

Introducción

En las organizaciones institucionales, usualmente generan sus propios documentos, almacenándolos en carpetas, en discos duros de PC o en algún lugar del universo del e-mail. Actualmente, la información de esos documentos es extremadamente valiosa es la base para el trabajo cotidiano, algo esencial para el éxito de cualquier organización. Es incomprensible que se siga perdiendo tiempo buscando fragmentos de información que se ha “archivado” en algún lugar. Sin mencionar toda la información valiosa que no se sabe que existe. La búsqueda es tediosa. Así, en un mundo donde la información es cada vez más abundante, la desinformación es actualmente un patrón común.

Alternativamente el almacenamiento electrónico y la gestión documental transforma información valiosa en algo infinitamente más útil, haciéndola más accesible. Los documentos de todo tipo, sin importar su origen, se almacenan conjuntamente en un repositorio central de documentos, una base de datos documental. Este proceso es automatizado en cada estación de trabajo gracias a una tecnología inteligente. Toda esta información se reúne y ordene lógicamente, poniéndola al alcance de todos. Con unos pocos clics del ratón, la información está disponible en cualquier momento y desde cualquier lugar. La gestión documental es una tecnología madura y probada que ha permitido a miles de empresas incrementar sus beneficios y reducir sus costos. Es hora de que también la Corte Departamental Electoral de Tarija se beneficie de una moderna gestión documental, por lo que el presente trabajo busca realizar el Diseño de un Sistema de Gestión documental para su posterior implementación y puesta en marcha, con los beneficios que ello conlleva.

I. El Proyecto

I.1. Presentación del Proyecto

I.1.1. Título

Implementación de un Sistema Informático Gestión Documental para la Corte Departamental Electoral de Tarija.

I.1.2. Área del Proyecto

Gestión Documental.

I.1.3. Responsable del Proyecto

Carrera de Ingeniería Informática – Taller III.

I.1.4. Entidades asociadas

Las entidades que se ven involucradas en el presente proyecto son:

- Corte Departamental Electoral Tarija.
- UAJMS (Universidad Autónoma Juan Misael Saracho).

I.1.5. Compromiso del director del Proyecto

<p>Yo, Adrian Mansilla Salas Director del grupo acepto las bases y condiciones del concurso, asimismo asumo la responsabilidad de cumplir los compromisos de ejecución del proyecto “Implementación de un Sistema de Gestión Documental para la Corte Departamental Electoral de Tarija” en caso de aprobarse.</p>	<p>_____ Univ. Adrián Mansilla Salas</p>
<p>Nombre del Director</p>	<p>Firma del Director</p>

Tabla 1: Compromiso del Director del Proyecto

I.1.6. Grupo responsable del Proyecto

El grupo responsable del proyecto está conformado por:

- Mansilla Salas Adrian.
- Bejarano Mauricio Rodrigo.

I.1.7. Talleres asociados

Taller III – INF-501

I.1.8. Duración (meses)

La duración para la ejecución del proyecto es de nueve meses.

I.1.9. Director responsable del Proyecto

NOMBRE	TALLER / GRUPO	TALLERES PROGRAMADOS
Mansilla Salas Adrian	Taller III/Grupo 2	Taller III
E MAIL	TELEFONO	
mansilladrian@gmail.com	Tel:46632540 Cel:70217486	

Tabla 2: Director responsable del proyecto

I.1.10. Director alterno del Proyecto

NOMBRE	TALLER/GRUPO	TALLERES PROGRAMADOS
Bejarano Mauricio Rodrigo	Taller III/Grupo 2	Taller III
E MAIL	TELEFONO	
mauriciorodrigo007@gmail.com	71894997	

Tabla 3: Director alterno del proyecto

I.1.11. Resumen del Proyecto

En las organizaciones institucionales, usualmente generan sus propios documentos, almacenándolos en carpetas, en discos duros de PC o en algún lugar del universo del e-mail. Actualmente, la información de esos documentos es extremadamente valiosa es la base para el trabajo cotidiano, algo esencial para el éxito de cualquier organización. Es incomprensible que se siga perdiendo tiempo buscando fragmentos de información que se ha “archivado” en algún lugar. Sin mencionar toda la información valiosa que no se sabe que existe. La búsqueda es tediosa. Así, en un mundo donde la información es cada vez más abundante, la desinformación es actualmente un patrón común.

Alternativamente el almacenamiento electrónico y la gestión documental transforma información valiosa en algo infinitamente más útil, haciéndola más accesible. Los documentos de todo tipo, sin importar su origen, se almacenan conjuntamente en un repositorio central de documentos, una base de datos documental. Este proceso es automatizado en cada estación de trabajo gracias a una tecnología inteligente. Toda esta información se reúne y ordene lógicamente, poniéndola al alcance de todos. Con unos pocos clics del ratón, la información está disponible en cualquier momento y desde cualquier lugar. La gestión documental es una tecnología madura y probada que ha permitido a miles de empresas incrementar sus beneficios y reducir sus costos. Es hora de que también la Corte Departamental Electoral de Tarija se beneficie de una moderna gestión documental, por lo que el presente trabajo busca realizar el Diseño de un Sistema de Gestión documental para su posterior implementación y puesta en marcha, con los beneficios que ello conlleva.

I.1.12. Síntesis vinculación entre objetivos, macroactividades y recursos

OBJETIVOS	MACROACTIVIDADES	RECURSOS
La Gestión Documental en la Corte Departamental Electoral de Tarija mejora.	Desarrollo y Programación del Sistema de Gestión Documental SigD.	11120 Bs.
Establecer los elementos centrales necesarios para el Diseño e Implementación de un Sistema de Gestión Documental para la Corte Departamental Electoral de Tarija.	Implementación y Capacitación del Sistema de Gestión Documental SigD.	<ul style="list-style-type: none">• Capacitación del SigD: 630 Bs.• Implementación del SigD: 200 Bs.

Tabla 4: Síntesis vinculación entre objetivos, macroactividades y recursos

I.2. Plan estratégico de la unidad beneficiaria

I.2.1. Misión

Administrar el registro civil, los procesos electorales y la toma de decisiones colectivas e impulsar una cultura democrática contribuyendo a la consolidación y profundización de la democracia boliviana.

I.2.2. Conclusiones del análisis FODA (de los factores externos e internos).

<p>Fortalezas</p> <ul style="list-style-type: none"> • Control de las oficialías de registro civil. • Número total de votantes para procesos electorales. • Organización y ejecución de la elección participativa, libre y transparente de los representantes departamentales, municipales y constituyentes. • Autoridad administrativa y disciplinaria sobre las autoridades electorales y personal de su jurisdicción. 	<p>Debilidades</p> <ul style="list-style-type: none"> • Incumplimiento del calendario electoral, oficial de la C.N.E. • Incremento en la incidencia de partidas anuladas.
<p>Oportunidades</p> <ul style="list-style-type: none"> • Interés que se ha despertado en la población por los procesos electorales. • Interés de la población por formar parte en las decisiones de estado. 	<p>Amenazas</p> <ul style="list-style-type: none"> • Rechazo de la asignación de jurados electorales. • Alto porcentaje de denuncias por delitos electorales. • Acusaciones contra jueces y notarios electorales. • Elevado número de depurados en los procesos electorales.

Tabla 5: Conclusiones del análisis FODA (de los factores internos y externos)

I.2.3. Objetivos de la organización (beneficiaria)

- Convertir el Organismo Electoral en un poder del estado que garantice el ejercicio pleno de los derechos a la identidad y la participación política de todas las personas.
- Administración del Registro Civil.
- Organización y ejecución de la elección participativa, libre y transparente de los representantes nacionales, departamentales, municipales y constituyentes, así como de la votación en las consultas populares.
- Promover programas de educación cívica y ciudadana. Esta tarea constituye una responsabilidad medular no sólo para la participación ciudadana en los procesos electorales, sino para la consolidación misma de la democracia y sus instituciones.

I.2.4. Estrategias y planes de acción

La recolección de información para la puesta en marcha del Sistema de Gestión Documental será realizada a través de observación a los directamente involucrados en las diferentes direcciones de la C.D.E. (Corte Departamental Electoral) de Tarija.

Se realizara sondeos y/o preguntas a los directores y a sus respectivas secretarias, que son los sujetos que más documentación producen en este tipo de organización.

La metodología a utilizarse para la investigación supone las siguientes etapas.

1.- Identificación de necesidades.

La etapa de identificación de las necesidades de información: Es una etapa que permite identificar las necesidades de información en base a demandas planteadas, esta identificación debe cubrir esencialmente a

los usuarios de la información, para este caso en particular, solamente se trato de usuarios internos(los funcionarios en general).

Para este objetivo se elaboro un cuestionario para los usuarios internos de la Corte Departamental Electoral cuyos resultados están reflejados en las conclusiones expuestas a continuación:

Conclusiones:

- El personal de la Corte Departamental Electoral no conoce totalmente los tipos de documentos que se producen en cada dirección o área.
- Las áreas o direcciones que elaboran más documentos son las de administración, presidencia e informática.
- En la Corte Departamental Electoral no existe una estructura que administre los roles en las fases de creación, revisión, aprobación y publicación de los documentos, por lo tanto se genera una desorganización en el control y clasificación de los archivos.
- Por esta falta de control en el archivo de los documentos se pierde tiempo en su búsqueda e incluso el extravío de los mismos o se realizan copias innecesarias, que en algunos casos son confundidas con los originales.

Recomendaciones:

- Se recomienda a las Direcciones o Áreas de la Corte Departamental Electoral realizar una única estructura organizativa digital en la red informática institucional, que administre la generación y control de los archivos en todas sus fases (creación,

revisión, aprobación y publicación) para facilitar al personal de la institución el uso, archivo y clasificación de los documentos.

2.- Sistema Informático de Gestión Documental implantado.

2.1 Análisis del sistema

2.2 Diseño del sistema

2.3 Programación del sistema

2.4 Implantación del sistema

3.- Capacitación al personal de la Corte Departamental Electoral de Tarija.

3.1 Elaboración de manuales de usuario.

3.2 Selección de hora y lugar para la capacitación

3.3 Selección del material didáctico.

3.4 Desarrollo del curso de capacitación a los usuarios del sistema.

I.2.5. Plan de desarrollo del personal

Las personas que son beneficiadas con la ejecución del proyecto son:

- El personal de la dirección de Informática de la Corte Departamental Electoral de Tarija.

I.2.6. Plan de asistencia

El personal que apoyó en la realización del proyecto fueron:

Docentes guías:

1. Lic. Efraín Torrejón T.
2. Ing. Silvana Paz R.

I.3. Vinculación del plan estratégico y del Proyecto

I.3.1. Coherencia del Proyecto con el contexto

ATRIBUCIONES DE LAS CORTES DEPARTAMENTALES ELECTORALES

ARTÍCULO 35°.- (Código Electoral)

Son atribuciones de las Cortes Departamentales Electorales:

a) Cumplir y hacer cumplir el presente Código y las resoluciones y reglamentos de la Corte Nacional Electoral.

b) Dirigir y administrar el Registro Civil y el Padrón Electoral en su jurisdicción en el marco de las directivas de la Corte Nacional Electoral.

c) Designar a los jueces y notarios e inspectores electorales, removerlos por faltas o delitos electorales que hubieran cometido en el ejercicio de sus funciones.

d) Efectuar, en sesión pública, el sorteo para la designación de jurados electorales según calendario oficial de la Corte Nacional Electoral.

e) Publicar en periódicos departamentales, carteles u otros medios de comunicación, la ubicación de las mesas de sufragio con especificación de recinto, asiento y circunscripción uninominal a la que pertenece, la cantidad de inscritos en cada mesa y en cada

circunscripción uninominal, así como el total de los inscritos en todo el Departamento.

f) Efectuar, en sesión pública, el cómputo departamental de las elecciones para Presidente, Vicepresidente, Senadores, Diputados, Alcaldes, Concejales Municipales y Agentes Cantonales y elevarlo a la Corte Nacional Electoral.

g) Conocer los recursos de apelación interpuestos ante el Jurado Electoral sobre los actos de escrutinio y cómputo realizados en la mesa de sufragio, no pudiendo actuar de oficio en la materia. Resolver, por mayoría absoluta de sus miembros y en grado de apelación, las causas de nulidad de mesas.

h) Dirimir las competencias que se suscitaren entre los organismos electorales de su jurisdicción.

i) Denunciar y promover las causas de responsabilidad ante la autoridad competente, por delitos electorales que cometieran en el ejercicio de sus funciones, los Prefectos, Jueces Electorales, Subprefectos, Alcaldes y Concejales Municipales, Jueces, Fiscales, Autoridades Militares y Policiales.

j) Conocer las denuncias efectuadas por los Jueces Electorales contra las autoridades señaladas en el artículo anterior y remitirlas a la autoridad competente.

k) Conocer y resolver los recursos de nulidad contra las resoluciones de los jueces electorales, sobre admisión o exclusión de partidas de inscripción en el registro cívico o en el Padrón Electoral.

l) Conocer y resolver las denuncias, tachas, excusas y recusaciones contra los jueces y notarios electorales.

ll) Conocer las denuncias e irregularidades cometidas en el proceso electoral e iniciar las acciones legales que correspondan.

m) Conocer, procesar y sancionar a todos los ciudadanos que incurrieran en faltas electorales.

n) Requerir el concurso de las instituciones y/o funcionarios del Poder Ejecutivo y Judicial, encomendándoles comisiones en el servicio electoral; y requerir de la fuerza pública para hacer cumplir sus resoluciones.

ñ) Programar, organizar, dirigir, coordinar, ejecutar, controlar y evaluar las actividades técnicas y administrativas del proceso electoral en su jurisdicción, en el marco de las directivas de la Corte Nacional Electoral. Asimismo, administrar el diez por ciento (10%) de papeleta única de sufragio, con notificación a los delegados de los partidos políticos que intervinieran en la elección.

o) Designar, promover y destituir al personal administrativo de la Corte Departamental Electoral. Asignar responsabilidades y evaluar el desempeño de sus funciones.

p) Formular proyectos y consultas ante la Corte Nacional Electoral para la mejor atención del servicio electoral.

q) Comunicar a la Corte Nacional Electoral el resultado de las elecciones de su jurisdicción en el plazo máximo de diez días.

r) Elevar anualmente informe de sus labores a la Corte Nacional Electoral.

s) Conservar adecuadamente la documentación relativa a la inscripción de ciudadanos e informar permanentemente a la Corte Nacional Electoral respecto a altas, bajas y cambios de domicilio de los ciudadanos. Conservar la correspondencia local de partidos y toda aquella que se procese en su jurisdicción.

t) Ejercer autoridad administrativa y disciplinaria sobre las autoridades electorales y personal de su jurisdicción.

u) Proponer a la Corte Nacional Electoral su presupuesto de egresos, administrar los recursos que se le asignen y presentar sus estados financieros anuales en los plazos establecidos por la Corte Nacional Electoral.

v) Inspeccionar periódicamente las oficinas de su jurisdicción, atender sus consultas, reclamaciones y necesidades.

w) Registrar las credenciales de los delegados de los partidos políticos o alianzas acreditados ante la Corte.

x) Extender credenciales a los Alcaldes elegidos directamente, Concejales Municipales y Agentes Cantonales. (Texto modificado por Ley N°. 2282 de 4 de diciembre de 2001).

y) Autorizar la inscripción y voto en centros de internación y penitenciarías, de acuerdo al reglamento.

z) Promover programas de educación cívica y ciudadana.

aa) Inscribir a los candidatos a Alcaldes, Concejales Municipales y Agentes Cantonales presentados por partidos políticos o alianzas y publicar las listas. (Texto incorporado por Ley N°. 2282 de 4 de diciembre de 2001).

bb) Inhabilitar, a denuncia de parte, a los candidatos a Alcaldes, Concejales Municipales y Agentes Cantonales que tengan auto de procesamiento o pliego de cargo ejecutoriados (texto incorporado por Ley N°. 2282 de 4 de diciembre de 2001).

Analizando cada uno de los incisos se puede apreciar que la información y su respectiva documentación son prioritarias para el cumplimiento de las atribuciones de la C.D.E. Los errores en el manejo de esta información podrían ser gravitantes en la ejecución de todas las actividades.

Entonces para poder coadyuvar la labor que realiza la Corte Departamental Electoral, el proyecto propone la implementación de un Sistema Informático de Gestión Documental, ofreciendo así a todo el Personal y a la Institución de la C.D.E. un mejoramiento en la calidad de servicio de tratamiento documental.

I.3.2. Vinculaciones de los problemas que reconoce la entidad y los problemas que busca resolver el Proyecto

PROBLEMAS DEFINIDOS COMO PRIORITARIOS	PROBLEMAS QUE ABORDA EL PROYECTO
Gestión Documental Ineficiente.	Gestión Documental Ineficiente.
Baja Calidad del Servicio de tratamiento documental.	Baja Calidad del Servicio de tratamiento documental.
Pérdida de Documentos.	Pérdida de Documentos.
Inexistencia de Seguridad en el Acceso y Protección de la Información.	Inexistencia de Seguridad en el Acceso y Protección de la Información.
Problemas de Localización y Búsqueda de Documentos.	Problemas de Localización y Búsqueda de Documentos.
Desconocimiento de la ubicación de la Documentación que se Genera.	Desconocimiento de la ubicación Documentación que se Genera

Ocupación de grandes espacios en almacenamiento de documentación.	Ocupación de grandes espacios en almacenamiento de documentación.
Elevado Número de Copias Innecesarias.	Elevado Número de Copias Innecesarias.
Confusión entre Documentos originales y Copias.	Confusión entre Documentos originales y Copias.

Tabla 6: Vinculaciones de los problemas que reconoce la entidad y los problemas que busca resolver el proyecto

I.4. El Proyecto

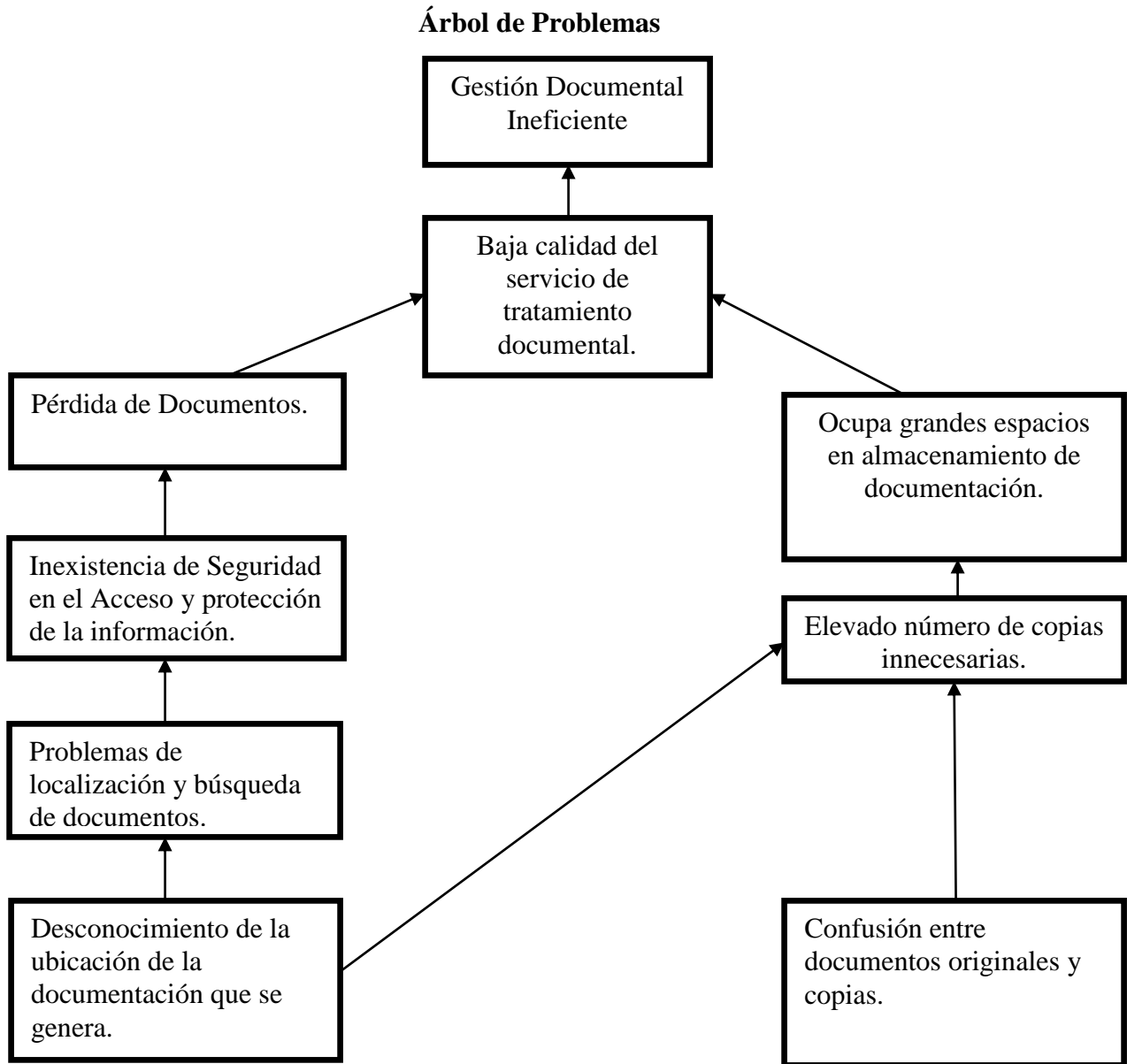


Figura 1: Árbol de Problemas

Árbol de Objetivos

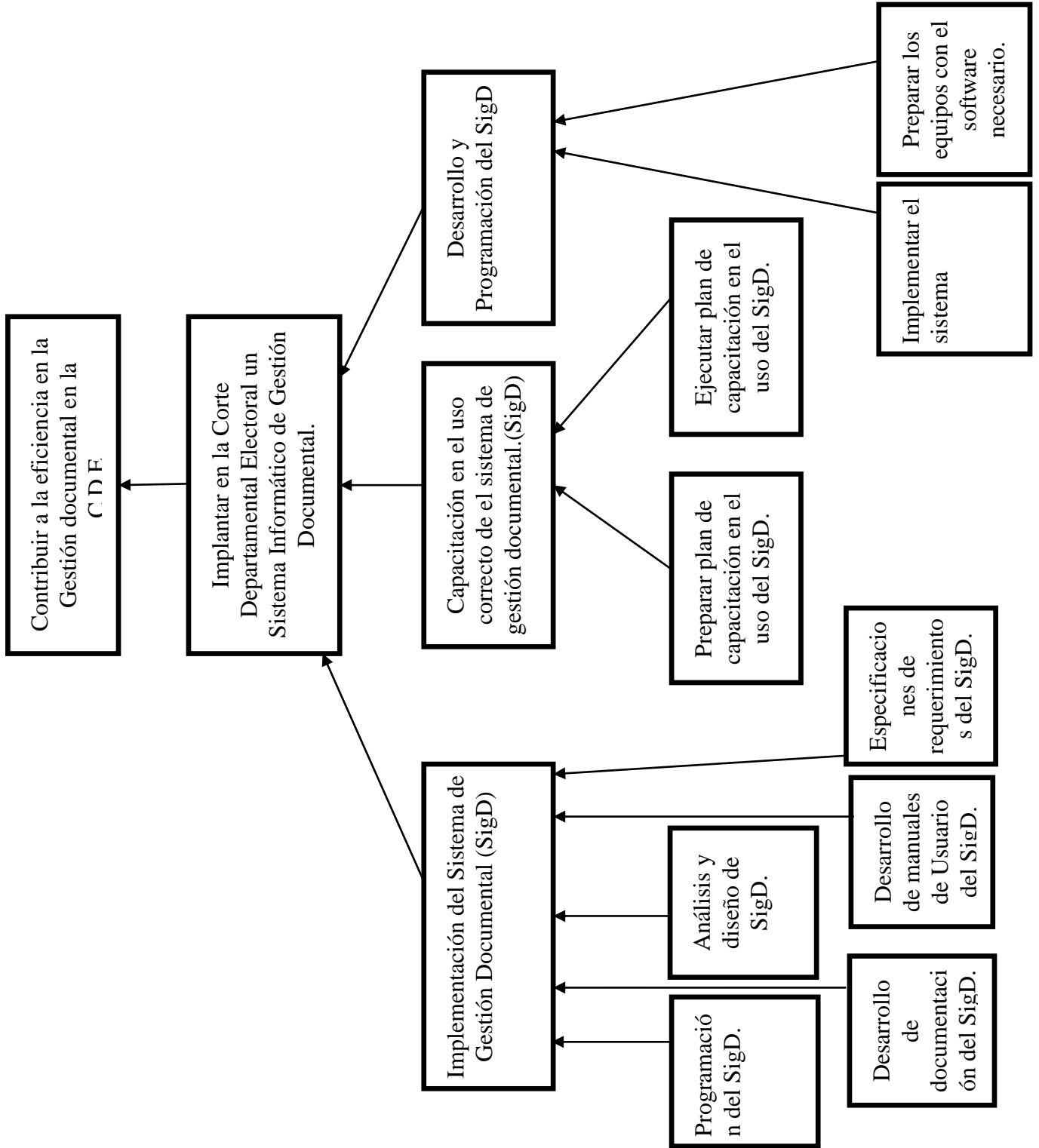


Figura 2: Árbol de Objetivos

I.4.1. Objetivo general, específicos e indicadores de resultados

I.4.1.1. Objetivo general

Implantar en la Corte Departamental Electoral un Sistema Informático de Gestión Documental para contribuir a una eficiente gestión documental y mejorar la calidad en el servicio de tratamiento documental.

I.4.1.2. Objetivos específicos

Los objetivos específicos del proyecto son:

- Desarrollo y Programación del Sistema Informático de Gestión Documental. (SigD)

- Capacitación en el uso correcto del Sistema Informático de Gestión Documental.(SigD)

- Implementación del Sistema Informático de Gestión Documental (SigD)

I.4.1.3. Vinculaciones de objetivos específicos e indicadores de resultados

OBJETIVOS ESPECIFICOS	INDICADORES DE RESULTADOS
1. Desarrollo y Programación del Sistema Informático de Gestión Documental. (SigD)	<ul style="list-style-type: none"> • Desarrollo del Sistema de Gestión Documental SigD en un plazo de 9 meses a partir del 26 de marzo de 2009.
2. Capacitación en el uso correcto del Sistema Informático de Gestión Documental. (SigD).	<ul style="list-style-type: none"> • Al finalizar el proyecto el personal de la dirección de informática de la C.D.E. estará capacitado en el uso del SigD en un período comprendido entre el 2 de diciembre hasta el 15 de diciembre.
3. Implementación del Sistema Informático de Gestión Documental (SigD).	<ul style="list-style-type: none"> • Un sistema (SigD) implementado en la red de la Corte Departamental Electoral

Tabla 7 : Objetivos específicos e indicadores de resultados

I.4.2. Equipo del Proyecto, actividades y recursos

I.4.2.1. Equipo del Proyecto

Director del Proyecto: Adrián Mansilla Salas

Codirector del Proyecto: Mauricio Rodrigo Bejarano

I.4.2.2. Componentes y actividades

- Desarrollo y Programación del Sistema de Gestión Documental SigD:
 - Especificaciones de requerimientos del SigD mediante cuestionarios aplicados al personal.
 - Análisis y diseño del SigD a través de UML.
 - Programación del sistema SigD.
 - Desarrollo del manual de instalación del SigD.
 - Desarrollo de manuales de Usuario del SigD.
- Capacitación del SigD.
 - Planificación plan de capacitación en el uso del SigD.
 - Ejecutar plan de capacitación en el uso del SigD.
- Implementación del SigD
 - Preparar los equipos con el software necesario.
 - Implementar el SigD.
 - Pruebas del sistema durante dos días a dos funcionarios con la previa capacitación al personal.

I.5. Planificación y diagramas de Gantt

Id	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos	Costo fijo
1	SISTEMA SIG-D	193 días	jue 26/03/09	lun 21/12/09		Proyecto	\$b 11.950,00
2	Perfil SIG-D	38 días	jue 26/03/09	lun 18/05/09			\$b 0,00
3	Presentacion perfil	18 días	jue 26/03/09	lun 20/04/09			\$b 0,00
4	Correcciones perfil	20 días	mar 21/04/09	lun 18/05/09	3		\$b 0,00
5	Desarrollo y programacion del SIG-D	136 días	mar 26/05/09	mar 01/12/09		Desarrollo y programacion del SIG-D	\$b 11.120,00
6	Especificacion de requerimientos del sistema	7 sem.	mar 26/05/09	lun 13/07/09	4	Ingeniero Software	\$b 450,00
7	Analisis y diseño del sistema	7 sem.	mar 14/07/09	lun 31/08/09	6	Analista de Sistemas	\$b 5.180,00
8	Programación del sistema	8 sem.	lun 07/09/09	vie 30/10/09	7	Programador	\$b 2.640,00
9	Pruebas del sistema	11 días	lun 02/11/09	lun 16/11/09	8	Ingeniero de Software	\$b 450,00
10	Desarrollo de manuales de usuario	11 días	mar 20/10/09	mar 03/11/09		Analista de Sistemas	\$b 1.200,00
11	Desarrollo de Documentacion del sistema	71 días	mar 25/08/09	mar 01/12/09		Analista de Sistemas	\$b 1.200,00
12	Capacitacion del SIG-D	10 días	mié 02/12/09	mar 15/12/09		Capacitacion para el uso del SIG-D	\$b 630,00
13	Preparacion de capacitacion en uso del SIG-D	1 sem	mié 02/12/09	mar 08/12/09		Preparacion Capacitacion SIG-D	\$b 365,00
14	Ejecucion de capacitacion en uso del SIG-D	1 sem	mié 09/12/09	mar 15/12/09	13	Ejecucion Capacitacion SIG-D	\$b 265,00
15	Implementacion del Sistema de Gestion Documental SIG-D	4 días	mié 16/12/09	lun 21/12/09		Implementacion del Sistema de Gestion Documental SIG-D	\$b 200,00
16	Preparar los equipos con el software necesario	2 días	mié 16/12/09	jue 17/12/09		Preparar software	\$b 100,00
17	Implementar el SIG-D	2 días	vie 18/12/09	lun 21/12/09	16	Implementar el SIG-D	\$b 100,00

Tabla 8: Planificación y presupuesto

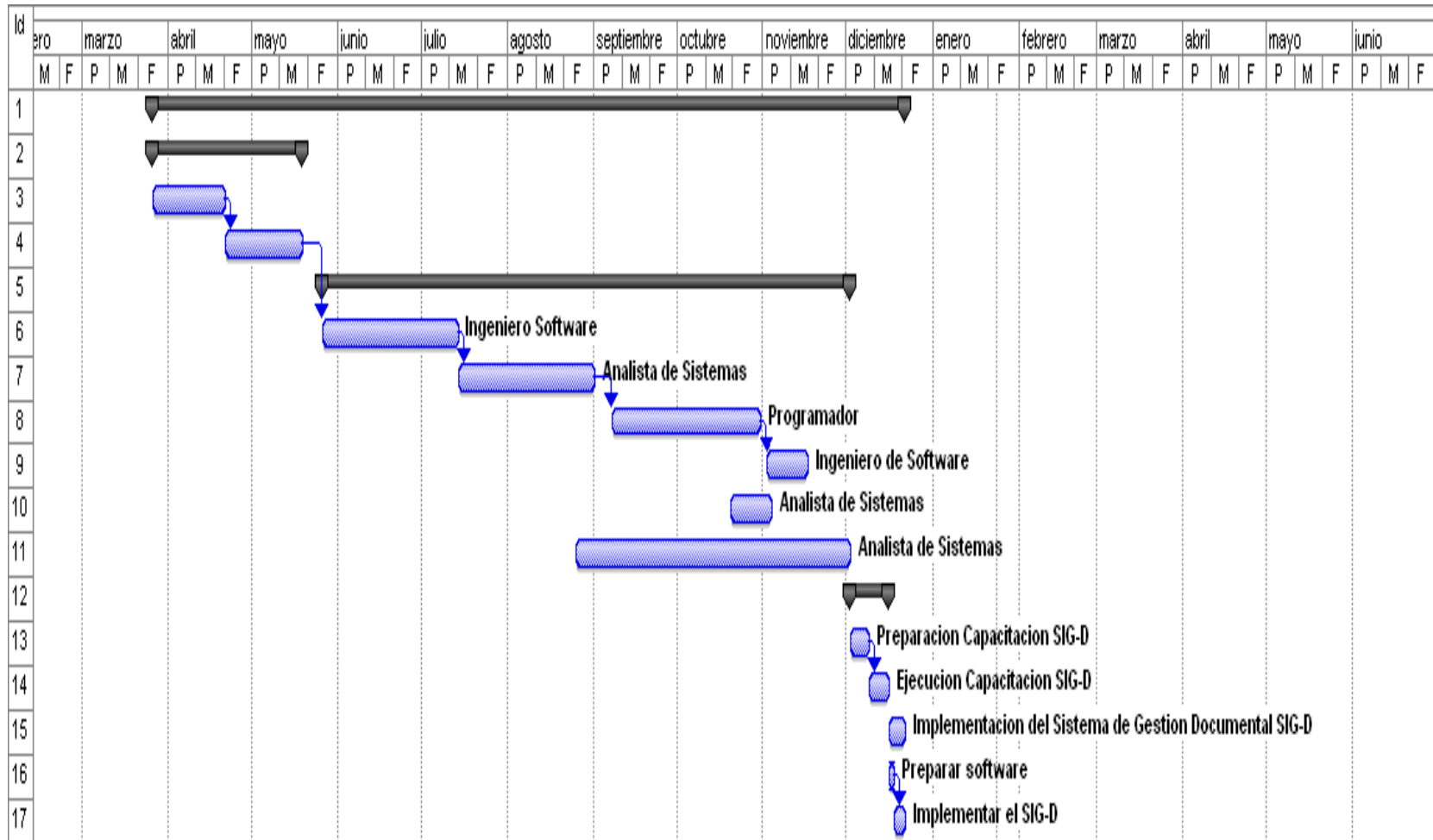


Figura 3: Diagrama de Gantt

I.6. Matriz de marco lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
<p>Fin:</p> <p>1.- Contribuir a la eficiencia en la Gestión documental en la C.D.E.</p>	<ul style="list-style-type: none"> • La incorporación del sistema de gestión documental a partir de diciembre del 2009 ahorrara en cada empleado el tiempo de búsqueda de documentación que será entre un 10% y 20% menos que en la actualidad. 	<ul style="list-style-type: none"> • Prueba de Medio Término a la Implementación del SigD. • Informe anual del desempeño del Departamento de Informática y Sistemas de la C.D.E. 	<ul style="list-style-type: none"> • Apoyo de las autoridades de la C.D.E. • Estabilidad Política.
<p>Propósito:</p> <p>2.- Implantar en la Corte Departamental Electoral un Sistema Informático de</p>	<ul style="list-style-type: none"> • El espacio de archivo físico puede reducirse hasta un 35%. 	<ul style="list-style-type: none"> • Prueba de Medio Término a la Implementación del SigD. 	<ul style="list-style-type: none"> • Apoyo de las autoridades de la C.D.E.

