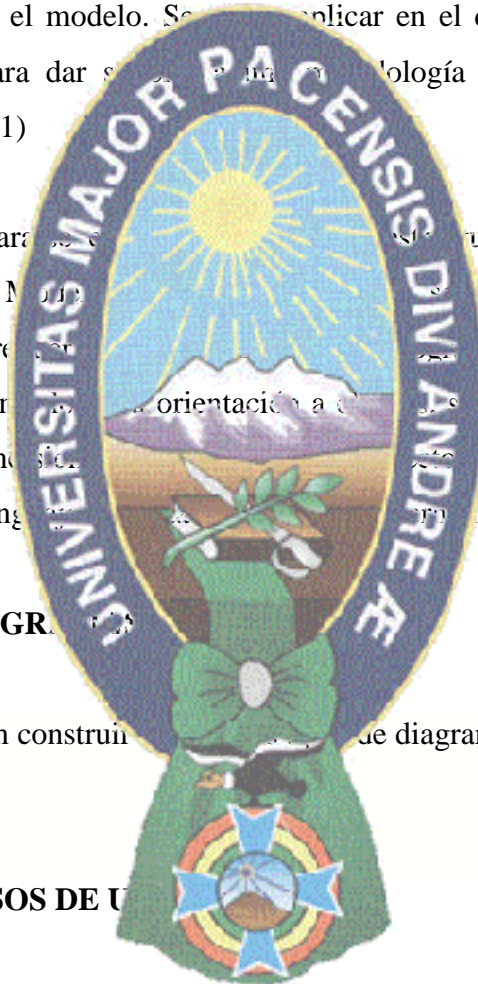
2.8.1. TIPOS DE DIAGRAMAS

Usando UML se pueden construir diferentes tipos de diagramas. Vamos a citar algunos: (UML, 2013)

DIAGRAMA DE CASOS DE USO

Se emplean para visualizar el comportamiento de un sistema, un subsistema o una clase, de forma que los usuarios puedan comprender cómo utilizar ese elemento y de forma que los desarrolladores puedan implementarlo. (UML, 2013)

Muestran un conjunto de casos de uso, actores y sus relaciones, estas pueden ser relaciones de inclusión o extensión. (UML, 2013)



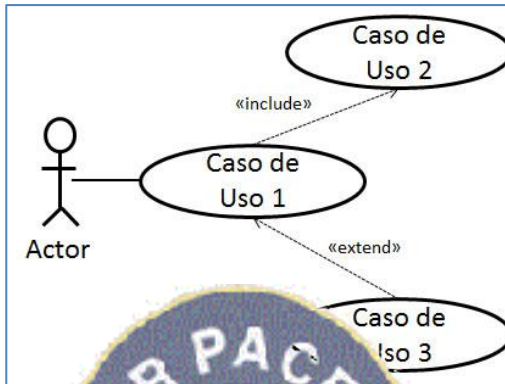


Figura 2.7 Ejemplo de Casos de Uso

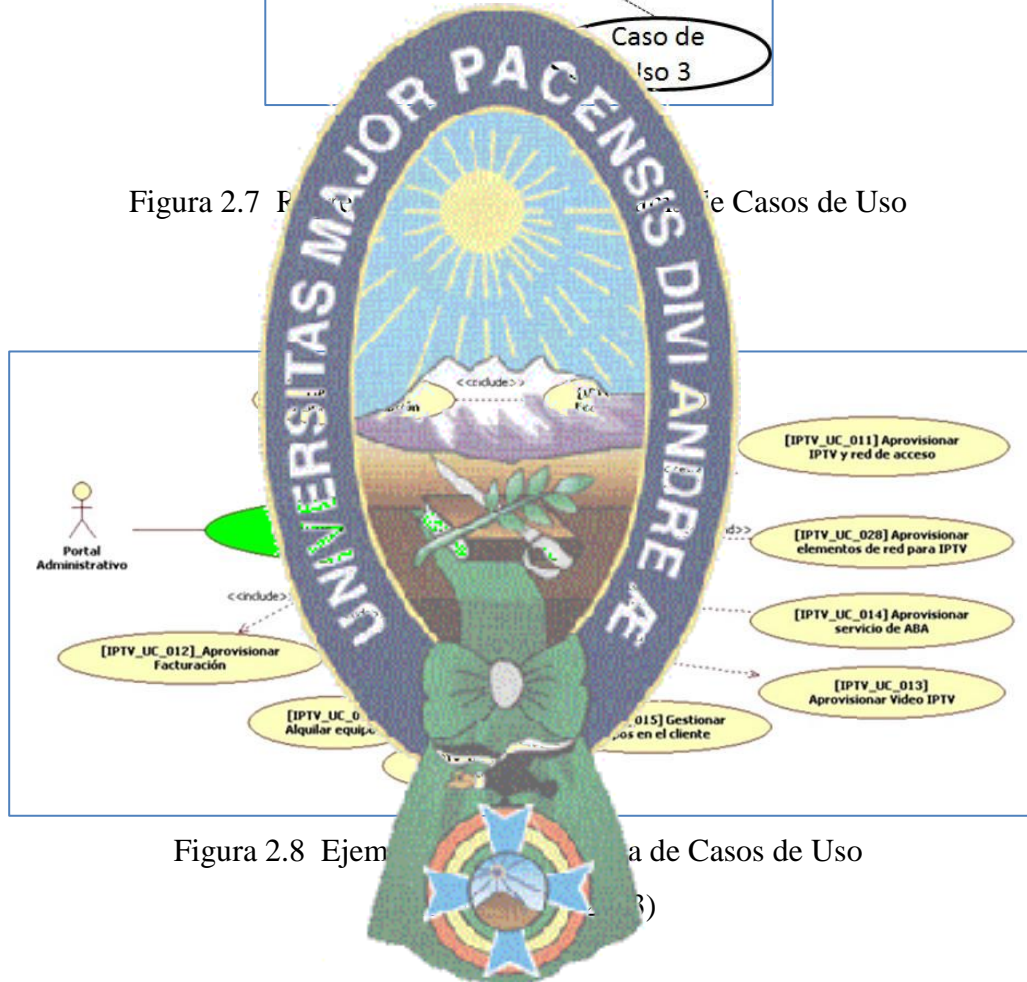


Figura 2.8 Ejemplo de Casos de Uso

DIAGRAMA DE CLASES

Muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista de diseño estática o la vista de procesos estática (sí incluyen clases activas). (UML, 2013)

Representación de una Clase

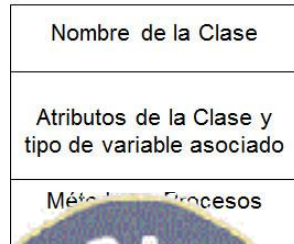


Figura 2.9 Representación de una Clase

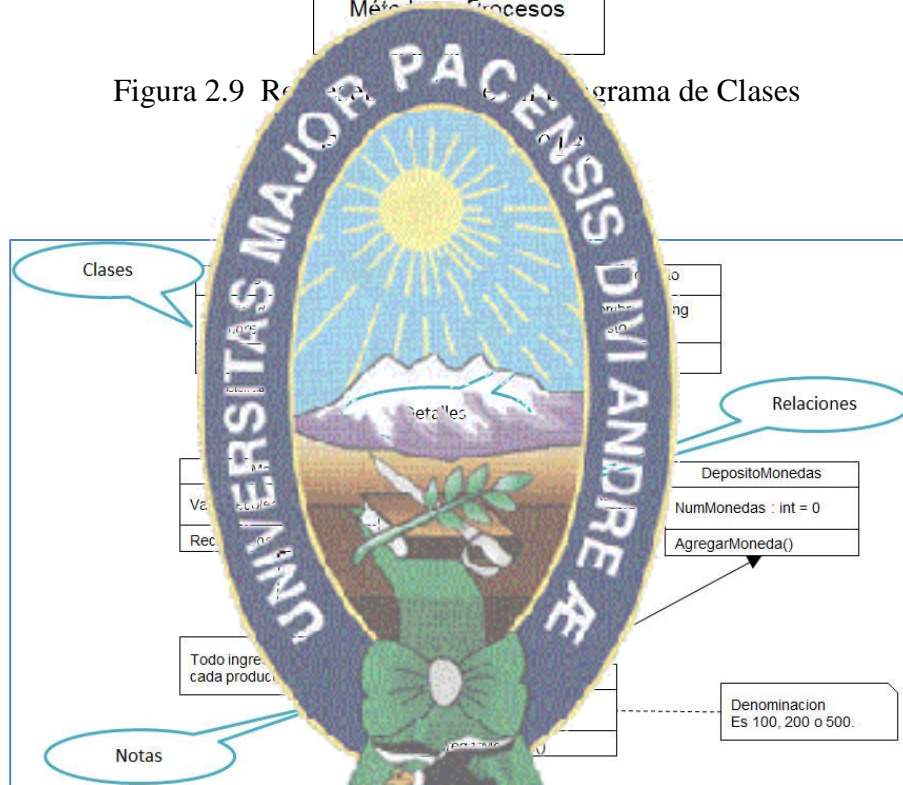


Figura 2.10 Diagrama de Clases

DIAGRAMA DE PAQUETES

Muestra cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema. (UML, 2013)

Sus elementos son:

- **Paquetes:** Agrupación de elementos, que pueden ser
- **Dependencias:** Indican que un elemento de un paquete requiere a otro paquete distinto

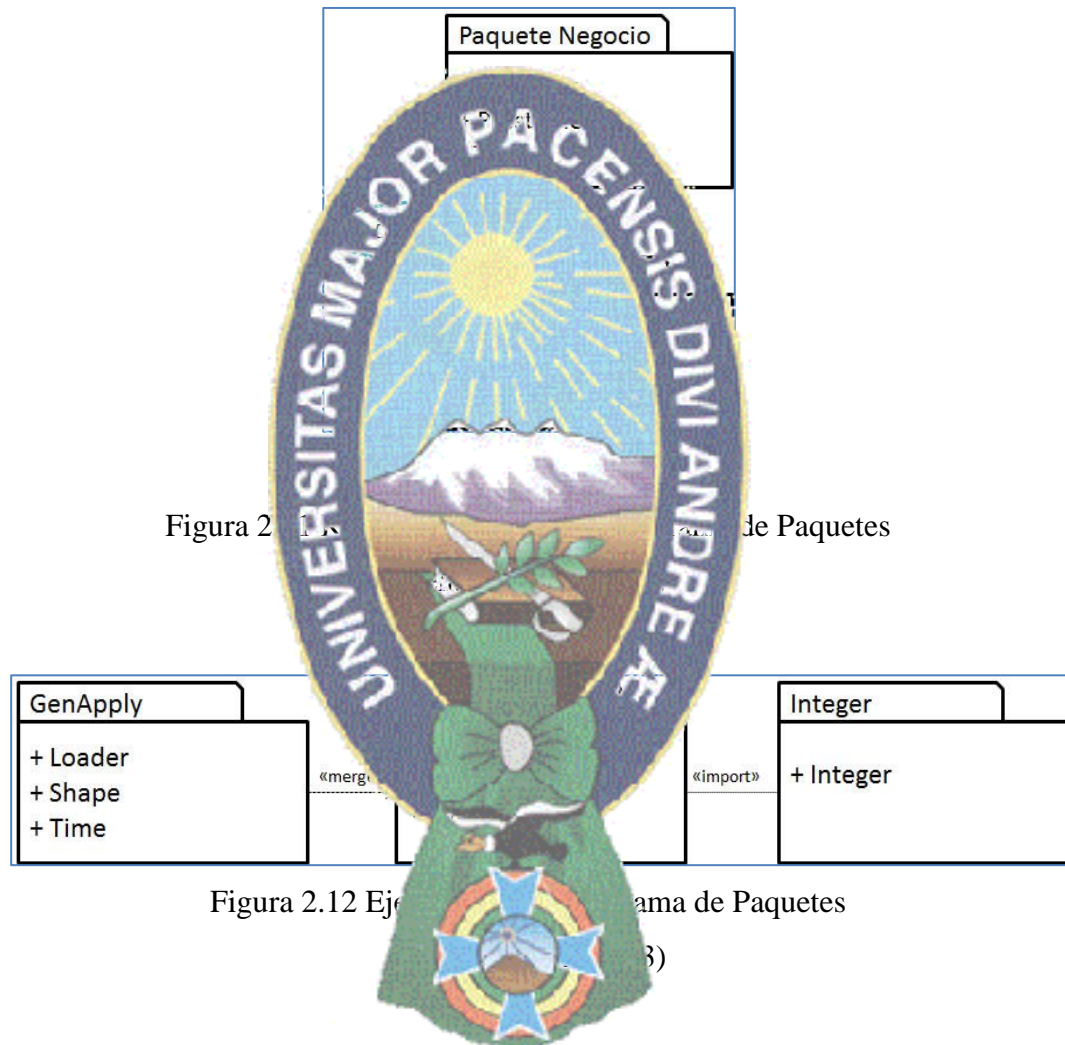


Figura 2.11 Ejemplo de Paquetes

Figura 2.12 Ejemplo de Paquetes

CAPITULO 3

MARCO APLICATIVO

3.1. INTRODUCCIÓN

El presente capítulo se divide en dos partes que se describen a continuación:

Análisis del dominio e identificación del proceso que se desarrollará. Para comprender el dominio y la identificación del proceso que se desarrollará.

Implementar la metodología que se utilizará en el desarrollo de software, además de utilizar el Lenguaje de Modelado de Procesos.

3.2 ANÁLISIS DEL DOMINIO E IDENTIFICACIÓN DEL PROCESO

El flujo de trabajo de los procesos correspondiente modelado fueron tomados, de procesos que se desarrollan en la Unidad de Catastro, de los cuales fue tomado el Proceso de Aprobación de Planos para el desarrollo de este proyecto.

3.2.1 ANALISIS DEL DOMINIO DE LOS PROCESOS

Para tener una concepción clara de los procesos que se desarrollan en la Unidad de Catastro, otros tipos de procesos son: Autorizaciones menajeras, Registro catastral (Ver anexo A).

3.2.2 IDENTIFICACION DEL PROCESO

Este punto describe las actividades del proceso planteado en el anterior subtitulo, la unidad encargada y la descripción de las actividades se muestran en la Tabla 3.1.

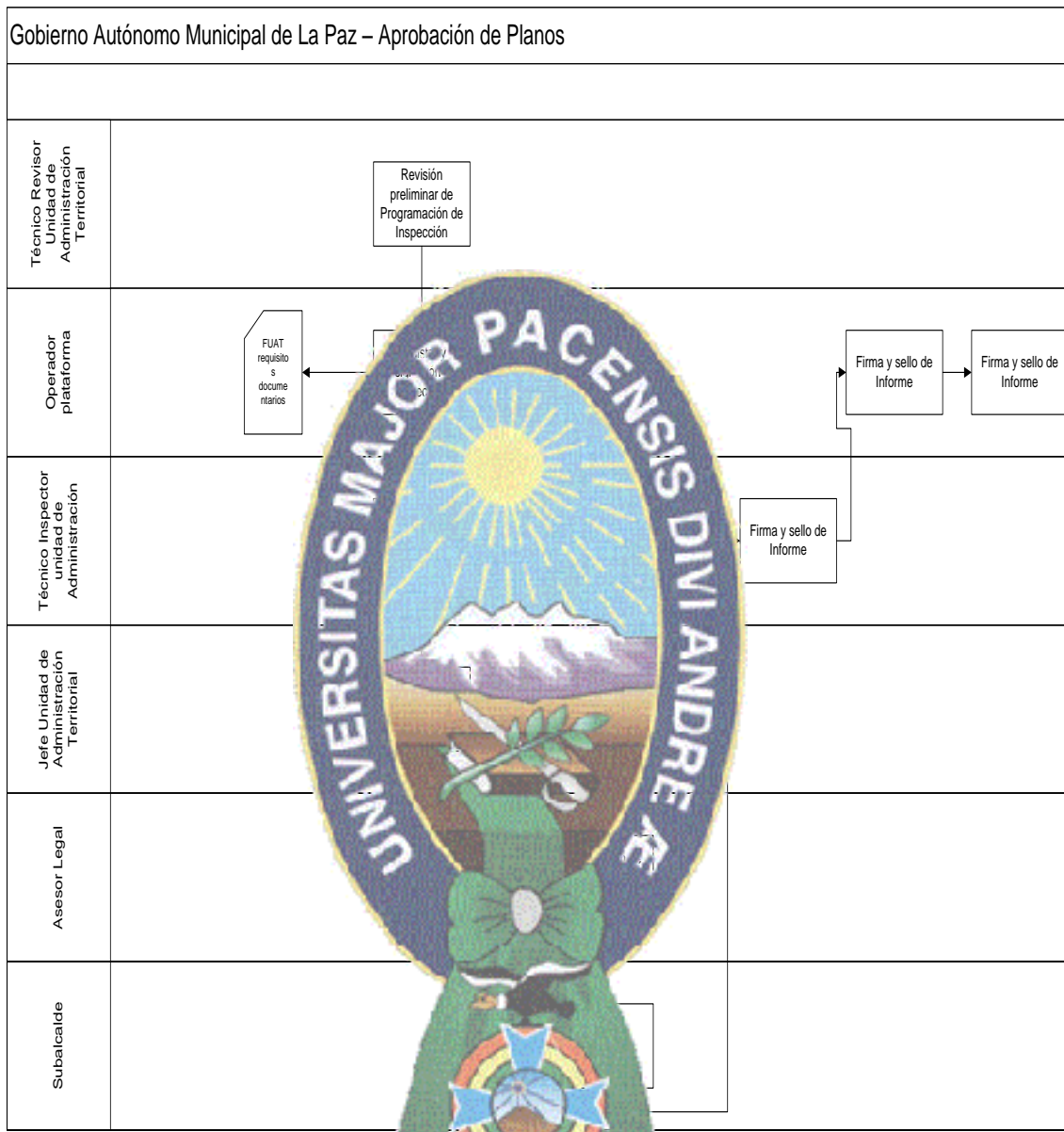


Figura 3.1 Proceso de aprobación de planos (Proceso tipo)

Fuente: (Adaptado) Unidad de Catastro GAMLP

NOMBRE DE LA ACTIVIDAD	UNIDAD ENCARGADA	DESCRIPCION
Revisión Preliminar y Programación de Inspección	Unidad de Catastro	Actividad inicial, es la actividad en la cual se elabora como

		inicio de la recepción de documentos y verificación de documentos, esta actividad se encarga de decepcionar todos los requisitos además de la información concerniente al usuario final.
10 ¹ .- Registro y asignación Inspección		Esta actividad describe a los requerimientos del usuario, como desarrollo de aplicaciones para realizar el respectivo trámite.
Requisitos documentarios		El Catastrista se encarga de elaborar una lista de los requisitos documentarios que debe presentar el usuario final
20 ² .- Inspección Técnica Elaboración de informe		Esta etapa comprende la revisión de documentos, posteriormente el técnico realiza una revisión y evaluación de los mismos
Informe de Aprobación/Rechazo/Observación		El técnico encargado de realizar la revisión, procede a evaluar un informe del estado actual del trámite, aprobando, rechazando la documentación presentada
Visto Bueno	Unidad de Catastro	En caso de que cumpla los requisitos emitidos en la anterior etapa, el Trámite es

¹ 10 = Actividad Inicial del proceso Aprobación de Planos

² 20, = Actividad Tarea del proceso Aprobación de Planos

		derivado a la siguiente actividad (30), caso contrario será devuelta al rol Plataforma (Actividad 10)
30 ³ .- Elaboración de resolución Administrativa	Unidad de Catastro	Según el informe de la actividad anterior (30) el Jefe de Unidad de Admiración territorial, realiza un informe autorizando los requerimientos solicitados.
Elaboración de Resolución Administrativa		Elaboración de informe
40 ⁴ .- Desglose y Emisión de Proforma de pago	Catastro	Una vez autorizada la Resolución Administrativa, en esta actividad se elaboran la resolución y proforma de pago, luego posteriormente ser derivado a la actividad Archivo
50 ⁵ .- Archivo		Esta actividad recepciona toda la documentación concerniente a Trámites ya finalizados, después de esta etapa solo queda cerrar el trámite para su Archivo.