Gestión Documental.			
<p>Componentes:</p> <p>1. Desarrollo y Programación del SigD (26/03/09 al 21/12/09).</p> <p>2.- Capacitación en el uso correcto de el sistema de gestión documental.(SigD).</p>	<ul style="list-style-type: none"> • Desarrollo del Sistema de Gestión Documental SigD en un plazo de 9 meses a partir del 26 de marzo de 2009. • Al finalizar el proyecto el personal de la C.D.E. estará capacitado en el uso del SigD en un período comprendido entre el 2 de diciembre hasta el 15 de diciembre. • La capacitación se realizará los días 	<ul style="list-style-type: none"> • Manual de usuario del SigD. • Lista de personal Capacitado en el uso del SigD. • Certificado de la C.D.E. • Sistema digital en la red de la Corte 	<ul style="list-style-type: none"> • Los plazos se cumplen según lo programado. • Desarrolladores cuentan con las herramientas necesarias. • El personal de la C.D.E. brinda el apoyo necesario para el desarrollo del sistema.

<p>3.-Implementación del Sistema de Gestión Documental (SigD)</p>	<p>viernes y sábado de horas 07:00 p.m. a 08:00 p.m.</p> <ul style="list-style-type: none"> • Un sistema (SigD) implementado en la red de la Corte Departamental Electoral 	<p>Departamental Electoral</p> <ul style="list-style-type: none"> • Certificado de implementación del sistema. 	
<p>Actividades:</p> <p>1.1.- Especificaciones de requerimientos del SigD.</p> <p>1.2.- Análisis y diseño de SigD.</p> <p>1.3.- Programación del SigD.</p>	<p>- Especificación de los requerimientos del sistema: 450 Bs. Se realizó una encuesta para determinar los requerimientos del SigD.</p> <p>- Análisis y Diseño del sistema: 5180 Bs. Se elaboraron 43 diagramas UML para el análisis y diseño del SigD.</p> <p>- Programación del sistema:</p>		

1.4.- Desarrollo de documentación del SigD.	2640 Bs. Se programo 1 sistema para la instalación en la red de la Corte Electoral Departamental.		- Disponibilidad del personal de la C.D.E. para ser capacitados en el uso del SigD.
1.5.- Desarrollo de manuales de Usuario del SigD.	- Pruebas del sistema: 450 Bs. - Se desarrollo 2 manuales de usuario: 1200 Bs.		- Disponibilidad del uso de las instalaciones de la C.D.E. para la capacitación.
2.1- Preparar plan de capacitación en el uso del SigD.	- Desarrollo de documentación del sistema: 1200 Bs.		
2.2.- Ejecutar plan de capacitación en el uso del SigD.	- Preparación de la capacitación del uso del SigD: 365 Bs.	• Certificados de capacitación	
3.1 Preparar los equipos con	- Ejecución de la capacitación para el uso del SigD: 265 Bs.		

<p>el software necesario.</p> <p>3.2 Implementar el sistema.</p>	<p>Total: 630 Bs.</p> <ul style="list-style-type: none"> - Se capacitaron 5 miembros del personal de la Corte Electoral Departamental - Preparar los equipos con el software necesario: 100 Bs. - Implementar del sistema SigD: 100 Bs. <p>Total: 200 Bs.</p> <p>Costo Total: 11950 Bs.</p>		
--	---	--	--

Tabla 9: Matriz de Marco Lógico

I. Componentes del Proyecto

I.1. Componente 1: Desarrollo y Programación del SigD

I.1.1. Plan de desarrollo de software

I.1.1.1. Introducción

Este Plan de Desarrollo del Software es la versión final para ser incluida en la propuesta elaborada como respuesta al proyecto final de la asignatura de Taller III del Programa de Informática de la Universidad Autónoma “Juan Misael Saracho”. Este documento provee una visión global del enfoque de desarrollo propuesto.

El proyecto ha sido ofertado por Adrian Mansilla Salas y Mauricio Rodrigo Bejarano, basado en una metodología de Rational Unified Process en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología, constando únicamente en la tercera fase de dos iteraciones. Es importante destacar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de Inicio y Elaboración y adicionalmente se esbozarán las fases posteriores de Construcción y Transición para dar una visión global de todo proceso.

El enfoque de desarrollo propuesto constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados. Este documento es a su vez uno de los artefactos de RUP.

I.1.1.2. Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para controlar el proyecto. En él se describe el enfoque de desarrollo del software.

Los usuarios del Plan de Desarrollo del Software son:

- El jefe del proyecto lo utiliza para organizar la agenda y necesidades de recursos, y para realizar su seguimiento.
- **Los miembros del equipo de desarrollo lo usan para entender lo qué deben hacer, cuándo deben hacerlo y qué otras actividades dependen de ello.**

I.1.1.3. Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del “Sistema Informático Gestión Documental”. El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada. Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base para la planificación de las iteraciones. Para la versión 1.0 del Plan de Desarrollo del Software, nos hemos basado en la captura de requisitos por medio del Stakeholder representante de la empresa para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el

seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

I.1.1.4. Resumen

Después de esta introducción, el resto del documento está organizado en las siguientes secciones:

Vista General del Proyecto — proporciona una descripción del propósito, alcance y objetivos del proyecto, estableciendo los artefactos que serán producidos y utilizados durante el proyecto.

Organización del Proyecto — describe la estructura organizacional del equipo de desarrollo.

Gestión del Proceso — explica los costos y planificación estimada, define las fases e hitos del proyecto y describe cómo se realizará su seguimiento.

Planes y Guías de aplicación — proporciona una vista global del proceso de desarrollo de software, incluyendo métodos, herramientas y técnicas que serán utilizadas.

I.1.1.5. Vista general del Proyecto

I.1.1.5.1. Marco teórico

- **Rational Unified Process (RUP)**

Proceso Unificado de Rational

El **Proceso Unificado de Rational** (*Rational Unified Process* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la

metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el **Rational Method Composer (RMC)**, que permite la personalización de acuerdo a necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el ***Rational Unified Process***, que se vendiera como producto independiente.

Principios de desarrollo

El RUP está basado en 5 principios clave que son:

1.1.1 Adaptar el proceso

El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto.

1.1.2 Balancear prioridades

Los requerimientos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. *Debe encontrarse un balance que satisfaga los deseos de todos.* Debido a este balanceo se podrán corregir desacuerdos que surjan en el futuro.

1.1.3 Demostrar valor interactivamente

Los proyectos se entregan, aunque sea de un modo interno, en **etapas iteradas**. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados

1.1.4 Elevar el nivel de abstracción

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia (frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la mejor manera los requerimientos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

1.1.5 Enfocarse en la calidad

El control de calidad no debe realizarse al final de cada iteración, sino en **todos** los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Ciclo de vida

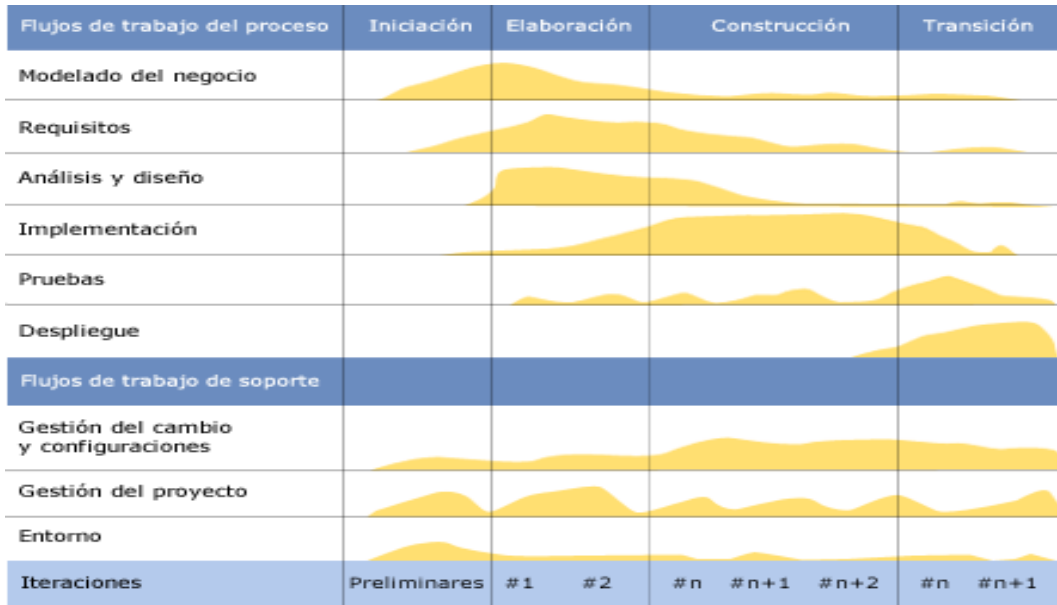


Tabla 1: Ciclo de vida RUP

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las

disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una baseline (Línea Base) de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requerimientos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Fases

- Establece oportunidad y alcance
- Identifica las entidades externas o actores con las que se trata
- Identifica los casos de uso.Rup.

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

Proceso: Las etapas de esta sección son:

- Modelado de negocio
- Requisitos
- Análisis y Diseño
- Implementación
- Pruebas
- Despliegue

Soporte: En esta parte nos conseguimos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP es la que permite que este sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las 4 fases descritas anteriormente:

- Inicio(También llamado Incepción)
- Elaboración

- Desarrollo(También llamado Implementación, Construcción)
- Cierre (También llamado Transición)

Artefactos

RUP en cada una de sus fases (pertenecientes a la estructura estática) realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema (entre otros). Estos artefactos (entre otros) son los siguientes:

Inicio:

- Documento Visión
- Especificación de Requerimientos

Elaboración:

- Diagramas de caso de uso

Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

Vista Lógica:

- Diagrama de clases
- Modelo E-R (Si el sistema así lo requiere)

Vista de Implementación:

- Diagrama de Secuencia
- Diagrama de estados
- Diagrama de Colaboración

Vista Conceptual:

- Modelo de dominio

Vista física:

- Mapa de comportamiento a nivel de hardware.

- **UML**

Lenguaje Unificado de Modelado

Lenguaje Unificado de Modelado (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 de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, 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 puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa (Lengua de Modelación Unificada), no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la orientación a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

En UML 2.0 hay 13 tipos diferentes de diagramas. Para comprenderlos de manera concreta, a veces es útil categorizarlos jerárquicamente, como se muestra en la figura de la derecha.

Los *Diagramas de Estructura* enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases
- Diagrama de componentes
- Diagrama de objetos

- Diagrama de estructura compuesta (UML 2.0)
- Diagrama de despliegue
- Diagrama de paquetes

Los *Diagramas de Comportamiento* enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades
- Diagrama de casos de uso
- Diagrama de estados

Los *Diagramas de Interacción* son un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia
- Diagrama de comunicación, que es una versión simplificada del Diagrama de colaboración (UML 1.x)
- Diagrama de tiempos (UML 2.0)
- Diagrama de vista de interacción (UML 2.0)

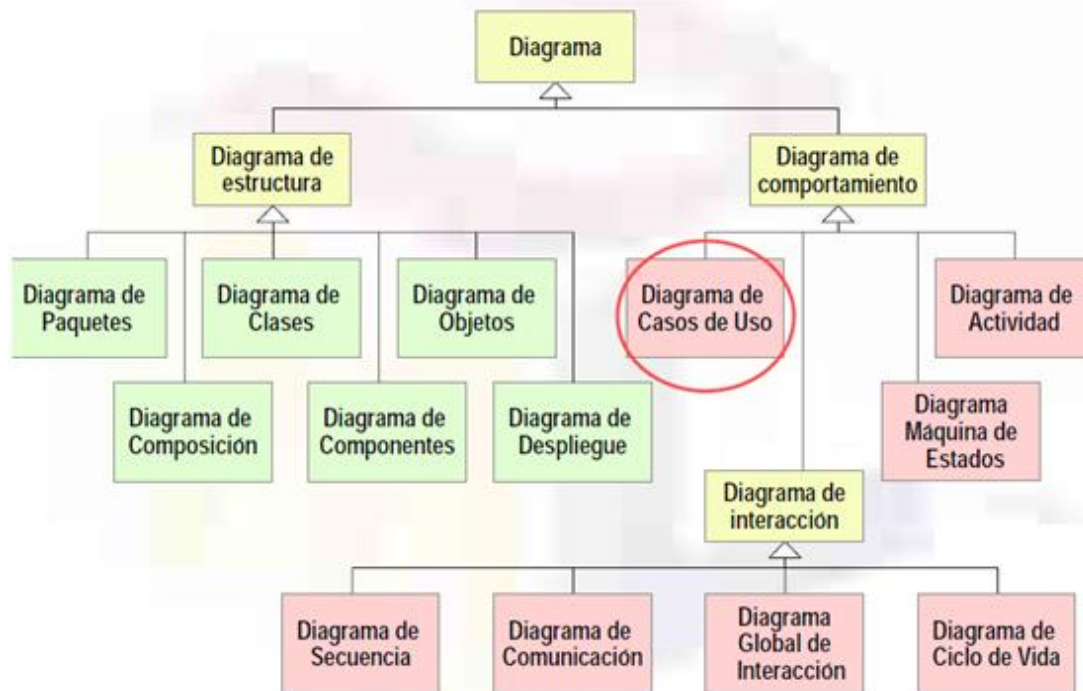


Figura 1: Estructura diagramas UML

Software para modelado en UML

A continuación, se listan algunos de los programas más populares para el modelado en **UML**

Software Libre

Estos programas están bajo licencias libres, siendo posible su libre uso, estudio y modificación.

- ArgoUML, Herramienta de modelado UML escrito en Java.
- BOUML, Ligera herramienta de modelado UML y generación de código C++, Java e IDL. Disponible para Windows, Unix/Linux y Mac OS X.

- Fujaba, No solo sirve para modelar sino que puede generar código Java automáticamente. También es capaz de hacer ingeniería inversa y crear los diagramas a partir del código Java.
- Dia Puede ser usado para modelar varios tipos de diagramas UML.
- gModeler Herramienta para modelado de UML basada en Flash (utilizable desde el navegador), que permite generar código Action Script 2.0 Compatible.
- MonoUML Herramienta CASE para la plataforma mono.
- Papyrus, Herramienta gráfica basada en Eclipse para el modelado con UML2, es de código abierto y se ofrece bajo licencia EPL.
- StarUML Herramienta de modelado para Windows desarrollada en Delphi. Bastante estable y utilizable.
- TCM, Toolkit for Conceptual Modeling, herramienta para crear diversos tipos de diagramas incluidos UML.
- Umbrello Herramienta para modelado UML para el entorno KDE.
- UMLet Herramienta para modelado rápido de UML también escrita en Java.
- Netbeans módulo UML.

- Open ModelSphere Herramienta de Modelado gratuita, para modelado de datos, procesos y UML. Disponible Como Open Source Software, Released Under GPL (GNU Public License).

Freeware para modelado en UML

Aunque gratuitos, estos programas se encuentran bajo licencias que no permiten el estudio y modificación de los mismos.

- JUDE Community Herramienta de modelado UML.
- Omondo plugin para Eclipse. Herramienta de modelado UML para Java.
- Oracle JDeveloper Un IDE para Java con soporte de diagramas UML.
- Visual Paradigm for UML, Herramienta de modelado UML y herramienta CASE que cuenta con una versión gratuita denominada Community Edition.

Otro software

Software comercial de modelado UML:

- Enterprise Architect de Sparx Systems
- Borland Together
- Corel iGrafx
- Microsoft Visio
- PowerDesigner de Sybase

- Rational Rose de IBM
- Poseidon for UML de GentleWare
- MagicDraw UML

Estandarización de UML

Además de haberse convertido en un estándar *de facto*, UML es un estándar industrial promovido por el grupo OMG al mismo nivel que el estándar CORBA para intercambio de objetos distribuidos. Para la revisión de UML se formaron dos "corrientes" que promovían la aparición de la nueva versión desde distintos puntos de vista. Finalmente se impuso la visión más industrial frente a la académica. Recientemente se ha publicado la versión 2.0 en la que aparecen muchas novedades y cambios que, fundamentalmente, se centran en resolver carencias prácticas. Además, esta versión recibe diversas mejoras que provienen del lenguaje SDL.

- **Microsoft SQL Server**

Microsoft SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje Transact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Así de tener unas ventajas que a continuación se pueden describir.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de

datos como son *Oracle*, *Sybase ASE*, *PostgreSQL*, *Interbase*, *Firebird* o *MySQL*

Características de *Microsoft SQL Server*

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma *gratuita*.

Es común desarrollar completos proyectos complementando *Microsoft SQL Server* y Microsoft Access a través de los llamados **ADP** (Access Data Project). De esta forma se completa la base de datos (*Microsoft SQL Server*), con el entorno de desarrollo (VBA Access), a través de la implementación de

aplicaciones de dos capas mediante el uso de formularios Windows.

Para el desarrollo de aplicaciones más complejas (tres o más capas), *Microsoft SQL Server* incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

- **Microsoft Visual Studio**

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

Versiones

- Microsoft Visual Studio 5
- Microsoft Visual Studio 6
- Microsoft Visual Studio .NET (2002)

- Microsoft Visual Studio .NET 2003
- Microsoft Visual Studio 2005
- Microsoft Visual Studio 2008

A partir de la versión 2005 Microsoft ofrece gratuitamente las *Express Editions*. Estas son varias ediciones básicas separadas por lenguajes de programación o plataforma enfocadas para novatos y entusiastas. Estas ediciones son iguales al entorno de desarrollo comercial pero sin características avanzadas. Las ediciones que hay son:

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual J# Express Edition (Desapareció en Visual Studio 2008)
- Visual Web Developer Express Edition (para programar en ASP.NET)

Adicionalmente, Microsoft ha puesto gratuitamente a disposición de todo el mundo una versión reducida de MS SQL Server llamada SQL Server Express Edition cuyas principales limitaciones son que no soporta bases de datos superiores a 4 GB de tamaño, únicamente utiliza un procesador y un Gb de Ram, y no cuenta con el Agente de SQL Server.

En el pasado se incluyeron los siguientes productos:

- Visual InterDev

- Visual J++
- Visual FoxPro
- Visual SourceSafe

Historia

Visual Studio 5.0

Microsoft presentó la primera versión de Visual Studio en 1997, incluyendo por primera vez en el mismo paquete muchas de sus herramientas de programación. Visual Studio 5.0 fue lanzado al mercado en dos ediciones: Professional y Enterprise. Incluía Visual Basic 5.0 y Visual C++ 5.0, para programación en Windows principalmente; Visual J++ 1.1 para programación en Java y Windows; y Visual FoxPro 5.0 para programación en xBase. Introdujo Visual Interdev para la creación dinámica de sitios web mediante ASP (Active Server Pages). Se incluía una réplica de la librería Microsoft Developer Network a modo de documentación.

Visual Studio 5.0 supuso el primer intento de Microsoft para que varios lenguajes utilizaran el mismo entorno de desarrollo. Visual C++, Visual J++, Interdev y MSDN Library hacían uso de un único entorno, denominado Developer Studio. Por otro lado, Visual Basic y Visual FoxPro usaban diferentes entornos.

Visual Studio 6.0

La siguiente versión, la 6.0, se lanzó en 1998 y fue la última versión en ejecutarse en la plataforma Win9x.^[1]

Los números de versión de todas las partes

constituyentes pasaron a 6.0, incluyendo Visual J++ y Visual InterDev que se encontraban en las versiones 1.1 y 1.0 respectivamente. Esta versión fue la base para el sistema de desarrollo de Microsoft para los siguientes 4 años, en los que Microsoft migró su estrategia de desarrollo al Framework .NET.

Visual Studio 6.0 fue la última versión en que Visual Basic se incluía de la forma en que se conocía hasta entonces; versiones posteriores incorporarían una versión muy diferente del lenguaje con muchas mejoras, fruto de la plataforma .NET. También supuso la última versión en incluir Visual J++, que proporcionaba extensiones de la plataforma Java, lo que lo hacía incompatible con la versión de Sun Microsystems. Esto acarreó problemas legales a Microsoft, y se llegó a un acuerdo en el que Microsoft dejaba de comercializar herramientas de programación que utilizaran la máquina virtual de Java.

Aunque el objetivo a largo plazo de Microsoft era unificar todas las herramientas en un único entorno, esta versión en realidad añadía un entorno más a Visual Studio 5.0: Visual J++ y Visual Interdev se separaban del entorno de Visual C++, al tiempo que Visual FoxPro y Visual Basic seguían manteniendo su entorno específico.

Visual Studio .NET (2002)

En esta versión se produjo un cambio sustancial, puesto que supuso la introducción de la plataforma .NET de Microsoft. .NET es una plataforma de ejecución

intermedia multilenguaje, de forma que los programas desarrollados en .NET no se compilan en lenguaje máquina, sino en un lenguaje intermedio (CIL - Common Intermediate Language) denominado Microsoft Intermediate Language (MSIL). En una aplicación MSIL, el código no se convierte a lenguaje máquina hasta que ésta se ejecuta, de manera que el código puede ser independiente de plataforma (al menos de las soportadas actualmente por .NET). Las plataformas han de tener una implementación de Infraestructura de Lenguaje Común (CLI) para poder ejecutar programas MSIL. Actualmente se pueden ejecutar programas MSIL en Linux y Mac OS X usando implementaciones de .NET que no son de Microsoft, tales como Mono y DotGNU.

Visual Studio .NET 2002 supuso también la introducción del lenguaje C#, un lenguaje nuevo diseñado específicamente para la plataforma .NET, basado en C++ y Java. Se presentó también el lenguaje J# -sucesor de J++- el cual, en lugar de ejecutarse en una máquina virtual de Java, se ejecuta únicamente en el framework .NET. El lenguaje Visual Basic fue remodelado completamente y evolucionó para adaptarse a las nuevas características de la plataforma .NET, haciéndolo mucho más versátil y dotándolo con muchas características de las que carecía. Algo similar se llevó a cabo con C++, añadiendo extensiones al lenguaje llamadas Managed Extensions for C++ con el fin de que los programadores pudieran crear programas en .NET.

Por otra parte, Visual FoxPro pasa a comercializarse por separado.

Todos los lenguajes se unifican en un único entorno. La interfaz se mejora notablemente en esta versión, siendo más limpia y personalizable.

Visual Studio .NET puede usarse para crear programas basados en Windows (usando Windows Forms en vez de COM), aplicaciones y sitios web (ASP.NET y servicios web), y dispositivos móviles (usando el .NET Compact Framework).

Esta versión requiere un sistema operativo basado en NT. La versión interna de Visual Studio .NET es la 7.0.

Visual Studio .NET 2003

Visual Studio .NET 2003 supone una actualización *menor* de Visual Studio .NET. Se actualiza el .NET Framework a la versión 1.1. También se añade soporte con el fin de escribir aplicaciones para determinados dispositivos móviles, ya sea con ASP.NET o con el .NET Compact Framework. Además el compilador de Visual C++ se mejora para cumplir con más estándares, el Visual C++ Toolkit 2003.

Visual Studio 2003 se lanza en 4 ediciones: Academic, Professional, Enterprise Developer, y Enterprise Architect. La edición Enterprise Architect incluía una implantación de la tecnología de modelado Microsoft Visio, que se centraba en la creación de representaciones visuales de la arquitectura de la aplicación basadas en UML. También se introdujo

"Enterprise Templates", para ayudar a grandes equipos de trabajo a estandarizar estilos de programación e impulsar políticas de uso de componentes y asignación de propiedades.

Microsoft lanzó el *Service Pack 1* para Visual Studio 2003 el 13 de Septiembre de 2006.

La versión interna de Visual Studio .NET 2003 es la 7.1 aunque el formato del archivo es 8.0.

Microsoft ha anunciado que Studio 2003 no funciona, ni será soportado en su sistema operativo Windows Vista, pero aclaró recientemente que los titulares de licencias Visual Studio 2003 recibirán el beneficio de hacer un "upgrade" (actualización) a Visual Studio 2008 sin cargo.

Visual Studio 2005

Visual Studio 2005 se empezó a comercializar a través de Internet a partir del 4 de Octubre de 2005 y llegó a los comercios a finales del mes de Octubre en inglés. En castellano no salió hasta el 4 de Febrero de 2006. Microsoft eliminó .NET, pero eso no indica que se alejara de la plataforma .NET, de la cual se incluyó la versión 2.0.

La actualización más importante que recibieron los lenguajes de programación fue la inclusión de *tipos genéricos*, similares en muchos aspectos a las plantillas de C#. Con esto se consigue encontrar muchos más errores en la compilación en vez de en tiempo de ejecución, incitando a usar comprobaciones estrictas en

áreas donde antes no era posible. C++ tiene una actualización similar con la adición de C++/CLI como sustituto de C# manejado.

Se incluye un diseñador de implantación, que permite que el diseño de la aplicación sea validado antes de su implantación. También se incluye un entorno para publicación web y pruebas de carga para comprobar el rendimiento de los programas bajo varias condiciones de carga.

Visual Studio 2005 también añade soporte de 64-bit. Aunque el entorno de desarrollo sigue siendo una aplicación de 32 bits Visual C++ 2005 soporta compilación para x86-64 (AMD64 e Intel 64) e IA-64 (Itanium). El SDK incluye compiladores de 64 bits así como versiones de 64 bits de las librerías.

Visual Studio 2005 tiene varias ediciones radicalmente distintas entre sí: Express, Standard, Professional, Tools for Office, y 5 ediciones Visual Studio Team System. Éstas últimas se proporcionaban conjuntamente con suscripciones a MSDN cubriendo los 4 principales roles de la programación: Architects, Software Developers, Testers, y Database Professionals. La funcionalidad combinada de las 4 ediciones Team System se ofrecía como la edición Team Suite.

Tools for the Microsoft Office System está diseñada para extender la funcionalidad a Microsoft Office.

Las ediciones Express se han diseñado para principiantes, aficionados y pequeños negocios, todas

disponibles gratuitamente a través de la página de Microsoft^[2] se incluye una edición independiente para cada lenguaje: Visual Basic, Visual C++, Visual C#, Visual J# para programación .NET en Windows, y Visual Web Developer para la creación de sitios web ASP.NET. Las ediciones express carecen de algunas herramientas avanzadas de programación así como de opciones de extensibilidad.

Se lanzó el service Pack 1 para Visual Studio 2005 el 14 de Diciembre de 2006.

La versión interna de Visual Studio 2005 es la 8.0, mientras que el formato del archivo es la 9.0.

Visual Studio 2008 [

Visual Studio 2008 fue publicado (RTM) el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de Febrero de 2008.

El nuevo framework (.Net 3.5) está diseñado para aprovechar las ventajas que ofrece el nuevo sistema operativo "Windows Vista" a través de sus subsistemas "Windows Communication Foundation" (WCF) y "Windows Presentation Foundation" (WPF).El primero tiene como objetivo la construcción de aplicaciones orientadas a servicios mientras que el último apunta a la creación de interfaces de usuario más dinámicas que las conocidas hasta el momento.

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan entre otras, las siguientes novedades.

- La mejora en las capacidades de Pruebas Unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para depurar y optimizar.
- Con Visual Studio Tools for Office (VSTO) integrado con Visual Studio 2008 es posible desarrollar rápidamente aplicaciones de alta calidad basadas en la interfaz de usuario (UI) de Office que personalicen la experiencia del usuario y mejoren su productividad en el uso de Word, Excel, PowerPoint, Outlook, Visio, InfoPath y Project. Una completa compatibilidad para implementación con ClickOnce garantiza el entorno ideal para una fácil instalación y mantenimiento de las soluciones Office.
- Visual Studio 2008 permite incorporar características del nuevo Windows Presentation Foundation sin dificultad tanto en los formularios de Windows existentes como en los nuevos. Ahora es posible actualizar el estilo visual de las aplicaciones al de Windows Vista debido a las mejoras en Microsoft

Foundation Class Library (MFC) y Visual C++. Visual Studio 2008 permite mejorar la interoperabilidad entre código nativo y código manejado por .NET. Esta integración más profunda simplificará el trabajo de diseño y codificación.