Tabla 3.1 Actividades del Proceso

Fuente: Elaboración propia

³ 30 = Actividad Tarea del proceso Aprobación de Planos

⁴ 40 = Actividad Tarea del proceso Aprobación de Planos

⁵ 50 = Actividad Final Proceso Aprobación de Planos

3.3 FASES DE LA METODOLOGIA XP

3.3.1 FASE DE PLANIFICACION

En esta fase se define el equipo de trabajo y se plantea las historias de usuario para la primera entrega. Además se dividirá por iteraciones la planificación partiendo de las historias de usuario.

3.3.1.1 EQUIPO DEL PROYECTO

La Tabla 3.2 describe el equipo de trabajo que estarán continuamente en el desarrollo del proyecto.



DESCRIPCION	ROL
Coordinador de Actividades	ENCARGADO DE PRUEBAS (BIG BOSS)
Clientes	ENCARGADO DE PRUEBAS (BIG BOSS)
Equipo de desarrollo	PROGRAMADOR ENCARGADO DE PRUEBAS (BIG BOSS) ENCARGADO DE SEGUIMIENTO (TRACKER)

Tabla 3.2 Equipo del Proyecto

Fuente: Elaboración Propia

3.3.1.2 HISTORIAS DE USUARIO

MODULO SEGURIDAD Y ACCESOS

La Tabla 3.3 describe el modulo de seguridad y control de accesos que permitirá administrar y controlar los accesos y la seguridad del sistema, además detalla las diferentes tareas necesarias para el módulo. Las demás historias de usuario referentes a este modulo de detallan en el ANEXO B.

HISTORIA DE USUARIO	
Nº 1	Titulo: Seguridad y control de
Usuario: Todos los usuarios	13
Prioridad en negocio: Alta	Plazo: 3 Días
Descripción: Existirán varios roles que se asignarán a los usuarios, cada rol tendrá los accesos (permisos) necesarios para interactuar con el sistema.	
Tareas: 1.- Diseñar estructura de base de datos 2.- Registro de usuarios, además de asignar roles 3.- Modificar de usuarios – roles 4.- Eliminación de usuarios - roles 5.- Implementar interfaz	

Tabla 3.3 Historia del usuario: Seguridad y Control de Acceso

Fuente: Elaboración propia

MODELADO DE FORMULARIO

El módulo permite modelar el Formulario, mediante la intervención de un usuario (Modelador), que realizara diferentes pasos para adicionar los campos a la plantilla.

La Tabla 3.4 describe el primer paso de generación del Formulario, además de las tareas necesarias en el modulo.

HISTORIA DEL USUARIO	
Nº 5	Actividad: Creación de Repositorio
Usuario: Modelador	Fecha: 03/01/2013
Prioridad en negocio: Alto	Tiempo: 6 días
Descripción: Para iniciar el modelado de formularios se debe crear un Repositorio (el repositorio es crear una tabla de datos) el cual está compuesto por los siguientes datos Id_Usuario, Id_Campo, Id_Formulario, Id_Usuario, Id_Campo, Id_Formulario.	
Tareas: 1.- Diseñar la estructura de base de datos para los repositorios. 2.- Desarrollar una interfaz para la administración un repositorio 3.- Implementar interfaz para la creación de repositorios.	

Tabla 3.4 Historia de la creación de Repositorio

La Tabla 3.5 describe la creación de Campos Físicos⁶, así como las tareas necesarias para el modulo.

HISTORIA DEL USUARIO

⁶ Campos Físicos: Campos que son almacenados directamente en la Base de Datos

Nº 6	Nombre Historia: Creación de Campos Físicos.
Usuario: Modelador	Fecha: 09/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 6 días
Descripción: Una vez creado el repositorio procedemos a la creación de Campos Físicos (son campos que están almacenados directamente en el repositorio de Datos con finalidad de usarlos en los campos lógicos).	
Tareas: 1.- Diseñar estructura de base de datos para Campos Físicos 2.- Desarrollar una interfaz para crear el campo físico 3.- Implementar interfaz para crear el campo físico	

Tabla 3.5 Descripción de las tareas para la creación de Campos Físicos.

La Tabla 3.6 describe la interfaz de usuario para las tareas necesarias para el modulo.

HISTORIA DEL USUARIO	
Nº 7	Nombre Historia: Adición de Proceso
Usuario: Modelador	Fecha: 09/03/2013
Prioridad en negocio: Muy Alta	Tiempo estimado: 6 días
Descripción: Para el inicio del modelado de un formulario previamente se creara un proceso que nos ayuda a identificar el proceso al cual pertenecerán los formularios que posteriormente se crearan, está compuesto por los siguientes datos Código_Proceso, Proceso, Servicio, Repositorio_Físico.	
Tareas: 1.- Diseñar la estructura de base de datos del modulo	

- 2.- Desarrollar las interfaces para listar, adicionar, modificar un proceso
- 3.- implementar las interfaces del modulo

Tabla 3.6 Historia del usuario: Adición de Proceso

Fuente: Elaboración propia

La Tabla 3.7 describe la creación de Campos Lógicos⁷, así como las tareas necesarias para el modulo.

HISTORIA DEL USUARIO	
Nº 8	Historia: Agregación de Campos
Usuario: Modelador	13
Prioridad en negocio: Alto	6 días
Descripción:	
El usuario (Arquitecto) crea el proceso pasamos a la agregación de los campos y formato a cada campo.	
Tareas:	
1.- Diseñar estructura de base de datos	
2.- Desarrollar una interfaz para la creación de los campos creados en repositorio.	
3.- Implementar las interfaces para la eliminación, modificación, listar los campos Lógicos.	

Tabla 3.7 Historia de creación de Campos Lógicos

Fuente: Elaboración propia

La Tabla 3.8 describe la creación de Actividades de un Proceso, así como las tareas necesarias para el modulo.

⁷ Campos Lógicos: Representación lógica de Campos Físicos

HISTORIA DEL USUARIO	
Nº 9	Nombre Historia: Creación de Actividades
Usuario: Modelador	Fecha: 09/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 6 días
Descripción: Previa creación de procesos y campos lógicos pasamos a la creación de la actividad para darle el orden dentro del proceso ad... nos ayudara almacenar las vistas del formulario.	
Tareas: 1.- Diseñar estructura de b... es. 2.- Desarrollar las interfa... modificar de actividades. 3.- implementar las inter... ificación de actividades.	

Tabla 3

La Tabla 3.9 describe la... actividad, así como las tareas necesarias para el modulo.

HISTORIA DEL USUARIO	
Nº 10	Historia: Vistas
Usuario: Modelador	09/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 6 Días
Descripción: Previa creación de las actividades procedemos a la adición de vistas para nuestro formulario.	
Tareas: 1.- Diseñar estructura de base de datos de la tabla Vistas	

- 2.- Desarrollar las interfaces para listar, adicionar eliminación, modificación de vistas.
- 3.- Implementar las interfaces para la adición, eliminación, modificación de Vistas.

Tabla 3.9 Historia del usuario: Vistas

Fuente: Elaboración propia

La Tabla 3.10 describe la creación de Dominios de una Vista, así como las tareas necesarias para el modulo.

HISTORIA DEL USUARIO	
Nº 11	Historia: Dominios
Usuario: Modelador	2013
Prioridad en negocio: Alto	Plazo: 3 Días
Descripción:	
Previa creación de actividades de campo, se crearon dominios el cual nos ayudara a direccionar los campos e	
Tareas:	
1.- Diseñar estructura de base de datos	
2.- Desarrollar las interfaces para listar, adicionar, eliminar y modificar dominios	
3.- Implementar el modulo con las interfaces para listar, adicionar, eliminar.	

Tabla 3.10 Historia del usuario: Dominios

Fuente: Elaboración propia

SEGUIMIENTO DE FORMULARIOS

El modulo Seguimiento de Formulario permite ver el estado de los formularios (Tramite), el modulo realiza operaciones del tipo: llenar, borrar, modificar y grabar. El actor podrá enviar el formulario a otro rol para que siga el flujo de trabajo.

La Tabla 3.11 describe el Modulo de Seguimiento de Formulario, así como las tareas necesarias para el modulo.

HISTORIA DEL USUARIO	
Nº 12	Nombre Historia: Seguimiento de Formularios
Usuario: Todos los usuarios	Fecha: 02/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 6 Días
Descripción: Para realizar el seguimiento de formularios en esta etapa se deberá verificar la existencia del modelado (Modelado de Datos), usuario deberá verificar los datos necesarios para el desarrollo del formulario.	
Tareas: 1.- Diseñar estructura de base de datos para el seguimiento de Formularios 2.- Crear el inicio de trámite de un formulario 3.- Crear el actuar un trámite de un formulario 4.- Implementar interfaz de usuario para el seguimiento de Formularios	

Tabla 3.11 Historia de Usuario para el Modulo de Seguimiento de Formularios

La Tabla 3.12 describe el Modulo de Reportes, así como las tareas necesarias para el modulo.

HISTORIA DEL USUARIO	
Nº 13	Nombre Historia: Reportes
Usuario: Todos los usuarios	Fecha: 02/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 4 Días
Descripción:	

Para realizar los reportes se tomara en cuenta los procesos y las fechas para tener un preciso reporte por fechas

Tareas:

- 1.- Diseñar una consulta en la base de datos que nos permita leer algunas tablas.
- 2.- crear el icono en el menú
- 3.- realizar los reportes de acuerdo a los requerimientos.
- 4.- Implementar interfaz



3.3.1.3 ITERACIONES

3.3.1.3.1 CRONOGRAMA

La Tabla 3.13 describe la secuencia de actividades de usuario trabajadas en el proyecto, los módulos necesarios y la siguiente situación:

- Seguridad y Control de Acceso
- Modelado de Formularios
- Seguimiento de Formularios
- Reportes

3.3.1.4 PLAN DE ENTREGAS

La Tabla 3.14 describe el Plan de Entrega del Sistema, la cual toma en cuenta las distintas iteraciones que fueron desarrolladas en el sistema.

NRO	HISTORIAS DE USUARIO	ENTREGA	ITERACION	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
SEMANAS								
SEGURIDAD								
1	Seguridad y control de acceso	08/07/2013	Primera					
2	Seguridad y control de acceso – Personas	15/07/2013	Primera					
3	Seguridad y control de acceso - Usuarios	22/07/2013	Primera					
4	Seguridad y control de acceso - Roles	29/07/2013	Primera					
MODELADO DE FORMULARIO								
5	Creación de Repositorio	08/08/2013	Primera					
6	Creación de Campos Físicos	15/08/2013	Primera					
7	Adición de Proceso	22/08/2013	Primera					
8	Agregación de Campos Lógicos	29/08/2013	Primera					
9	Creación de Actividades	05/09/2013	Primera					
10	Vistas	12/09/2013	Primera					
11	Dominios	19/09/2013	Primera					
SEGUIMIENTO DE FORMULARIO								
12	Seguimiento de Formularios	26/09/2013	Primera					
REPORTES								
13	Reportes	03/10/2013	Primera					

Tabla 3.13 Cronograma por Historias de Usuario

Fuente: Elaboración propia

NRO	ITERACION	ENTREGA	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
SEMANAS							
1ra	Primera	02/09/2013	█	█	█	█	█
2da	Segunda	14/10/2013			█	█	█
3ra	Tercera	11/11/2013				█	█

Tabla 3.14 Plan de entregas

Fuente: Elaboración propia

3.3.1.5 VELOCIDAD DEL PROYECTO

La Tabla 3.15 y Figura 3.2 muestran la velocidad del proyecto en cada iteración.

	Iteración 2	Iteración 3
Días	33	27
Semanas	5,5	4,5
Historias de Usuario (Velocidad del proyecto)	4	2

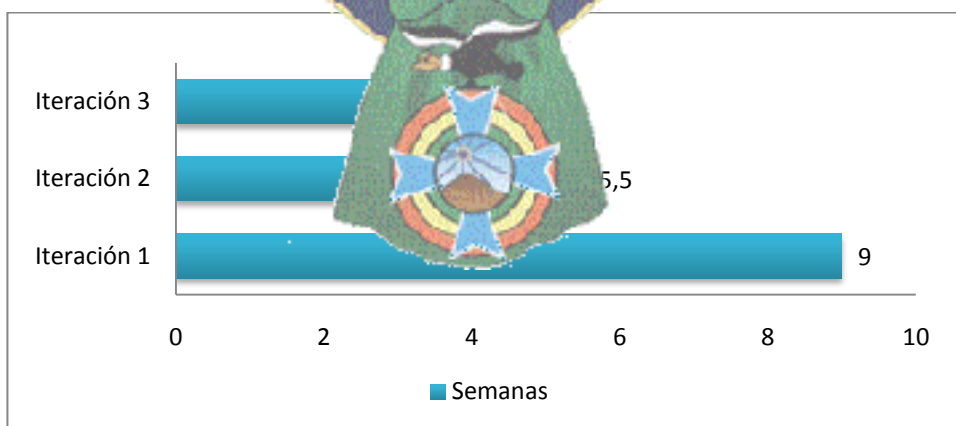


Figura 3.2 Velocidad del proyecto

Fuente: Elaboración propia

3.3.1.6 REUNIONES

La Tabla 3.16 muestra un detalle de reuniones (cronograma las reuniones de entrega), las cuales son realizadas de manera periódica.

NRO	ITERACION	ENTREGA	DURACION	JULIO	AGOSTO	SEPTIEMBRE	OCTUBRE	NOVIEMBRE
SEMANAS								
1ra	Primera			■	■	■		
2da	Segunda					■	■	
3ra	Tercera						■	■



3.3.2 FASE DE DISEÑO

En esta fase se toma en cuenta el resultado del análisis, mediante Lenguaje Unificado de Modelado – UML, como un lenguaje que proporciona un vocabulario de reglas para crear modelos de las funcionalidades y por último se presenta modelos y diagramas necesarios para especificar el sistema completo.

3.3.2.1 METAFORA DEL SISTEMA

La Tabla 3.17 describe la metáfora del sistema que es una descripción simple de cómo funciona el sistema en general.

3.3.2.2 ACTORES DEL NEGOCIO

DESCRIPCION DE LOS ACTORES

Los actores identificados para el desarrollo del presente proyecto se detallan en la Tabla 3.18.

Identificación de la Metáfora	
MODELADOR DE FORMULARIO	AUTONOMO MUNICIPAL DE LA PAZ
Metáfora del Sistema	
<p>El usuario modelador de formularios se registra en el sistema con un nombre de usuario y contraseña si el modelador está registrado en el sistema. El usuario puede acceder al sistema con diferentes opciones del menú de modelador de formularios. El usuario puede seleccionar la opción de crear un formulario, lo que se muestra en los pasos que se enumeran a continuación:</p> <p>Del menú de modelador de formularios, el usuario puede seleccionar la opción de crear un repositorio. Una vez creado el repositorio, el usuario puede acceder al repositorio. Después de crear el repositorio y los formularios, el usuario puede acceder al repositorio que creó. Teniendo el proceso creado, el usuario puede acceder al proceso utilizando los campos físicos creados.</p> <p>Nombra un formulario adicional y crea un formulario vacío. Cuando el formulario vacío puede modelar sus campos y los campos se muestran en el formulario. Para finalizar el modelador de formularios, el usuario puede acceder al formulario con la creación de un trámite con el nombre del trámite. El usuario puede acceder al trámite creado. Una vez modelado el formulario, el usuario puede acceder al trámite creado. Además de poder emitir reportes.</p>	
Información de aprobación de la	
Firma del Gestor (Big Boss)	Firma del cliente

Tabla 3.17 Metáfora del sistema

Fuente: Elaboración propia

ACTOR	DESCRIPCION
-------	-------------

DIRECTOR	Es el encargado de toda la Dirección, bajo sus manos tiene la responsabilidad en el manejo de todas las unidades que la conforman.
JEFE UNIDAD	Persona encargada de la unidad, sus actividades son planificar, controlar, administrar todos los requerimientos de la unidad (Una dirección puede tener varias
ADMINISTRADOR	... administrar todo el sistema, puede ...uario así como también dar su
OPERADOR	Formularios y hacer el
MODELADOR	ir, además de diseñar el ... es que se realizaran en la



ACTIVIDADES DE LOS AC

Descripción de las actividades de lo ... bla 3.19.

ACTOR	ACTIVIDADES
DIRECTOR	<ol style="list-style-type: none"> 1.- Realizar el requerimiento (Movimiento) de nuevo personal en la Dirección 2.- Definir los lineamientos de la Dirección

	<p>3.- Tomar todas las decisiones más importantes de la Dirección</p> <p>4.- Verificar los reportes Generados por el sistema</p>
JEFE UNIDAD	<p>1.- Realizar movimientos de personal en la unidad</p> <p>2.- Tomar las decisiones que conciernen a la unidad</p> <p>3.- Controlar los movimientos de la unidad</p> <p>4.- Controlar los recursos económicos de la unidad</p> <p>5.- Verificar los reportes generados por el sistema</p>
ADMINISTRADOR	<p>1.- Dar de alta un usuario al sistema</p> <p>2.- Eliminar un usuario del sistema</p>
OPERADOR	<p>1.- Registrar los datos para su seguimiento.</p>
MODELADOR	<p>1.- Recopilar información para la generación de un flujo de trabajo</p> <p>2.- Modelar un flujo de trabajo (Formulario)</p> <p>3.- Petición de nuevos requerimientos</p>

Tabla 3.19 Actividades de los Actores

Fuente: Elaboración propia

3.3.2.3 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso (ver Figura 3.3), describe una breve vista para el usuario del trabajo que se realizara en el sistema.