- LINQ (Language Integrated Query) es un nuevo conjunto de herramientas diseñado para reducir la complejidad del acceso a Base de Datos, a través de extensiones para C++ y Visual Basic así como para Microsoft .NET Framework. Permite filtrar, enumerar, y crear proyecciones de muchos tipos y colecciones de datos utilizando todos la misma sintaxis, prescindiendo del uso de lenguajes especializados como SQL o XPath.
- Visual Studio 2008 ahora permite la creación de soluciones multiplataforma adaptadas para funcionar con las diferentes versiones de .Net Framework: 2.0. (Incluido con Visual Studio 2005), 3.0 (incluido en Windows Vista) y 3.5 (incluido con Visual Studio 2008).
- .NET 3.5 incluye biblioteca ASP.NET AJAX para desarrollar aplicaciones web más eficientes, interactivas y altamente personalizadas que funcionen para todos los navegadores más populares y utilicen las últimas tecnologías y herramientas Web, incluyendo Silverlight y Popfly.

- **Asp.net**

Active Server Pages (ASP) es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services (IIS).

Descripción

La tecnología ASP está estrechamente relacionada con el modelo tecnológico de su fabricante. Intenta ser solución para un modelo de programación rápida ya que *programar en ASP es como programar en Visual Basic*, por supuesto con muchas limitaciones.

Lo interesante de este modelo tecnológico es poder utilizar diversos componentes ya desarrollados como algunos controles ActiveX así como componentes del lado del servidor, tales como CDONTS, por ejemplo, que permite la interacción de los scripts con el servidor SMTP que integra IIS.

Se facilita la programación de sitios web mediante varios objetos integrados, como por ejemplo un objeto de sesión basada en cookies, que mantiene las variables mientras se pasa de página a página.

Versiones

Ha pasado por cuatro iteraciones mayores:

- ASP 1.0 (distribuido con IIS 3.0)
- ASP 2.0 (distribuido con IIS 4.0)
- ASP 3.0 (distribuido con IIS 5.0)

- ASP.NET (parte de la plataforma .NET de Microsoft).

Las versiones pre-.NET se denominan actualmente (desde 2002) como ASP *clásico*.

En el último ASP clásico, ASP 3.0, hay siete objetos integrados disponibles para el programador: Application, ASPError, Request, Response, Server, Session y ObjectContext. Cada objeto tiene un grupo de funcionalidades frecuentemente usadas y útiles para crear páginas web dinámicas.

Desde 2002, el ASP *clásico* está siendo reemplazado por ASP.NET, que, entre otras cosas, reemplaza los lenguajes interpretados como VBScript o JScript por lenguajes *compilados* a código intermedio (llamado MSIL o Microsoft Intermediate Language) como Microsoft Visual Basic, C#, o cualquier otro lenguaje que soporte la plataforma .NET. El código MSIL se compila con posterioridad a código nativo.

- **C#**

C# (pronunciado "*si sharp*") es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de

Java aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi).

El símbolo # viene de sobreponer "+" sobre "+" y eliminar las separaciones, indicando así su descendencia de C++.

C#, como parte de la plataforma.NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 19 de noviembre de 2007 salió la versión 3.0 de C# destacando entre las mejoras los tipos implícitos, tipos anónimos y el LINQ (Language Integrated Query).

Aunque C# forma parte de la plataforma.NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el Framework de DotGNU - Mono que no genera programas para dicha plataforma, sino para una plataforma diferente como Win32 o UNIX / Linux.

- **Visual Paradigm for UML**

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software

de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Lista de características:

Soporte de UML versión

2.1 Diagramas de Procesos de Negocio - Proceso, Decisión, Actor de negocio, Documento Modelado colaborativo con CVS y Subversión (nueva característica)

Interoperabilidad con modelos UML2 (meta modelos UML 2.x para plataforma Eclipse) a través de XMI (nueva característica).

Ingeniería de ida y vuelta Ingeniería inversa - Código a modelo, código a diagrama Ingeniería inversa Java, C++, Esquemas XML, XML, .NET exe/dll, CORBA IDL Generación de código - Modelo a código, diagrama a código Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso. Diagramas EJB - Visualización de sistemas EJB. Generación de código y despliegue de EJB's - Generación de beans para el desarrollo y despliegue de aplicaciones.

Diagramas de flujo de datos. Soporte ORM -
Generación de objetos Java desde la base de datos
Generación de bases de datos - Transformación de
diagramas de Entidad-Relación en tablas de base de
datos Ingeniería inversa de bases de datos - Desde
Sistemas Gestores de Bases de Datos (DBMS)
existentes a diagramas de Entidad-Relación
Generador de informes para generación de
documentación

Distribución automática de diagramas - Reorganización
de las figuras y conectores de los diagramas UML

Importación y exportación de ficheros XMI
Integración con Visio - Dibujo de diagramas UML con
plantillas (stencils) de MS Visio
Editor de figuras

Otras herramientas y plugins de modelado UML:

Plataforma Java (Windows/Linux/Mac OS X):

SDE para Eclipse

SDE para NetBeans

SDE para Sun ONE

SDE para Oracle JDeveloper

SDE para JBuilder

SDE para IntelliJ IDEA

SDE para WebLogic Workshop

Plataforma Windows:

SDE para Microsoft Visual Studio

- **Windows Server 2003**

Windows Server 2003 es un sistema operativo de la familia Windows de la marca Microsoft para servidores que salió al mercado en el año 2003. Está basada en tecnología NT y su versión del núcleo NT es la 5.2.

En términos generales, Windows Server 2003 se podría considerar como un Windows XP modificado, no con menos funciones, sino que estas están deshabilitadas por defecto para obtener un mejor rendimiento y para centrar el uso de procesador en las características de servidor, por ejemplo, la interfaz gráfica denominada *Luna* de Windows XP viene desactivada y viene con la interfaz clásica de Windows. Sin embargo, es posible volver a activar las características mediante comandos *services.msc*. En internet existen varios trucos para hacerlo semejante a Windows XP.

Características

Sus características más importantes son:

- Sistema de archivos NTFS:
 1. Cuotas.
 2. cifrado y compresión de archivos, carpetas y no unidades completas.
 3. permite montar dispositivos de almacenamiento sobre sistemas de archivos de otros dispositivos al estilo unix.

- Gestión de almacenamiento, backups... incluye gestión jerárquica del almacenamiento, consiste en utilizar un algoritmo de caché para pasar los datos menos usados de discos duros a medios ópticos o similares más lentos, y volverlos a leer a disco duro cuando se necesitan.
- Windows Driver Model: Implementación básica de los dispositivos más utilizados, de esa manera los fabricantes de dispositivos sólo han de programar ciertas especificaciones de su hardware.
- ActiveDirectory Directorio de organización basado en LDAP, permite gestionar de forma centralizada la seguridad de una red corporativa a nivel local.
- Autenticación Kerberos5.
- DNS con registro de IP's dinámicamente.
- Políticas de seguridad.

Servidores

Los servidores que maneja Windows 2003 son:

- Servidor de archivos.
- Servidor de impresiones.
- Servidor de aplicaciones.
- Servidor de correo (SMTP/POP).
- Servidor de terminal.

- Servidor de Redes privadas virtuales (VPN) (o acceso remoto al servidor).
- Controlador de Dominios (mediante Active Directory).
- Servidor DNS.
- Servidor DHCP.
- Servidor de Streaming de Video.
- Servidor WINS.

Servidor de Impresión

Teniendo ya en cuenta que para activar el servidor de impresión en Windows Server 2003 tiene que tener instalado el Windows Server, luego implementar una red cliente servidor y configurar la impresora en las PC's y está listo para que la pueda utilizar, ya sea del servidor o de una "PC hija".

Mejoras Respecto Windows 2000 Server

Diferencias principales con Windows 2000 server

1. Durante la instalación arranca con el mínimo de servicios activados para no comprometer la seguridad del sistema.
2. Mejoras en el manejo de políticas de seguridad.
3. Active Directory ya no utiliza NetBIOS sino que es necesaria la presencia de un DNS que soporte Service Records (detección de servicios ofrecidos por una máquina a través de un DNS).

Versiones

Actualmente existen cuatro versiones de Windows 2003, aunque todas ellas cuentan a su vez con versiones de 32 y 64 bits (excepto Web Edition). Las versiones son:

- **Web Edition** Diseñado para los servicios y el hospedaje Web.
- **Standard Edition** El más versátil de todos, ofrece un gran número de servicios útiles para empresas de cualquier tamaño.
- **Enterprise Edition** Para empresas de mayor tamaño que la *Standard Edition*.
- **Datacenter Edition** Para empresas que requieran bases de datos más escalables y un procesamiento de transacciones de gran volumen.

Las diferencias entre las versiones, explicadas en mayor detalle, pueden encontrarse en la Web de Microsoft.

Windows Server 2003 RTM

Soporte para Windows Server 2003 sin Service Pack finalizó el 10 de abril de 2007.

Service Pack 1 (SP1)

El 30 de marzo de 2005, Microsoft lanza (Service Pack 1), para todas las versiones de Windows 2003. Con él, dotan al Sistema operativo de las mejoras incluidas en el SP2 de Windows XP, tales como una nueva interfaz para el Cortafuegos (aunque al tratarse de un servidor,

el cortafuegos estará deshabilitado por defecto), o la corrección de todos los bugs aparecidos hasta la fecha en Windows 2003. El soporte de Windows Server 2003 Service Pack 1 finalizará el 14 de abril de 2009.

Service Pack 2 (SP2)

El 12 de marzo de 2007 se lanzó el Service Pack 2 de Windows Server 2003. Este SP2 está concebido como una actualización para Windows Server 2003 R2, a su vez una actualización del Server 2003 original que Microsoft lanzó en diciembre del 2005. No obstante, este Service Pack se instala tanto sobre versiones R2 del sistema como sobre la versión original.

Entre las novedades que podemos encontrar en este Service Pack destacamos:

- Microsoft Management Console (MMC) 3.0, que hace del proceso de creación de directivas (policy) de grupos introducido en el anterior service pack, algo más intuitivo y manejable.
- Windows Deployment Services en sustitución de Remote Installation Services para la realización de instalaciones remotas del sistema (sin encontrarse delante de la computadora en la cual se va a instalar ni tener el DVD del sistema en el lector de esta).
- Scalable Networking Pack (SNP) permite escalar las redes corporativas (hacerlas crecer y controlar dicho crecimiento en la dirección que queramos) para hacer frente a las crecientes

demandas de ancho de banda por parte de algunas aplicaciones concretas.

- El cliente de conexión a redes inalámbricas soporta ahora autenticación WPA2.
- Incluye todas las actualizaciones de seguridad y parches lanzados hasta la fecha.

Este Service Pack ya puede descargarse para su instalación o en formato de imagen ISO para grabar en CD o DVD para las plataformas de 32 y 64 bits. El Soporte Técnico para este Service Pack finalizará 12 ó 24 meses presentado el próximo Service Pack, o cuando finalice el ciclo de vida del producto, lo que ocurra primero.

- **HTML**

HTML, siglas de **HyperText Markup Language** (*Lenguaje de Marcas de Hipertexto*), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores). Por convención, los archivos de formato HTML usan la extensión `.htm` o `.html`.

- **Alfresco**

Alfresco es un sistema de administración de contenidos libre, basado en estándares abiertos y de escala empresarial para sistemas operativos tipo Unix y Otros. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Alfresco incluye un repositorio de contenidos, un framework de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS que provee compatibilidad de sistemas de archivos en Windows y sistemas operativos tipo Unix, un sistema de administración de contenido web capacidad de virtualizar aplicaciones web y sitios estáticos vía Apache Tomcat, búsquedas vía el motor Lucene y flujo de trabajo en jBPM. Alfresco está desarrollado en Java.

Historia

Alfresco fue fundado en 2005 por John Newton, cofundador de Documentum y John Powell, ex COO de Business Objects. Entre sus inversionistas se incluyeron

SAP, Accel Partners y Mayfield Fund. El staff técnico original estaba compuesto por ingenieros provenientes de Documentum y Oracle. Mientras que el producto inicial de Alfresco Inc. estaba enfocado inicialmente en la gestión de documentos, en mayo de 2006 se anunció la intención de expandirlo hasta abarcar la gestión de contenido web tras contratar personal técnico y directivo de Interwoven. En 2007, Alfresco contrató al principal ingeniero de ventas de Vigente.

- **Microsoft Sharepoint**

Microsoft Office SharePoint Portal Server 2003 es una plataforma web de trabajo colaborativo y gestión documental, especialmente orientada a documentos Microsoft Office. Se apoya directamente en SQL Server 2000/2005 a partir de SP2 y Windows Server 2003 con IIS para el rol de presentación. Estos roles pueden estar en el mismo servidor así como llegar a arquitecturas mas escalables con varios servidores. Requiere Active Directory para la validación de Windows Server 2003.

Como parte cliente SPS 2003 se integra en su totalidad con MS Office 2003. Desde Excel o Word en la creación y edición de documentos, hasta la integración total de Outlook 2003 pudiendo compartir entre cliente y servidor, calendario, contactos o mandar correos a librerías de documentos para su publicación en el portal.

Microsoft Office SharePoint Portal Server 2003 tiene su continuación con Microsoft Office Sharepoint Server 2007 o "MOSS 2007".

I.1.1.6. Propósito, alcance y objetivos

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han celebrado con el stakeholder de la empresa desde el inicio del proyecto, Ing. Fernando Cortez.

La Corte Departamental Electoral de Tarija tiene como funciones administrar el registro civil, los procesos electorales y la toma de decisiones colectivas e impulsar una cultura democrática contribuyendo a la consolidación y profundización de la democracia boliviana.

Al ser esta una entidad dedicada a los procesos electorales se genera mucha documentación la cual se almacena en archivos y folders, es hora de que también la Corte Departamental Electoral de Tarija se beneficie de una moderna gestión documental, por lo que el presente trabajo busca realizar el Diseño de un Sistema de Gestión documental para su posterior implementación y puesta en marcha, con los beneficios que ello conlleva.

El proyecto tiene como objetivo implantar en la Corte Departamental Electoral un Sistema Informático de Gestión Documental para contribuir a una eficiente gestión documental y mejorar la calidad en el servicio de tratamiento documental.

El sistema se divide en los siguientes bloques:

- Reportes
 - Reportes de caratula del documento.

- Administración
 - Administración de usuarios.
 - Administración de la documentación almacenada.

I.1.1.7. Suposiciones y restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder de la empresa son:

Debe contemplarse las implicaciones de los siguientes puntos críticos:

- Seguridad de transacciones e intercambio de información.
- Adaptación a la normativa de Protección de Datos.

I.1.1.8. Entregables del Proyecto

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto y que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la filosofía de RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una

versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

3) Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

4) Visión

Este documento define la visión del producto desde la perspectiva del cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema.

5) Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, post-condiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

6) Especificaciones Adicionales

Este documento capturará todos los requisitos que no han sido incluidos como parte de los casos de uso y se refieren requisitos no-funcionales globales. Dichos requisitos incluyen: requisitos legales o normas, aplicación de estándares, requisitos de calidad del producto, tales como: confiabilidad, desempeño, etc., u otros requisitos de ambiente, tales como: sistema operativo, requisitos de compatibilidad, etc.

7) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos

ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán desechados. Asimismo, este artefacto, será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

8) Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

Dentro de los Diagramas que presentaremos, solo consideramos los más importantes dentro de cada grupo de diagramas, por ejemplo, con los casos de uso y diagramas de actividad representamos el diagramas de comportamiento, con los diagramas de secuencia es para los diagramas de iteración, y final mente los diagramas de clases que es el más representativo de los diagramas de estructura.

9) Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los

datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases (donde se utiliza un profile UML para Modelado de Datos, para conseguir la representación de tablas, claves, etc.).

10) Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema. (Este modelo es sólo una versión preliminar al final de la fase de Elaboración, posteriormente tiene bastante refinamiento).

11) Modelo de Despliegue

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

12) Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

13) Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación.

14) Producto

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente, obteniéndose una nueva reléase al final de cada iteración.

I.1.1.9. Evolución del plan de desarrollo del software

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

I.1.1.10. Organización del Proyecto

I.1.1.10.1. Participantes en el Proyecto

De momento no se incluye el personal que interactuará con el Sistema de Gestión Documental que se estimen convenientes para proporcionar los requisitos y validar el sistema.

El resto del personal del proyecto, considerando las fases de Inicio, Elaboración y dos iteraciones de la fase de Construcción, estará formado por los siguientes puestos de trabajo y personal asociado:

- **Jefe de Proyecto.** Esta labor es ejercida por Adrian Mansilla Salas., alumno del último curso del Programa de Ing. Informática. Con una experiencia modesta en metodologías de desarrollo, herramientas CASE y notaciones, en particular la notación UML y el proceso de desarrollo RUP.
- **Analista de Sistemas.** Esta labor será ejercida por los estudiantes: Adrian Mansilla Salas, Mauricio Rodrigo Bejarano.
- **Analistas - Programadores.** Este trabajo ha sido encomendado a Adrian Mansilla Salas, Mauricio Rodrigo Bejarano.
- **Ingeniero de Software.** Encargada de las pruebas funcionales del sistema, realizará la labor de Adrian Mansilla Salas, Mauricio Rodrigo Bejarano.

I.1.1.10.2. Interfaces externas

La Corte Departamental Electoral de Tarija definirá los participantes del proyecto que proporcionarán los requisitos del sistema, y entre ellos quiénes serán los encargados de evaluar los artefactos de acuerdo a cada parte del sistema y según el plan establecido.

El equipo de desarrollo interactuará activamente con los participantes de la Corte Departamental Electoral de Tarija para la especificación y validación de los artefactos generados.

I.1.1.10.3. Roles y responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Jefe de Proyecto	El jefe de proyecto asigna los recursos, gestiona las prioridades, coordina las interacciones con los clientes y usuarios, y mantiene al equipo del proyecto enfocado en los objetivos. El jefe de proyecto también establece un conjunto de prácticas que aseguran la integridad y calidad de los artefactos del proyecto. Además, el jefe de proyecto se encargará de supervisar el establecimiento de la arquitectura del sistema. Gestión de riesgos. Planificación y control del proyecto.
Analista de Sistemas	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Elaboración del Modelo de Análisis y Diseño. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Construcción de prototipos. Colaboración en la elaboración de las pruebas funcionales, modelo de datos y en las validaciones con el usuario

Ingeniero de Software	Gestión de requisitos, gestión de configuración y cambios, elaboración del modelo de datos, preparación de las pruebas funcionales, elaboración de la documentación. Elaborar modelos de implementación y despliegue.
-----------------------	---

Tabla 2: Roles y responsabilidades

I.1.1.11. Gestión del proceso

I.1.1.11.1. Estimaciones del Proyecto

El presupuesto del proyecto y los recursos involucrados se mencionan al inicio del proyecto.

I.1.1.11.2. Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

I.1.1.11.3. Plan de las fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar).

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	5 semanas

Fase de Elaboración	2	19 semanas
Fase de Construcción	2	2 semanas
Fase de Transición	-	3 semanas

Tabla 3: Plan de las fases

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su

	realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.
--	---

Tabla 4: Hitos de las fases

I.1.1.11.4. Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

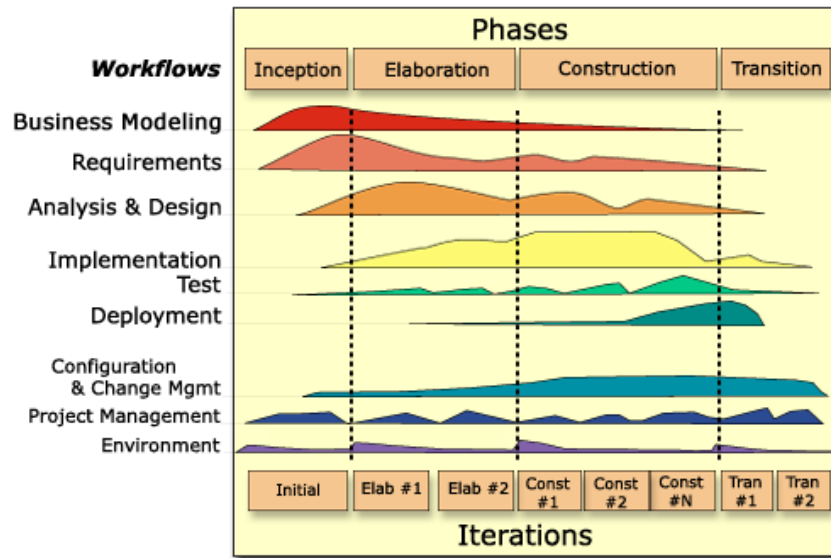


Figura 2: Fase del Workflow

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	07/08/2009 —	08/12/2009

	12/08/2009	
Requisitos		
Glosario	25/08/2009 – 08/12/2009	08/12/2009
Visión	25/08/2009 -08/12/2009	08/12/2009
Modelo de Casos de Uso	25/08/2009 -20/09/2009	08/12/2009
Especificación de Casos de Uso	21/09/2009 -25/09/2009	08/12/2009
Análisis / Diseño	10/08/2009 – 25/09/2009	
Modelo de Análisis / Diseño	10/08/2009 – 30/08/2009	08/12/2009
Modelo de Datos	03/09/2009 – 22/09/2009	08/12/2009
Implementación		
Prototipos de Interfaces de Usuario	11/09/2009 –	Revisar en cada iteración

	25/09/2009	
Modelo de Implementación	27/09/2009 -08/12/2009	Revisar en cada iteración
Pruebas		
Casos de Pruebas Funcionales	11/12/2009 – 13/12/2009	Revisar en cada iteración
Despliegue		
Modelo de Despliegue	11/12/2009 – 13/12/2009	Revisar en cada iteración
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	09/07/2009 -08/12/2009	Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Tabla 5: Calendario del proyecto

I.1.1.12. Seguimiento y control del Proyecto

Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto y por el Comité de Seguimiento y Control.

Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y checklist (listas de verificación) incluidas en RUP.

Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá una baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada sólo por una Solicitud de Cambio aprobada.

I.1.1.13. Glosario

I.1.1.13.1. Introducción

Este documento recoge todos y cada uno de los términos manejados a lo largo de todo el proyecto de desarrollo del Sistema de Gestión Documental SigD. Se trata de un diccionario informal de datos y definiciones de la nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

I.1.1.13.2. Propósito

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo de un sistema para la gestión de documentos. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

I.1.1.13.3. Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos por la Corte Departamental Electoral.

I.1.1.13.4. Referencias

El presente glosario hace referencia a los siguientes documentos:

- Documento Plan de Desarrollo Software
- Documento Visión
- Documentos de Especificación de Casos de Uso.

I.1.1.13.5. Organización del glosario

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según el alfabeto.

I.1.1.13.6. Definiciones

A continuación se presentan todos los términos manejados a lo largo de todo el proyecto de desarrollo de un sistema:

C.D.E.

Corte Departamental Electoral.

Direcciones

Se denomina dirección a cualquiera de las cuatro áreas a nivel ejecutivo en que la Corte Electoral divide sus funciones y funcionarios, por ejemplo la dirección de Informática, Registro Civil, Administración, etc.

Director

El director es la autoridad máxima dentro de las distintas direcciones, superado en jerarquía solo por los vocales.

Respaldo de la Base de Datos

Copia de los datos de la base de datos, para su almacenamiento y uso posterior en caso de que los datos originales hayan sufrido algún daño. Esta operación deberá realizarse periódicamente, para asegurar la utilidad de los datos almacenados.

Asignación

Acción de relacionar una cosa con otra.

Banner

Imagen o logo utilizado en los sitios web como presentación de los mismos, a través de la utilización de imágenes, animaciones o Frases que se utilizan para presentar al sitio.

Campos

Atributos o campos de texto que pertenecen a una entidad, por ejemplo una entidad persona puede tener el campo nombre.

Caracteres

Letras, símbolos, números, etc. utilizado para la creación de nombres, palabras mensajes a utilizar en el llenado de campos de texto.

Check box

Componente utilizado por aplicaciones, páginas web, programas, etc., para poder realizar el marcado de opciones que se realizaban sobre papel y se utilizaban las marcas en X o “tickeos” en cajas o cuadros.

Click derecho

Acción que se presenta en la navegación por las PCs gracias a los mouses o ratones en las computadoras, se denomina así a la acción cuando se aprieta o presiona la tecla o botón derecho del mouse.

Correo Electrónico

Conjunto de palabras que conforman una dirección perteneciente a una sola entidad persona, empresa, etc., y es único que siempre trae el formato: nombrecorreo@nombreservidordecorreo.com.

Data display

Artefacto electrónico que permite la proyección de las imágenes generadas por una computadora es decir lo que se muestra en el monitor se puede proyectar sobre superficies planas elegidas preferentemente blancas, o colores muy claros.

Digital

Se refiere a toda la información que se maneja en las computadoras o en soportes magnéticos como CD, DVD, Diskettes, etc.

Directorio

Espacios existentes en las computadoras que pueden contener o no archivos digitales como imágenes, documentos, música, etc., que constan de un nombre y se ubican en discos duros o soportes magnéticos.

Extensión

Parte del nombre de los archivos digitales y que identifica al archivo es decir lo distingue de otros como una imagen de un sonido.

Lógicamente

Se refiere a acciones que se realizan sobre la computadora.

Login

Palabra que define al usuario como único puede constar de letras, números, caracteres y le permite identificarse puede ser ante aplicaciones que lo requieren.

Nullable

Se refiere a espacios o variables y valores que pueden no contener nada o estar vacíos.

Password

Término inglés que se refiere a utilización de claves que permiten identificar a entidades.

Pulsar

Presionar, apretar.

Recoger

Capturar datos que se han ingresado en campos de entrada de datos.

Release

Versión de algo que es público, que esta liberado.

Ruta

Dirección de un archivo o carpeta es decir su ubicación al explorarlo.

Servidor de archivos

PCs que permiten el almacenamiento de archivos de forma remota desde cualquier otra parte del mundo.

Sistema operativo

Programa que permite el funcionamiento y utilización de una computadora.

Tipo Password

Caracteres o palabras que se muestran solo con un tipo de caracteres es decir están encriptados, no se pueden leer.

SigD

Sistema Informático de Gestión Documental.

Usuario

Persona registrada para usar el sistema.

Publicar Documento

Documento que ya ha pasado por la etapa de revisión y aprobación y esta visible para todos los usuarios registrados en el sistema.

Cadena de Aprobación

Flujo que sigue un documento para pasar de un estado a otro, de rechazado a aprobado.