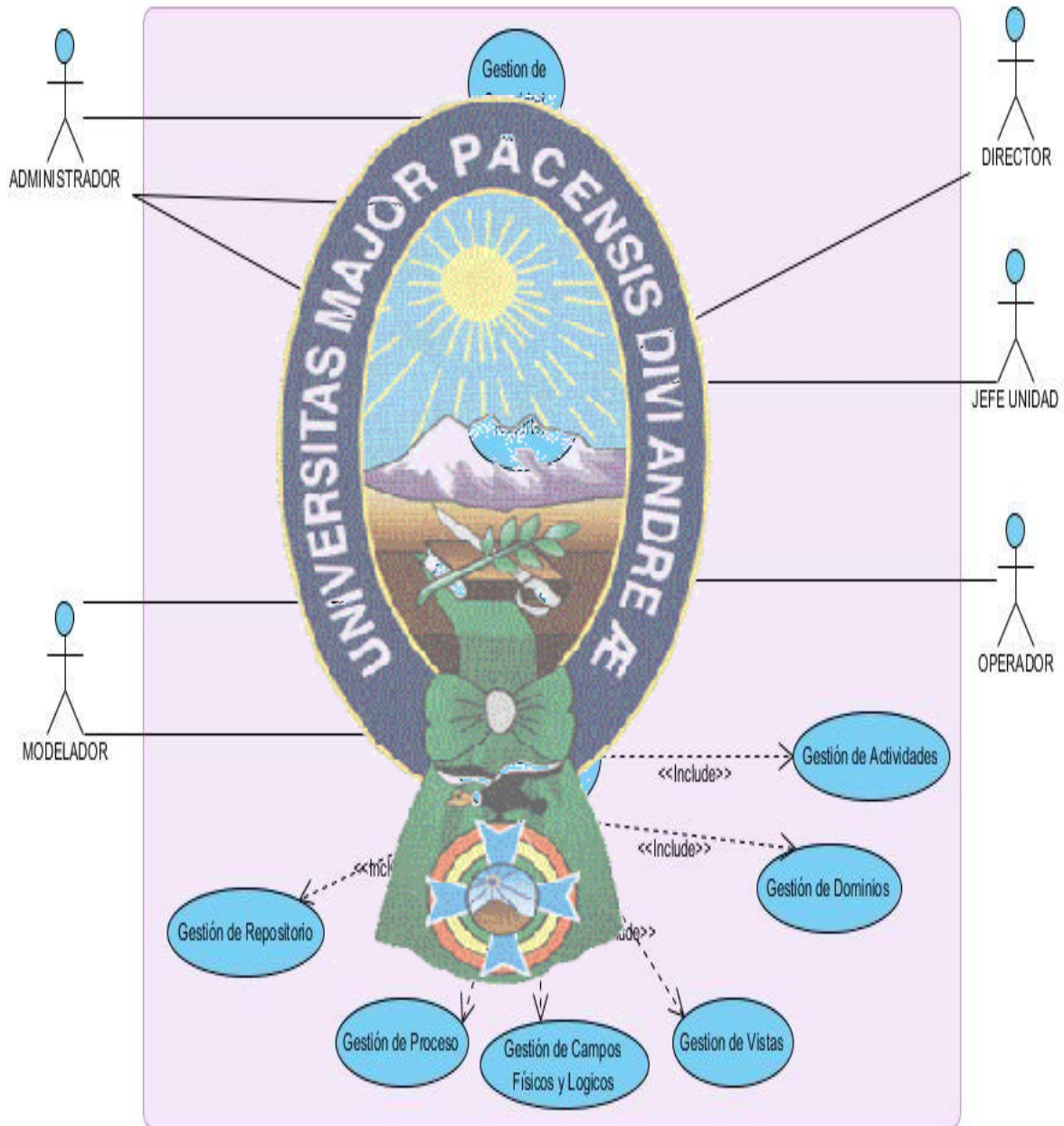







Figura 3.3 Casos de Uso del Sistema

Fuente: Elaboración propia

La Tabla 3.20 describe cada uno de los actores del caso de uso general del sistema.

CASOS DE USO DEL SISTEMA	
Nombre	Casos de uso del sistema
Autor	Elaborado por el equipo de desarrollo
Fecha	01/01/2020
DESCRIPCION	
<p>Los diferentes roles involucrados en el sistema (de acuerdo al nivel del permiso), tienen el acceso a la seguridad, gestión de usuarios, gestión de propiedades, gestión de documentos y modelado del sistema.</p>	
ACTORES	
 Director	Encargado de la gestión general del sistema
 Jefe de Unidad	Encargado de la gestión de unidades del sistema
 Administrador	Encargado de administrar el sistema
 Operador	Realiza la gestión de los Formularios, además de verificar la información
 Modelador	Modifica los datos del sistema (Gestión Catastral)
PRECONDICIONES	
<p>Los actores deberán estar registrados en el sistema SMF con todos los privilegios</p>	
FLUJO NORMAL	

<ol style="list-style-type: none"> 1. El actor (usuarios) introduce su nombre de usuario y contraseña pulsa ingresar 2. El sistema realiza una comprobación de usuario rol, permisos y la unidad organizacional a la que pertenece el usuario 3. El sistema muestra el menú de opciones y accesos que este usuario tiene. 4. El actor selecciona del menú de opciones que le corresponda 5. El sistema responde con el modelo correspondiente 	
FLUJO ALTERNATIVO	
<ol style="list-style-type: none"> 6. El sistema no encuentra el usuario y devuelve al punto de inicio. 	
POSTCONDICIONES	
<p>Accesos según el rol</p>	

Tabla 3.2 Flujo de datos de inicio del Sistema



Para un mayor detalle, los diferentes flujos que conciernen al sistema se describen en el ANEXO C.

3.3.2.4. MODELO FISICO

La Figura 3.4 muestra el diseño de la base de datos y las relaciones entre las diferentes tablas del sistema, así como los tipos de atributos que la componen.

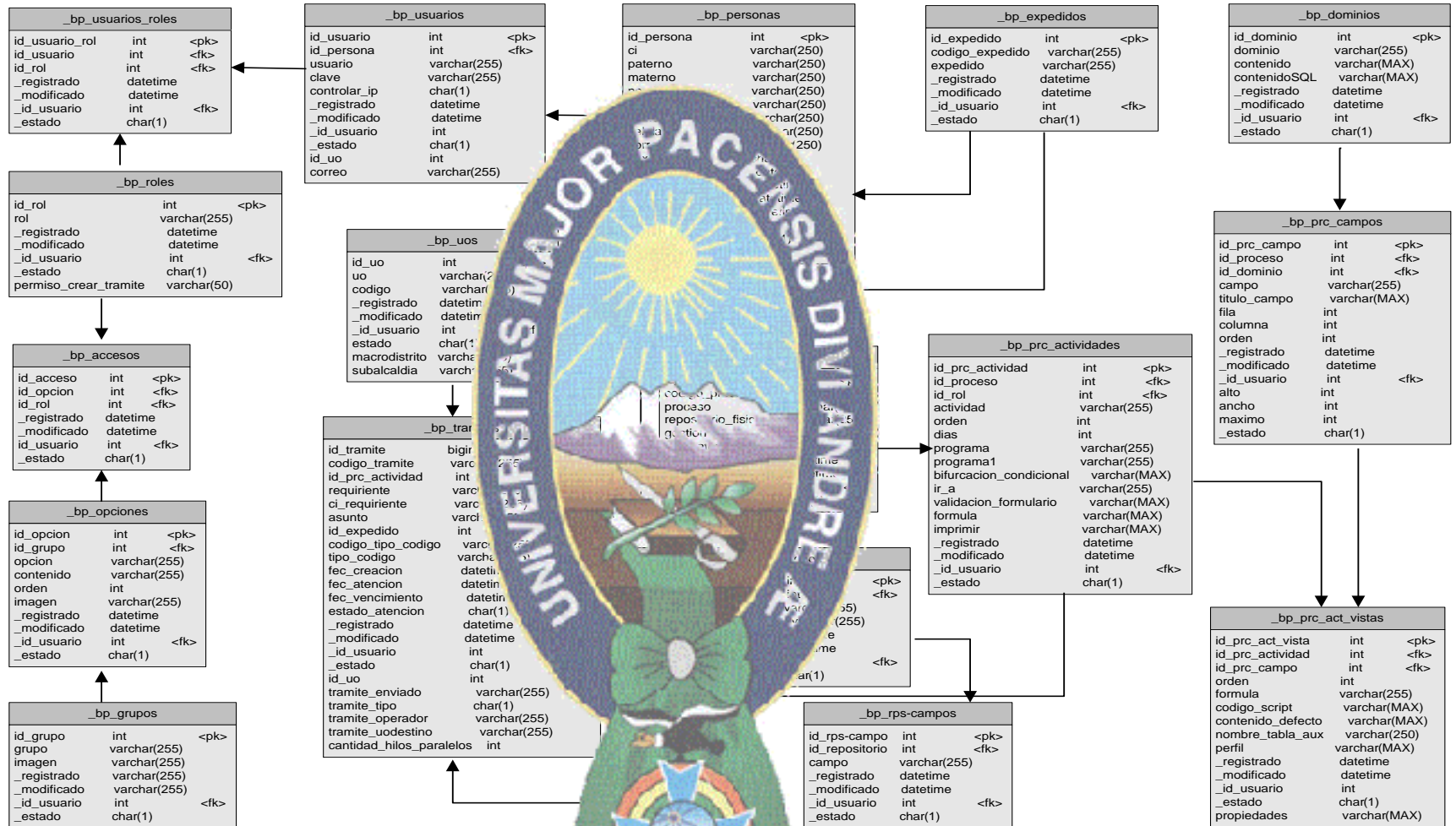


Figura de Proceso Físico
 Fuente: Elaboración propia

3.3.2.5 DIAGRAMA DE CLASES

El diagrama de clases (Figura 3.5), visualiza las relaciones entre las clases que existen en el sistema, además de los diferentes métodos que la comprenden.

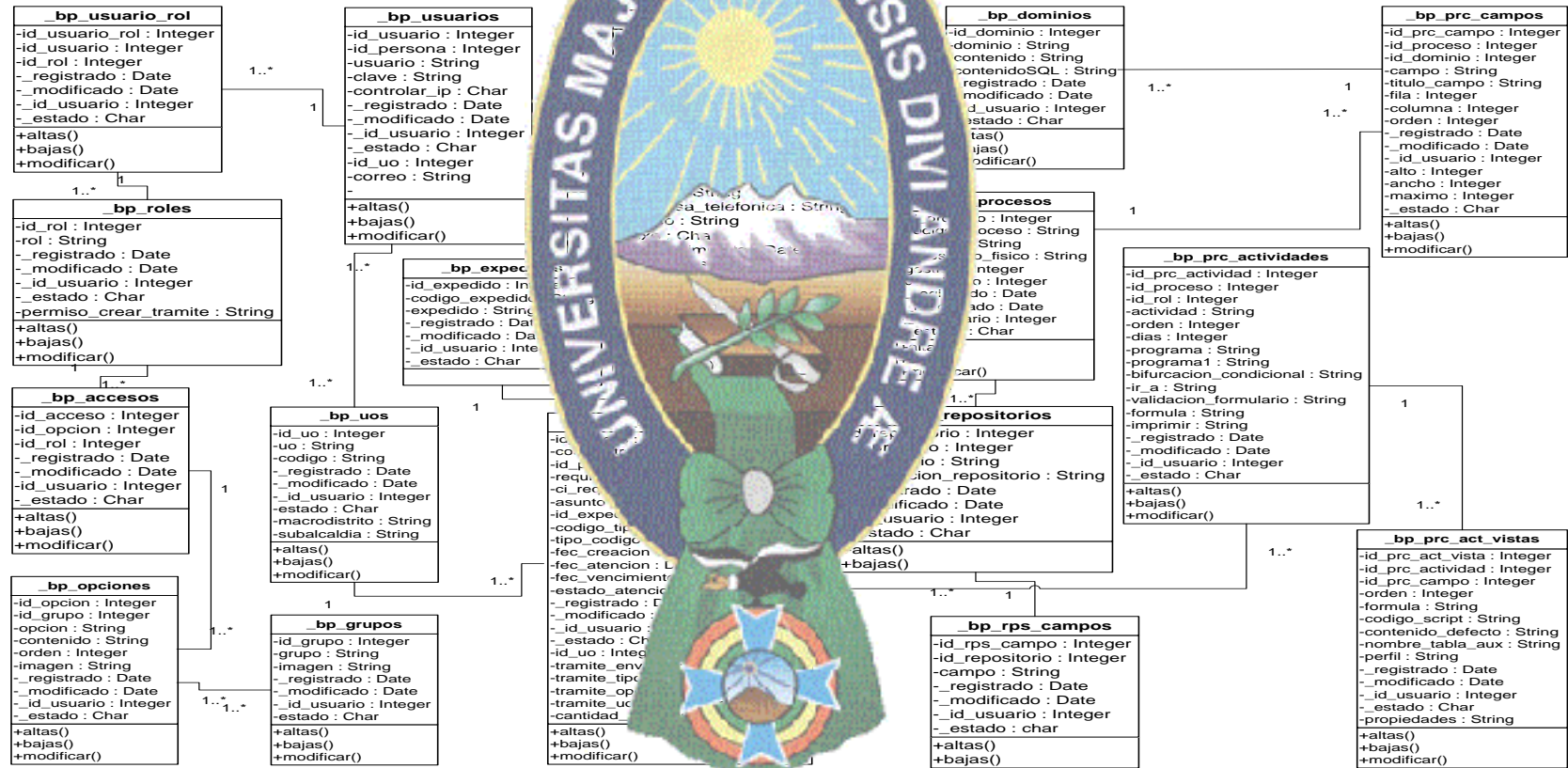


Figura 3.5 Diagrama de clases

Fuente: Elaboración

3.3.2.6 TARJETAS CLASE, RESPONSABILIDAD Y COLABORACION (CRC)

Se describen a continuación las tarjetas C.R.C del sistema.

MODULO SEGURIDAD Y ACCESOS

_bp_personas	
Objetivos y Responsabilidades Administradores	
Registra las personas que acceden a la parte del sistema	_bp_personas
_bp_usuarios	
Objetivos y Responsabilidades Administradores	
Adicionar usuarios y registrar el nombre con el cual va acceder	_bp_usuarios_roles
_bp_roles	
Objetivos y Responsabilidades Administradores	
Adicionara información de los roles con el cual el usuario está registrado	_bp_usuarios_roles _bp_usuarios

Tabla 3.23 Usuarios
Fuente: Elaboración Propia

Tabla 3.23 tarjeta CRC de Roles

Fuente: Elaboración Propia

_bp_usuarios_rols	
Objetivos y Responsabilidades	Colaboradores
Adiciona roles a los usuarios.	_bp_rols

Tabla 3.24 tarjeta CRC de Usuarios - Roles

Fuente: Elaboración Propia

_bp_accesos	
Objetivos y Responsabilidades	Colaboradores
Adicionar accesos a los usuarios y registrar los accesos permitidos.	

_bp OPCIONES	
Objetivos y Responsabilidades	Colaboradores
Adicionar opciones de formulario dependientes a los grupos y accesos ya definidos en el modelado de los formularios.	

Tabla 3.25 tarjeta CRC de Opciones

Fuente: Elaboración Propia

_bp_grupos	
Objetivos y Responsabilidades	Colaboradores
Adicionar grupos para el modelado de formularios.	_bp OPCIONES

Tabla 3.27 tarjeta CRC de Grupos

Fuente: Elaboración Propia

_bp_ous	
Objetivos y Responsabilidades	Colaboradores
Adicionar las diferentes unidades que intervienen en los procesos y así identificar al usuario por unidad.	_bp_usuarios

Tabla 3.28 tarjeta CRC de las Unidades

_bp_expedidos	
Objetivos y Responsabilidades	Colaboradores
Almacena y registra los expedientes No en los expedientes ya que en los expedientes se utiliza una tabla paramétrica.	

MODULO GENERACION DE FORMULARIOS, SEGUIMIENTO DE FORMULARIOS

_bp_repositorios	
Objetivos y Responsabilidades	Colaboradores
Adicionar información de los repositorios para que se liste en los procesos	_bp_rps-campos _bp_usuarios

Tabla 3.30 tarjeta CRC de Repositorio

Fuente: Elaboración Propia

_bp_tramites	
Objetivos y Responsabilidades	Colaboradores
Adicionar trámites para los cuales se realizan los formularios.	_bp_expedidos _bp_usuarios _bp_uos _bp_prc_actividades

Tabla 3.31 tarjeta CRC de Tramites

Fuente: Elaboración Propia

_bp_procesos	
Objetivos y Responsabilidades	Colaboradores
Adicionar procesos corresponden los	otros elementos.

_bp_prc_actividades	
Objetivos y Responsabilidades	Colaboradores
Adicionar actividades procesos procesos.	roles

Tabla 3.3 tarjeta CRC de Actividades

Fuente: Elaboración Propia

_bp_rps_campo	
Objetivos y Responsabilidades	Colaboradores
Adicionar campos físicos al repositorio.	_bp_repositorios

Tabla 3.34 tarjeta CRC de Campos Fisicos

Fuente: Elaboración Propia

_bp_prc_campos	
Objetivos y Responsabilidades	Colaboradores
Adicionar campos lógicos al proceso correspondiente.	_bp_procesos _bp_dominios

Tabla 3.35 tarjeta CRC de Campos Lógicos

Fuente: Elaboración Propia

_bp_prc_act_vistas	
Objetivos y Responsabilidades	Colaboradores
Adicionar vistas (controles y otros) a la vista del formulario	_bp_procesos _bp_dominios

_bp_dominios	
Objetivos y Responsabilidades	Colaboradores
Almacena y adiciona información general (datos paramétricos) al modelado del formulario	_bp_procesos _bp_dominios

Tabla 3.36 tarjeta CRC de Dominios

Fuente: Elaboración Propia

3.3.2.7 DIAGRAMA DE PAQUETES

La Figura 3.6 describe el Diagrama de paquetes del sistema.



Figura 3.6 Diseño de paquetes del sistema

Fuente: Autor, 2019

3.3.3. FASE DE DESARROLLO

Los pasos claves en el proceso son comenzar con una implementación simple de las primeras historias de usuario y de ahí obtener requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

En esta fase se desarrollo las funcionalidades (historias de usuario). Se realizó tres iteraciones, para luego obtener un resultado en una iteración final.

3.3.3.1 ITERACION FINAL DEL SISTEMA

La idea principal detrás de mejorar el desarrollo iterativo es desarrollar un sistema de programas de manera incremental para aprovechar ventaja de lo que se ha aprendido a lo largo del desarrollo antes de desarrollar nuevas versiones entregables del sistema. El desarrollo final de las historias de usuario es la continuación.

3.3.3.1.1 SEGURIDAD

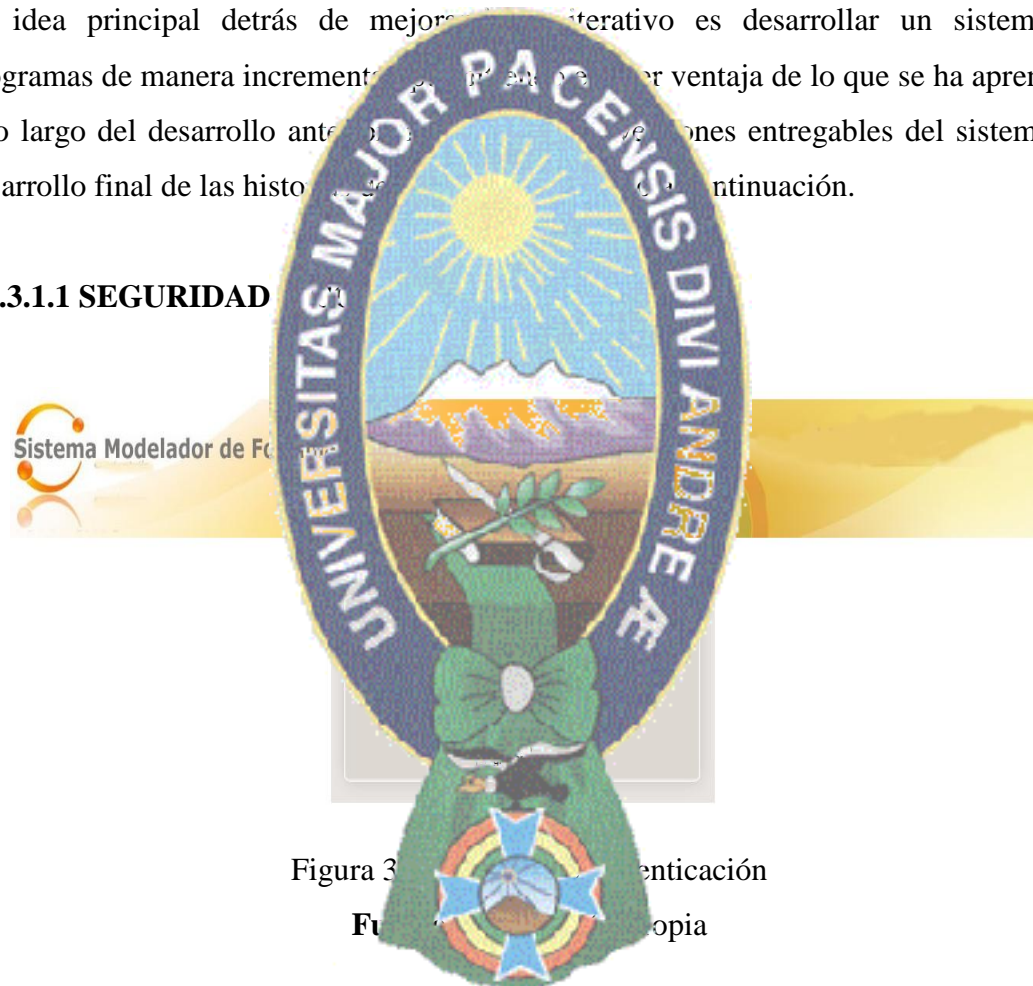


Figura 3 Autenticación
Fuente: Wikipedia

TAREAS Y PRUEBAS DE ACEPTACION SEGURIDAD Y CONTROL DE ACCESO

TAREAS

- 1.- Diseñar estructura de base de datos del modulo.

2.- Registro de usuarios, además de la asignación de roles

Adicionar usuario: nos permitirá registrar al usuario y asignarle una contraseña para que pueda acceder al sistema.



3.- Modificar de usuarios – roles

4.- Eliminación de usuarios - roles

5.- Implementar interfaz

PRUEBAS DE ACEPTACION SE Y CONTROL DE ACCESO

a) Identificar todos los posibles resultados de la Historia.

Información que debe ser registrado para que la persona pueda ser parte del sistema.

b) Identificar los resultados que terminen la historia y los que permiten continuar dentro de la historia

Termina la historia cuando la persona es registrado correctamente, caso contrario no se podrá asignar un usuario.

c) Identificar los caminos posibles de ejecución

Se realizara el registro sin ningún problema en el caso de dar de baja a uno de las personas se le debe dar de baja también a todos los accesos que la persona.

d) Asignar un conjunto de valores validos y valores del entorno a cada camino de ejecución para obtener el resultado

El conjunto de valores nos permite definir diferentes tablas para que de esta manera se pueda llevar el control de acceso.

e) Eliminar caminos redundantes

No existen caminos redundantes.

Para un mayor detalle, (ver Anexo 1: Seguridad y Control de Acceso) se describen en el ANEXO 1.

3.3.3.1.2 MODELADO

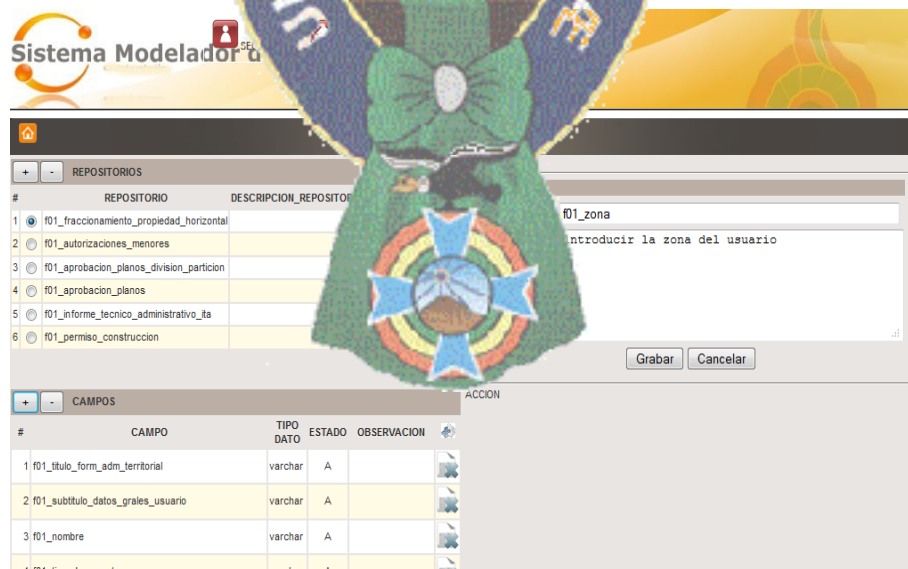


Figura 3.9 Pantalla Adicionar Repositorio

Fuente: Elaboración propia

TAREAS Y PRUBAS DE ACEPTACION CREACION DE REPOSITORIO

TAREAS

- 1.- Diseñar la estructura de base de datos de la tabla Repositorios.
- 2.- Desarrollar una interfaz para la adición, eliminación un repositorio.
- 3.- Implementar interfaz para la creación de repositorios.

PRUEBAS DE ACEPTACION CREACION DE REPOSITORIO

- a) Identificar todos los posibles caminos de ejecución para la creación de un repositorio.
Datos requeridos por el sistema se ingresan en el formulario adicionalmente se ingresan los datos de un repositorio en la base de datos.
b) Identificar los resultados de la ejecución de la historia que permiten continuar dentro de la historia.
Termina la historia cuando se ingresan los datos de un repositorio caso contrario revisamos el módulo de aceptación de repositorio.
c) Identificar los caminos posibles para la creación de un repositorio.
Mediante la adición de repositorio se ingresan los datos de un repositorio para asegurar que no haya datos repetidos en la base de datos.
d) Asignar un conjunto de valores de los alrededores del entorno a cada camino de ejecución para obtener el resultado.
El conjunto de valores está dado por los datos de la tabla de repositorios.
e) Eliminar caminos redundantes.
No existen caminos redundantes.

Para un mayor detalle, las demás historias de usuario (Creación de Repositorio) se describen en el ANEXO C.

3.3.3.1.3 SEGUIMIENTO DE FORMULARIO

Usuario administrador | Rol PLATAFORMA - INGRESO | Admin Admin, Administrador | UO DIRECCION DE ADMINISTRACION GENERAL | CERRAR SE

Sistema Modelador de Formularios

Selecionar Proceso

INICIAR UN TRAMITE ASOCIADO AL SERVICIO DE: Aprobaciones y Autorizaciones de Administración Territorial
INICIAR UN TRAMITE ASOCIADO AL PROCESO DE: APROBACION DE PLANOS

VENTANILLA Calle Colon
OPERADOR administrador
SERVICIO Aprobaciones y Autorizaciones de Administración Territorial
PROCESO

TIPO DE HOJA L*

DATOS PERSONALES DEL

DdeTP
Dirección de Desarrollo Organizacional
y Tecnologías de Información

APROBACIONES Y AUTORIZACIONES DE ADMINISTRACION TERRITORIAL

DATOS GENERALES DEL INMUEBLE

MACRODISTRITO
--SELECCIONAR--

CODIGO CATASTRAL

ZONA

TIPO DE

NOMBRE DE VIA

NUMERO DE DOMICILIO

NUMERO DE INMU

CORREO ELEC

TELEFONO / CELULAR

CORREO ELEC

SERVICIO REQUERIDO POR EL SOLICITANTE

SUSTITUCION LEGALIZACION
 REMODELACION AMPLIACION REMODELACION

REQUISITOS PRESENTADOS POR EL SOLICITANTE

CERTIFICACION CATASTRAL VIGENTE (FOTOCOPIA)

RESUMEN
HISTORICO
ADJUNTOS

Numero de Trámite: 1092013
Solicitante: SANCESID
CODICE
CI: 12312122_LR2
Proceso: APROBACION DE PLANOS
Actividad: FUNDACION Y ENVIO DE DOCUMENTOS

ACTUAR
MODIFICAR
BLOQUEAR
AVANZAR
VOLVER

Figura 3.11 Pantalla de Formulario creado

Fuente: Elaboración propia

Usuario administrador | Rol PLATAFORMA - INGRESO | Admin Admin, Administrador | UO DIRECCION DE ADMINISTRACION GENERAL | [CERRAR SESION](#)

Sistema Modelador de Formularios

PLATAFORMA - INGRESO | ADICIONAR | CLONAR

PENDIENTES | BLOQUEADOS | RECIBIDOS | SIN DATOS | FIRMADOS | FIRMAS OBSERVADAS | PARA FIRMA | FINALIZADOS | BUSCAR TRAMITE:

#	Numero de Tramite	Estado Atencion	Solicitante tipo Documento/No	Atendido	Procesos/Actividad
3512	109/2013	✓	MARCELO CONDE CI 12312122 - LPZ	Inicia: 14/11/2013 13:12:54 Plazo: 1 Vence: 18/11/2013 13:12:54	APROBACION DE PLANOS 10 - FILIACION Y ENVIO DE DOCUMENTOS Rol: PLATAFORMA - INGRESO
3510	100/2013	✓	SHIRLEY CI 123 - LPZ	Inicia: 19/04/2013 9:57:30 Plazo: 1 Vence el: 23/04/2013 9:57:30	PERMISO DE CONSTRUCCION 10 - FILIACION Y ENVIO DE DOCUMENTOS Rol: PLATAFORMA - INGRESO
3509	107/2013	✓	18042013 CI 18042013 - LPZ	Inicia: 19/04/2013 9:57:30 Plazo: 1 Vence el: 23/04/2013 9:57:30	APROBACION DE PLANOS 10 - FILIACION Y ENVIO DE DOCUMENTOS Rol: PLATAFORMA - INGRESO

Figura 3.1. Interfaz de Seguimiento de Formularios

TAREAS Y PRUEBAS DE ACEPTACION DEL SISTEMA DE FORMULARIOS

TAREAS

- 1.- Diseñar estructura de base de datos para el desarrollo de Formularios.
- 2.- Crear el inicio de trámite de un trámite.
- 3.- Crear el actuar un trámite de un trámite de ruta.
- 4.- Implementar interfaz.

PRUEBAS DE ACEPTACION DEL SISTEMA DE FORMULARIOS

a) Identificar todos los posibles errores de la historia

el registro de los formularios se debe tener un código de tramite visible en la tabla de seguimiento de formulario para de esta manera el usuario pueda identificar el estado del formulario.

b) Identificar los resultados que terminen la historia y los que permiten continuar dentro de la historia

Termina la creación de un nuevo trámite cuando este es fue llenado correctamente los campos sin ningún error al guardarlo.

c) Identificar los caminos posibles de ejecución

El camino de ejecución es con la creación de la hoja de ruta para el inicio del nuevo trámite el cual nos permitirá visualizarlo en la tabla de trámites pendientes.

d) Asignar un conjunto de valores validos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

El conjunto de valores esta dado por los datos recuperados por el sistema y los tramites nuevos será por el registro a través de una documentación de identificación.

e) Eliminar caminos redundantes

No existen caminos redundantes

Para un mayor detalle, las (ver seguimiento de Formularios) se describen en el ANEXO C

3.3.3.1.4 REPORTEES



Admin, Administrador | UO DIRECCION DE ADMINISTRACION GENERAL | CERRAR SESION

REPORTES | MATRIZ | FORMULARIOS DE INSPECCION

N°	MACRODISTRITO	NOMBRE DEL TECNICO	FECHA DE INICIO	FECHA DE FIN	ESTADO	INFORME TECNICO	FECHA DE INFORME	NUMERO DE TRAMITE
1	COTAHUMA	16						
2	COTAHUMA	TEC. MARCIA TICONA QUISBERT	05/05/2012 0:00:00				11/05/12	1612
3	COTAHUMA	16	05/05/2012 0:00:00			51	06/07/2012	1823
4	COTAHUMA	TEC. MARCIA TICONA QUISBERT	22/03/2012 0:00:00			102	09/05/12	1758
5	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	05/05/2012 0:00:00					
6	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	01/01/1900 0:00:00			SACO-UAT-PCH 117/12	18-05-12	2907
7	COTAHUMA	TEC. MARCIA TICONA QUISBERT	05/05/2012 0:00:00			113	16/05/12	2988
8	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	01/01/1900 0:00:00			SACO-UAT-PCH 38/2012	09-05-12	2972
9	COTAHUMA	TEC. MARCIA TICONA QUISBERT	05/05/2012 0:00:00			93	10/05/12	2455
10	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	20/03/2012 0:00:00					
11	COTAHUMA	TEC. MARCIA TICONA QUISBERT	05/05/2012 0:00:00			37	08/05/12	2606
12	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	01/01/1900 0:00:00			SACO-UAT-PCH 100/12	27/04/12	2778
13	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	05/05/2012 0:00:00			SACO-UAT-PCH 112/12	14-05-12	2727
14	COTAHUMA	16	05/05/2012 0:00:00			92	03/05/2012	2784
15	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	05/05/2012 0:00:00			SACO-UAT-PCH 118/12	21-05-12	2807
16	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	05/05/2012 0:00:00			SACO-UAT-PCH-106/12	11-05-12	2862
17	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	05/05/2012 0:00:00			SACO-UAT-PCH 39/12	10-05-12	2900
18	COTAHUMA	TEC. MARCIA TICONA QUISBERT	08/05/2012 0:00:00			127	31/05/12	3024
19	COTAHUMA	TEC. MARCIA TICONA QUISBERT	09/05/2012 0:00:00			40	11/05/12	3036
20	COTAHUMA	TEC. MARCIA TICONA QUISBERT	09/05/2012 0:00:00			123	24/05/12	3035
21	COTAHUMA	TEC. MARCIA TICONA QUISBERT	10/05/2012 0:00:00		APROBADO	42	18/05/12	3086
22	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	10/05/2012 0:00:00		APROBADO	SACO-UAT-PCH 41/12	17-05-12	3084
23	COTAHUMA	16	15/05/2012 0:00:00		OBSERVADO	110	15/05/2012	3141
24	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	11/05/2012 0:00:00		OBSERVADO	SACO-UAT-PCH 125/12	30-05-12	3114
25	COTAHUMA	TEC. PILAR CHOQUE CHAMBILLA	14/05/2012 0:00:00		OBSERVADO	SACO-UAT-PCH 122/12	22/05/12	3155
26	COTAHUMA	16	16/05/2012 0:00:00		OBSERVADO	47	01/06/2012	3228
27	COTAHUMA	16	16/05/2012 0:00:00		OBSERVADO	121	24/05/2012	3225
28	COTAHUMA	16	16/05/2012 0:00:00		OBSERVADO	110	18/05/2012	320
29	COTAHUMA	16	16/05/2012 0:00:00		OBSERVADO	124	04/06/2012	3207
30	COTAHUMA	16	16/05/2012 0:00:00		OBSERVADO	115	31/05/2012	3224

Figura 3.13 Pantalla de Reportes estáticos

Fuente: Elaboración propia

TAREAS Y PRUEBAS DE ACEPTACION REPORTES

TAREAS

- 1.- Diseñar una consulta en la base de datos que nos permita leer algunas tablas.
- 2.- Crear el icono en el menú
- 3.- Realizar los reportes de acuerdo a los procesos creados.
- 4.- Implementar interfaz.

PRUEBAS DE ACEPTACION

a) Identificar todos los procesos

Se identificara los procesos que se ejecutan en el sistema en un reporte

b) Identificar los resultados de los procesos que permiten continuar dentro de la historia

Termina el reporte cuando se ejecuta el proceso

c) Identificar los caminos de los procesos

Ingresamos al icono reporte de un proceso específico y la fecha que necesita el reporte.

d) Asignar un conjunto de valores de los procesos del entorno a cada camino de ejecución para obtener el resultado

El conjunto de valores está dado por los datos recuperados por el sistema y los trámites nuevos serán por el registro mediante una documentación de identificación.

e) Eliminar caminos redundantes

No existen caminos redundantes.



3.3.4 FASE DE PRUEBAS

3.3.4.1 PRUEBAS DE ACEPTACION

PRUEBA AL MODULO DE GENERACION DE FORMULARIO

Este documento cubre el conjunto de pruebas funcionales relacionadas con el modulo Modelado de Formulario.

En este módulo se podrá modelar formularios cada uno de estos formularios serán registrados por diferentes pasos que conforman el modelado y podrán ser seguidos en su seguimiento.

DESCRIPCION

El usuario modelador de formularios debe ingresar con usuario y contraseña si el modelador está registrado en el sistema. Después de ingresar al menú con diferentes opciones del cual selecciona la opción de modelado de formularios. Los pasos que se enumeran a continuación:

Del menú de modelado elige la acción de crear repositorio del cual adiciona un repositorio. Una vez creado el repositorio adiciona un campo al repositorio. Después de crear el repositorio y los campos adicionados al repositorio que creo. Teniendo el proceso procede a crear campos lógicos al proceso utilizando los campos físicos ya creados.

Nombrar un formulario adicionando una actividad. Teniendo el formulario vacío puede modelar sus campos adicionando una vista en el formulario.

Para finalizar el modelador podrá visualizar el formulario con la creación de un trámite con el nombre del proceso creado.



Una vez modelado el formulario se puede realizar el seguimiento de este además de poder emitir reportes.

CONDICIONES DE EJECUCION

El usuario debe estar registrado en el sistema.

ENTRADA

- El actor ingresa al sistema y contraseña pulsa aceptar.
- El sistema verifica los datos ingresados en el sistema.
- El sistema le muestra una interfaz con diferentes opciones.
- El actor selecciona una opción y elige la acción repositorios.
- El actor adiciona un repositorio.
- El actor adiciona campos al repositorio creado.
- El actor una vez de tener los campos adiciona un proceso con el repositorio que se creó.
- El actor adiciona campos al proceso utilizando los campos físicos creados en el repositorio.
- El actor nombra un formulario a la actividad.
- El actor modela los campos adicionales a una vista en el Formulario.

RESULTADO ESPERADO

Formulario modelado con datos exactos.



EVALUACION DE LA PRUEBA

Prueba satisfactoria.

3.4 ARQUITECTURA DEL SISTEMA

En la Figura 3.14 se describe la Arquitectura del Sistema.