I.1.1.14. Modelo de casos de uso

Caso de Uso alto Nivel: General

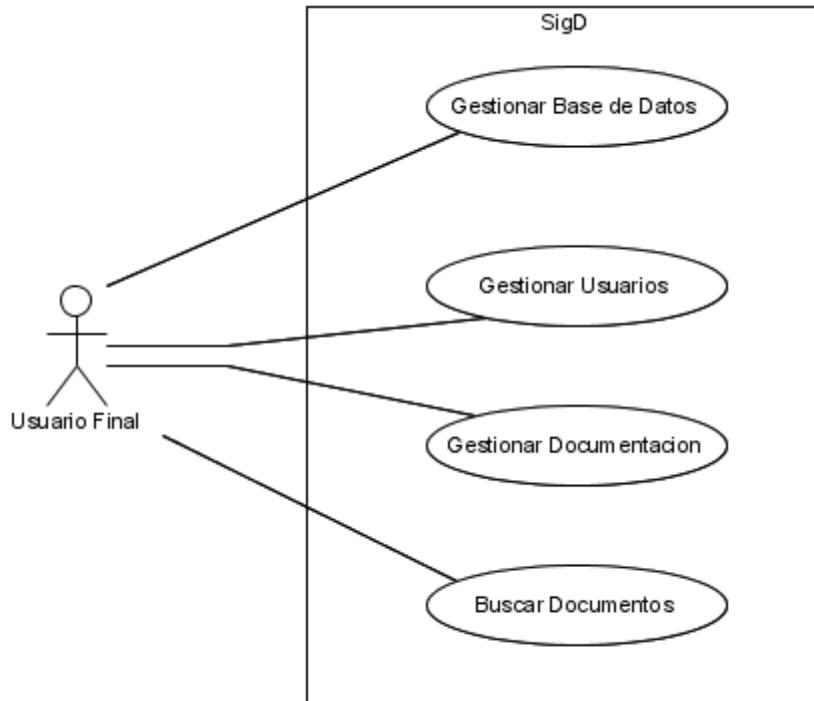


Figura 3: Caso de uso general

Caso de Uso alto Nivel: Usuarios

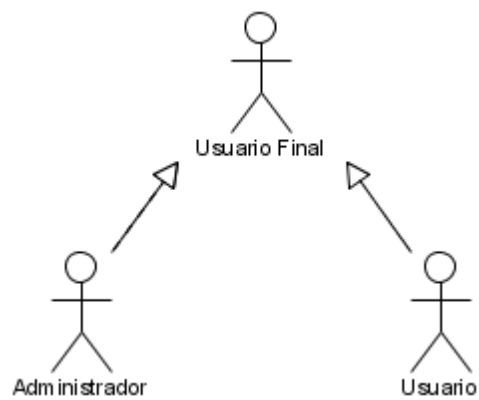


Figura 4: Caso de uso usuarios

Caso de Uso: Gestionar Usuarios

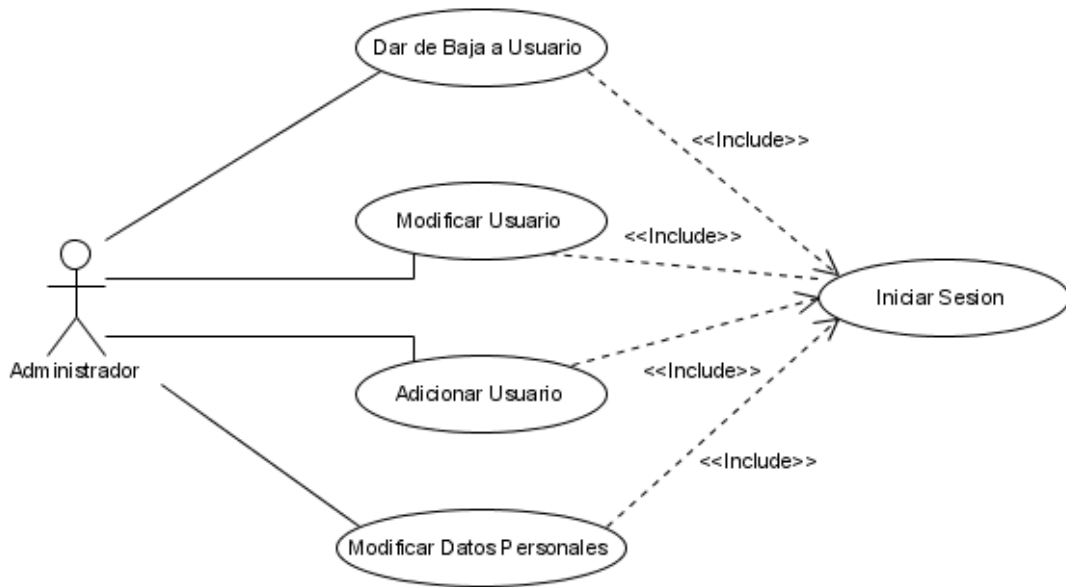


Figura 5: Caso de uso gestionar usuarios

Caso de Uso: Gestionar Base de Datos

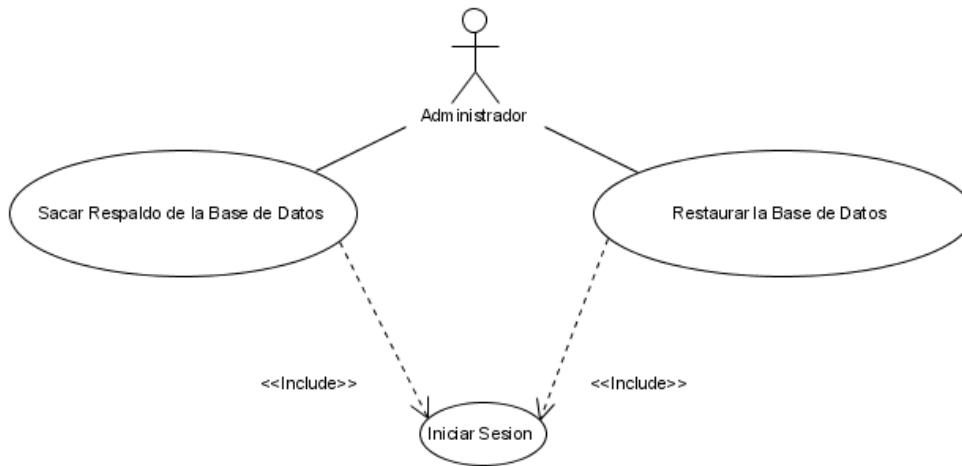


Figura 6: Caso de uso gestionar base de datos

Caso de Uso: Gestionar Documentos

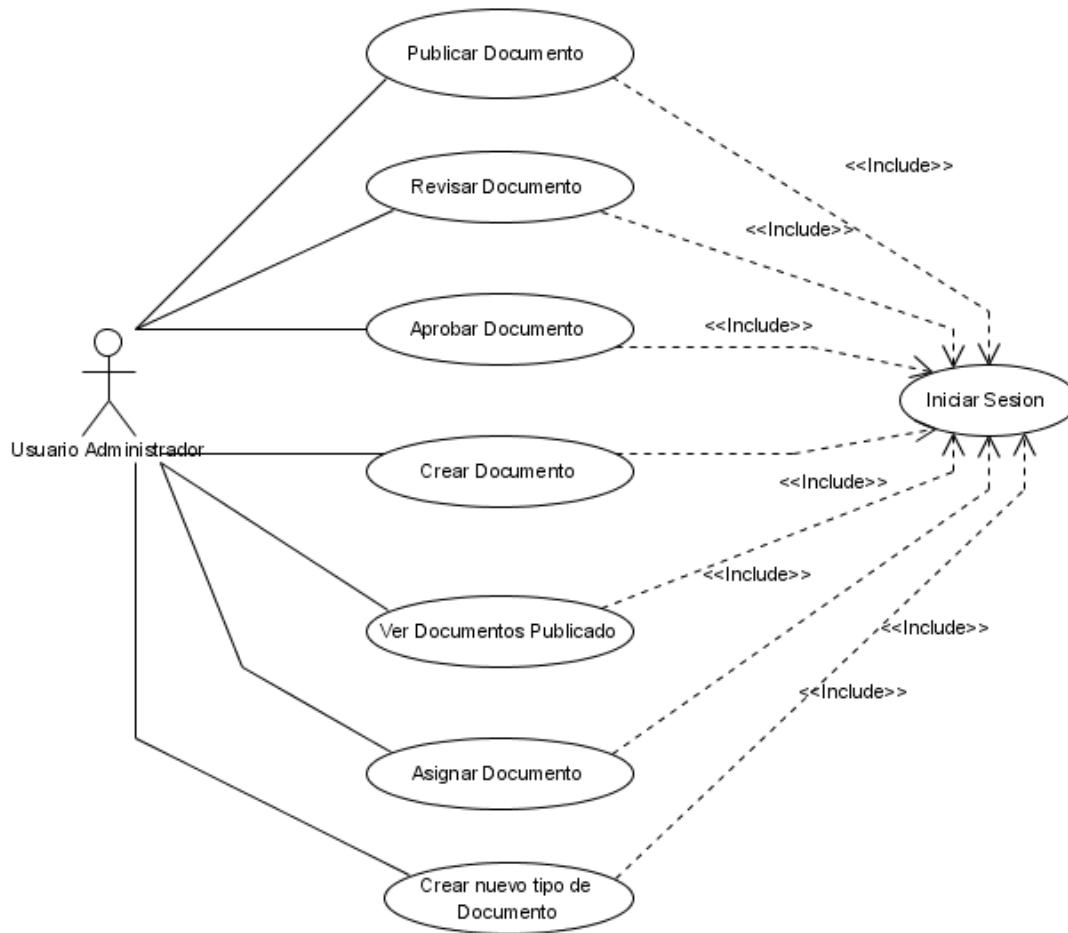


Figura 7: Caso de uso gestionar documentos

Caso de Uso: Realizar Búsquedas

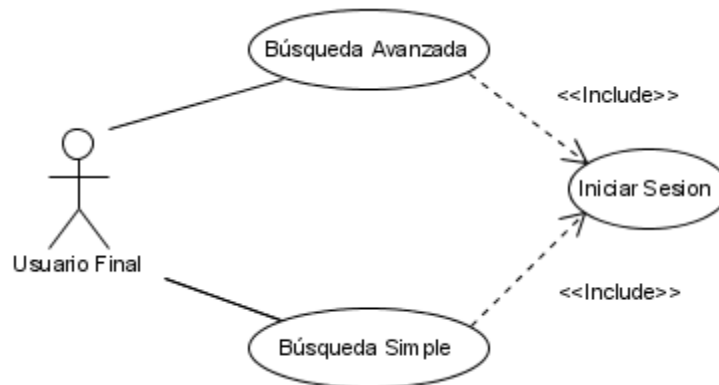


Figura 8: Caso de uso realizar búsquedas

I.1.1.15. Visión

I.1.1.15.1. Introducción

El propósito de éste documento es recoger, analizar y definir las necesidades de alto nivel y las características del Sistema de Gestión de Documental de la Corte Departamental Electoral de Tarija. El documento está centrado en la funcionalidad requerida por el Director de Informática y los participantes del Proyecto.

La funcionalidad se basa principalmente en la Administración y Generación de Reportes de la documentación de la Corte Departamental Electoral de Tarija. Los detalles de cómo cubre los requerimientos se pueden observar en la especificación de los casos de uso y otro documentos adicionales.

I.1.1.15.2. Alcance

El documento Visión se ocupa, como ya se mencionó en el punto anterior de un Sistema de Gestión de Documental para la Corte Departamental Electoral de Tarija.

El Sistema permitirá a la Corte Departamental Electoral de Tarija tener la información necesaria del proceso de administración de la documentación que se genera en la organización, tales como reportes de caratulas para documentos oficiales y tener a disposición toda la documentación que se genera en la organización, etc.

El sistema tiene la base sobre las siguientes características:

Gestión de Base de Datos.- En esta parte el administrador puede sacar respaldos de la Base de Datos del sistema, o puede restaurar la Base de Datos.

Gestión de Usuarios.- Esta parte cubre la gestión de los usuarios que van a utilizar el sistema, como ser altas bajas y modificaciones.

Gestión de Documentos.- En esta parte gestiona toda la documentación, en la cual podemos subir documentos, asignar usuarios, y controlar la versión del documento y su cadena de aprobación.

I.1.1.15.3. Definiciones, acrónimos y observaciones

RUP: Son las siglas de Rational Unified Process. Se trata de una metodología para describir el proceso de desarrollo de software.

SigD: Son la Siglas del Sistema de Gestión Documental.

I.1.1.15.4. Referencias

- Documento Glosario
- Documento Plan de Desarrollo de Software
- Diagramas de Casos de Uso

I.1.1.15.5. Posicionamiento

- **Oportunidad de negocio**

Este sistema permitirá a la Corte Departamental Electoral de Tarija automatizar el control y administración de la documentación que se genera en la organización, lo cual permitirá un acceso rápido y sencillo a los datos, gracias a la implementación de interfaces graficas de usuario sencillas e intuitivas, además de que el acceso a los datos estarán siempre disponibles y actualizados, lo cual es un factor muy importante para poder llevar un control adecuado en la organización.

I.1.1.15.6. Descripción de stakeholders (participantes en el Proyecto) y usuarios

- **Resumen de stakeholders**

Nombre	Descripción	Responsabilidades
Ing. Fernando Cortez Michel	Director y Representante de la Dirección de Informática de la Corte Departamental Electoral de Tarija.	El stakeholder realiza: Representa a todos los usuarios posibles del sistema. Seguimiento del desarrollo del proyecto. Aprueba y desaprueba los requisitos y funcionalidades

Tabla 6: Resumen de stakeholders

I.1.1.15.7. Resumen de usuarios

Nombre	Descripción	Stakeholder
Administrador	Es el encargado de Administrar el Sistema, tiene asignado todos los roles y puede acceder a todas las funcionalidades del Sistema	Administración de Usuarios Administración de la Documentación
Usuario Normal	Es el encargado de revisar y administrar los documentos que tenga asignados y pendientes. También puede generar reportes e imprimir la documentación.	Administración de la Documentación

Tabla 7 : Resumen de usuarios

I.1.1.15.8. Entorno de usuarios

El Sistema será desarrollado en el lenguaje de

Programación Asp.net con C# y el entorno de desarrollo será Visual Studio 2008.

Por motivo de que los ordenadores de la Corte Departamental Electoral de Tarija cuentan con el Sistema Operativo Microsoft Windows Server 2003 la plataforma de ejecución de las interfaces será la plataforma Ms. Windows Server 2003.

- **Perfil de los stakeholders: Administración**

Representante	Ing. Fernando Cortez Michel
Descripción	Director y Representante de la Dirección de Informática de la Corte Departamental Electoral de Tarija.
Tipo	Administrador.
Responsabilidades	Representa a todos los usuarios posibles del sistema. Seguimiento del desarrollo del proyecto. Aprueba y desaprueba los requisitos y funcionalidades
Criterio de Éxito	A definir por el cliente.
Grado de participación	Revisión de requerimientos, estructura del sistema

Tabla 8: Perfil de los stakeholders

I.1.1.15.9. Descripción global del producto

- **Perspectiva del producto**

El producto a desarrollar es un Sistema global para la gestión y administración de la documentación en la Corte Departamental Electoral de Tarija, con la intención de agilizar y automatizar los reportes que genera dicha gestión y administración. Las áreas a tratar por el Sistema son: La Administración de Usuarios, Administración de la Documentación.

- **Resumen de características**

Los beneficios que obtendrá la Corte Departamental Electoral de Tarija a partir del producto son:

Beneficios para la Jefatura	Características que lo apoyan
Mayor agilidad en la elaboración de Reportes	Aplicación que le permite generar de forma automática los reportes.
Facilidad para la gestión de usuarios del Sistema de Gestión Documental	Base de datos con la información de todos los usuarios del Sistema de Gestión Documental.
Mayor agilidad para la gestión y Administración de la Documentación.	Base de datos con la información de toda la documentación almacenada en el sistema.

Tabla 9: Resumen de características

I.1.1.16. Especificaciones de casos de uso

Caso de Uso:	Iniciar sesión
Actores:	Usuario
Descripción: <p>Este caso de uso hace referencia al control de acceso del usuario (administrador, normal) autorizado al manejo del sistema, teniendo que introducir el usuario y clave, dependiendo del tipo de usuario se habilitaran o deshabilitaran algunas tareas si el rol es igual a Administrador se habilitaran todas las tareas, si es igual a normal se restringen tareas.</p>	
Precondición: <p>El personal autorizado debe Introducir usuario y clave.</p>	
Flujo Principal: <ol style="list-style-type: none">1.- El usuario introduce usuario y clave.2.- Se verifica la existencia del usuario y clave.3.- Si existe, se muestra la pantalla principal del sistema.	
Flujo de Excepción: <p>Mensaje de error: usuario o clave incorrectos (se da cuando no se encuentra el usuario registrado en el sistema o cuando los datos son incorrectos).</p>	

Tabla 10: E.C.U. Iniciar sesión

Caso de Uso:	Adicionar Usuario
Actores:	Administrador
Descripción:	
En este caso de uso el administrador realiza la adición de un nuevo usuario a la base de datos del SigD.	
Propósito:	
Registrar a un nuevo usuario, que puede ser tanto administrador como usuario normal.	
Referencia Cruzada:	
No tiene	
Precondición:	
1.- El administrador haya iniciado sesión en el sistema.	
Flujo Principal:	
1.- El administrador selecciona adicionar usuario.	
2.- El administrador ingresa los datos del nuevo usuario.	
3.- Se verifica si los datos son correctos.	
4.- Si son correctos se verifica si el usuario ya está registrado.	
5.- Si no lo esta se almacenan los datos del nuevo personal.	
Subflujo :	
1.- Mensaje: El usuario ha sido registrado.	
Flujo de Excepción:	

1.- Mensaje error: Datos incorrectos.

2.- Mensaje error: Personal ya existe.

Tabla 11: E.C.U. Adicionar usuario

Caso de Uso:	Modificar usuario
Actores:	Administrador
Descripción: En este caso de uso el administrador realiza la modificación de uno de los registros del usuario de la base de datos SigD.	
Propósito: Modificar el registro de un usuario, que puede ser tanto administrador como usuario normal.	
Referencia Cruzada: No tiene.	
Precondición: 1.- El administrador haya iniciado sesión en el sistema.	
Flujo Principal: 1.- El administrador selecciona Modificar Usuario. 2.- El administrador selecciona un usuario. 3.- El administrador edita los datos (Nombre, apellido paterno, apellido materno) del usuario escogido. 4.- El administrador selecciona Update.	

5.- Si son correctos se modifican los datos del usuario escogido.
Subflujo : 1.- Mensaje: personal modificado
Flujo de Excepción: 1.- Mensaje error: datos incorrectos

Tabla 12: E.C.U. Modificar usuario

Caso de Uso:	Dar de baja Usuario
Actores:	Administrador
Descripción: En este caso de uso el administrador realiza la acción de dar de baja a un usuario de la base de datos SigD.	
Propósito: Dar de baja a un usuario, que puede ser tanto administrador como usuario normal.	
Referencia Cruzada: No tiene	
Precondición: 1.- El administrador haya iniciado sesión en el sistema.	
Flujo Principal: 1.- El administrador selecciona dar de baja. 2.- El administrador selecciona un usuario.	

<p>3.- Se le pide al administrador confirmar la acción.</p> <p>4.- Si es confirmada se da de baja al usuario escogido.</p>
<p>Subflujo :</p> <p>1.- Dialogo: Está seguro de dar de baja al usuario.</p>
<p>Flujo de Excepción:</p> <p>No tiene</p>

Tabla 13: E.C.U. Dar de baja usuario

Caso de Uso:	Modificar Datos Personales
Actores:	Usuario
<p>Descripción:</p> <p>En este caso de uso el usuario puede modificar sus datos personales, como nombre de usuario, contraseña, y dirección de correo electrónico.</p>	
<p>Propósito:</p> <p>Modificar cualquier dato que el administrador del sistema haya errado al crear al usuario.</p>	
<p>Precondición:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p> <p>2.- El usuario haya ingresado a la página modificar datos.</p>	
<p>Flujo Principal:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p>	

<p>2.- El usuario ingresa a la página de modificar datos.</p> <p>3.- El usuario actualiza sus datos.</p>
<p>Flujo de Excepción:</p> <p>1.- Nombre de usuario repetido.</p>

Tabla 14: E.C.U. Modificar datos personales

Caso de Uso:	Crear Documento
Actores:	Usuario
<p>Descripción:</p> <p>En este caso de uso el usuario tiene la opción de crear un documento que desee desde su PC al servidor (Base de Datos).</p>	
<p>Propósito:</p> <p>Subir un archivo digital desde su computadora al servidor, y agregarlo el mismo a algún usuario como revisor, aprobador, o publicador quedando en estado de creador el usuario que sube el documento al sistema.</p>	
<p>Referencia Cruzada:</p> <p>No tiene</p>	
<p>Precondición:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p> <p>2.- El usuario seleccione nuevo documento.</p>	
<p>Flujo Principal:</p> <p>1.- El usuario selecciona un tipo de documento.</p>	

<p>2.- El usuario escribe el nombre del documento.</p> <p>3.- El usuario selecciona la cadena de aprobación.</p> <p>4.- El usuario escribe las palabras claves para la búsqueda del documento.</p> <p>5.- El usuario selecciona al revisor del documento.</p> <p>6.- El usuario escoge una fecha límite de revisión.</p> <p>7.- El usuario selecciona al aprobador del documento.</p> <p>8.- El usuario escoge una fecha límite de aprobación.</p> <p>9.- El usuario selecciona al publicador del documento.</p> <p>10.- El usuario escoge una fecha límite de publicación.</p> <p>11.- El usuario sube el archivo mediante una búsqueda en su ordenador personal.</p> <p>12.- El usuario selecciona Inserta Documento.</p>
<p>Subflujo :</p> <p>1.- Mensaje: Archivo subido exitosamente.</p>
<p>Flujo de Excepción:</p> <p>1.- Datos incorrectos.</p>

Tabla 15: E.C.U Crear documento

Caso de Uso:	Revisar documento
Actores:	Usuario
<p>Descripción:</p> <p>Este caso de uso hace referencia al usuario que tiene en sus documentos pendientes a</p>	

una revisión y entra a la pantalla revisión para poder ver el documento, aprobarlo y hacerlo subir de nivel, o en su defecto rechazarlo y que vuelva al creador
<p>Precondición:</p> <ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario tenga un documento para revisar
<p>Flujo Principal:</p> <ol style="list-style-type: none"> 1.- Verifica el estado del documento, si está en revisión. 2.- Si existe se coloca en la sección revisión de la pantalla de inicio. 3.- El usuario escoge el documento para revisarlo. 4.- Entra a pantalla de revisión. 5.- Descargar documento.
<p>Flujo de Excepción:</p> <p>No tiene.</p>

Tabla 16: E.C.U. Revisar documento

Caso de Uso:	Aprobar documento
Actores:	Usuario
<p>Descripción:</p> <p>Este caso de uso hace referencia al usuario que tiene en sus documentos pendientes una aprobación y entra a la pantalla aprobación para poder ver el documento, aprobarlo y hacerlo subir de nivel, o en su defecto rechazarlo y que vuelva al creador</p>	

<p>Precondición:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p>
<p>Flujo Principal:</p> <p>1.- Verifica el estado del documento, si está en aprobación.</p> <p>2.- Si existe se coloca en la sección aprobación de la pantalla de inicio.</p> <p>3.- El usuario escoge el documento para aprobarlo.</p> <p>4.- Entra a pantalla de revisión.</p> <p>5.- Descargar documento.</p>
<p>Flujo de Excepción:</p> <p>No tiene.</p>

Tabla 17: E.C.U. Aprobar documento

Caso de Uso:	Publicar Documento
Actores:	Usuario
<p>Descripción:</p> <p>Este caso de uso hace referencia al usuario ya ha escogido el documento a publicar, el usuario tiene que descargar el documento, y publicar el documento, para que este documento pueda ser público, o en su defecto rechazarlo y que vuelva al creador.</p>	
<p>Precondición:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p> <p>2.- El usuario tenga un documento para publicar.</p>	
<p>Flujo Principal:</p>	

<p>1.- El usuario escoge un documento para publicar.</p> <p>2.- Entra a la pantalla de publicación.</p> <p>3.- El usuario descarga el documento.</p> <p>4.- Aprobar el documento.</p>
<p>Flujo de Excepción:</p> <p>No tiene.</p>

Tabla 18: E.C.U. Publicar documento

Caso de Uso:	Ver Documentos Publicados
Actores:	Usuario
<p>Descripción:</p> <p>Este caso de uso hace referencia al usuario que necesita usar un documento ya publicado, y entra a la pantalla donde están todos los documentos publicados.</p>	
<p>Precondición:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p>	
<p>Flujo Principal:</p> <p>1.- El usuario escoge un documento publicado.</p> <p>2.- Entra a la pantalla de ver documento.</p> <p>3.- El usuario descarga el documento.</p>	

<p>Flujo de Excepción:</p> <p>No tiene.</p>
--

Tabla 19: E.C.U. Ver documentos publicados

Caso de Uso:	Asignar Documento
Actores:	Usuario
<p>Descripción:</p> <p>Este caso de uso hace referencia al usuario que quiere crear un flujo de aprobación en el sistema donde otro usuario sea el creador del documento.</p>	
<p>Precondición:</p> <p>1.- El usuario haya iniciado sesión en el sistema.</p>	
<p>Flujo Principal:</p> <p>1.- El usuario entra a la pantalla de asignación de documentos.</p> <p>2.- Introduce los datos del documento, y el usuario asignado a realizar.</p> <p>3.- Asigna el documento.</p>	
<p>Flujo de Excepción:</p> <p>No tiene.</p>	

Tabla 20: E.C.U. Asignar documento

Caso de Uso:	Búsqueda Simple
Actores:	Usuario
Descripción:	
En este caso de uso el usuario realiza una búsqueda simple, solamente por nombre del documento y por palabras claves del documento.	
Propósito:	
Buscar un archivo digital de que se encuentre la base de datos y haya sido publicado.	
Precondición:	
<ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El documento a buscar sea un documento público. 	
Flujo Principal:	
<ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario selecciona opción buscar. 	
Flujo de Excepción:	
No tiene	

Tabla 21: E.C.U. Búsqueda simple

Caso de Uso:	Búsqueda Avanzada
Actores:	Usuario
Descripción:	
En este caso de uso el usuario ha escogido la búsqueda avanzada donde tendrá mas parámetros para realizar una búsqueda mas especifica.	
Propósito:	
Buscar cualquier archivo que no puedan ser encontrados con una búsqueda simple.	
Precondición:	
<ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario haya ingresado a la página de búsqueda avanzada. 3.- El documento a buscar sea un documento público. 	
Flujo Principal:	
<ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario ingresa a la página de búsqueda avanzada. 3.- El usuario selecciona opción buscar. 	
Flujo de Excepción:	
No tiene	

Tabla 22: E.C.U. Búsqueda avanzada

Caso de Uso:	Crear nuevo tipo de Documento
Actores:	Administrador
Descripción:	
Este caso de uso hace referencia a la facultad del administrador del sistema de adicionar un nuevo tipo de documento al sistema a requerimiento del personal	
Precondición:	
<ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario tiene que ser administrador del sistema. 	
Flujo Principal:	
<ol style="list-style-type: none"> 1.- El usuario ingresa al sistema. 2.- Entra a la pantalla administrador. 3.- Entra a la pantalla de nuevo tipo de Documento. 4.- Adiciona un Nuevo tipo de Documento. 	
Flujo de Excepción:	
No existe	

Tabla 23: E.C.U. Crear nuevo tipo de documento

Caso de Uso:	Sacar Respaldo a la Base de Datos
Actores:	Administrador
Descripción:	
En este caso de uso el Administrador del sistema, saca un Backup a la base de datos del sistema SigD.	
Propósito:	
Tener un respaldo de la base de datos del sistema, para cualquier eventualidad.	
Precondición:	
<ol style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario sea un administrador del sistema. 3.- El usuario haya ingresado a la página de administración del sistema. 	
Flujo Principal:	
<ol style="list-style-type: none"> 1.- El usuario inicia sesión en el sistema. 2.- El usuario ingresa a la página de administración del sistema. 3.- El usuario saca un respaldo de la base de datos. 4.- El usuario descarga el respaldo. 	
Flujo de Excepción:	
No tiene	

Tabla 24: E.C.U. Sacar respaldo a la base de datos

Caso de Uso:	Restaurar Base de Datos
Actores:	Administrador
Descripción:	
En este caso de uso el administrador sube un archivo de respaldo al sistema y realiza la sustitución de la base de datos.	
Propósito:	
Restaurar el sistema desde un respaldo.	
Precondición:	
<ul style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario sea un administrador del sistema. 2.- El usuario haya ingresado a la página de administración del sistema 	
Flujo Principal:	
<ul style="list-style-type: none"> 1.- El usuario haya iniciado sesión en el sistema. 2.- El usuario ingresa a la página de administración del sistema. 3.- El usuario selecciona el archivo de respaldo. 4.- El usuario respalda el sistema. 	
Flujo de Excepción:	
Subir un archivo que no es de respaldo.	

Tabla 25: E.C.U. Restaurar base de datos

I.1.1.17. Especificaciones adicionales

I.1.1.17.1. Requisitos funcionales

A continuación se muestra gráficamente los requisitos funcionales del sistema, agrupados de acuerdo a los módulos descritos en el plan de desarrollo del software.

Módulo Administración de usuarios

- Adicionar Usuarios
- Modificar Usuarios
- Dar de Baja a Usuarios.
- Sacar Respaldo de la Base de Datos del Sistema.
- Restaurar la Base de Datos.
- Adicionar Tipo de Documento.

Módulo Administración de documentos

- Crear Documento.
- Revisar Documento.
- Aprobar Documento.
- Publicar Documento.
- Ver Documentos Publicados.
- Subir Nueva Versión del Documento.
- Eliminar Documento.
- Eliminar Versión.
- Modificar Datos Personales.
- Buscar Documentos.

I.1.1.17.2. Requisitos no funcionales

A continuación se muestran los requisitos no funcionales del sistema, agrupados de acuerdo a características de Usabilidad, Portabilidad y Documentación. No se definieron requisitos para otros aspectos no funcionales.

Usabilidad

- Visual Studio 2008. SP1. Para el desarrollo del sistema con el lenguaje C#.
- Sql Server 2005. Motor de Base de Datos.
- Windows Server 2003. SP2. Plataforma donde se alojara el sistema SigD.
- ISS 6. Internet Information Services. Servidor Web de Microsoft con soporte para paginas “.aspx” y alojamiento del sistema SigD.
- AjaxToolKit.
- VMWare 6.0. Para ejecutar Windows server 2003, durante el desarrollo del sistema.
- Visual Paradigm 4.0. Para realizar diagramas UML.
- Crystal Reports 9. Para realizar los reportes del sistema.
- Una red local a través de la cual funciona el sistema.

Soporte de sistemas operativos

El sistema deberá poder instalarse sobre Windows Server 2003 y ejecutarse sobre operativos la tecnología asp.net y el gestor de base de datos SQL SERVER 2005.

Documentación

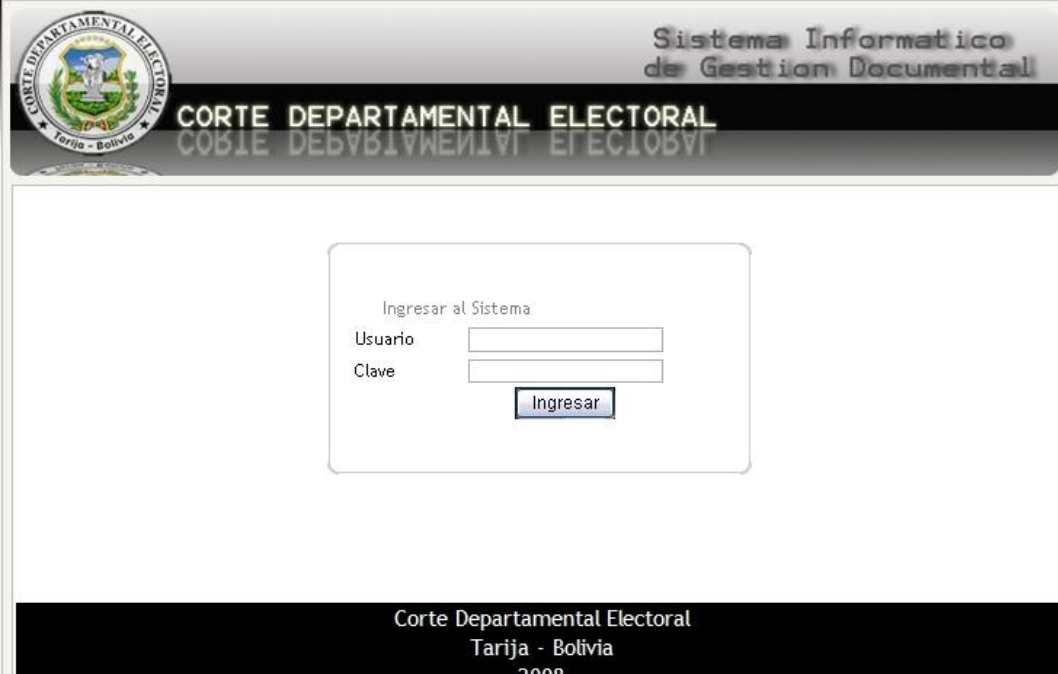
- Manual de Usuario.
- Manual de Instalación.

I.1.1.18. Interfaces de usuario.

El sistema será un sistema web que se ejecutara en el navegador, dentro de la intranet de la Corte Departamental Electoral de Tarija

Iniciar Sesión.

Esta ventana es la que aparece cuando el usuario (administrador o normal) quiere ingresar al sistema.



The screenshot shows a web interface for the 'Sistema Informático de Gestión Documental' of the 'Corte Departamental Electoral Tarija - Bolivia'. The interface features a header with the organization's logo and name, and a main content area with a login form. The form includes fields for 'Usuario' and 'Clave', and an 'Ingresar' button. The footer contains the text 'Corte Departamental Electoral Tarija - Bolivia 2008'.

Sistema Informático de Gestión Documental

CORTE DEPARTAMENTAL ELECTORAL
Tarija - Bolivia

Ingresar al Sistema

Usuario

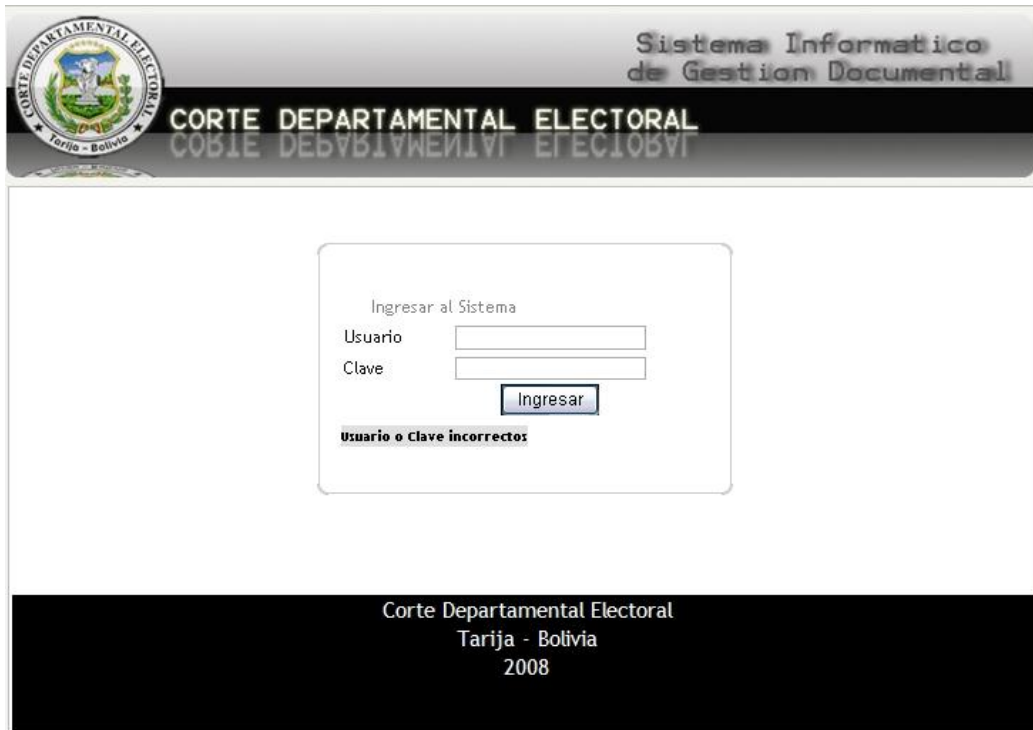
Clave

Ingresar

Corte Departamental Electoral
Tarija - Bolivia
2008

Figura 9: Interface Iniciar sesión

Esta pantalla aparecerá cuando el usuario equivoque, el usuario o la clave



The screenshot displays the login interface for the 'Sistema Informatico de Gestion Documental' of the 'CORTE DEPARTAMENTAL ELECTORAL Tarija - Bolivia'. The interface includes a header with the organization's logo and name, and a footer with contact information. The main content area features a login form with the title 'Ingresar al Sistema', input fields for 'Usuario' and 'Clave', and an 'Ingresar' button. Below the button, a red error message reads 'Usuario o clave incorrectos'.

Sistema Informatico de Gestion Documental

CORTE DEPARTAMENTAL ELECTORAL

Tarija - Bolivia

Ingresar al Sistema

Usuario

Clave

Usuario o clave incorrectos

Corte Departamental Electoral
Tarija - Bolivia
2008

Figura 10: Interface Mensaje de error

Pantalla entrada.

Esta pantalla aparecerá una vez que el usuario ha ingresado correctamente, donde tendrá le aparecerán los documentos que tiene asignado el usuario en los distintos.

Sistema Informatico de Gestion Documental

CORTE DEPARTAMENTAL ELECTORAL
Tarija - Bolivia

Bienvenido Señor(a): Adrian Mansilla Salas [Busqueda Avanzada](#)

- Inicio
- Nuevo Documento ▶
- Mis Documentos
- Publicados
- Mis Datos

Documentos con Prioridad Alta

No existen Documentos Con prioridad Alta para mostrar

Documentos Normales

	Nombre del Documento	Fecha de Creacion	Fecha Limite	Estado	Accion
Seleccionar	Carta a presidencia III	23/01/2010 04:47:46 p.m.	25/01/2010 12:00:00 a.m.	Pendiente	Aprobar

Documentos por Crear, Asignados por terceros

No existen Documentos Asignados por Terceros para mostrar

Documentos Rechazados

No existen Documentos Rechazados para mostrar

Figura 11: Interface Pantalla de entrada

En la pantalla siguiente es la que le sale a un usuario Administrador, la diferencia con la siguiente pantalla es que en esta no aparece el botón administrador.

Sistema Informatico de Gestion Documental

CORTE DEPARTAMENTAL ELECTORAL
Tarija - Bolivia

Bienvenido Señor(a): Aldo Jorge Coro Sellis Busqueda Avanzada Buscar

Salir del Sistema

- Inicio
- Nuevo Documento
- Mis Documentos
- Publicados
- Mis Datos

Documentos con Prioridad Alta

	Nombre del Documento	Fecha de Creacion	Fecha limite	Estado	Accion
Seleccionar	Informe de software padron biometrico 2010	12/09/2010 p.m.	10:28:27 12/09/2010 p.m.	10:28:27	Pendiente Revisar
Seleccionar	Informe de Formularios de empadronamiento 2010	17/09/2010 p.m.	06:26:54 17/09/2010 p.m.	06:26:54	Pendiente Revisar

Documentos Normales

	Nombre del Documento	Fecha de Creacion	Fecha Limite	Estado	Accion
Seleccionar	Carta a oficiales de registro civil departamento de tarija	29/09/2010 p.m.	07:54:05 30/09/2010 a.m.	12:00:00	Pendiente Aprobar

Documentos por Crear, Asignados por terceros

No existen Documentos Asignados por Terceros para mostrar

Documentos Rechazados

	Nombre del Documento	Observaciones	Estado	version	Condicion
Seleccionar	Listas de empadronamiento	mala ortografia	Revision	1	Reprovado

Figura 12: Interface Pantalla de entrada administrador

Pantalla Adicionar Documento

En la siguiente pantalla se muestra como el usuario creara un nuevo documento, asignándole a este el tipo de documento, nombre y los usuarios respectivos para la cadena de aprobación, está en especial es para una cadena de aprobación normal.

The screenshot shows a web application interface for adding a new document. The user is logged in as 'Adrian Mansilla Salas' with the role of 'Administrador'. The page title is 'Insertar un Nuevo Documento al Sistema'. The interface includes a navigation menu on the left with options like 'Inicio', 'Nuevo Documento', 'Mis Documentos', 'Publicados', and 'Mis Datos'. The main content area contains a form with the following fields:

Tipo de documento	Informes
Nombre del Documento	Introducir el nombre del Documento
Cadena de Aprobacion	Normal
Palabras claves para buscar este documento	Las palabras clave son de vital importancia para poder ubicar este documento posteriormente
Revisor	Admin admin
Fecha limite de Revision	Fecha limite de Revision
Aprobador	Mauricio Bejarano
Fecha limite de Aprobacion	Fecha limite de Aprobacion
Publicador	Aldo Jorge Coro Sellis
Fecha limite de Publicacion	Fecha limite de Publicacion
Prioridad	Alta
Cargar Documento	Examinar... Insertar Documento

Figura 13: Interface Adicionar documento aprobación normal

Cuando la cadena de aprobación es corta el sistema muestra la siguiente pantalla.

The screenshot shows a web application window titled 'Bienvenido Señor(a): Adrian Mansilla Salas'. The main heading is 'Insertar un Nuevo Documento al Sistema'. On the left, there is a navigation menu with icons and labels: 'Inicio', 'Nuevo Documento', 'Mis Documentos', 'Publicados', and 'Mis Datos'. The main content area contains several form fields: 'Tipo de documento' (dropdown menu with 'Informes' selected), 'Nombre del Documento' (text input with placeholder 'Introducir el nombre del Documento'), 'Cadena de Aprobacion' (dropdown menu with 'Corta' selected), and 'Palabras claves para buscar este documento' (text input with placeholder 'Las palabras clave son de vital importancia para poder ubicar este documento posteriormente'). Below these are 'Publicador' (dropdown menu with 'Admin admin' selected), 'Fecha limite de Publicacion' (text input), 'Prioridad' (dropdown menu with 'Alta' selected), and 'Cargar Documento' (text input). There are two buttons: 'Examinar...' and 'Insertar Documento'.

Figura 14: Interface Adicionar documento aprobación corta

Cuando se ha creado correctamente un documento el sistema muestra el siguiente mensaje de confirmación.

The screenshot shows a confirmation message box. The text reads: 'Bienvenido Sr(a): Aldo Jorge Coro Sellis' followed by 'El documento se ha creado Correctamente'. Below the text is a button labeled 'Volver al Inicio'.

Figura 15: Interface Mensaje de confirmación

Asignar Documentos

Título del Documento	<input type="text" value="Escribir el Título del Documento"/>
Usuario a Creador	<input type="text" value="Admin admin"/>
Fecha limite de presentacion	<input type="text"/>
Tipo de Documento	<input type="text" value="Informes"/>
Notas en este Documento	<input type="text" value="Escribir parametros para realizar el documento"/>
Prioridad	<input type="text" value="Alta"/>

Figura 16: Interface Asignar documentos

Ver Documento

En la siguiente pantalla el usuario podrá aprobar el documento que este en estado revisión, aprobación o publicación.

Nombre del Documento:	Contrato operadores
Fecha de Creacion:	1/23/2010 4:46:54 PM
Creador:	Mauricio Bejarano

Observaciones:

Figura 17: Interface Ver documento

Al hacer clic en el botón descargar documento el usuario podrá ver el enlace hacia su documento, como se muestra a continuación.



Figura 18: Interface Descargar documento

Al rechazar un documento el usuario tiene que confirmar esta acción y el sistema le muestra el siguiente mensaje



Figura 19: Interface Mensaje rechazar documento

Al darle clic al botón aceptar y si no se ha llenado el campo observaciones, el sistema muestra el siguiente mensaje en la pantalla.



Figura 20: Interface Mensaje datos incompletos

Cuando el documento ha sido rechazado correctamente, el sistema mostrara la siguiente pantalla

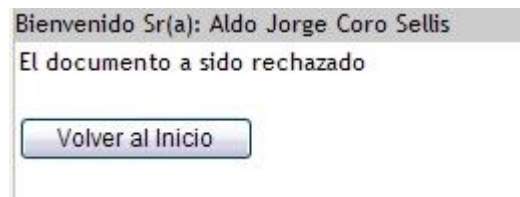


Figura 21: Interface Mensaje confirmación de rechazo

Y al aprobar el documento también necesita una confirmación del usuario, que se muestra a continuación



Figura 22: Interface Mensaje aprobar documento

El hecho de aprobar un documento, muestra un mensaje de confirmación con la pantalla siguiente.

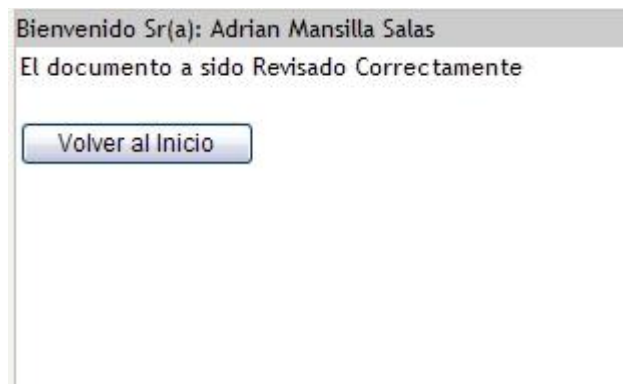


Figura 23: Interface Mensaje confirmación de aprobación

Cuando estando en la pantalla principal se hace clic en el botón *Mis documentos* el sistema desplegara la lista de documentos que el usuario a creado, donde se muestra el estado del documento y fecha de creación, como se muestra en la siguiente pantalla.

Ver Mis Documentos

The screenshot shows a web application interface for a user named Adrian Mansilla Salas. The main content area is titled 'Mis documentos' and displays a table with the following data:

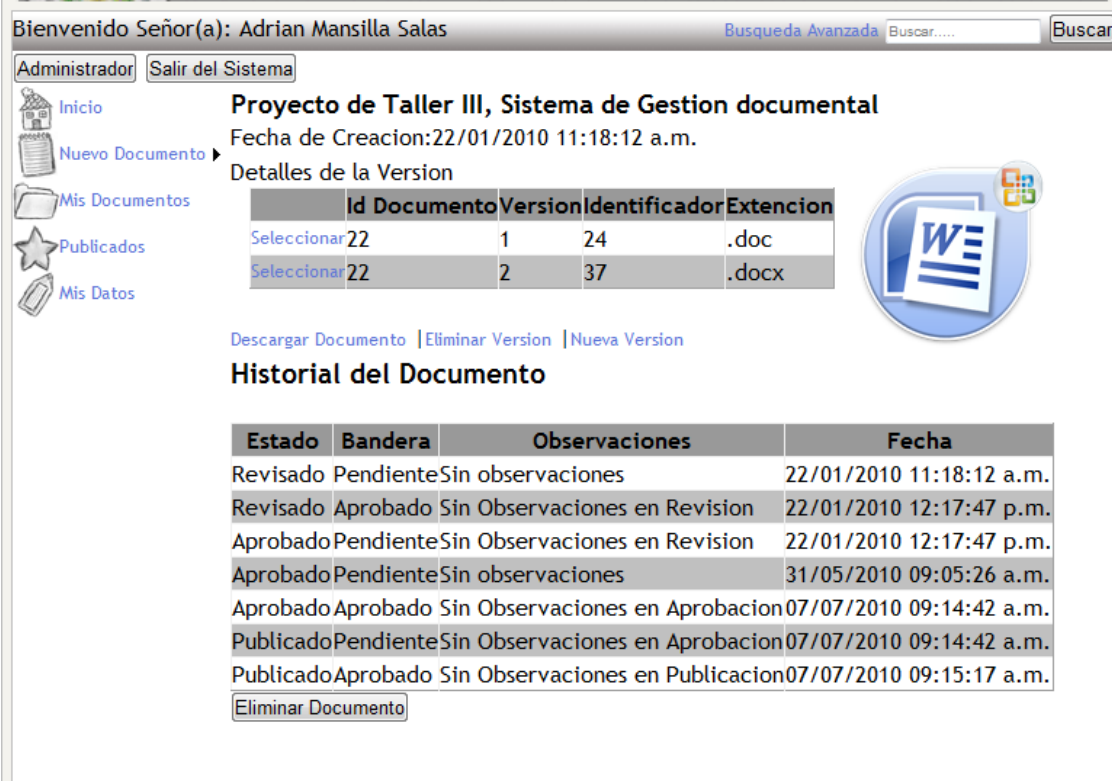
	Nombre del Documento	Observaciones	Version	Rol Usuario	Estado Documento
Seleccionar	informe de gestion 2009	Sin Observaciones en Publicacion	1	Publicar	Aprobado
Seleccionar	Proyecto de Taller III, Sistema de Gestion documental	Sin Observaciones en Publicacion	2	Publicar	Aprobado
Seleccionar	Circular II/2009	Sin observaciones	2	Revisar	Pendiente
Seleccionar	Carta a prefectura II/2010	Sin observaciones	2	Revisar	Pendiente

Below the table, the number '12' is displayed, likely representing the total number of documents.

Figura 24: Interface Ver mis documentos

Detalle del Documento

Al hacer clic en seleccionar el documento el usuario ingresa a la pantalla donde hay el detalle del estado del documento.



The screenshot shows a web application interface for document management. At the top, it displays the user's name 'Bienvenido Señor(a): Adrian Mansilla Salas' and a search bar with 'Busqueda Avanzada' and 'Buscar.....' buttons. Below this, there are navigation links for 'Administrador' and 'Salir del Sistema'. A sidebar on the left contains icons and links for 'Inicio', 'Nuevo Documento', 'Mis Documentos', 'Publicados', and 'Mis Datos'. The main content area features the document title 'Proyecto de Taller III, Sistema de Gestion documental' and its creation date 'Fecha de Creacion:22/01/2010 11:18:12 a.m.'. A section titled 'Detalles de la Version' contains a table with two rows of document versions. Below this table are links for 'Descargar Documento', 'Eliminar Version', and 'Nueva Version'. A section titled 'Historial del Documento' contains a table with six rows detailing the document's history, including states like 'Revisado', 'Aprobado', and 'Publicado', along with flags and observation notes. At the bottom of this section is a button labeled 'Eliminar Documento'.

Proyecto de Taller III, Sistema de Gestion documental
Fecha de Creacion:22/01/2010 11:18:12 a.m.

Detalles de la Version

	Id Documento	Version	Identificador	Extencion
Seleccionar	22	1	24	.doc
Seleccionar	22	2	37	.docx

[Descargar Documento](#) | [Eliminar Version](#) | [Nueva Version](#)

Historial del Documento

Estado	Bandera	Observaciones	Fecha
Revisado	Pendiente	Sin observaciones	22/01/2010 11:18:12 a.m.
Revisado	Aprobado	Sin Observaciones en Revision	22/01/2010 12:17:47 p.m.
Aprobado	Pendiente	Sin Observaciones en Revision	22/01/2010 12:17:47 p.m.
Aprobado	Pendiente	Sin observaciones	31/05/2010 09:05:26 a.m.
Aprobado	Aprobado	Sin Observaciones en Aprobacion	07/07/2010 09:14:42 a.m.
Publicado	Pendiente	Sin Observaciones en Aprobacion	07/07/2010 09:14:42 a.m.
Publicado	Aprobado	Sin Observaciones en Publicacion	07/07/2010 09:15:17 a.m.

[Eliminar Documento](#)

Figura 25: Interface Detalle del documento

Documentos Publicados

Documentos Publicados

	Nombre	Claves	Fecha	Tipo de Documento
Seleccionar	Carta a presidencia II/2010	cartas 2/2010	22/01/2010 11:24:10 a.m.	Cartas
Seleccionar	Certificado de trabajo Julio Vargas	Documento de administracion, Recursos Humanos	07/09/2010 08:44:21 a.m.	Certificados
Seleccionar	Contrato operadores	operadores	23/01/2010 04:46:54 p.m.	Contratos
Seleccionar	Cuadro Matriz Foda	foda	23/01/2010 04:49:50 p.m.	Cuadros
Seleccionar	Gestion Documental	Sistema Informatico de Gestion Documental diciembre 2010	21/09/2010 06:08:32 p.m.	Proyectos
Seleccionar	Informe de gestion 2006	2006, gestion anual empadronamiento	14/09/2010 10:13:12 p.m.	Informes
Seleccionar	informe de gestion 2009	informes de operadores durante el proceso de empadronamiento biometrico tarija 2009	25/01/2010 07:57:29 p.m.	Informes
Seleccionar	informe de gestion 2009	gestion 2009	22/01/2010 03:51:44 p.m.	Informes
Seleccionar	Proyecto de Taller III, Sistema de Gestion documental	sistema de gestion documental	22/01/2010 11:18:12 a.m.	Informes

Figura 26: Interface Documentos publicados

Pantalla Administrador

Al entrar al modulo del administrador, al usuario administrador le aparecerá la siguiente pantalla. En esta pantalla el Administrador del sistema podrá sacar un respaldo y restaurar la base de datos del sistema SigD.

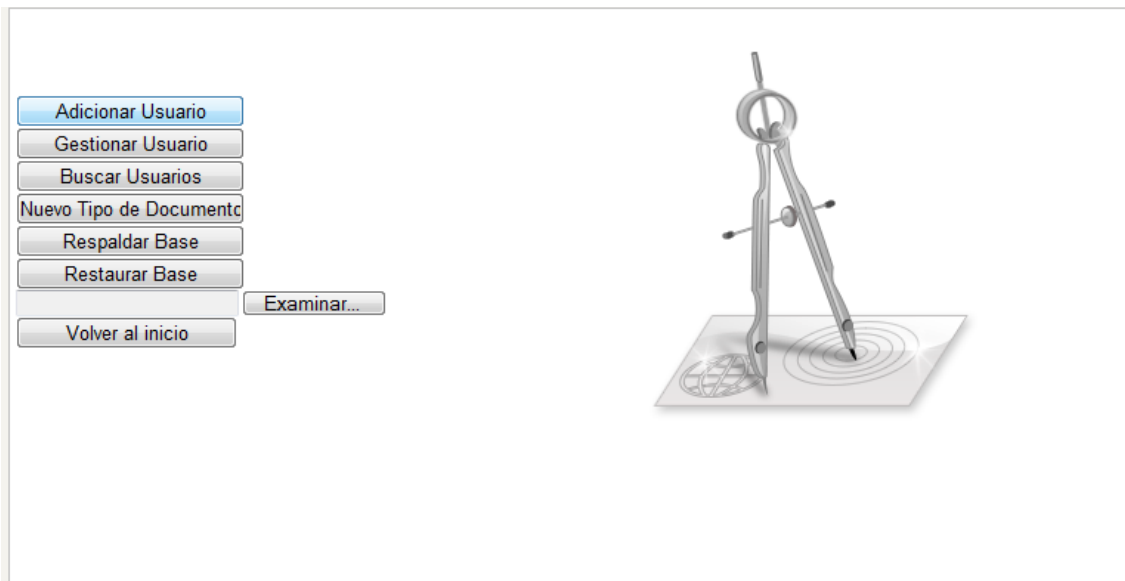


Figura 27: Interface Pantalla administrador

Pantalla Adicionar Usuario

Al ingresar a adicionar usuario el administrador tendrá que llenar los datos generales del usuario, para guardar estos, como muestra la siguiente pantalla.

Usuario: Adrian Mansilla Salas

Nombre:

Apellido Paterno:

Apellido Materno:

* Nombre de Usuario:

Rol Usuario:

Direccion de correo electronico - email:

Contraseña:

Repetir contraseña:

Dirección:

Tipo de usuario:

* Nota.- El nombre de usuario es un identificador unico para entrar al sistema

Figura 28: Interface Adicionar usuario

Al ingresar un usuario correctamente el sistema muestra la siguiente pantalla de confirmación.

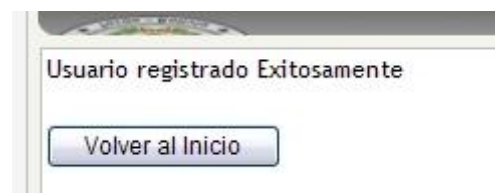


Figura 29: Interface Mensaje de confirmación datos registrados

Gestionar Usuario

En la sección de gestión de usuario, el sistema muestra la siguiente pantalla desde la cual el administrador podrá modificar datos de los usuarios o darlos de baja.

Usuario: Adrian Mansilla Salas			
	Nombre	Apellido Paterno	Apellido Materno
Editar Seleccionar	Mauricio		Bejarano
Editar Seleccionar	Admin	admin	
Editar Seleccionar	Aldo Jorge	Coro	Sellis
Editar Seleccionar	Adrian	Mansilla	Salas
Editar Seleccionar	Daniel	Salas	Mercado
Editar Seleccionar	Lucia	Salaz	Sanchez
Editar Seleccionar	Nicolas	Sanchez	Gonzales

Figura 30: Interface Gestionar usuario

Buscar Usuario

El sistema administrador también podrá hacer búsqueda de usuarios en la sección de búsquedas en esta se presentara la siguiente pantalla.

Buscar Usuario:

Nombre	apellido_paterno	apellido_materno
Adrian	Mansilla	Salas
Mauricio		Bejarano

Figura 31: Interface Buscar usuario

Nuevo Tipo de Documento

En esta pantalla el administrador del sistema podrá aumentar tipos de documentos, que requieran los usuarios del sistema SigD.

Agregar Nuevo tipo de documento

Nombre	
Actas	<input type="text"/> <input type="button" value="Insertar Nuevo Documento"/>
Cartas	
Certificados	
Circulares	
Contratos	
Cuadros	
Informes	
Instructivos	
Licitaciones	
Listas	
1	
2	

Figura 32: Interface Nuevo tipo de documento

Pantalla Imprimir Index

Cuando el usuario aprueba el documento para su publicación el sistema da la opción al usuario de imprimir la pagina index o caratula del documento como se muestra a continuación:



Informe de avance en empadronamiento biometrico 2010



Tipo de Documento: Informes
Version: 1
Fecha de Creacion: 25/01/2010 08:35:34p.m.

Adrian Mansilla Salas Revisor

Lucia Salaz Sanchez Aprobador

Aldo Jorge Coro Sellis Publicador

Mauricio Bejarano Creador

Figura 33: Interface Imprimir index

Modificar Datos Personales

En esta pantalla el usuario puede corregir cualquier error que haya habido durante el proceso de registro del nuevo usuario al sistema, o para que el usuario pueda modificar su nombre de usuario o contraseña asignada.

Nombre	<input type="text" value="Adrian"/>
Apellido Paterno	<input type="text" value="Mansilla"/>
Apellido Materno	<input type="text" value="Salas"/>
Nombre de Usuario	<input type="text" value="adrian.mansilla"/>
Correo electronico- Email	<input type="text" value="mansilladrian@gmail.com"/>
Contraseña Anterior	<input type="password"/>
Contraseña Nueva	<input type="password"/>
Repetir Contraseña	<input type="password"/>
Direccion	<input type="text" value="Administracion"/> ▼
	<input type="button" value="Guardar Cambios"/>

Figura 34: Interface Modificar datos personales

Búsqueda Simple

La búsqueda simple se realiza desde la barra de estado, esta búsqueda muestra resultados que coincidan en el nombre del documento publicado, o en las claves asignadas al documento.

Resultado de la Búsqueda

	Nombre	Claves	Fecha	Tipo Documento
Seleccionar	Gestion Documental	Sistema Informatico de Gestion Documental diciembre 2010	21/09/2010 06:08:32 p.m.	Proyectos
Seleccionar	Informe de gestion 2006	2006, gestion anual empadronamiento	14/09/2010 10:13:12 p.m.	Informes
Seleccionar	informe de gestion 2009	gestion 2009	22/01/2010 03:51:44 p.m.	Informes
Seleccionar	informe de gestion 2009	informes de operadores durante el proceso de empadronamiento biometrico tarija 2009	25/01/2010 07:57:29 p.m.	Informes
Seleccionar	Proyecto de Taller III, Sistema de Gestion documental	sistema de gestion documental	22/01/2010 11:18:12 a.m.	Informes

Figura 35: Interface Búsqueda simple

Búsqueda Avanzada

La búsqueda avanzada sirve para definir parámetros más específicos en la búsqueda, donde el usuario pueda encontrar exactamente el documento que pueda necesitar el usuario final.



The screenshot shows a web browser window with the title "Bienvenido Señor(a): Adrian Mansilla Salas". The page content includes a search bar with the text "Busqueda Avanzada" and a "Buscar...." button. Below this, there are several search criteria: "Con todas las palabras" with a text input field containing "sistema gestion"; "Fecha de publicacion entre" with two date input fields containing "01/01/2010" and "21/10/2010"; "Sin las palabras" with an empty text input field; and "Tipo de documento" with a dropdown menu set to "Todos". A "Buscar" button is located at the bottom left of the search area.