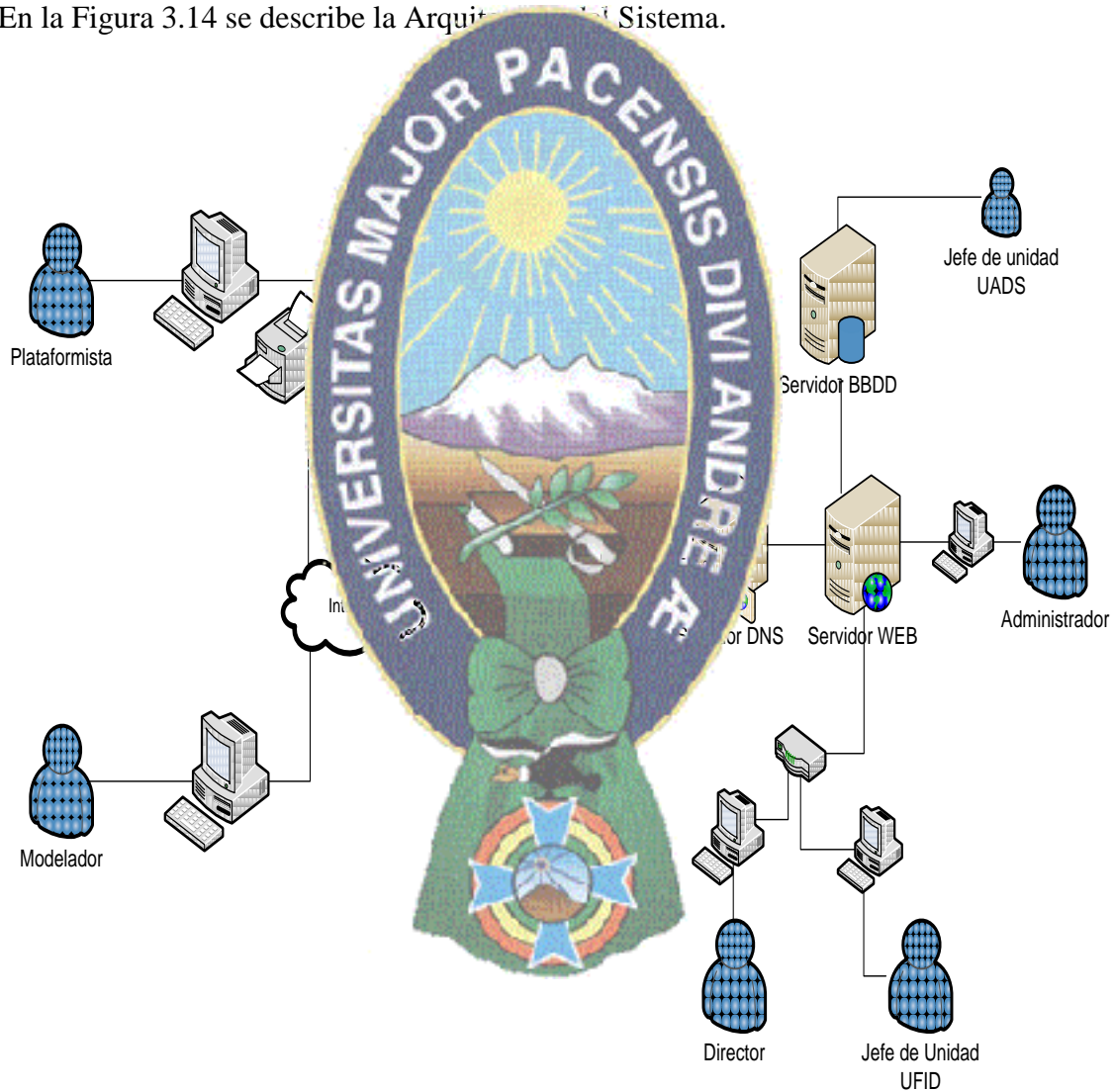


Figura 3.14 Arquitectura del sistema

Fuente: Elaboración propia

3.5 SEGURIDAD Y CONTROL DE ACCESO

3.5.1 CONTROL DE ACCESO

El control de acceso constituye un elemento importante para permitir o no el ingreso al sistema o solo a ciertos directorios generales e incluso el ingreso a ficheros o archivo delicados. Para la realización del proceso de control se lleva a cabo las siguientes acciones.

- La autenticación de usuarios para determinar si una persona está autorizada para llevar a cabo una acción. Una persona ajena al sistema o un usuario no autorizado debe ingresar mediante la interfaz inicial del sistema, debiendo ingresar el nombre de usuario y su contraseña correcta para poder acceder a los recursos del sistema.
- La definición de roles. Cada usuario autorizado, tiene definido un rol, lo que significa que el acceso único a los recursos del sistema y un conjunto de acciones que pueda realizar sobre las tablas de las base de datos.

3.5.2 SEGURIDAD EN LA BASE DE DATOS

La seguridad de la base de datos es implementada en varios niveles en los módulos:

- Seguridad y Acceso
- Generación de Formularios
- Seguimiento de formularios
- Reportes

Todos los ficheros almacenados en la base de datos están protegidos contra escritura por cualquier cuenta que no sea la del súper usuario SQL, el cual es la única vía de conexión permitida a la base de datos. Las conexiones de los clientes al sistema se restringen por dirección IP y por nombre de usuario.

Los usuarios están incluidos en grupos y el acceso a las tablas se encuentra restringido en base a esos grupos (Administración, Seguridad, Seguimiento y Reportes).

3.5.4 SEGURIDAD DE LA INFORMACIÓN

Otro mecanismo de seguridad es el de los respaldos realizados (backups) de la base de datos periódicamente, utilizando un script lanzado como una tarea programada, de esta manera se cuenta con un respaldo en caso de pérdida o alguna incidencia.

COSTO ESTIMADO DEL PROYECTO

El costo estimado del proyecto se determinó utilizando el modelo COCOMO básico donde se utilizan las siguientes variables:



$$E = 28,6 \text{ meses} - \text{hombre}$$

Esfuerzo: $E = 28,6 \text{ meses} - \text{hombre}$

Tiempo de desarrollo: $TDES = 2,5(28,6)^{0,35} = 8,08 \text{ meses}$

Productividad: $PR = \frac{LDC}{E} = \frac{7500}{28,6} = 262,2 \text{ LDC/mes} - \text{hombre}$

Personal Promedio: $P = \frac{E}{TDES} = \frac{28,6}{8,08} = 3,5 \text{ personas}$

Tomando un estimado de 2500 BS de sueldo por persona, se tiene un costo total del proyecto de:

$$\text{Costo} = (\text{num} - \text{pers}) * (\text{tiempo}) * (\text{sueldo}) = 3 * 8,08 * 2500 = 60,000 \text{ BS}$$



CAPITULO 4

CALIDAD DE SOFTWARE

4.1 METRICAS DE CALIDAD

Durante el proceso de evaluación, de acuerdo a los perfiles de usuario, mencionados con anterioridad y con metodología WEB site QEM se evaluara las características usabilidad, funcionabilidad, confiabilidad y eficiencia, características que son las más importantes según Olsina(OLSINA).

4.1.1 FUNCIONALIDAD

El punto función es la métrica de software y del proceso por el cual se desarrolla. Se centra en el programa, los puntos función se calculan realizando un análisis detallado para determinar los siguientes números.

- **Número de entradas** es la cantidad de pantalla de usuario que proporciona al software para interactuar con la aplicación.
- **Numero de salidas** es la cantidad de informes, mensajes de error y toda forma de interacción.
- **Número de peticiones** es la cantidad de peticiones que está definida como una entrada interactiva que resulta de la ejecución de algún tipo de respuesta en forma de salida interactiva.
- **Numero de archivos**, se cuenta el número de archivos maestro lógico.
- **Numero de interfaces externas** es la cantidad de interfaces legibles por el ordenador que son solicitados para transmitir información a otro sistema.



De acuerdo a lo mencionado es que se tiene los resultados en la siguiente tabla:

Entradas de usuario	26
Salidas de usuario	39
Peticiones de usuario	20
Numero de archivos	30
Interfaces externas	42

Tabla 4.1 Tabla síntesis para hallar el Punto función

Fuente: elaboración propia

Los puntos función se hallan en la siguiente fórmula:

$$PF = CUENTA_TOTAL * (GRADO DE CONFIABILIDAD + TASA DE ERROR * \sum Fi)$$

PF = MEDIDA DE FUNCIÓN

CUENTA_TOTAL es la suma de los errores, peticiones, interfaces externas y archivos.

GRADO DE CONFIABILIDAD es la medida de la calidad del sistema.

TASA DE ERROR: probabilidad de pérdida de dominio de la información (1%).

Fi= son los valores de ajuste de complejidad que se toman los valores de la tabla 4.4 y que dan respuesta a los siguientes datos:

Para hallar cuenta total se calcula en la siguiente tabla con los datos obtenidos, considerando un factor de ponderación:

Parámetro de medición	Cuenta	*	Factor de ponderación			Igual	total
			Simpl e	Medio	Complejo		
Entradas de usuario	26	*	4	X4	6	=	104
Salidas de usuario	39	*	3	X6	7	=	234

Peticiones de usuario	20	*	3	X4	6	=	80
Numero de archivos	30	*	7	X10	15	=	300
Interfaces externas	42	*	5	X7	10	=	294
Cuenta total							1012

Tabla 4.2 Tabla con datos obtenidos con el factor de ponderación medio

Fuente: Elaboración Propia

Para hallar la sumatoria de los Fi se debe utilizar la tabla 4.3 respondiendo a las preguntas de acuerdo a los niveles de importancia de 0 al 5 que se muestran en la siguiente tabla.



Factor	Escala
1.- ¿Requiere el Sistema Copias de seguridad de información fiables?	5
2.- ¿Se requiere comunicación de datos?	3
3.- ¿Existen funciones de procesos distribuidos?	2
4.- ¿Es crítico el rendimiento?	3
5.- ¿Sera ejecutado en el SO existente?	3
6.- ¿Requiere el sistema de entradas interactivas?	4
7.- ¿Requiere el sistema entrada de datos interactiva sobre múltiples ventanas?	4
8.- ¿Se actualizan los archivos maestros de forma interactiva?	3
9.- ¿Son complejas las entradas, salidas, los archivos y las peticiones?	2
10.- ¿es complejo el procesamiento interno?	3
11.- ¿Se ha diseñado el código para ser reutilizado?	4

12.- ¿Están incluidas en diseño la conversión y la instalación?	4
13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones?	4
14.- ¿ Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
TOTAL $\sum Fi$	47

Tabla 4.4 Tabla de factores y Valores

Fuente: Propia

Una vez obtenidos los datos y con una confiabilidad mínimo de 75% a continuación calculamos el

$$PF = CUENTA_TOTAL * (G + COEFICIENTE_DE_ERROR * \sum Fi)$$

$$PF = 1012 * (0.75 + 0.01 * 70)$$

$$PF = 1234.6$$

Si consideramos que el número de puntos a $\sum Fi = 70$

$$PF_{Max} = 1012 * (0.75 + 0.01 * 100)$$

$$PF_{Max} = 1467.4$$

Entonces si $\sum Fi$ es considerado de 70, la funcionalidad obtenida entre los puntos será.

$$PF / PF_{Max} = 1234.6 / 1467.4$$

$$PF / PF_{Max} = 0.84$$

Por lo tanto la funcionalidad del sistema es de 0.84.

4.1.2 PORTABILIDAD

Para el sistema desarrollado se tiene que el hardware en el que funciona de manera estable del lado del servidor esta dado por un equipo Microsoft Windows Server R2 el acceso a este servidor es por medio de sesiones de usuario, donde los usuarios autenticados pueden ingresar a la información que el sistema brinda, de acuerdo al rol que tenga. Las terminales de usuario puedan acceder al sistema pueden tener equipos desde Pentium IV.

En cuanto al software este es apto para funcionar bajo distintas plataformas, puede ser Windows, ya que esta aplicación está hecha en el lenguaje de programación CSharp ,ASPX, MVC3 y gestor de base de datos SQL server 2008 los cuales son compatibles con todo tipo de tecnología que nos brindan los sistemas operativos.

Para medir la portabilidad se analizan los siguientes atributos.

Usuario: el sistema es portable, debido a que como estará implementado en la web, el usuario no tendrá inconvenientes para

Servidor: También es portable ya que el instalador del sistema se lo entregara en CD lo cual

A nivel Sistema Operativo el sistema funciona bajo los siguientes sistemas operativos,Microsoft: Windows XP y Windows 2003, Linux.

A nivel de hardware el sistema requiere las características mínimas de hardware: microprocesador de nivel superior, RAM de 128, como mínimo, espacio de disco desde 1 GB.



4.1.3 .MANTENIBILIDAD

4.1.3.1 MANTENIMIENTO

Que se realiza para corregir errores que afectan al usuario final. Para hallar un tiempo medio entre fallas que pueden ocurrir en el software utilizamos la siguiente relación. $TMEF = TMDF + TMC$

Dónde:

TMEF: tiempo medio entre fallas que pueden ocurrir

TMDF: tiempo medio de fallas que ocurrirán.

TMD: tiempo medio de cambio que se tarda.

Para encontrar el valor de TMC se tiene tal siguiente ecuación:

$$TMC=TMAC+TMIC+TMPC+TMDC$$

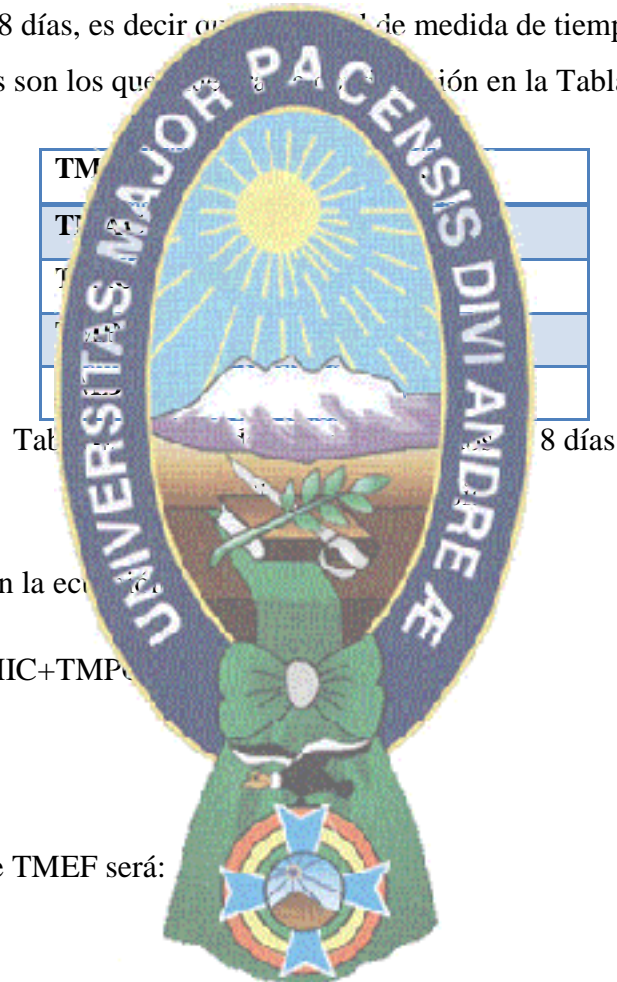
Dónde:

TMAC: tiempo medio de analizar cambios a realizar.

TMIC: tiempo medio de implementar los cambios a realizar.

TMPC: tiempo medio de probar los cambios realizados

Para la obtención de los valores de las anteriores variables se realizó una muestra durante los últimos 8 días, es decir que el periodo de medida de tiempo serán los días los resultados obtenidos son los que se muestran en la Tabla 4.4.



Entonces tenemos en la ecuación

$$TMC = TMAC + TMIC + TMPC$$

$$TMC = 2 + 2 + 2 + 1$$

$$TCM = 7$$

Entonces el valor de TMEF será:

$$TMEF = 8 + 7 = 15$$

Lo cual significa que 15 días es el tiempo estimado medio en el que pueden ocurrir fallas y realizar las respectivas correcciones a estas.

4.1.3.2 MANTENIMIENTO ADOPTIVO.

El mantenimiento adoptivo ocurrirá cuando:

Se cambian las políticas.

Se cambie la estructura organizacional.

Se cambie el personal.

Modificaciones que harán que el sistema cambie en poca o gran magnitud, cambios para los cuales el sistema está preparado para adaptarse a algunos de estos ajustes en la institución, pero para otros más complejos, se deberá hacer una revisión de los procesos y su adaptación con los nuevos cambios que se generen.

4.1.3.3 MANTENIMIENTO

El sistema está completamente actualizado con nuevas funcionalidades de acuerdo a los nuevos requisitos cuando sean relacionados con el servicio e información.

4.1.4 FACILIDAD DE

La medición de la facilidad de uso se realizó a través de la facilidad que el usuario tiene para conocer al sistema, aprenderlo y operarlo. A continuación se presenta la información obtenida en pequeños talleres de capacitación a los



Usuarios	Facilidad de aprendizaje	Facilidad de Operación
Director de la Institución	95%	87%
Encargado de Programa	85%	90%
Usuario	95%	90%
Administrador de usuarios	100%	100%

PROMEDIO	92.5%	93.7%	94.25%
----------	-------	-------	--------

Tabla 4.6 Tabla de valores en porcentajes de Facilidad de usabilidad

Fuente: Elaboración Propia

Por lo tanto de acuerdo a los resultados obtenidos, se puede apreciar que se tuvo una facilidad de uso del sistema del 93,48%



CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

La metodología XP brindó varias ventajas, la planificación basada en las iteraciones dio información actualizada de las versiones y avance del desarrollo del software, para tener un mejor control.

Los conceptos de la técnica de programación (Aprobación de Planos) que se aplicaron en el GAMLP. Además como método es una potente herramienta para analizar los procesos cualquiera sea su definición.

El diseño de base de datos se adaptó a las iteraciones de la metodología XP, teniendo en cuenta la actualización.

Se implementó el módulo de gestión de usuarios, el cual es dinámico y flexible, ya que podemos modelar su estructura de acuerdo a nuestros requerimientos.

Al implementar el sistema la información se almacena de forma más segura y confiable y nos permite tomar normas de seguridad para la Base de datos.

Se implementó un módulo adicional de gestión de usuarios.

Se ha reducido el tiempo empleado en actualización de los datos, gracias al dinamismo de los formularios.

Se desarrolló una interfaz amigable y dinámica para todos los usuarios.



Por todo lo expuesto anteriormente, se puede concluir que el sistema, ha cumplido satisfactoriamente con todos los requerimientos de la unidad de Catastro.

5.2 RECOMENDACIONES

En cuanto a la organización podemos recomendar que en área de Dirección de catastro se pueden ir adecuando los diferentes procesos modelando sus formularios lo cual almacena más datos de diferentes procesos para tener más funcionalidades y mejoras de acuerdo a las necesidades que se van generando en su automatización.

Se recomienda la actualización de los datos que se van generando en la implantación, esto para un correcto funcionamiento del sistema.

Realizar backups periódicos de los datos que se van generando y seguir políticas de seguridad acordadas para el sistema.



Núñez, S. A. (27 de 11 de 2005). *Estándares de seguridad en la información*. Obtenido de <http://www.enterate.unam.mx/Articulos/2005/febrero/seguridad.htm>

Pressman, R. (2002). *Ingeniería de Software*.