Figura 36: Interface Búsqueda avanzada

I.1.1.19. Modelo de análisis y diseño

I.1.1.19.1. Diagramas de actividad

Actividades: Iniciar sesión

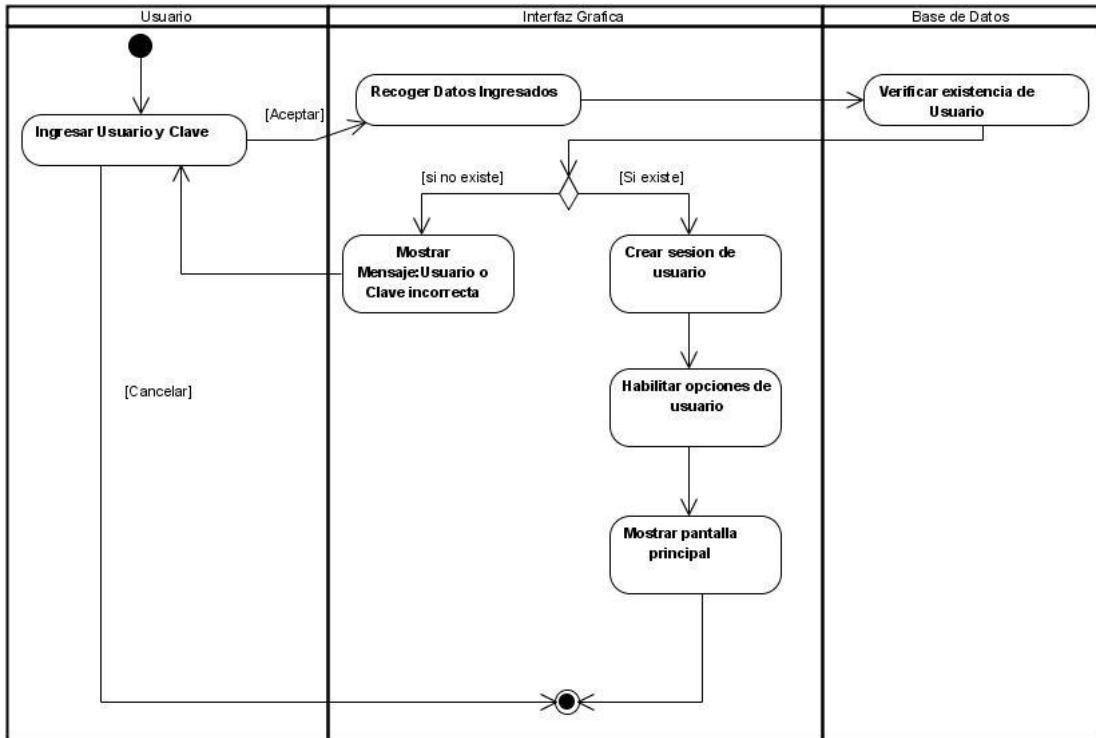


Figura 37: Diagrama de Actividades Iniciar sesión

Actividades: Dar Baja a Usuarios

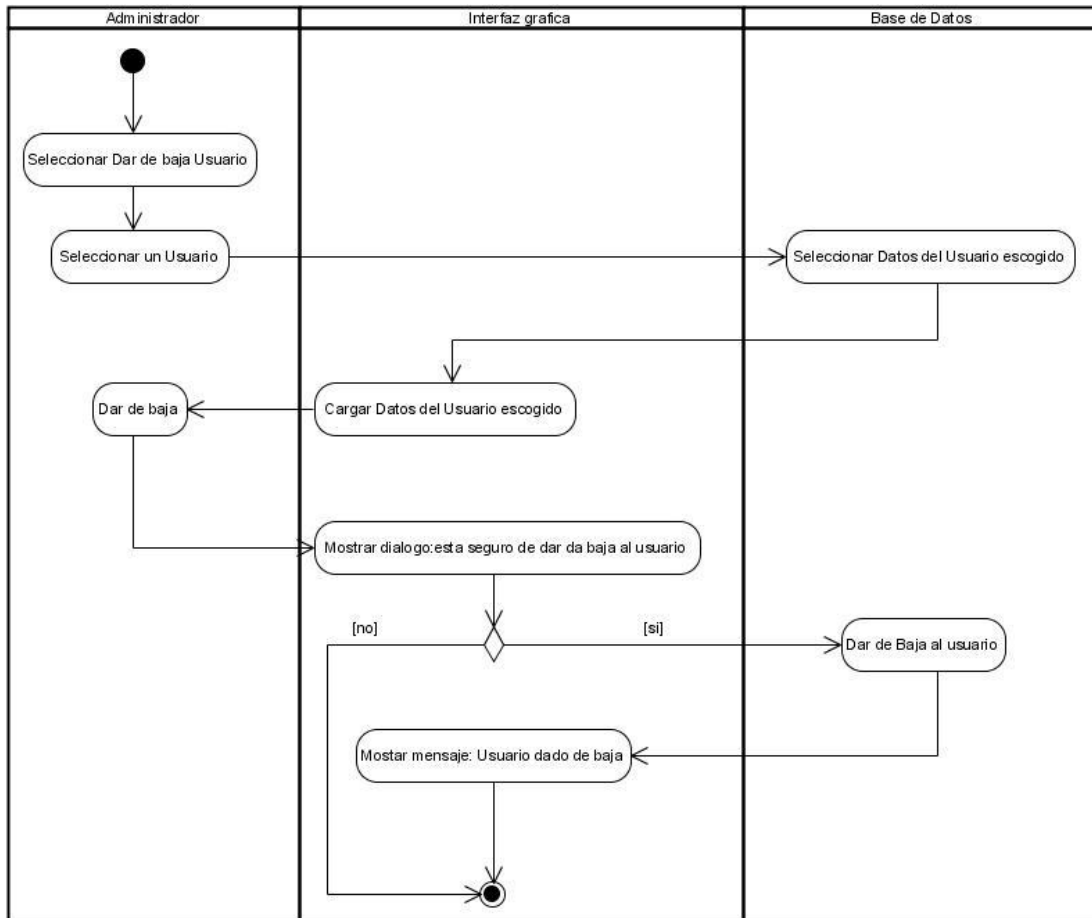


Figura 38: Diagrama de Actividades Dar baja a usuarios

Adicionar Usuarios

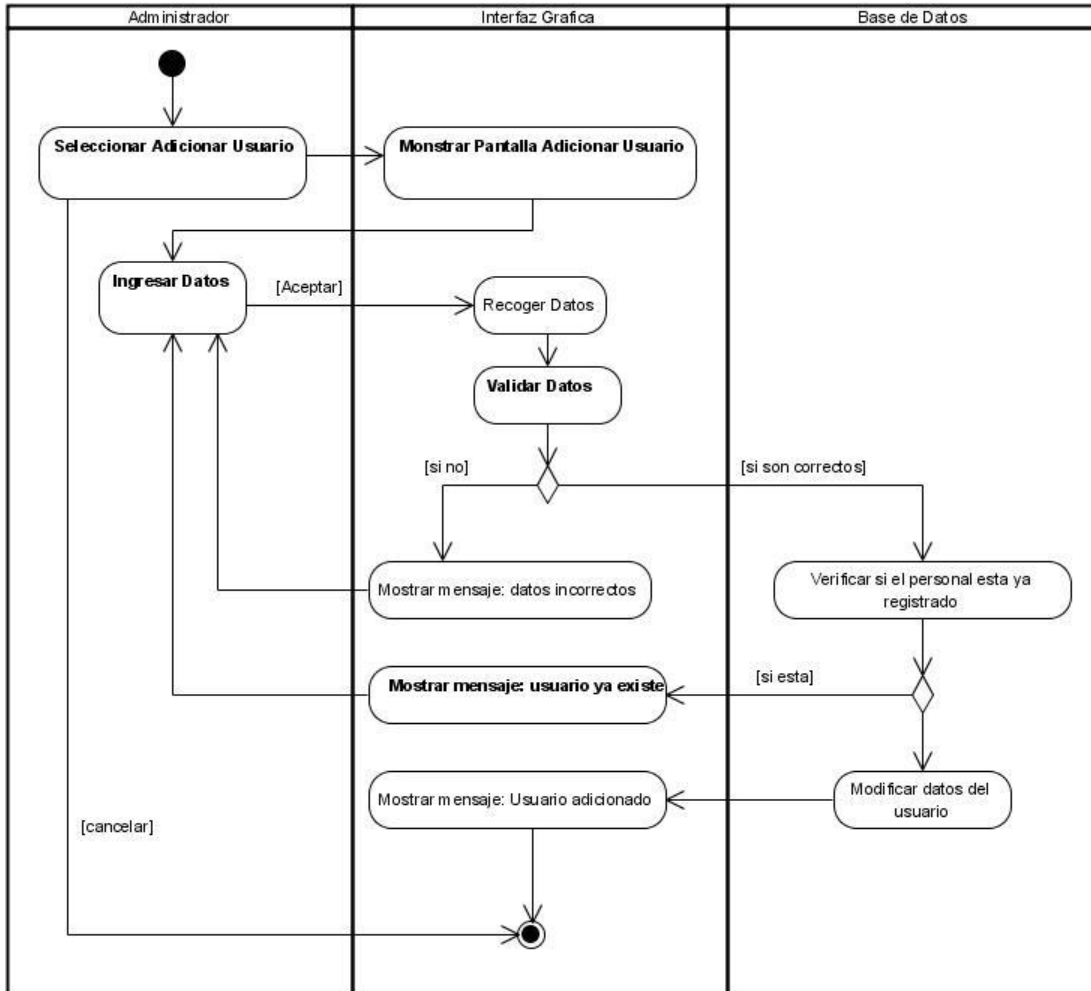


Figura 39: Diagrama de Actividades Adicionar usuarios

Actividades: Modificar Usuarios

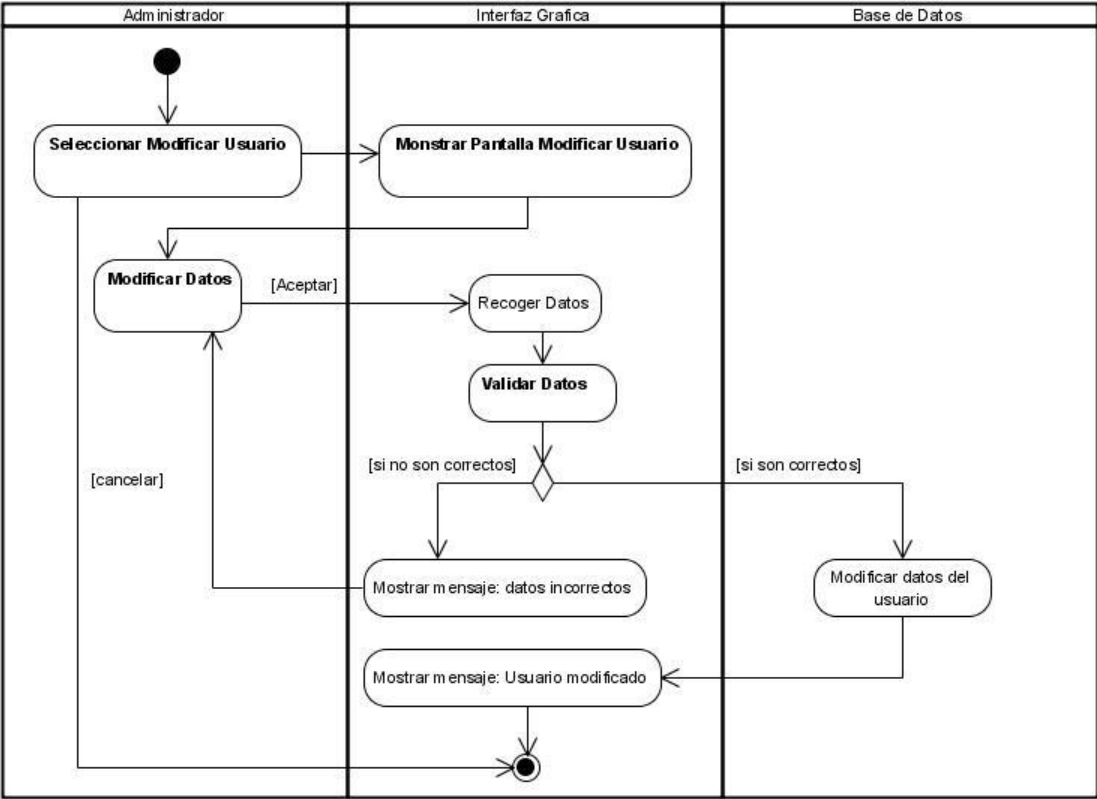


Figura 40: Diagrama de Actividades Modificar usuarios

Actividades: Crear Documentos

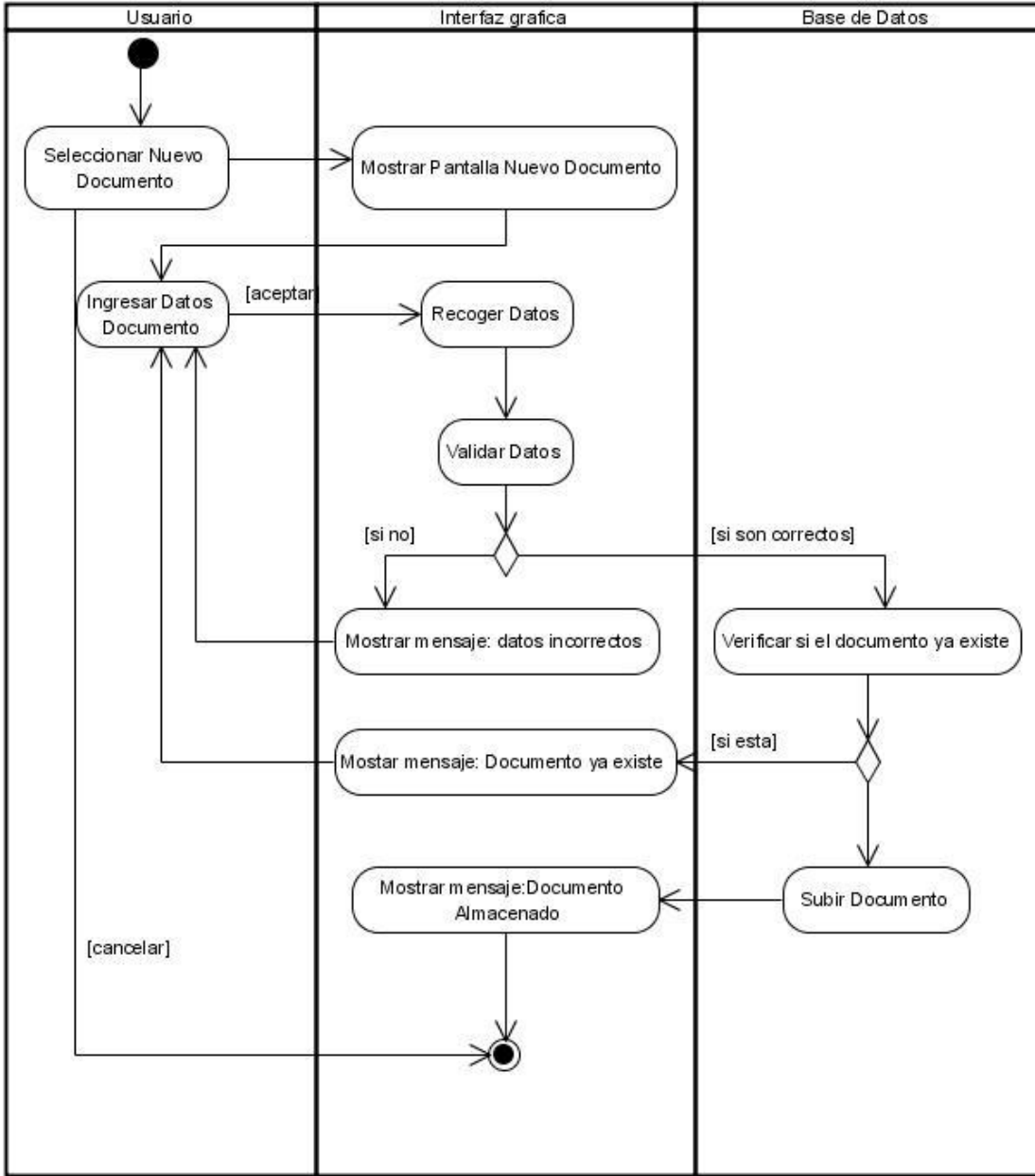


Figura 41: Diagrama de Actividades Crear documentos

Actividades: Ver Documento Publicado

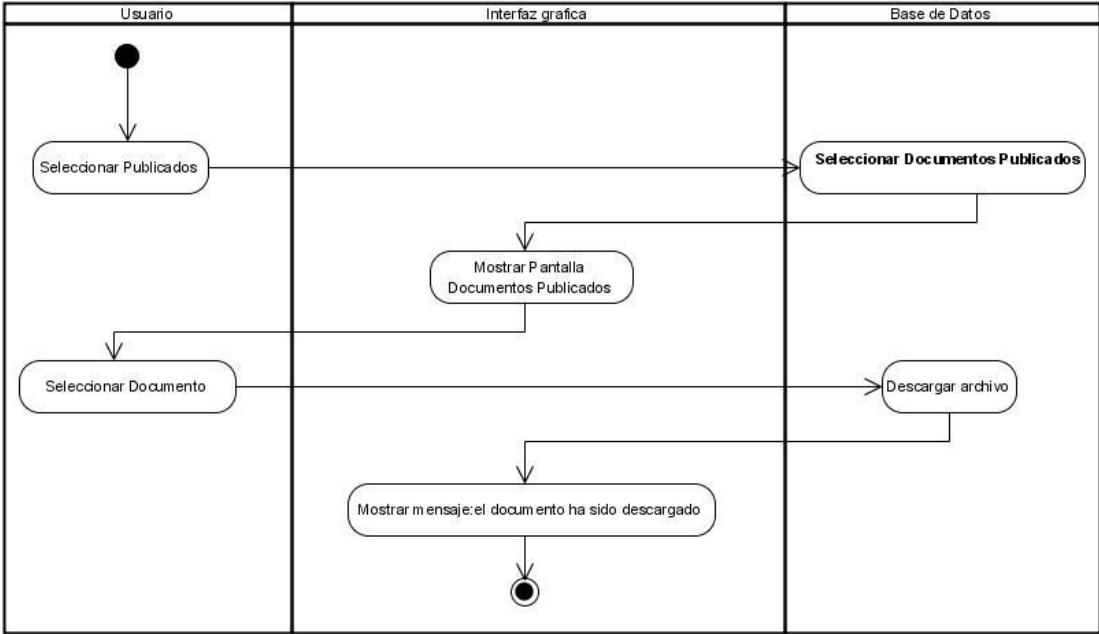


Figura 42: Diagrama de Actividades Ver documento publicado

Actividades: Revisar Documentos

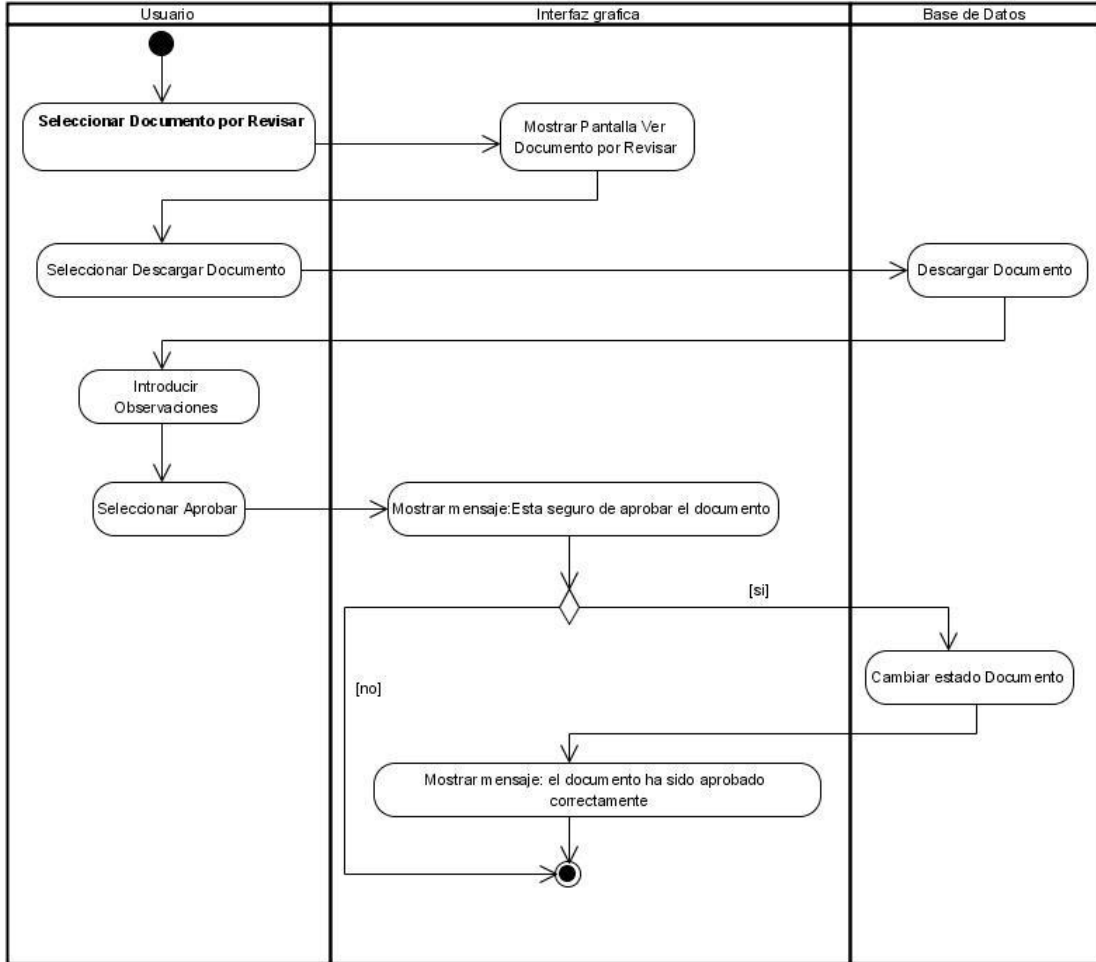


Figura 43: Diagrama de Actividades Revisar documentos

Actividades: Aprobar Documentos

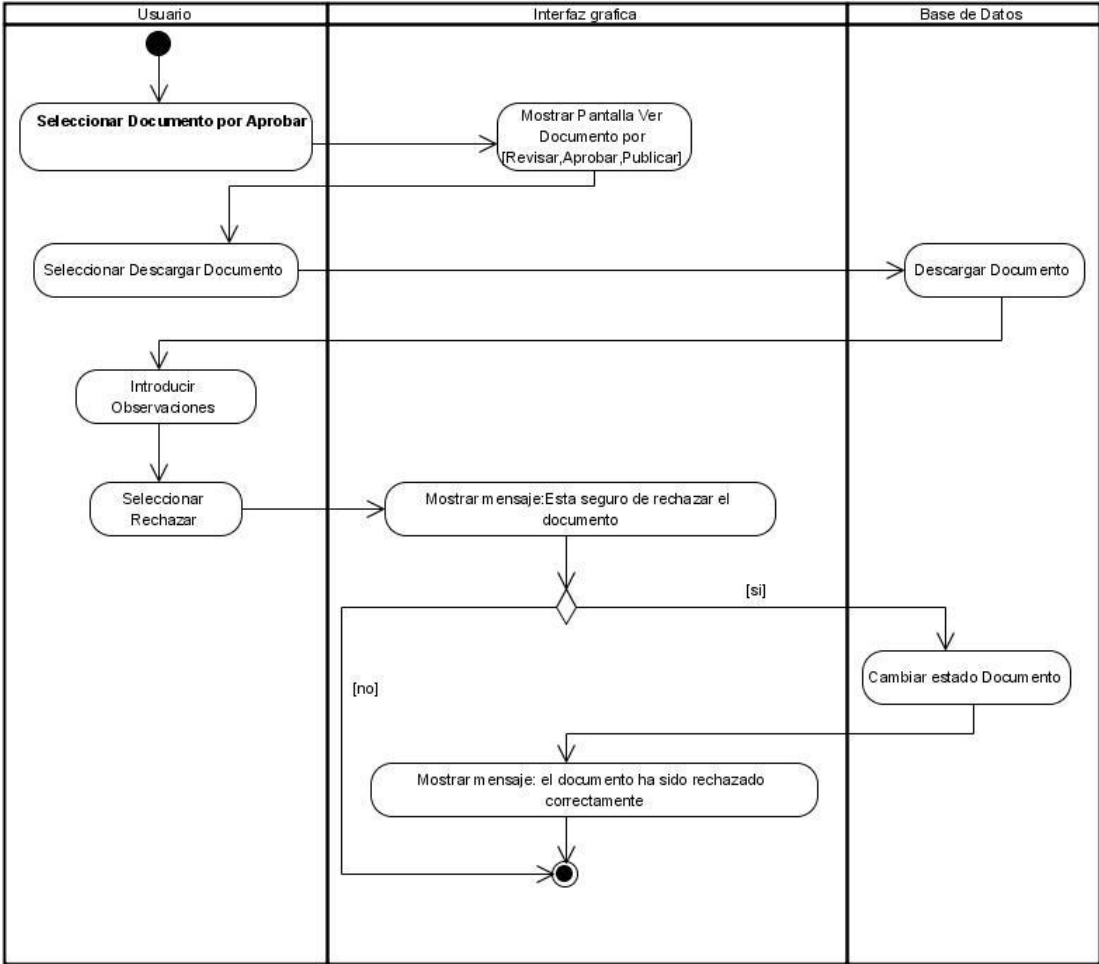


Figura 44: Diagrama de Actividades Aprobar documentos

Actividades: Publicar Documentos

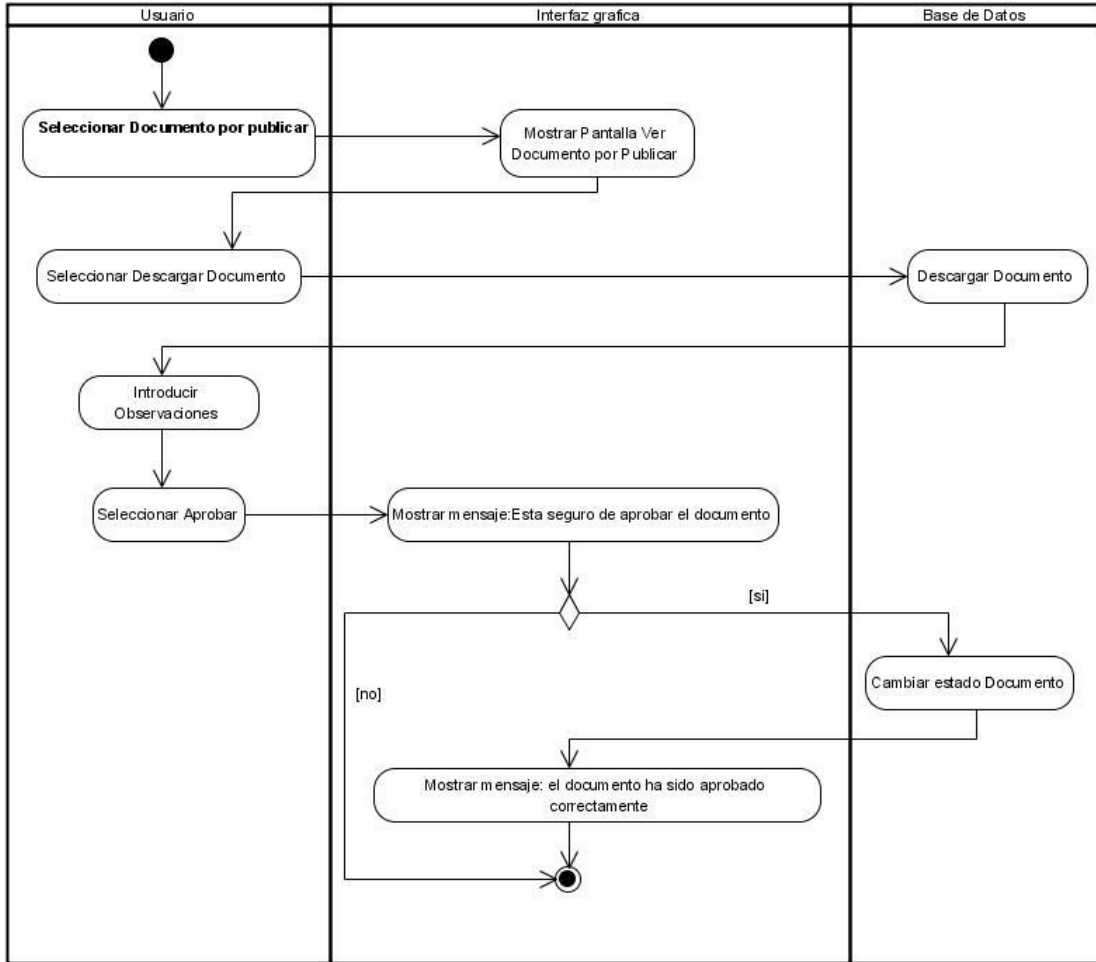


Figura 45: Diagrama de Actividades Publicar documentos

Actividades: Asignar Documentos

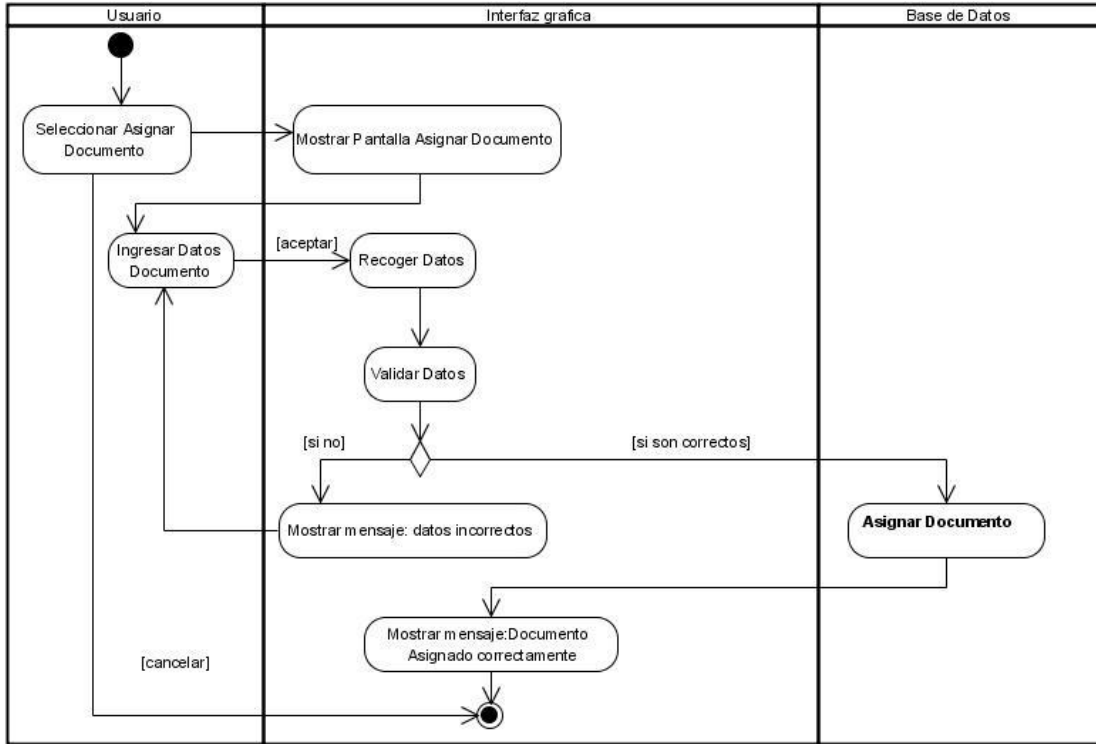


Figura 46: Diagrama de Actividades Asignar documentos

Actividades: Búsqueda Simple

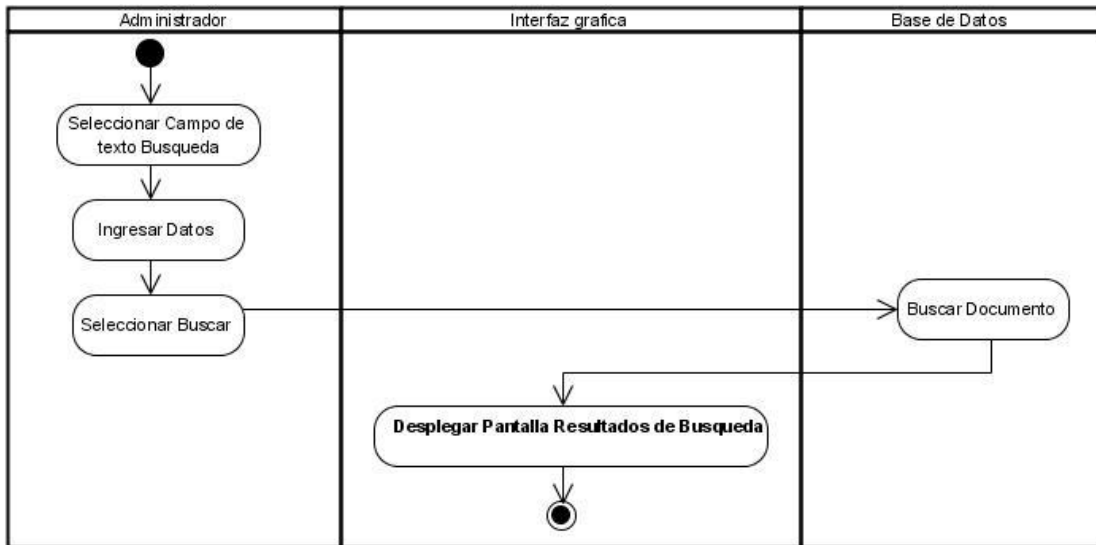


Figura 47: Diagrama de Actividades Búsqueda simple

Actividades: Búsqueda Avanzada

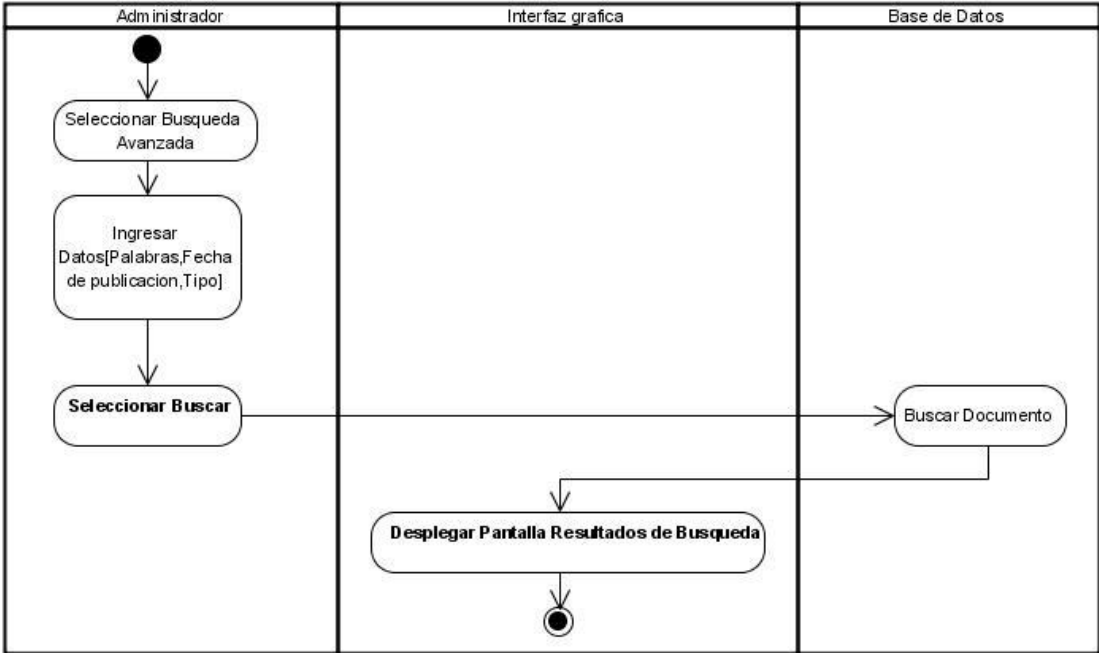


Figura 48: Diagrama de Actividades Búsqueda avanzada

Actividades: Nuevo Tipo de Documento

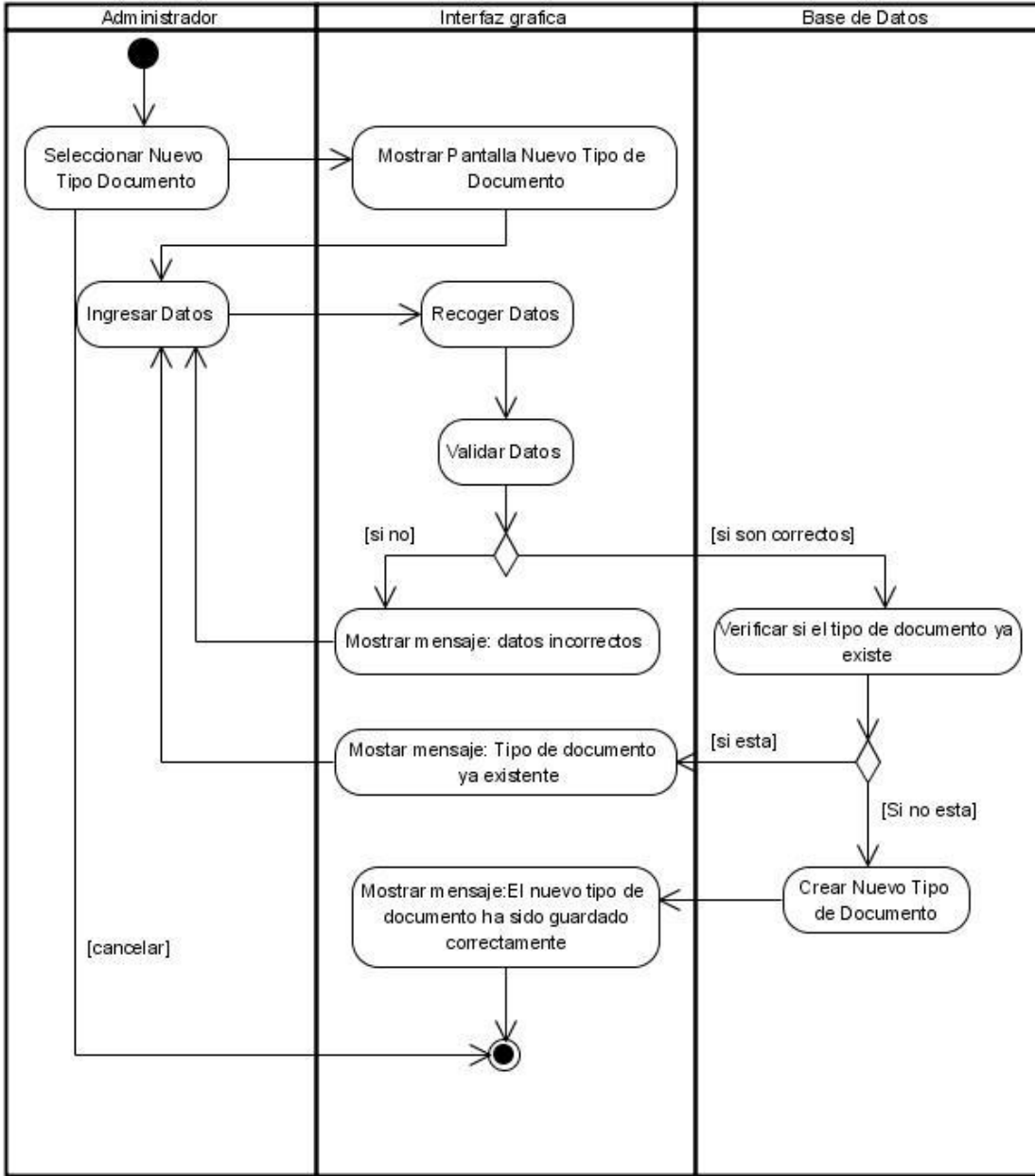


Figura 49: Diagrama de Actividades Nuevo tipo de documento

Actividades: Sacar Respaldo a la Base de Datos

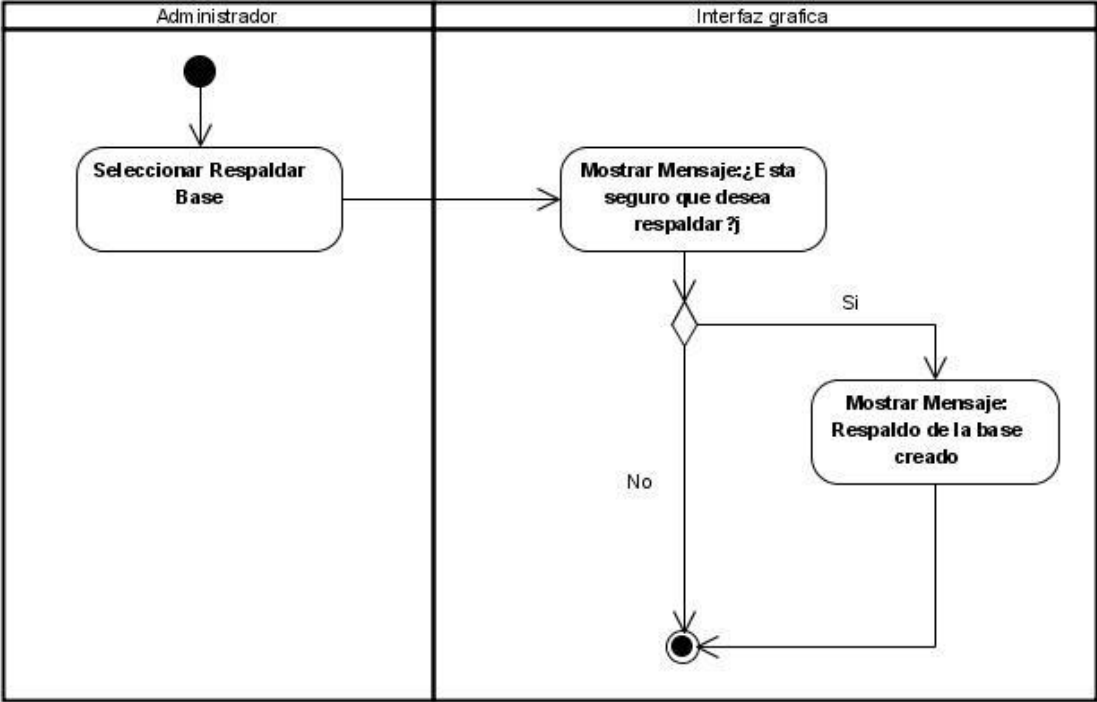


Figura 50: Diagrama de Actividades Sacar respaldo a la base de datos

Actividades: Restaurar Base de Datos

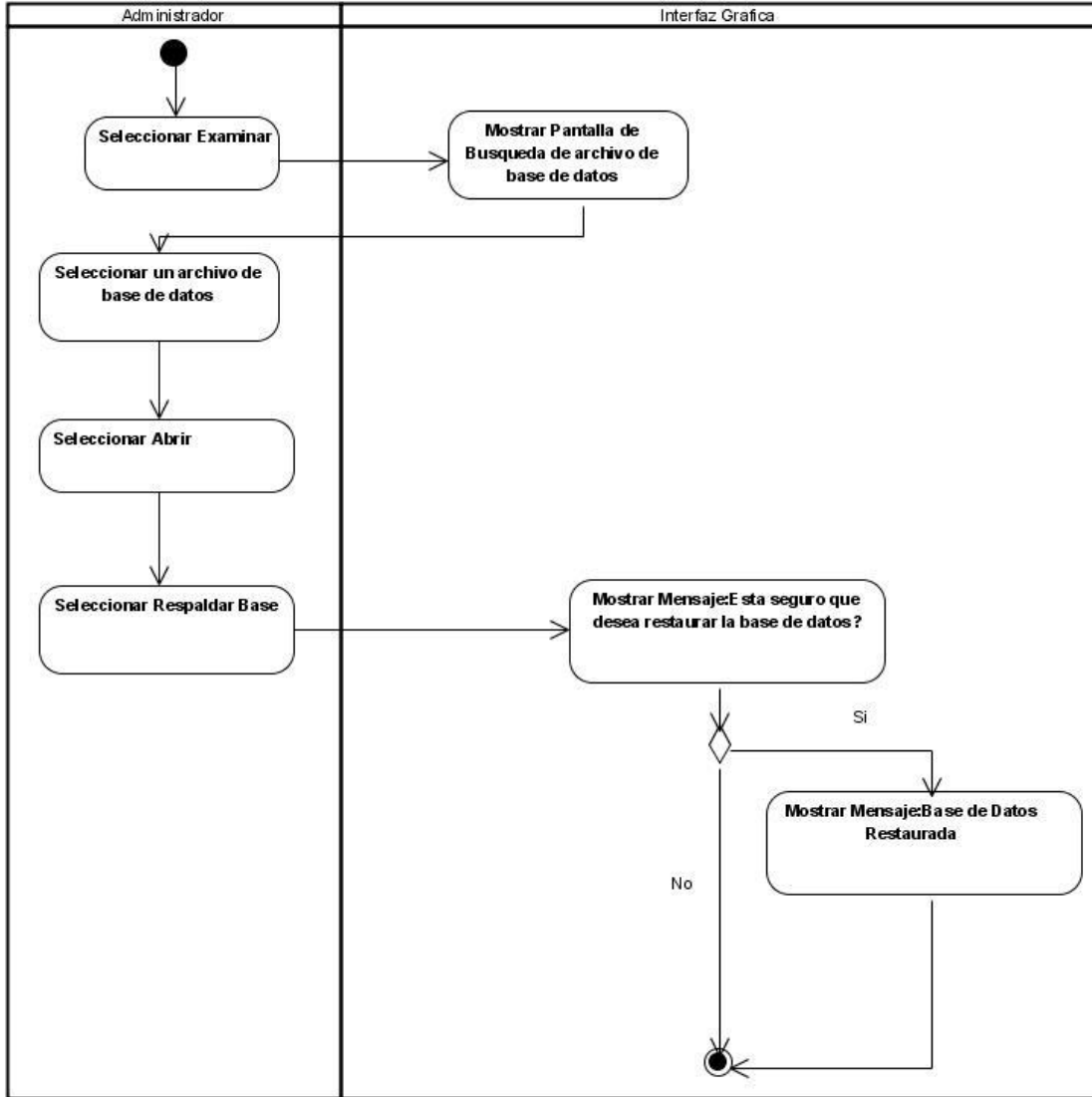


Figura 51: Diagrama de Actividades Restaurar base de datos

Actividades: Modificar Datos Personales.

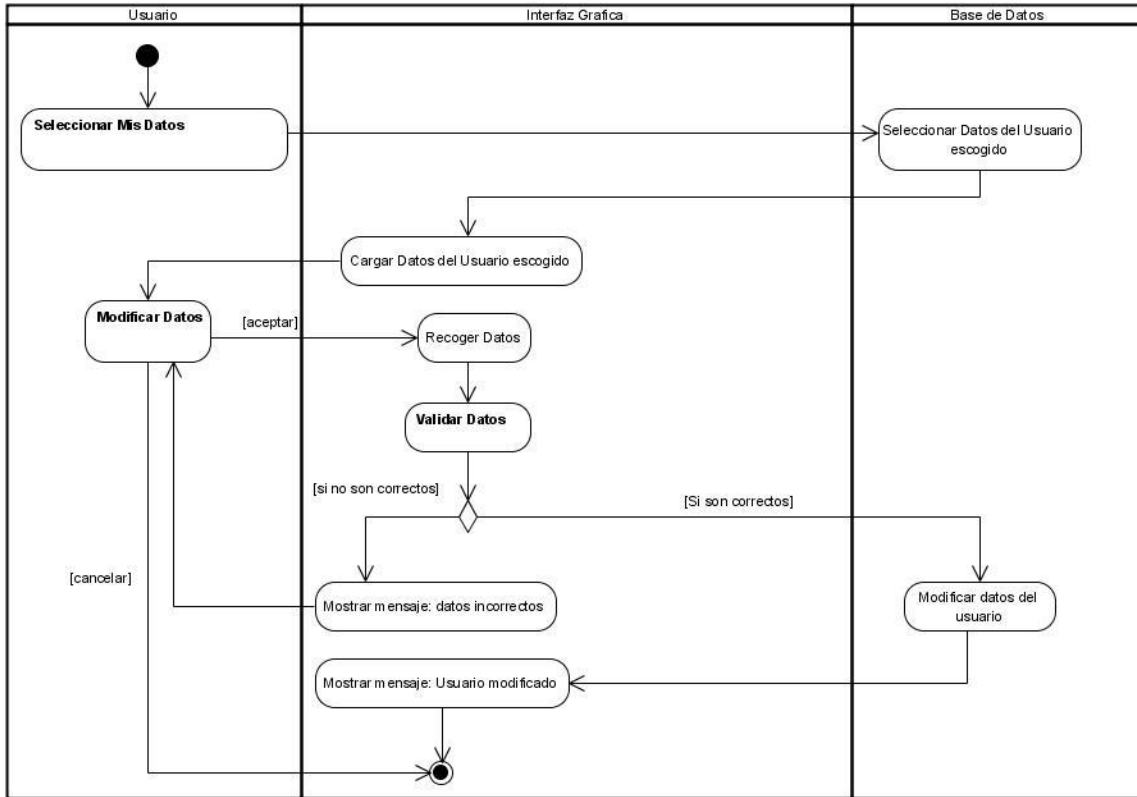


Figura 52: Diagrama de Actividades Modificar Datos Personales.

I.1.1.19.2. Diagramas de secuencia.

Secuencia: Iniciar Sesión

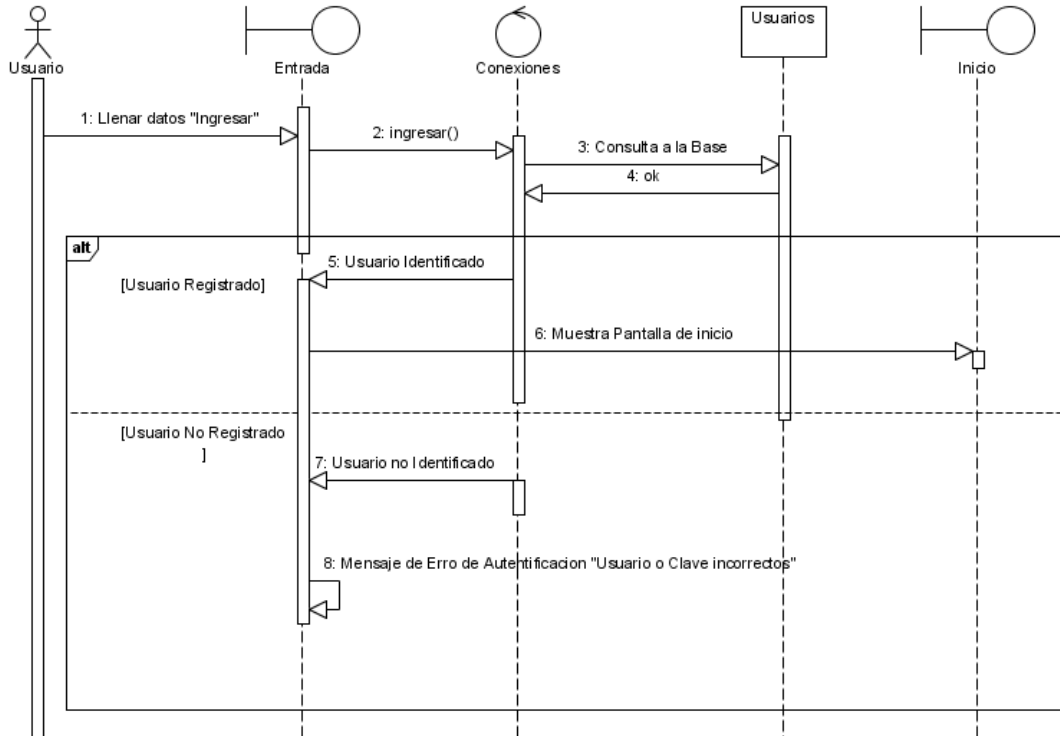


Figura 53: Diagrama de Secuencia Iniciar sesión

Secuencia: Dar Baja a Usuarios

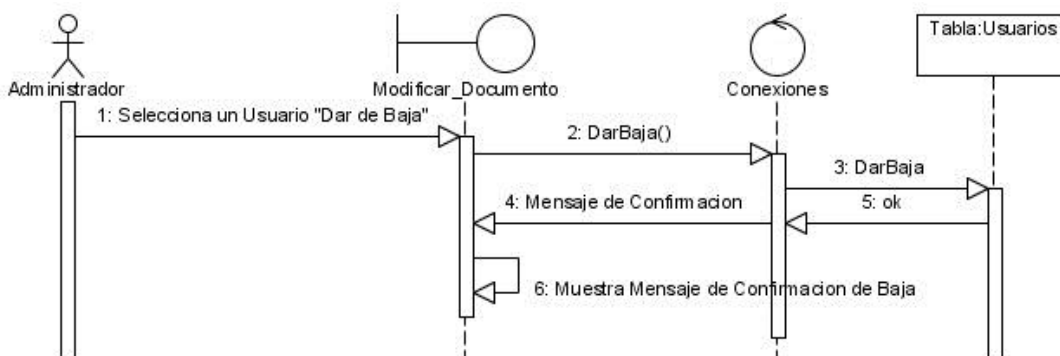


Figura 54: Diagrama de Secuencia Dar baja a usuarios

Secuencia: Adicionar Usuarios

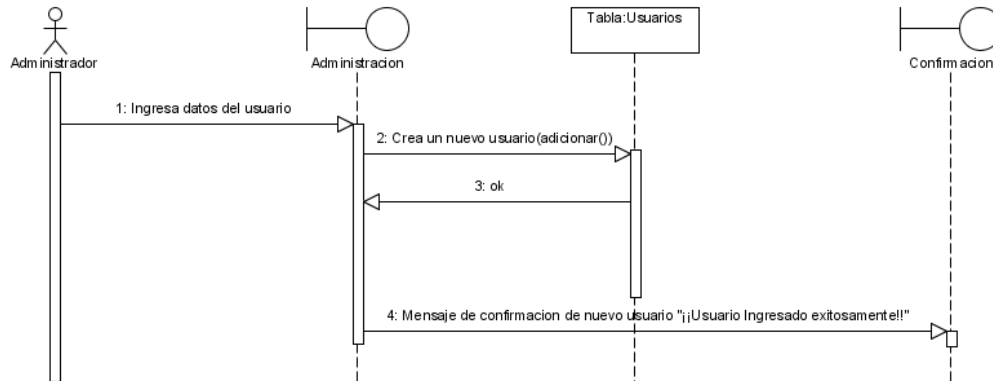


Figura 55: Diagrama de Secuencia Adicionar usuarios

Secuencia: Modificar usuarios

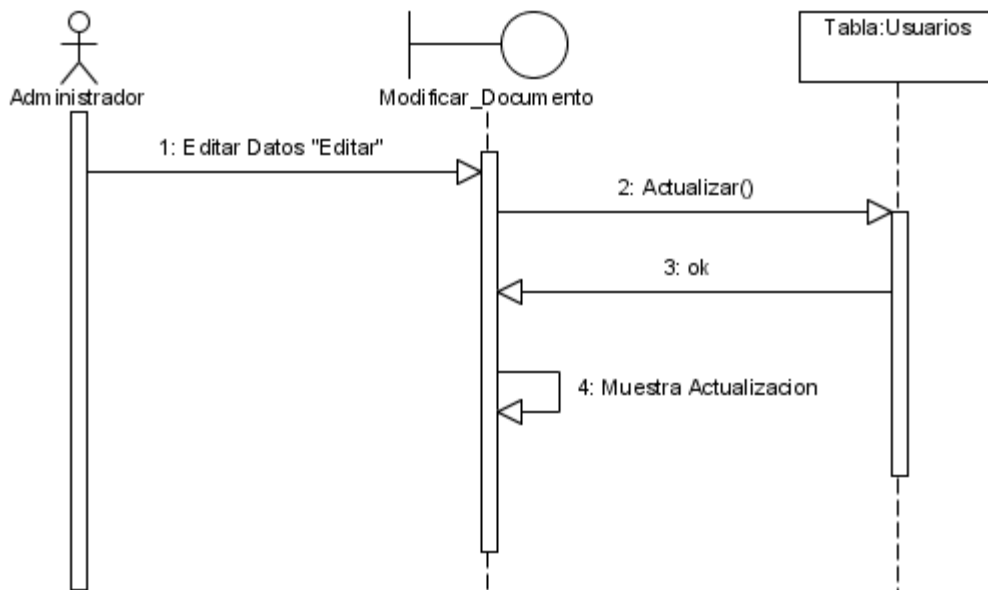


Figura 56: Diagrama de Secuencia Modificar usuarios

Secuencia: Crear Documentos

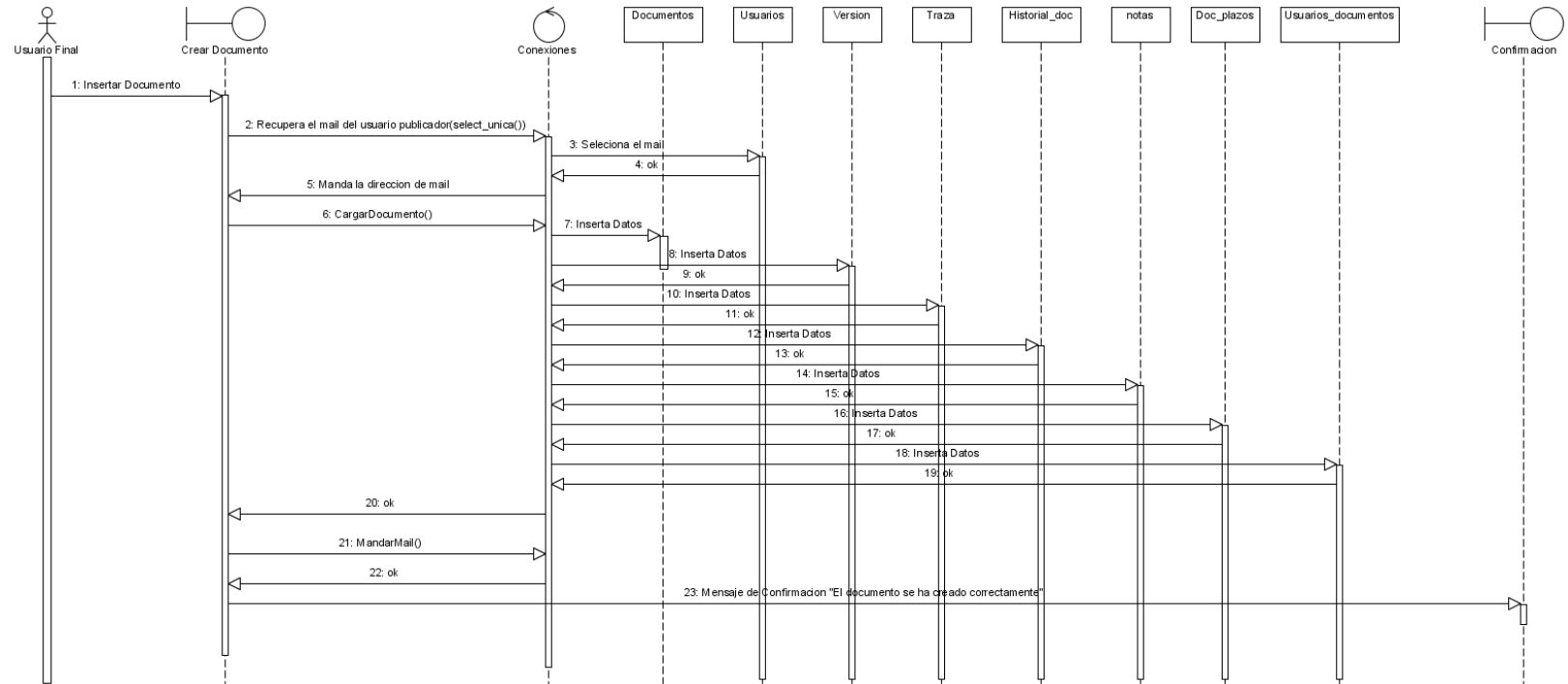


Figura 57: Diagrama de Secuencia Crear documentos

Secuencia: Ver documento publicado

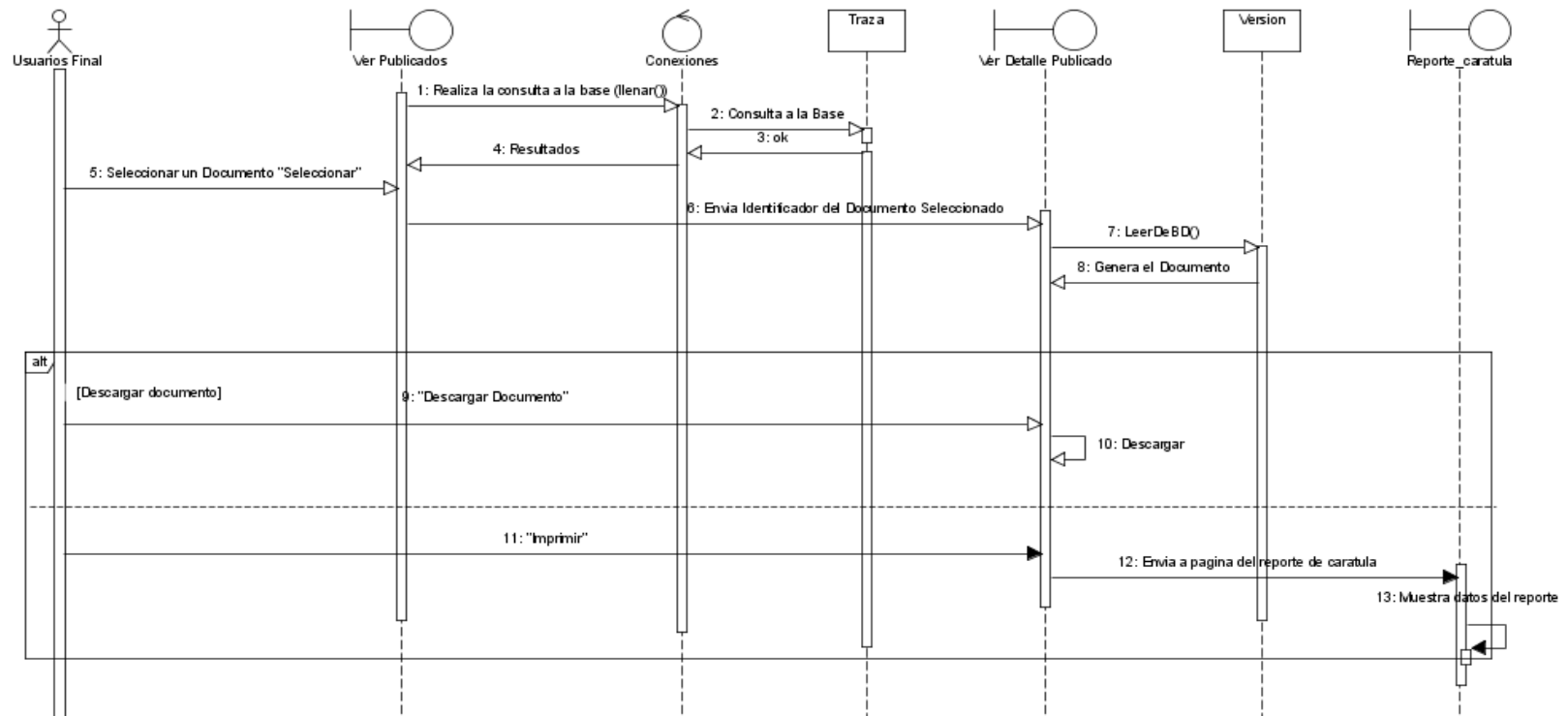


Figura 58: Diagrama de Secuencia Ver documento publicado

Secuencia: Revisar Documentos

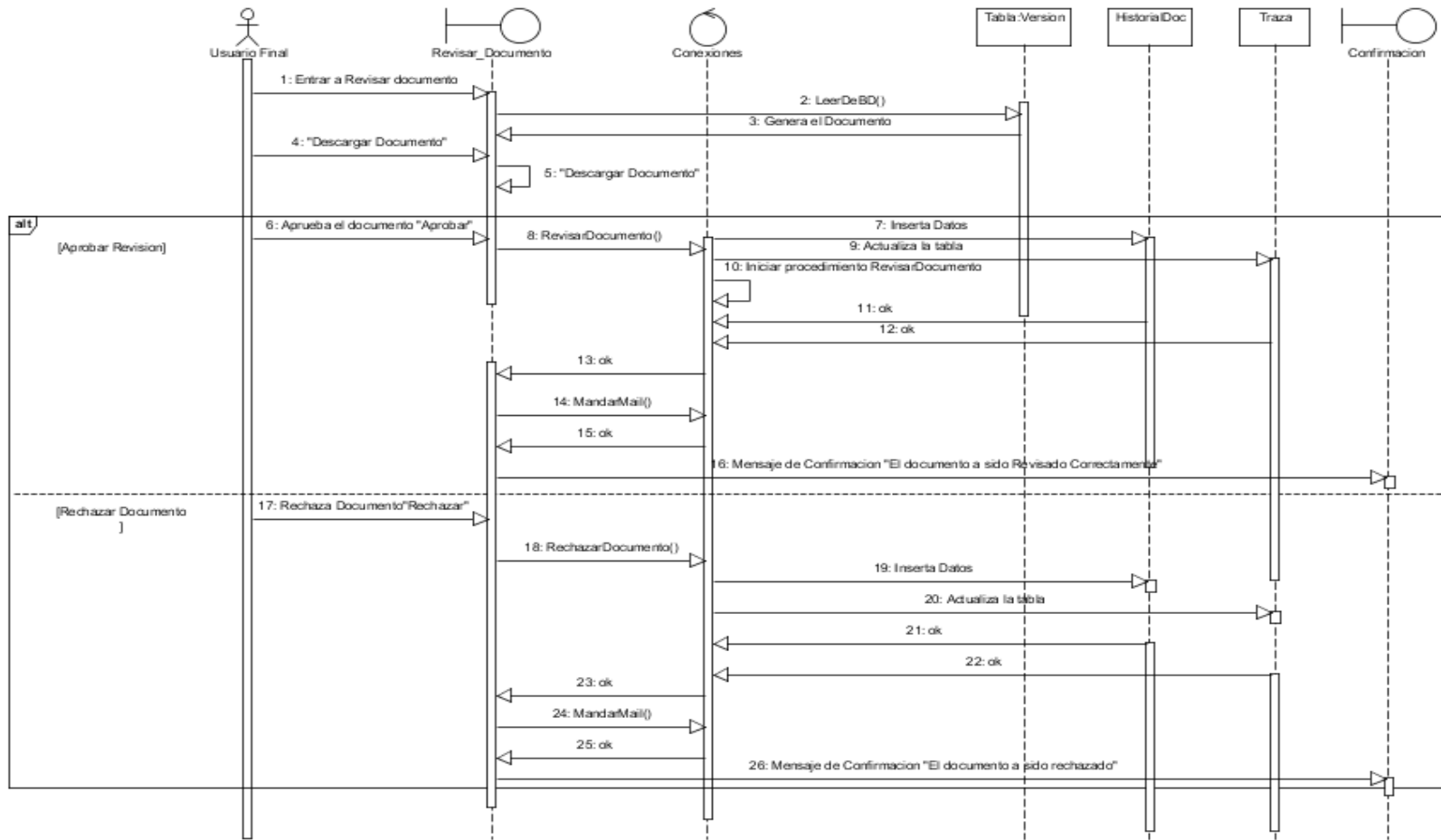


Figura 59: Diagrama de Secuencia Revisar documentos

Secuencia: Aprobar Documentos

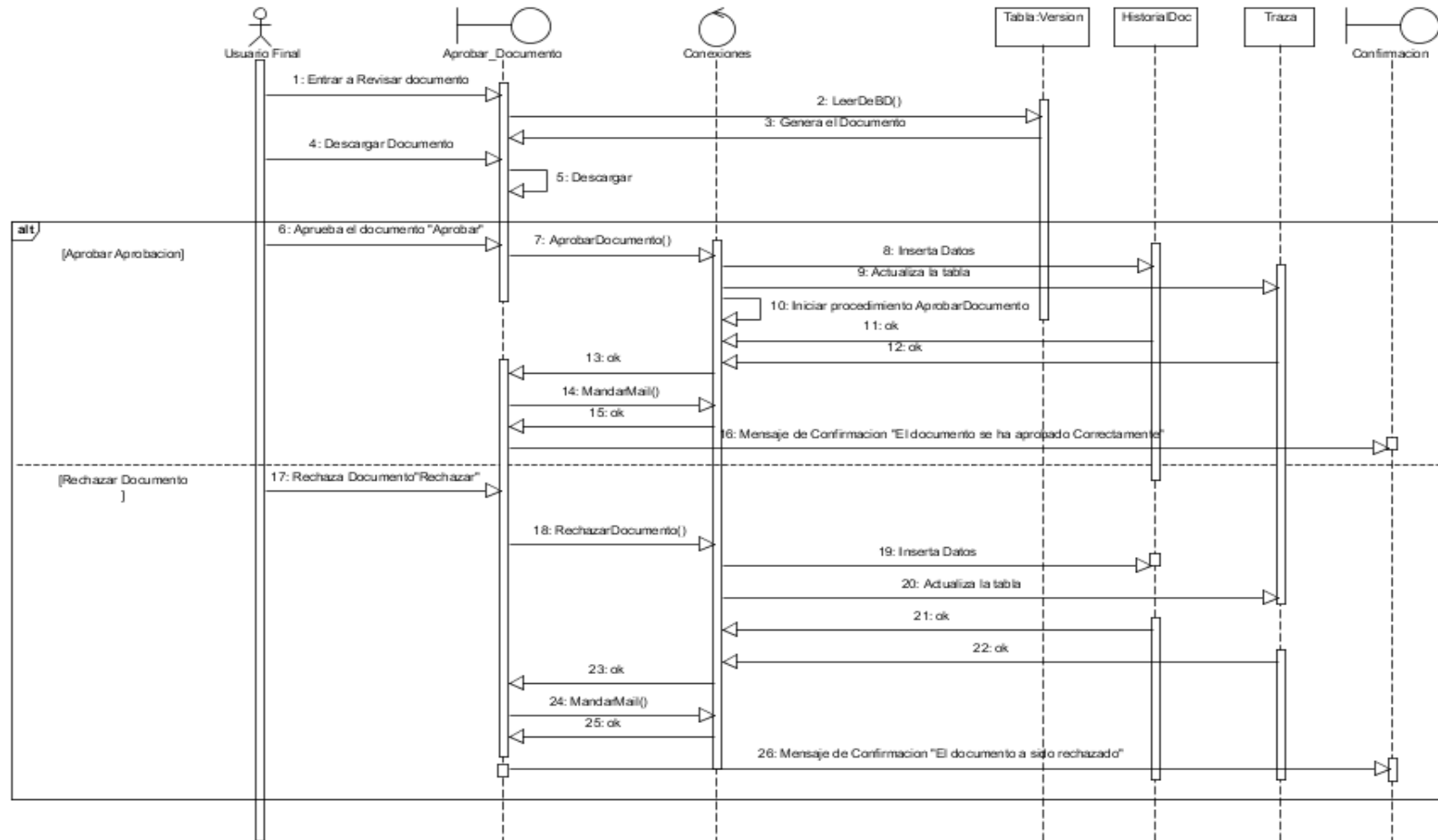


Figura 60: Diagrama de Secuencia Aprobar documentos

Secuencia: Publicar Documentos

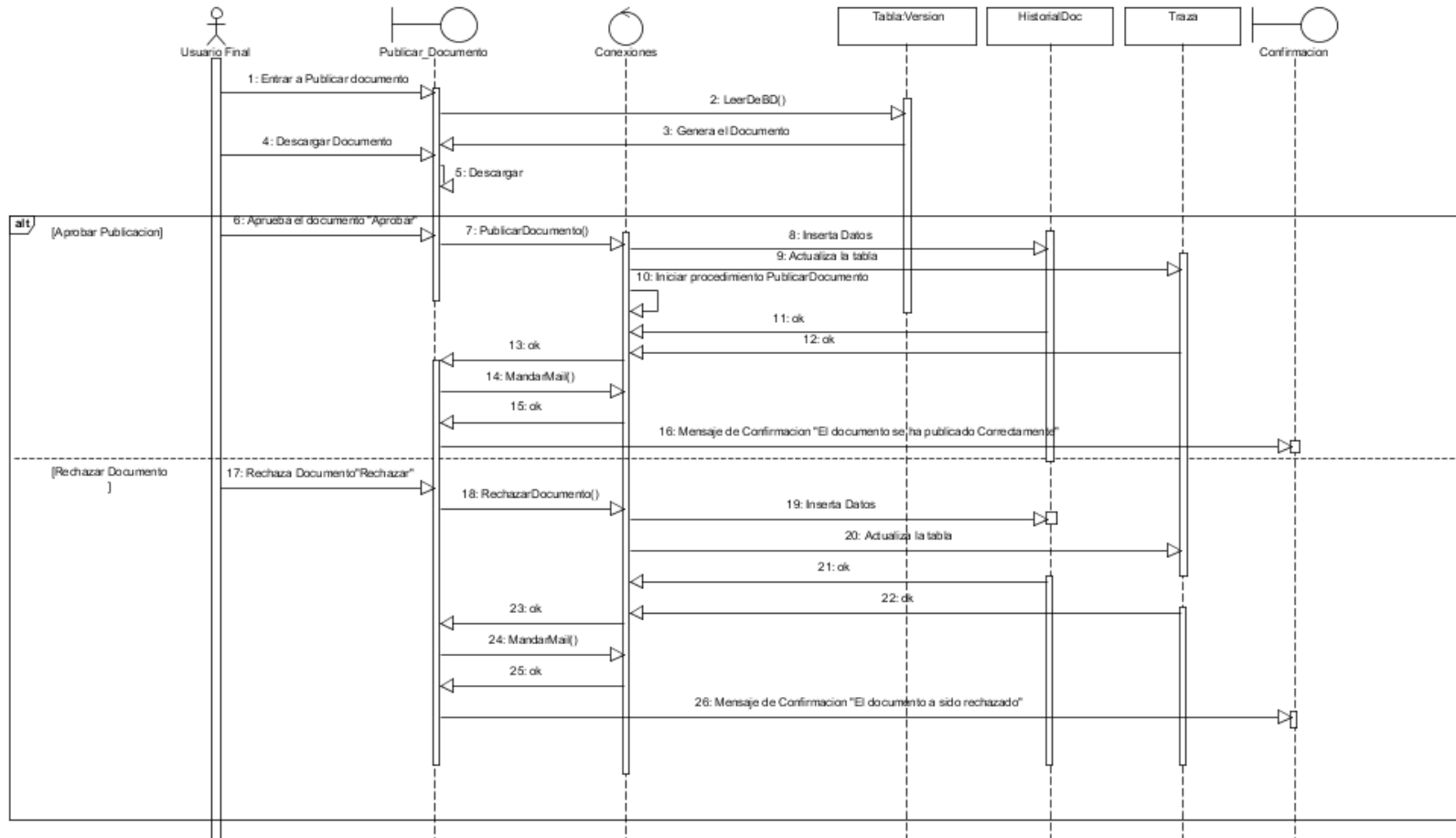


Figura 61: Diagrama de Secuencia Publicar documentos

Secuencia: Asignar Documentos

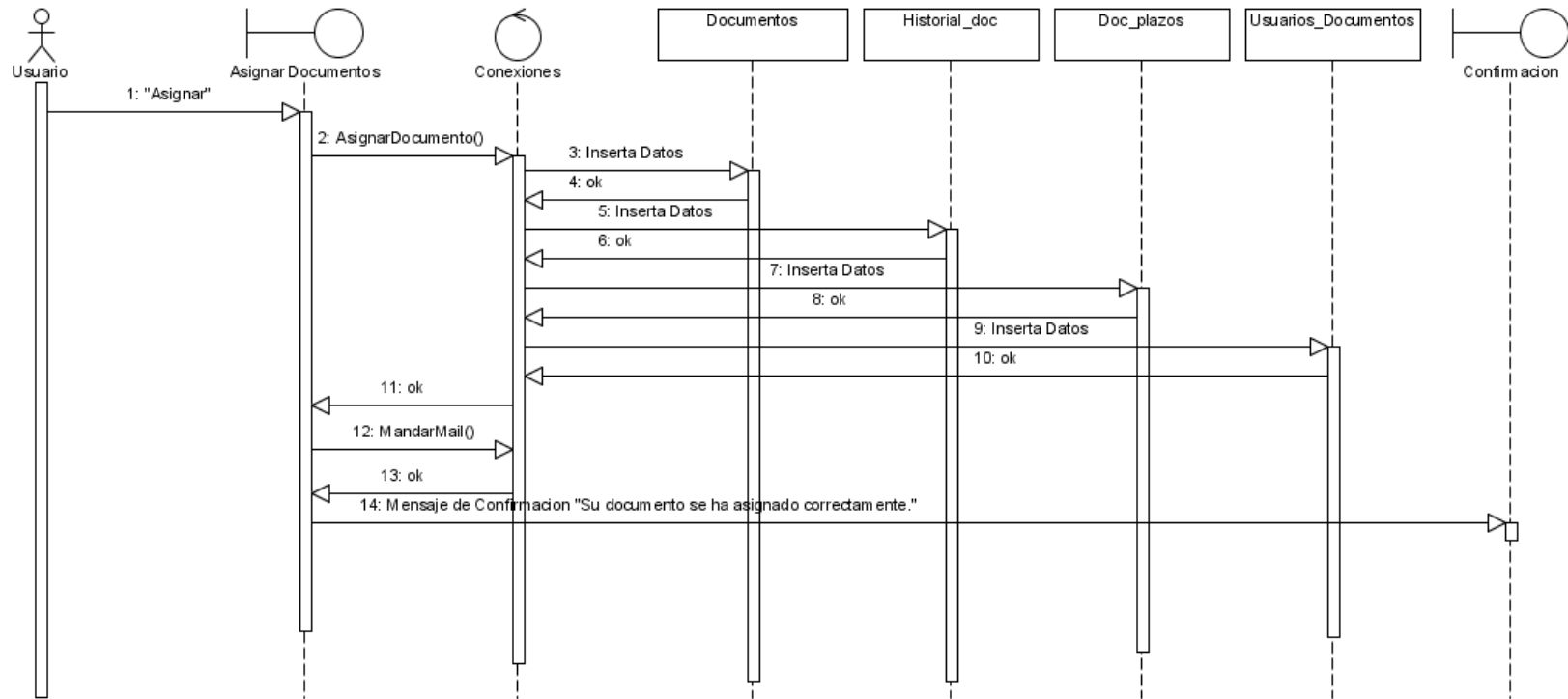


Figura 62: Diagrama de Secuencia Asignar documentos

Secuencia: Búsqueda Simple

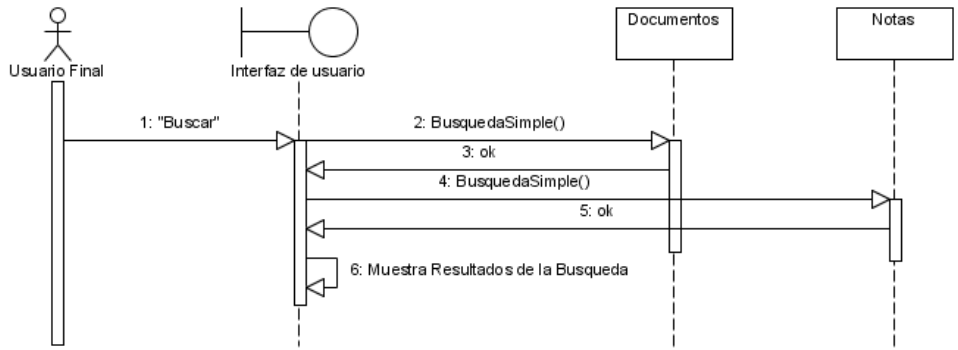


Figura 63: Diagrama de Secuencia Búsqueda simple

Secuencia: Búsqueda Avanzada

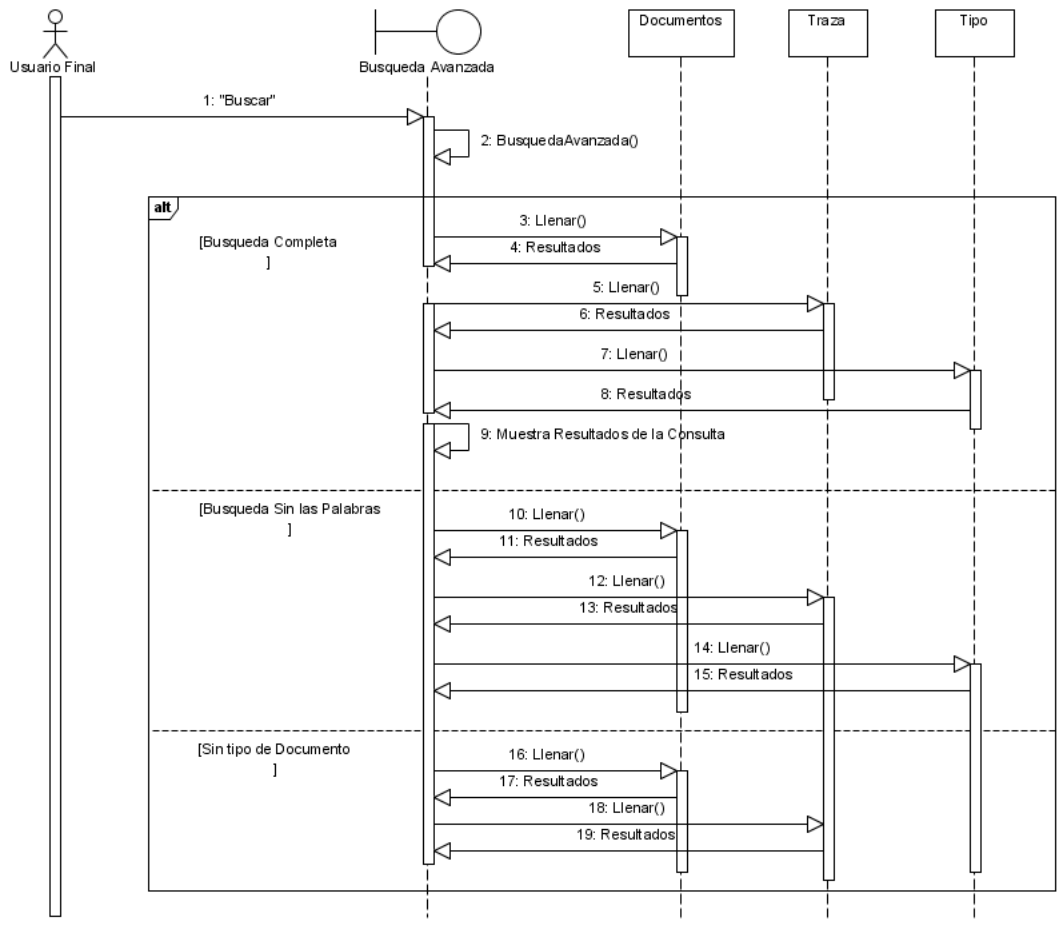


Figura 64: Diagrama de Secuencia Búsqueda avanzada

Secuencia: Nuevo tipo de Documento

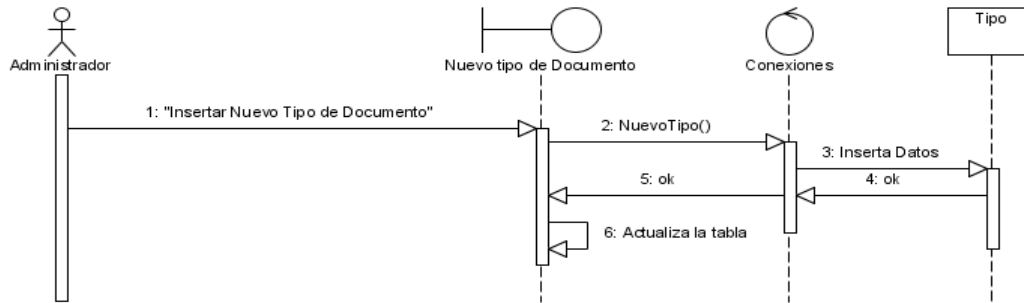


Figura 65: Diagrama de Secuencia Nuevo tipo de documento

Secuencia: Sacar Respaldo a la Base de Datos

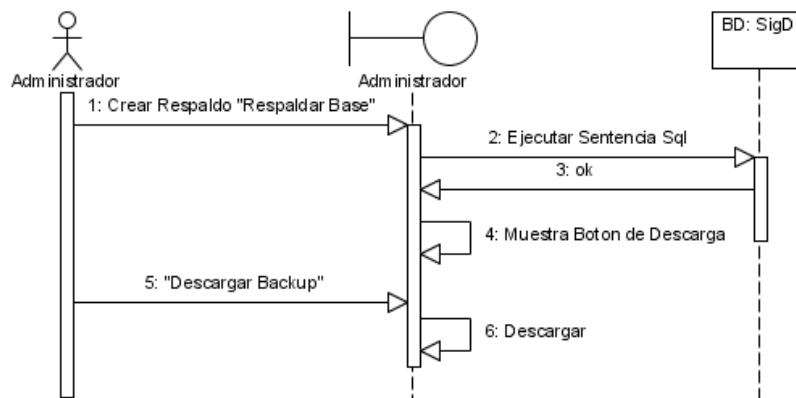


Figura 66: Diagrama de Secuencia Sacar respaldo a la base de datos

Secuencia: Restaurar Base de Datos

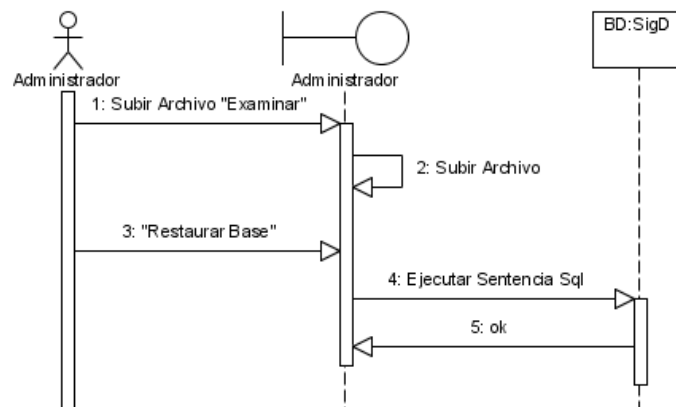


Figura 67: Diagrama de Secuencia Restaurar base de datos

Secuencia: Modificar Datos Personales

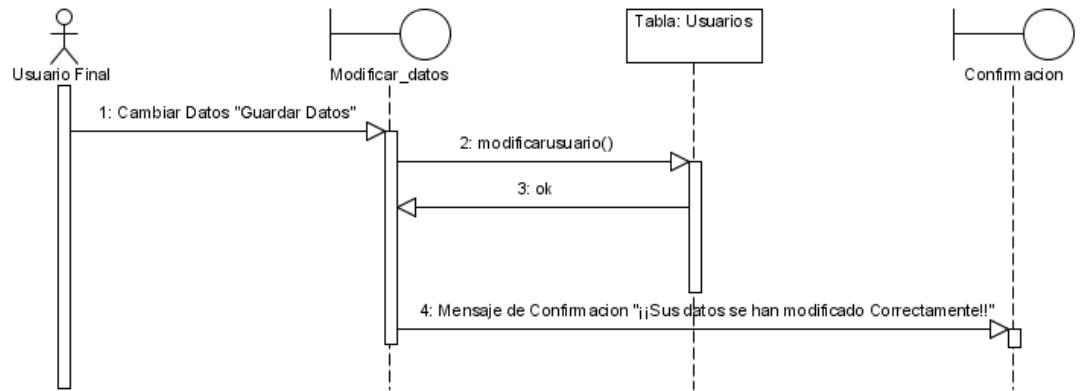


Figura 68: Diagrama de Secuencia Modificar Datos Personales.

I.1.1.20. Modelo de datos

I.1.1.20.1. Introducción

Base de Datos es el conjunto de datos almacenados con una estructura lógica. Es decir, tan importante como los datos, es la estructura conceptual con la que se relacionan entre ellos. En la práctica, podemos pensar esto como el conjunto de datos más los programas (o *software*) que hacen de ellos un conjunto consistente.

Si no tenemos los dos factores unidos, no podemos hablar de una base de datos, ya que ambos combinados dan la coherencia necesaria para poder trabajar con los datos de una manera sistemática.

I.1.1.20.2. Propósito

Usar el lenguaje estándar: Lenguaje Estructurado de Consultas-SQL para la crear la Base de datos.

I.1.1.20.3. Alcance

- Mostrar las relaciones de las tablas usando el modelo relacional.
- Definir de la Base de Datos con el Lenguaje Estructurado de Consultas-SQL.

I.1.1.20.4. Diagrama de clases

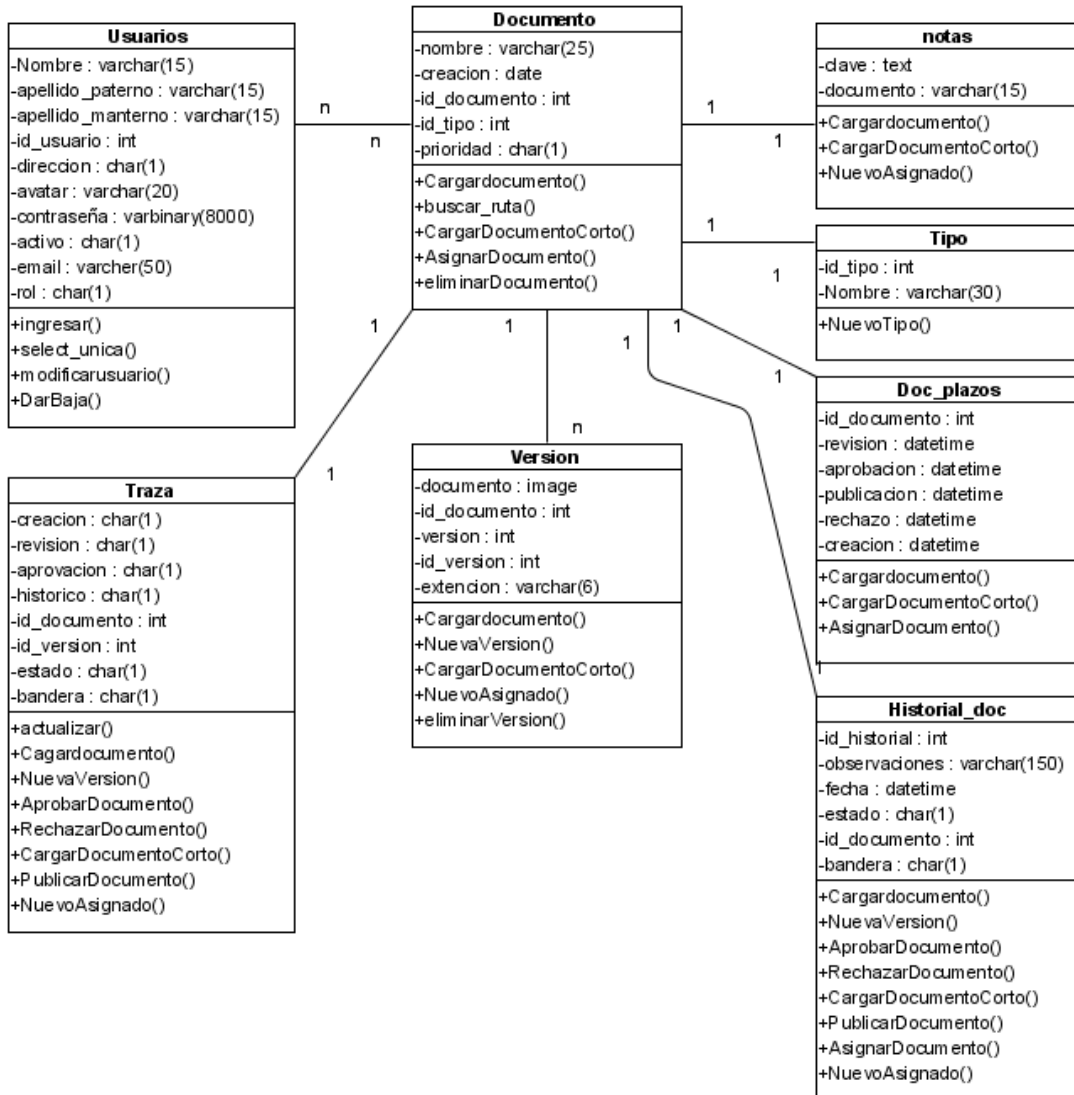


Figura 69: Diagrama de clases

I.1.1.20.5. Modelo relacional

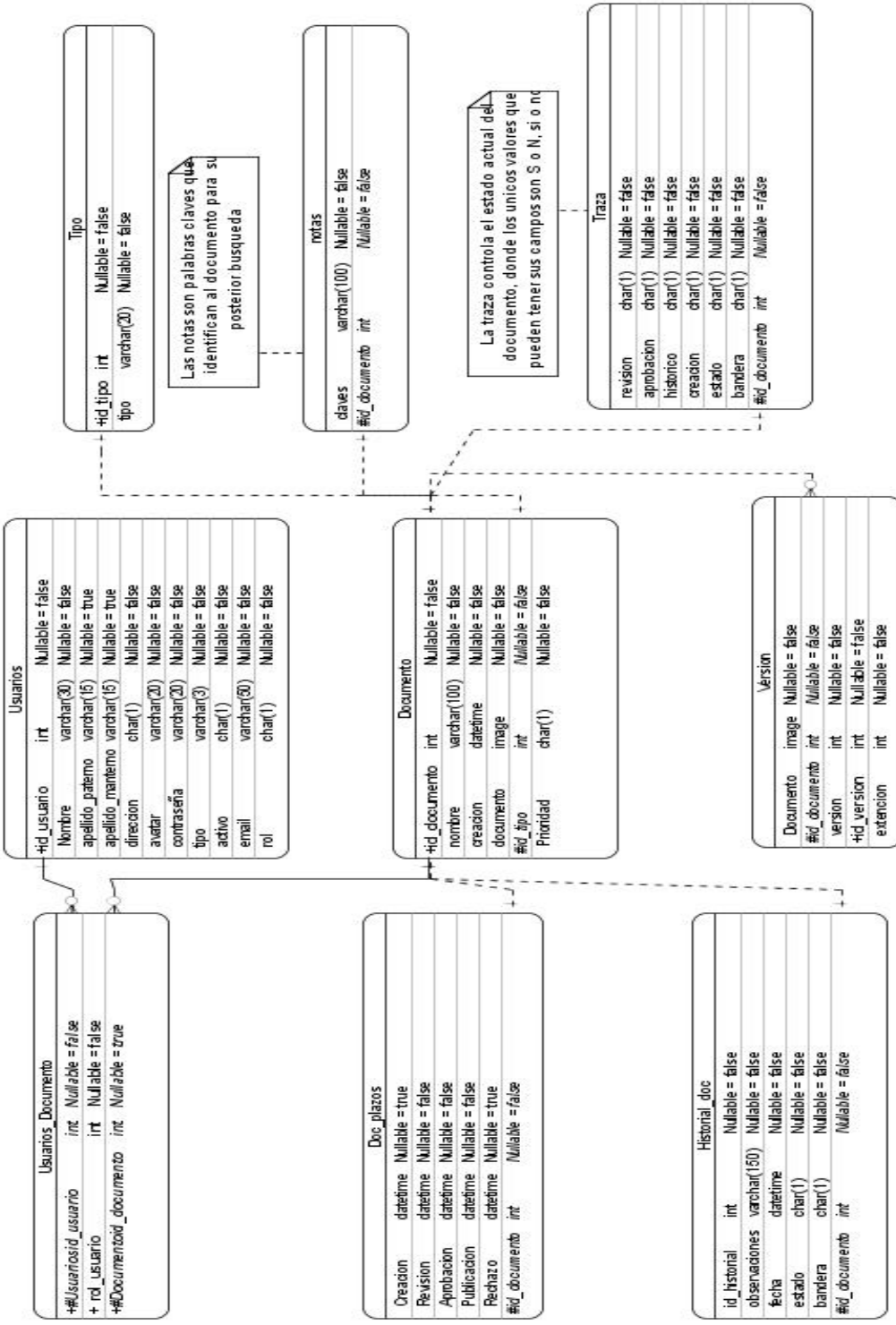