Puramagia, D. (09 de 12 de 2011). *SQL Server 2008 R2 – Principios Básicos de SQL Server 2008 R2*. Obtenido de <http://www.daf.com.ar/blog/2011/12/09/sql-server-2008-r2-principios-basicos-de-sql-server-2008-r2/>

qflow. (2013). Obtenido de <http://qflow.esi.uclm.es/doc/pgsi/doc/patrones.pdf>

Q-Flow. (28 de 03 de 2011). *Patrones Basicos de Workflow*. Obtenido de

[http://www.urudata.com/spanish/partners/download/PPQf-](http://www.urudata.com/spanish/partners/download/PPQf-Patrones%20de%20Workflow%20WfM.pdf)

[Patrones%20de%20Workflow%20WfM.pdf](http://www.urudata.com/spanish/partners/download/PPQf-Patrones%20de%20Workflow%20WfM.pdf)

Robert, M. (1990). *Metodología de*

UML. (2013). Recuperado el 5 de mayo de 2013 de [http://kuainasi.ciens.ucv.ve/ads2010-](http://kuainasi.ciens.ucv.ve/ads2010-2/uml/index.html#)

[2/uml/index.html#](http://kuainasi.ciens.ucv.ve/ads2010-2/uml/index.html#)





ANEXO A

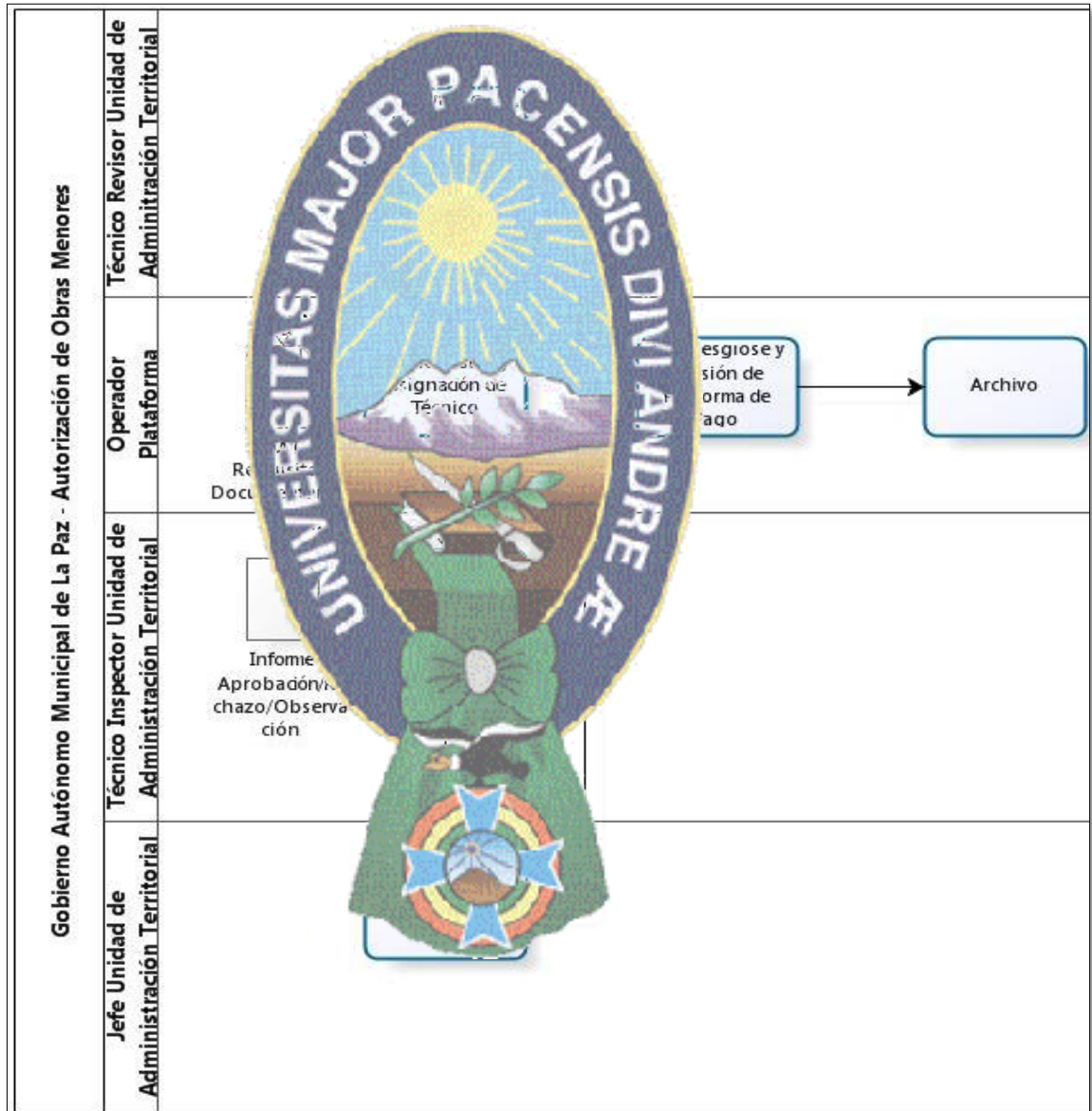


Figura Modelado del Proceso: Autorización de Obras Menores

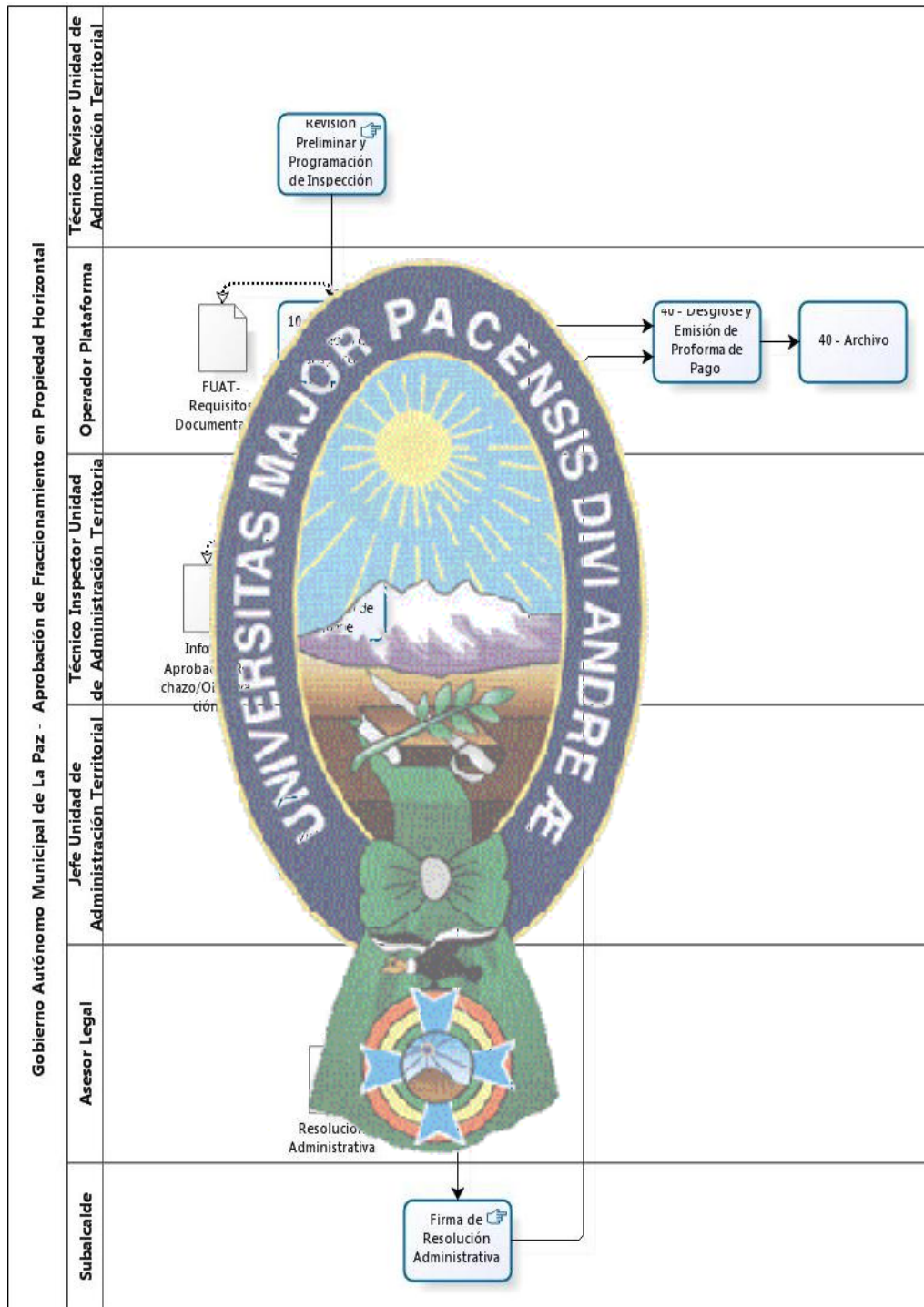


Figura Modelado del Proceso: Aprobación de Fraccionamiento en Propiedad Horizontal

ANEXO B

HISTORIA DE USUARIO MODULO SEGURIDAD Y CONTROL DE ACCESO

HISTORIA DE USUARIO	
Nº 2	Nombre Historia: Seguridad y control de acceso
Usuario: Administrador	Fecha: 02/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 3 Días
<p>Descripción:</p> <p>El inicio de dar alta a un usuario requiere registrar los datos correspondientes al usuario (Datos personales) según su hoja de vida, el usuario Administrador tendrá acceso a toda la información.</p>	
<p>Tareas:</p> <ol style="list-style-type: none"> 1.- Diseñar estructura de datos 2.- Dar de alta a una nueva persona (datos de la persona) 3.- Modificar datos de una persona 4.- Eliminar una persona en especial (si se le haya asignado un usuario) 5.- Implementar interfaz 	

Tabla: Historia de Usuario - Seguridad - Personas

HISTORIA DEL USUARIO	
Nº 3	Nombre Historia: Seguridad y control de acceso – Usuarios
Usuario: Administrador	Fecha: 02/03/2013
Prioridad en negocio: Alta	Tiempo estimado: 3 Días
<p>Descripción:</p> <p>Para realizar la asignación de usuarios se requiere previamente crear una persona (datos</p>	

correspondientes al usuario), para luego poder proporcionarle los accesos necesarios.

Tareas:

- 1.- Diseñar estructura de base de datos del módulo - Usuarios
- 2.- Dar de alta a un nuevo Usuario
- 3.- Modificar datos de un usuario existente
- 4.- Eliminar un usuario en específico (validar que no se le hayan asignado roles)
- 5.- Implementar interfaz

Tabla: Historia del usuario Seguridad - Usuarios

HISTORIA DEL USUARIO	
Nº 4	Objetivo: Seguridad y control de acceso
Usuario: Administrador	Fecha: 13
Prioridad en negocio: Alto	Tiempo: 3 Días
Descripción: Para realizar la asignación de roles, se debe crear un usuario, cada rol tendrá permisos a los módulos que le sean asignados.	
Tareas:	
1.- Diseñar estructura de base de datos del módulo - Roles	
2.- Dar de alta a un nuevo Rol	
3.- Adicionar accesos a un Rol	
3.- Modificar Rol	
4.- Eliminar Rol	
5.- Implementar interfaz	

Tabla: Historia del usuario Seguridad - Roles

ANEXO C

CASOS DE USO DEL SISTEMA

SEGURIDAD Y ACCESOS

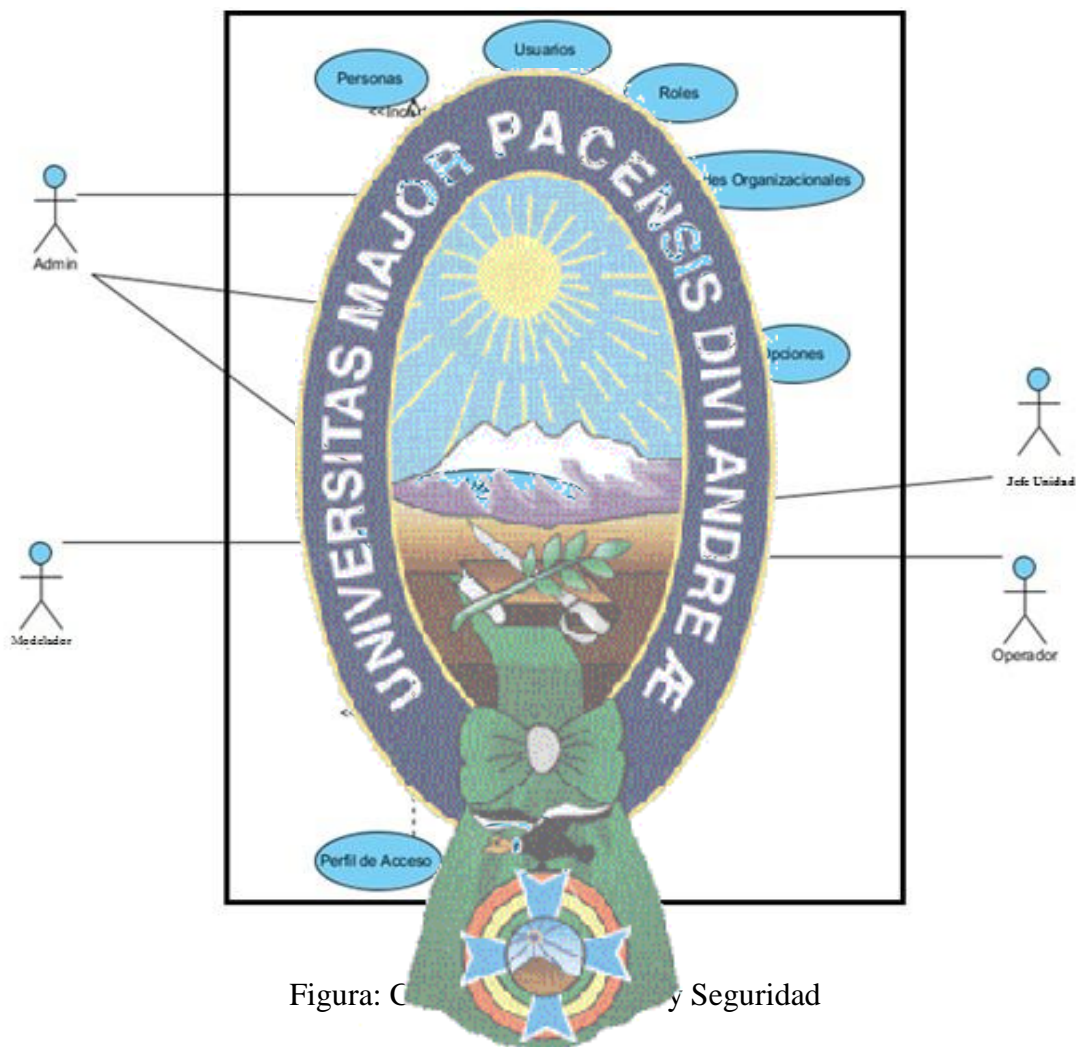



Figura: C y Seguridad



Figura: Caso de uso de Gestión de Acceso y Seguridad

GESTION DE ACCESO	
Nombre	
Autor	
Fecha	
DESCRIPCION	
El administrador puede	manejar diferentes usuarios
ACTORES	
 Admin	
PRECONDICIONES	
El admin deberá ser registrado en el sistema con todos los privilegios	
FLUJO NORMAL	
7. El actor Admin introduce su nombre y contraseña pulsa ingresar 8. El sistema realiza una comparación de usuario rol, permisos y la unidad organizacional a la que pertenece el usuario 9. El sistema muestra el menú de opciones y accesos que este usuario tiene. 10. El actor admin selecciona del menú de opciones usuarios y adiciona un usuario 11. El sistema responde con un mensaje adicionado 12. El actor admin selecciona del menú de opciones grupo y puede adicionar un grupo	

13. El sistema responde con un mensaje adicionado
FLUJO ALTERNATIVO
14. El sistema no encuentra al usuario no lo deja ingresar y devuelve al punto de inicio
POSTCONDICIONES
Accesos según el rol

Tabla: Descripción de Acceso y Seguridad



Figura: Diagrama de Acceso y Seguridad

GESTION DE SEGURIDAD	
Nombre	Ge...
Autor	Elabora...
Fecha	09-Ag...
DESCRIPCION	
El administrador puede admin... mediante la asignación de roles y unidades organizacionales.	
ACTORES	
Administrador	Administrador del sistema
PRECONDICIONES	


El administrador deberá ser registrado en el sistema, con todos los privilegios de súper usuario	
FLUJO NORMAL	
<ol style="list-style-type: none"> 1. El actor Admin introduce su nombre de usuario y contraseña pulsa ingresar 2. El sistema realiza una comprobación de usuario rol, permisos y la unidad organizacional a la que pertenece el usuario 3. El sistema muestra el menú de accesos que este usuario tiene. 4. El actor admin selecciona una persona y adiciona una persona 5. El sistema responde con éxito 6. El actor admin selecciona un usuario y adiciona un usuario 7. El sistema responde con éxito 8. El actor admin selecciona un rol y adiciona un nuevo rol 9. El sistema responde con éxito 10. El actor admin selecciona una unidad organizacional y adiciona una nueva unidad organizacional 11. El sistema responde con éxito 	
FLUJO ALTERNATIVO	
12. El sistema no encuentra el usuario y devuelve al punto de inicio	
POSTCONDICIONES	
Accesos según el rol	

Tabla: Descripción Caso de uso Acceso y Seguridad

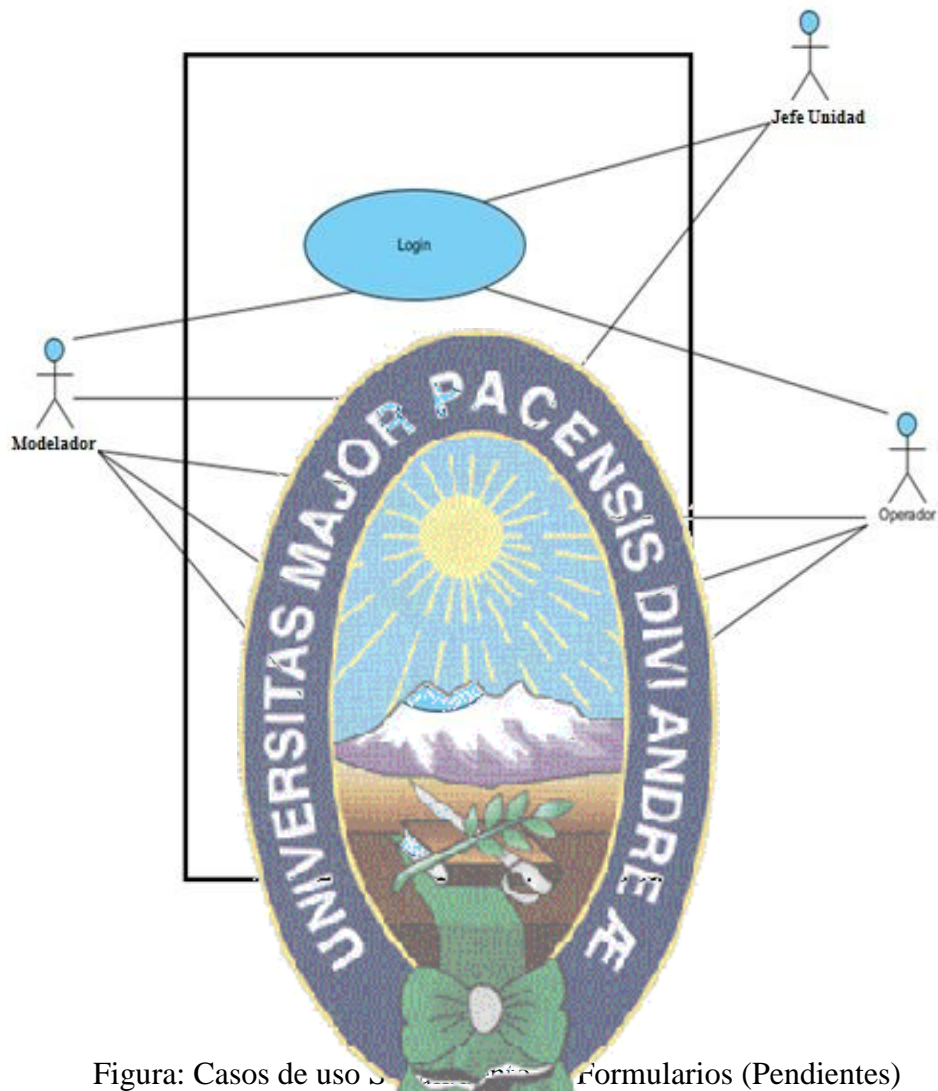


Figura: Casos de uso de Formularios (Pendientes)

CONTEXTO	
Nombre	Com...
Autor	Elaboración propia
Fecha	09-Agosto-2013
DESCRIPCION	
El arquitecto, Operador, Autoridad ingresan a un ambiente de trabajo con el fin de obtener los flujos lo más actualizado posible para no tener inconvenientes.	
ACTORES	

 Modelador	Este actor representa al usuario que realiza el modelado de procesos.
 Operador	Este actor representa al usuario que actúa sobre los procesos modelados
 Jefe de Unidad (Autoridad)	Este actor representa al usuario que autoriza el proceder de una actividad
PRECONDICIONES	
El usuario debería haberse autenticado.	
FLUJO NORMAL	
<ol style="list-style-type: none"> 1. El actor Arquitecto realiza la acción de seleccionar un proceso y el sistema realiza la acción de validar los permisos que este tiene dentro de los permisos que este tiene. 2. El sistema verifica si el usuario tiene acceso a la acción seleccionada. 3. El sistema responde con un mensaje de éxito. 4. El usuario realiza la acción de validar los permisos. 	
FLUJO ALTERNATIVO	
5. El sistema realiza la validación de los permisos y el rol que desempeña.	
POSTCONDICIONES	
se almacena los datos del usuario y se realiza el proceso que realizo dentro el sistema	

Tabla: Descripción del Proceso Contexto



MODELADOR

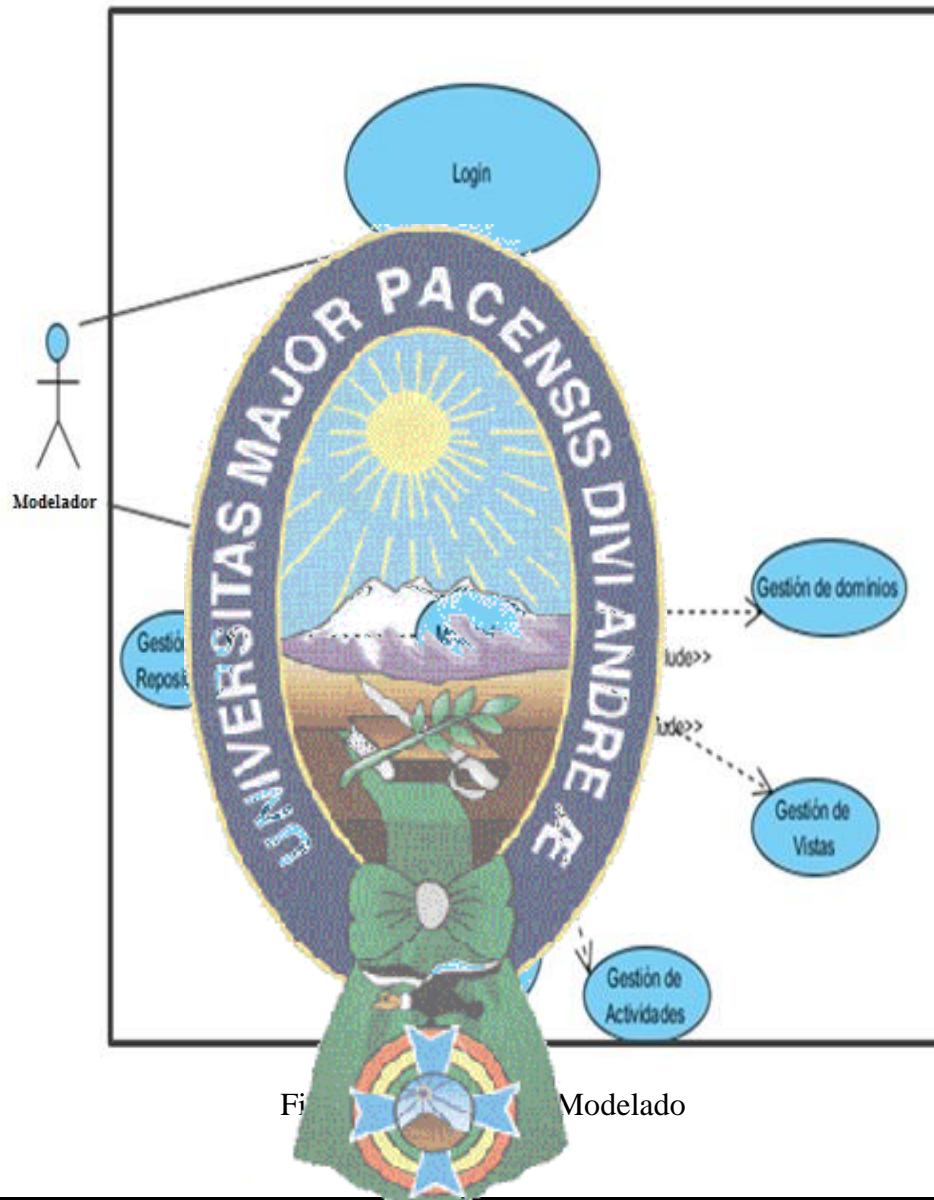



Fig. Modelado

MODELADO	
Nombre :	Modelado
Autor :	Elaboración propia
Fecha:	09-Agosto-2013
DESCRIPCION	

Permite modelar los formularios dinámicos.
ACTORES
 Modelador
PRECONDICIONES
El usuario tiene que estar registrado en el sistema.
FLUJO NORMAL
<ul style="list-style-type: none"> • El actor ingresa al sistema con su usuario y contraseña pulsa aceptar. • El sistema verifica si el usuario está registrado en el sistema. • El sistema le muestra los repositorios. • El actor selecciona un repositorio y elige la acción repositórios. • El actor adiciona un campo en la tabla de repositórios. • El actor adiciona un campo a un formulario. • El actor una vez se ha creado un formulario adiciona un proceso con el repositorio. • El actor adiciona campos físicos utilizando los campos físicos creados en el repositorio. • El actor nombra un formulario con una actividad. • El actor modela los campos físicos en la vista en el Formulario. • El actor una vez terminado el formulario podrá visualizar el formulario con la con la creación de un proceso con el nombre del proceso creado.
FLUJO ALTERNATIVO
<ol style="list-style-type: none"> 1. El actor no podrá ingresar al sistema en caso que este no este registrado en el sistema. 2. En caso que no se tenga ningún campo adicionado en el formulario no podrá visualizar nada en el formulario.
POSTCONDICIONES

El actor podrá ingresar al sistemas según al rol y los accesos asignados desde el administrador.

Tabla: Descripción Casos de uso Modelado

SEGUIMIENTO DE FORMULARIOS

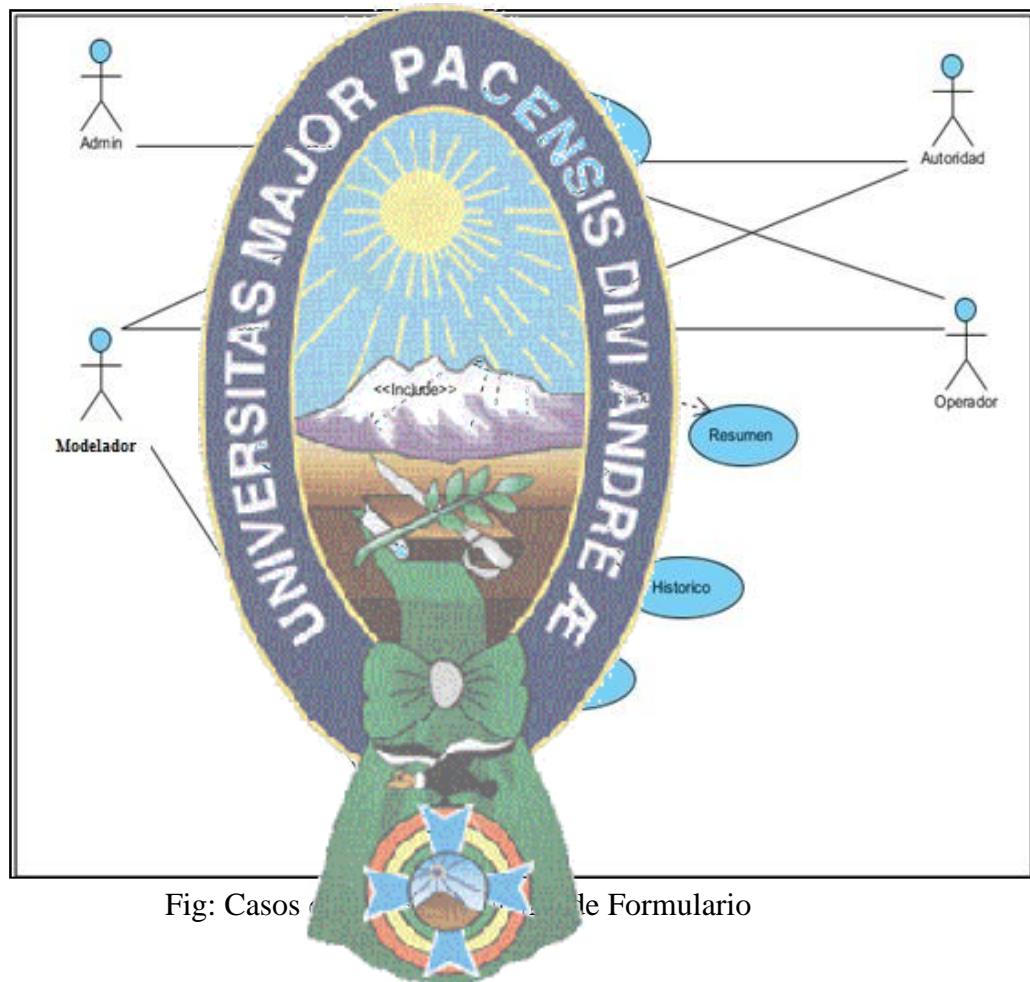






Fig: Casos de uso de Formulario

SEGUIMIENTO DE FORMULARIO	
Nombre	Pendientes
Autor	Unidad de Administración de Desarrollo de Sistemas.
Fecha	09-Agosto-2013
DESCRIPCION	
El Administrador, Arquitecto, Operador, Autoridad deberán tener acceso a los trámites pendientes del sistema.	
ACTORES	
 Modelador (Arquitecto)	
 Administrador	
 Operador	
 Jefe de Unidad (A)	
PRECONDICIONES	
El tramite debe existir c	
FLUJO NORMAL	
<ol style="list-style-type: none"> 1. El actor operador desp 2. El sistema verifica si el a 3. El actor selecciona la opción. 4. El sistema muestra una bandeja de trámites (trámites pendientes, tramites observados, nuevos tramites 5. El actor selecciona el trámite el icono que representa trámites (pendientes, recibidos, observados por firma) en la bandeja. 6. El sistema muestra el formulario en el cual puede actuar al mismo tiempo mostrara un menú de acceso al resumen, histórico, adjuntos. 7. El actor puede llenar, borrar, modificar, cargar, guardar en el formulario. 8. El sistema guarda todas las modificaciones realizadas en el formulario. 9. El actor podrá enviarlo a otro rol con la opción avanzar. 10. El sistema envía el trámite a otro rol. 	



11. El actor puede regresar a la bandeja de trámites o salir de su sesión.
FLUJO ALTERNATIVO
12. El actor no ingresa al sistema si este no está registrado en el sistema.
13. El sistema no acepta si el usuario o la contraseña no coinciden.
POSTCONDICIONES
El actor tendrá acceso según a los roles que se le asigne al rol.
El actor administrador es el único que tiene el completo control del sistema.

Tabla: Descripción de los casos de uso de Formulario

REPORTES

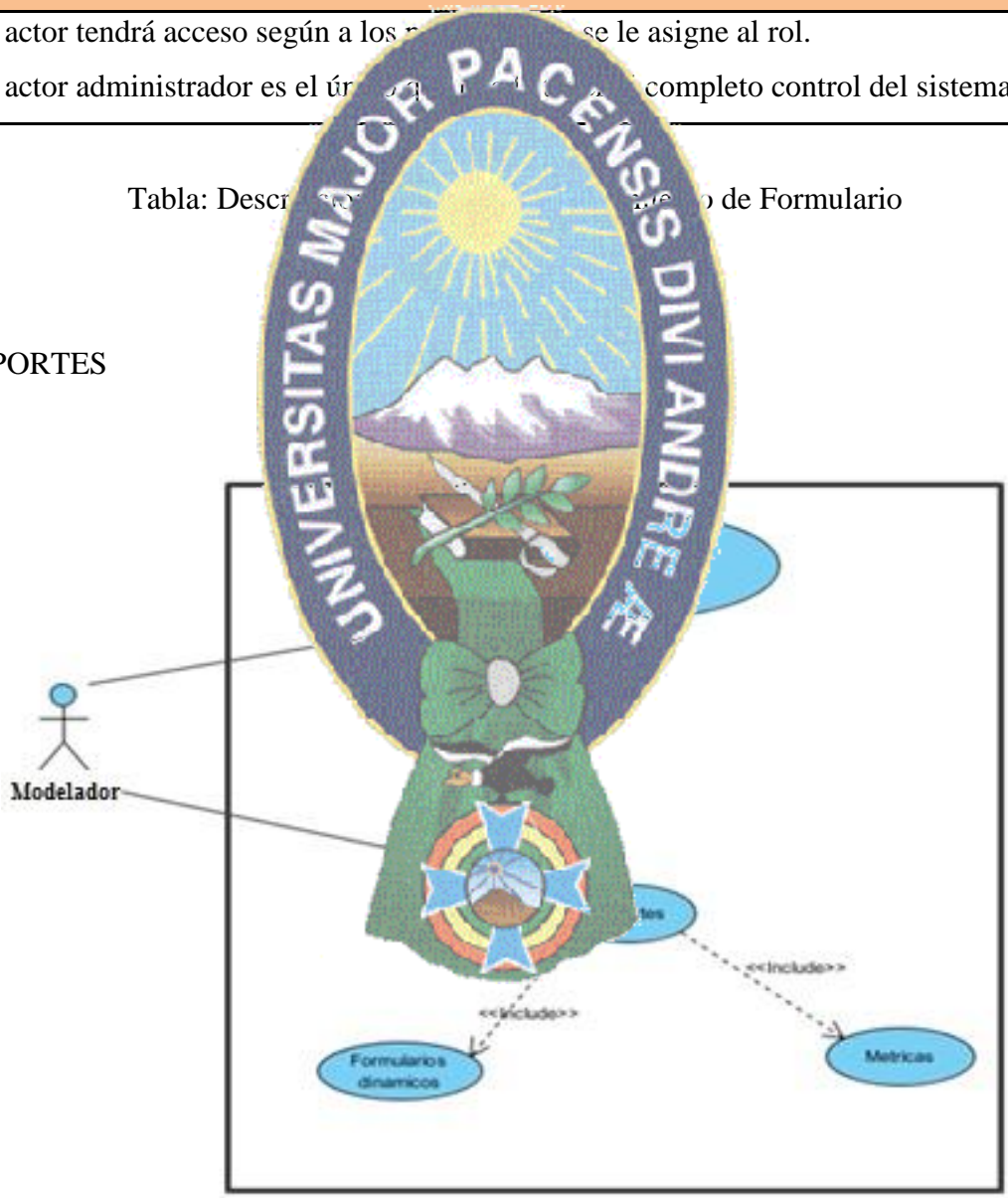


Figura: Casos de uso Reportes


REPORTES	
Nombre:	Reportes
Autor:	Unidad de Administración de Desarrollo de Sistemas.
Fecha:	09-Agosto-2013
DESCRIPCION	
Permite realizar reportes dinámicos de trámites activos o procesos activos.	
ACTORES	
 Modelador	
PRECONDICIONES	
El usuario tiene que estar registrado.	
FLUJO NORMAL	
<ol style="list-style-type: none"> 1. El actor ingresa al sistema con el nombre de usuario y contraseña con el cual fue registrado. 2. El sistema verifica los datos ingresados y permite el acceso al sistema. 3. El sistema le muestra el menú de opciones al actor. 4. El actor selecciona la opción Reportes. 5. El actor podrá ver los reportes de trámites activos o procesos activos. 6. El sistema le mostrará los reportes de trámites activos o procesos activos en el menú Reportes. 7. El actor selecciona el reporte que desea visualizar. 8. El sistema muestra el reporte de trámites activos o procesos activos. 9. El actor visualiza el reporte de trámites activos o procesos activos. 	
FLUJO ALTERNATIVO	
<ol style="list-style-type: none"> 1. El actor no podrá ingresar al sistema si no está registrado. 2. El actor no podrá ver reportes si este no tiene asignado la acción. 	
POSTCONDICIONES	
El usuario podrá visualizar a las Reportes siempre y cuando este tenga el acceso asignado a su rol.	