Figura 70: Diagrama entidad-relación

I.1.1.20.6. Definición de tablas

```
CREATE TABLE Version (  
    Documento image NOT NULL,  
    id_documento int NOT NULL,  
    version int NOT NULL,  
    id_version int NOT NULL,  
    extencion int NOT NULL,  
    PRIMARY KEY (id_version));  
  
CREATE TABLE Historial_doc (  
    id_historial int NOT NULL,  
    observaciones varchar(150) NOT NULL,  
    fecha datetime NOT NULL,  
    estado char(1) NOT NULL,  
    bandera char(1) NOT NULL,  
    id_documento int NOT NULL);  
  
CREATE TABLE Doc_plazos (  
    Creacion datetime NULL,  
    Revision datetime NOT NULL,  
    Aprobacion datetime NOT NULL,  
    Publicacion datetime NOT NULL,  
    Rechazo datetime NULL,  
    id_documento int NOT NULL);  
  
CREATE TABLE Tipo (  
    id_tipo int NOT NULL,  
    nombre varchar(50) NOT NULL,  
    PRIMARY KEY (id_tipo));
```

```

id_tipo int IDENTITY NOT NULL,
tipo varchar(20) NOT NULL,
PRIMARY KEY (id_tipo));
CREATE TABLE Traza (
revision char(1) NOT NULL,
aprobacion char(1) NOT NULL,
historico char(1) NOT NULL,
creacion char(1) NOT NULL,
estado char(1) NULL,
bandera char(1) NULL,
id_documento int NOT NULL,
PRIMARY KEY (id_documento));
CREATE TABLE Usuarios_Documento (
Usuariosid_usuario int NOT NULL,
rol_usuario int IDENTITY NOT NULL,
Documentoid_documento int NOT NULL,
PRIMARY KEY (Usuariosid_usuario,
rol_usuario,
Documentoid_documento));
CREATE TABLE notas (
claves varchar(100) NOT NULL,
id_documento int NOT NULL);
CREATE TABLE Documento (
id_documento int IDENTITY NOT NULL,

```

```

nombre    varchar(100) NOT NULL,
creacion  datetime NOT NULL,
documento image NOT NULL,
id_tipo   int NOT NULL,
Prioridad char(1) NOT NULL,
PRIMARY KEY (id_documento));

CREATE TABLE Usuarios (
id_usuario    int IDENTITY NOT NULL,
Nombre        varchar(30) NOT NULL,
apellido_paterno varchar(15) NULL,
apellido_manterno varchar(15) NULL,
direccion     char(1) NOT NULL,
avatar        varchar(20) NOT NULL,
contraseña    varchar(20) NOT NULL,
tipo          varchar(3) NOT NULL,
activo        char(1) NOT NULL,
email         varchar(50) NOT NULL,
rol           char(1) NOT NULL,
PRIMARY KEY (id_usuario));

ALTER TABLE Traza ADD CONSTRAINT FKTraza419661 FOREIGN
KEY (id_documento) REFERENCES Documento (id_documento);

ALTER TABLE notas ADD CONSTRAINT FKnotas359360 FOREIGN
KEY (id_documento) REFERENCES Documento (id_documento);

ALTER TABLE Version ADD CONSTRAINT FKVersion357724
FOREIGN KEY (id_documento) REFERENCES Documento

```

```
(id_documento);

ALTER TABLE Historial_doc ADD CONSTRAINT FKHistorial_516719
FOREIGN KEY (id_documento) REFERENCES Documento
(id_documento);

ALTER TABLE Doc_plazos ADD CONSTRAINT FKDoc_plazos365952
FOREIGN KEY (id_documento) REFERENCES Documento
(id_documento);

ALTER TABLE Documento ADD CONSTRAINT FKDocumento784192
FOREIGN KEY (id_tipo) REFERENCES Tipo (id_tipo);

ALTER TABLE Usuarios_Documento ADD CONSTRAINT
FKUsuarios_D997277 FOREIGN KEY (Documentoid_documento)
REFERENCES Documento (id_documento);

ALTER TABLE Usuarios_Documento ADD CONSTRAINT
FKUsuarios_D684479 FOREIGN KEY (Usuariosid_usuario)
REFERENCES