Tabla: Descripción Casos de uso Reportes

DETALLE DE LAS HISTORIAS DE USUARIO

- 1.- Seguridad y Control de Acceso
- 2.- Seguridad y control de acceso – Personas

Sistema Modelador de Formularios

USUARIOS

ADICIONAR

Persona: Admin Admin
Usuario: marcelo
Clave: condori
Controlar ip: S
Unidad Organizacional: DIRECC
Espacio Físico: Calle
Correo:

- 3.- Seguridad y control de

Sistema Modelador de Formularios

ROLES

#	ROL	ACCION
1	Super Usuario	
2	PLATAFORMA - INGRESO	
3	TECNICO UAT	

ADICIONAR

ROL: PLATAFORMA
PERMISO FIRMA: VACIO
PERMISO CREAR TRAMITE: NO
Grabar Cancelar

Figura: Pantalla de registro de Usuario Rol

- 4.- Seguridad y control de acceso – Roles

81	<input type="radio"/>	RHUANIRI	prueba	S	SUB ALCALDIA MD7CENTRO	15	Dependiente	A	⌵
82	<input checked="" type="radio"/>	WILBER.SOLIZ	prueba	S	SUB ALCALDIA MD7CENTRO	15	Dependiente	A	⌵
83	<input type="radio"/>	ACALLE	prueba	S	SUB ALCALDIA MD7CENTRO	15	Dependiente	A	⌵
84	<input type="radio"/>	CARISMENDY	prueba	S	SUB ALCALDIA MD7CENTRO	15	Dependiente	A	⌵

ADICIONAR

Usuario: WILBER.SOLIZ

Rol:

Figura: Pantalla de Asignación de Usuario – Rol

6.- Creación de Campos Físicos

#	REPOSITO	ACCION
1	f01_fraccionamiento_pr...	A
2	f01_autorizaciones_mer...	A
3	f01_aprobacion_planos...	A
4	f01_aprobacion_planos...	A
5	f01_informe_tecnico_ad...	A
6	f01_permiso_construcci...	A

#	CAMPO	ACCION
472	f01_titulo_form_adm_territori...	A
473	f01_subtitulo_datos_grales_L...	A
474	f01_paterno	A
475	f01_materno	A
476	f01_nombre	A

Figura: Creación de Campos Físicos

7.- Adición de Proceso

Sistema Modelador de Formularios

SEGUIDAD MILESCRITORIO MODELADO

PROCESO ADICIONAR

Tipos Sistema: 01TS - Sistemas Adminis

Nombre de Sistema: --Seleccione un Sist

Servicio: Seleccione sist

Código Proceso: A01

Proceso: APROBACION

Repositorio Físico: f01_aproba

11_aprobacion_planos

Cancelar

8.- Campos Lógicos

Sistema Modelador de Formularios

SEGUIDAD MILESCRITORIO M.

	TITULO CAMPO	FILA COLUMNA	TIPO CAMPO	DOMINIO	E OBSERVACION
<input type="checkbox"/>	DATOS TECNICOS	370	1 varchar	UNVERSO A	
<input type="checkbox"/>	Nº DE INFORME TECNICO	380	1 varchar	UNVERSO A	
<input type="checkbox"/>	FECHA DE INFORME	380	2 varchar	UNVERSO A	
<input type="checkbox"/>	DJUNTOS	360	3 varchar	UNVERSO A	
<input type="checkbox"/>	TRON DE ENTAMIENTO	370	2 varchar	UNVERSO A	
<input type="checkbox"/>	NUMERO DE TRAMITE	380	3 varchar	UNVERSO A	

Figura: Pantalla Listar Campo lógicos

9.- Creación de Actividades

Sistema Modelador de Formularios

SEGURIDAD ESCRITORIO MODELADO

#	ORDEN - ACTIVIDAD	ROL	ANT	E	I	
1	10 - FILIACION Y ENVIO DE DOCUMENTOS	PLATAFORMA - INGRESO				A
2	20 - INSPECCION Y ELABORACION DE INFORME	TECNICO UAT	10			A
3	30 - ELABORACION RESOLUCION ADMINISTRATIVA		20			A
4	40 - FIRMA Y SELLO DE INFORME		30			A
5	50 - PARA ENTREGA AL CIUDADANO		40			A
6	60 - ARCHIVO DE GESTION		50			A
7	50000 - RESUMEN					A
8	80002 - REPORTES/MONITOREO					A
9	90020 - BUSQUEDA					A

10.- Vistas

FORMULARIOS

Doplar Campos a: 10-FILIACION

Cambiar Tipo de Representacion: Lectura

C	#	CAMPO	TIPO	TIPO CONTROL	COD.	CONT. DEFECTO
<input type="checkbox"/>	51	f01_nro_fojas_total	Text			
<input type="checkbox"/>	52	f01_descripcion	Text			
<input type="checkbox"/>	1	f01_titulo_form_adm_territorial	Lectura	Title		
<input type="checkbox"/>	35	f01_titulo_frae_prop_horizontal	Lectura	Title		
<input type="checkbox"/>	2	f01_subtitulo_datos_grales_usuario	Lectura	Subtitle		
<input type="checkbox"/>	20	f01_macrodistrito	Obligatorio	Dropdown List	*	
<input type="checkbox"/>	21	f01_distrito	Esritura	Dropdown List		
<input type="checkbox"/>	19	f01_codigo_catastral	Esritura	Text		

Figura: Pantalla Listar Vistas

11.- Dominios

Sistema Modelador de Formularios

SEGURIDAD ESCRITORIO MODELADO

DOMINIO	CONTENIDO	CONTENIDO_SQL	Est
1 UNIVERSO			A
5 TIPO DOCUMENTO GT	01-SELECCIONAR-#CII#NITIT#RUNI#PASAPORTE#PASAPORTE#LIBRETA SERVICIO MILITAR#LIBRETA SERVICIO MILITAR		A
7 MACRODISTRITO GT		select macrodistrito, macrodistrito from macrodistrito_gt	A
8 LUGAR EXPEDIDO GT	01-SELECCIONAR-#LPZ#LPZ#ORU#ORU#PTS#PTS#CBB#CBB#CHQ#CHQ		A
9 TIPO INFORME GT	01-SELECCIONAR-#API#APROBADO#RECHI#RECHAZADO#OBS/O		A
0 TECNICOS GT		select id_usuario, tecnico, macrodistrito from tecnicos where _estado = CHAR(85)	A
1 HORARIO_GT		select id, HORA from HORARIO	A
2 TIPO INMUEBLE GT	01-SELECCIONAR-#1#TERRENO#2#UNIFAMILIAR#3#DE SERVICIO#12#CEMENTERIO#13#MAUSOLEO#14#		A
7 LUGAR EXPEDIDO GT	01-SELECCIONAR-#LPZ#LPZ#ORU#ORU#PTS#PTS#		A

NOMBRE HISTORIA SEGURIDAD CONTROL DE ACCESO – PERSONAS

TAREAS

- 1.- Diseñar estructura de datos para el módulo – Personas.
- 2.- Dar de alta a una nueva persona.

Registrar Persona.- esta tarea permitirá registrar a la persona con sus datos personales para usarlos en el sistema.



LISTAR PERSONAS										ADICIONAR		
#	NOMBRES	PATERNO	MATERNO	CI	DIRECCION	TELEFONO	CELULAR	SEXO	CORREO	FEC-NACIMIENTO		
1	Administrador	Admin	Admin	1				M		19/12/2002 0:00:00		
2	MONITOREO	DESCONCENTRACION		1				M		19/12/2002 0:00:00		
3	Ariana	Peñaloza	Flores	1				M		19/12/2002 0:00:00		
4	Roberto	Valdenassi	Valdenassi	1				M		19/12/2002 0:00:00		
5	Jimena	Trujillo	Trujillo	1				M		19/12/2002 0:00:00		
6	Sergio	Gutierrez	Gutierrez	1				M		19/12/2002 0:00:00		
7	Luis	Tarqui	Tarqui	1				M		19/12/2002 0:00:00		
8	Mariana	Paredes	Pico	1				M		19/12/2002 0:00:00		
9	Alfonso	Trujillo	Trujillo	1				M		19/12/2002 0:00:00		

NOMBRE:

PATERNO:

MATERNO:

CI:

EXPEDIDO: LPZ

DIRECCION:

TELEFONO:

CELULAR:

SEXO: F

CORREO:

FEC NACIMIENTO:

Grabar Cancelar

de Personas

- 3.- Modificar datos
- 4.- Eliminar una persona (no se le haya asignado un usuario)
- 5.- Implementar

PRUEBAS DE ACCESO AL USUARIO

Seguridad y control de acceso

- a) **Identificar todos los posibles caminos de la Historia**
Información requerida por el sistema para el acceso.
- b) **Identificar los resultados de la historia y los que permiten continuar dentro de la historia**
Termina la historia cuando el usuario es dado correctamente, caso contrarios no podrá tener permisos de acceso.
- c) **Identificar los caminos posibles de ejecución**
Se realizara el registro sin ningún problema en el caso de dar de baja a uno de los usuarios se le debe dar de baja también a todos los accesos del usuario.
- d) **Asignar un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.**

El conjunto de valores nos permitirá acceder a diferentes tablas para llevar el control de acceso en el sistema.

e) Eliminar caminos redundantes

No existen caminos redundantes.

NOMBRE HISTORIA: SEGURIDAD Y CONTROL DE ACCESO – USUARIOS

TAREAS

- 1.- Diseñar estructura de roles – Usuarios.
- 2.- Dar de alta a un usuario.
- 3.- Modificar datos de un usuario.
- 4.- Eliminar un usuario (se le hayan asignado roles)
- 5.- Implementar roles.

PRUEBAS DE ACEPTACION DE LA HISTORIA USUARIO

- a) Identificar todos los posibles caminos de la historia**
Información que debe ser registrada para poder adicionar los roles necesarios dentro del sistema
- b) Identificar los resultados que permiten finalizar la historia y los que permiten continuar dentro de la historia**
Termina la historia cuando es registrado correctamente el rol, caso contrario no se contara con los roles necesarios.
- c) Identificar los caminos posibles de la historia**
Se realizara el registro sin ningún problema, en el caso de dar de baja a uno Rol, se le debe verificar si ningún usuario fue asignado ese rol.
- d) Asignar un conjunto de valores validos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.**
El conjunto de valores nos permitirá acceder a diferentes tablas para que de esta manera se pueda llevar el control de acceso en el sistema



e) Eliminar caminos redundantes

No existen caminos redundantes.

NOMBRE HISTORIA: SEGURIDAD Y CONTROL DE ACCESO – ROLES

TAREAS

1.- Diseñar estructura de base de datos de Rol – Roles.

2.- Dar de alta a un nuevo Rol

81	<input type="radio"/>	RHUANIRI	prueba	S	15	Dependiente	A	✓
82	<input checked="" type="radio"/>	WILBER.SOLIZ	prueba	S	15	Dependiente	A	✓
83	<input type="radio"/>	ACALLE	prueba	S	15	Dependiente	A	✓
84	<input type="radio"/>	CARISMENDY	prueba	S	15	Dependiente	A	✓

ADICIONAR

Usuario: WILBER.SOLIZ

Rol: Super U

Grabar Canceiar

Figura 1.1 Interfaz de Usuario - Rol

3.- Adicionar accesos a un Rol

4.- Modificar Rol

5.- Eliminar Rol

6.- Implementar interfaz

PRUEBAS DE ACEPTACION PARA EL USUARIO

a) Identificar todos los posibles resultados de la historia

Información que debe ser registrado para que el usuario cuente con un rol o varios roles dentro del sistema.

b) Identificar los resultados que terminen la historia y los que permiten continuar dentro de la historia

Termina la historia cuando es registrado correctamente el rol, caso contrarios no podrá tener permisos de acceso.

c) Identificar los caminos posibles de ejecución

Se realizara el registro sin ningún problema, en el caso de dar de baja a uno Rol, se le debe verificar si ningún usuario fue asignado ese rol.

d) Asignar un conjunto de valores válidos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.

El conjunto de valores nos permitirá acceder a diferentes tablas para que de esta manera se pueda llevar el control de acceso en el sistema.

e) Eliminar caminos redundantes

No existen caminos redundantes.

NOMBRE HISTORIA: **REPOSICIONAR DATOS DE LA TABLA DE CAMPOS FISICOS.**

TAREAS

- 1.- Diseñar estructura de los datos de la tabla de campos físicos.
- 2.- Desarrollar una interfaz para el campo físico.
- 3.- Implementar interfaz para el campo físico.

PRUEBAS DE ACEPTACION

a) Identificar todos los posibles caminos de ejecución de la historia

Datos requeridos por el sistema para los campos físicos creados en repositorio.

b) Identificar los resultados que se obtienen de la historia y los que permiten continuar dentro de la historia

Creamos campos físicos y terminamos con los nombres del campo.

c) Identificar los caminos posibles de ejecución

Ingresamos desde el listado de repositorio y seleccionamos uno para adicionar campos físicos en el repositorio.

d) Asignar un conjunto de valores validos y valores del entorno a cada camino de ejecución para obtener el resultado esperado.



El conjunto de valores esta dado por el valor asignado el repositorio para q de esta manera los campos sean identificados en un solo repositorio.

e) Eliminar caminos redundantes

No existen caminos redundantes.

NOMBRE HISTORIA: ADICIÓN DE PROCESO

TAREAS

- 1.- Diseñar la estructura de base de datos para almacenar un proceso.
- 2.- Desarrollar las interfaces de usuario para crear un proceso.
- 3.- implementar las interfaces de usuario para consultar un proceso.

PRUEBAS DE ACEPTACIÓN

a) Identificar todos los caminos posibles

Nos permitirá adicionar procesos desde la interfaz principal de procesos.

b) Identificar los resultados esperados y los que permiten continuar dentro de la historia

Termina la historia cuando se realice la selección correcta del proceso.

c) Identificar los caminos posibles

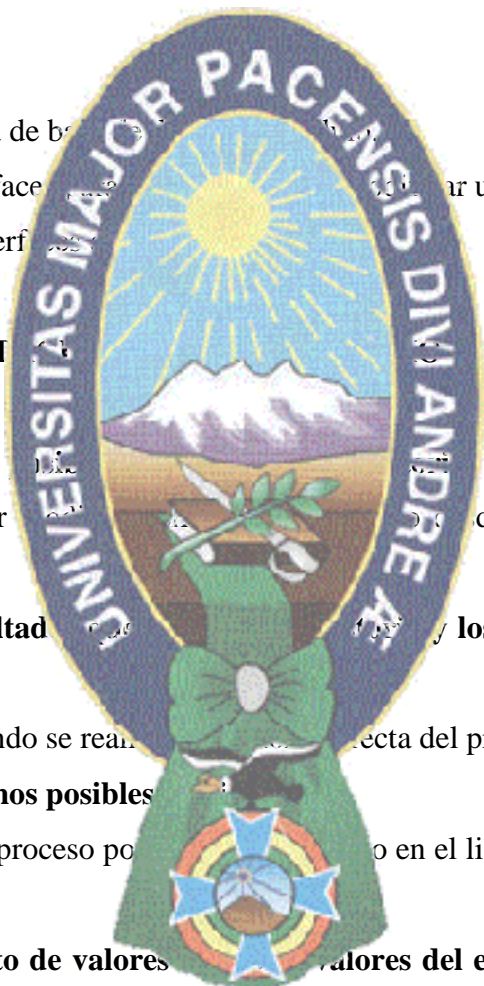
Mediante la adición de proceso por medio de la interfaz de usuario en el listado de nuestra tabla listar proceso.

d) Asignar un conjunto de valores de los valores del entorno a cada camino de ejecución para obtener el resultado esperado.

El conjunto de valores está dado por los datos recuperados de la tabla repositorios el cual será designado a un proceso.

e) Eliminar caminos redundantes

No existen caminos redundantes.



NOMBRE HISTORIA: AGREGACIÓN DE CAMPOS LÓGICOS

TAREAS

- 1.- Diseñar estructura de base de datos del modulo.
- 2.- Desarrollar una interfaz para diseñar la agregación de los campos creados en repositorio.
- 3.- Implementar las interfaces para la creación, eliminación, modificación, listar los campos Lógicos ver.

PRUEBAS DE ACEPTACIÓN

- a) Identificar todos los caminos**
Campos requeridos para la ejecución de un camino específico.
- b) Identificar los resultados que permiten continuar dentro de la historia**
Termina la historia cuando se encuentra el campo lógico en el proceso correspondiente.
- c) Identificar los caminos repetidos**
Mediante el registro de algunos caminos permitirá controlar campos repetidos los cuales tienen que ser eliminados.
- d) Asignar un conjunto de valores del entorno a cada camino de ejecución para obtener el resultado**
El conjunto de valores está dado por los valores de la tabla de campos físicos de un repositorio.
- e) Eliminar caminos redundantes**
No existen caminos redundantes.



NOMBRE HISTORIA: CREACIÓN DE ACTIVIDADES

TAREAS

- 1.- Diseñar estructura de base de datos de la tabla actividades.

- 2.- Desarrollar las interfaces para listar, adicionar, eliminar, modificar de actividades.
- 3.- Implementar las interfaces para adición eliminación y modificación de actividades

PRUEBAS DE ACEPTACION HISTORIA USUARIO

Actividades

a) Identificar todos los posibles resultados de la Historia

Las actividades requeridas para modelar un proceso específico.

b) Identificar los resultados que permiten continuar dentro de la historia y los que permiten continuar

Termina la historia cuando se ejecuta una actividad. En caso contrario revisamos el módulo de actividades.

c) Identificar los caminos de ejecución

Mediante el registro de actividades se realizará la adición de actividades únicas de un proceso.

d) Asignar un conjunto de valores de actividad dentro de cada camino de ejecución para obtener

El conjunto de valores está determinado por la tabla de actividades.

e) Eliminar caminos redundantes

No existen caminos redundantes.



NOMBRE HISTORIA: VISTAS

TAREAS

- 1.- Diseñar estructura de base de datos para Vistas.
- 2.- Desarrollar las interfaces para listar, adicionar eliminación, modificación de vistas.
- 3.- Implementar las interfaces para la adición, eliminación, modificación de Vistas.

PRUEBAS DE ACEPTACION HISTORIA USUARIO

a) Identificar todos los posibles resultados de la Historia

Datos requeridos por el sistema para realizar el modelado del formulario a través de las vistas que se visualizaran en el formulario.

b) Identificar los resultados que terminen la historia y los que permiten continuar dentro de la historia

Termina la historia cuando es asignado correctamente las vistas necesarias en una actividades caso contrario revisamos el modelado hasta que nos muestre en el formulario.

c) Identificar los caminos posibles

Mediante el registro de vistas se permitirá visualizar los campos seleccionados en el formulario.

d) Asignar un conjunto de valores de la tabla de vistas entorno a cada camino de ejecución para obtener

El conjunto de valores e la tabla de vistas de una actividad.

e) Eliminar caminos reducidos

No existen caminos reducidos.



NOMBRE HISTORIA: D

TAREAS

- 1.- Diseñar estructura de base de datos y dominios.
- 2.- Desarrollar las interfaces para agregar y modificar dominios.
- 3.- Implementar el modulo con la funcionalidad de agregar, modificar, adicional, eliminar.

PRUEBAS DE ACEPTACION HISTORIA USUARIO

a) Identificar todos los posibles resultados de la Historia

Los datos son almacenados en dominios serán requeridos por los campos con especial comportamiento.

b) Identificar los resultados que terminen la historia y los que permiten continuar dentro de la historia

Termina la historia cuando se haya registrado correctamente el dominio caso contrario lo revisamos en el código fuente.

c) Identificar los caminos posibles de ejecución

Mediante los campos el sistema nos permitirá realiza campos repetidos los cuales tienen que ser únicos en nuestras tablas.

d) Asignar un conjunto de valores del entorno a cada camino de ejecución para obtener el resultado

El conjunto de valores esta en la tabla dominios.

e) Eliminar caminos reducidos

No existen caminos reducidos

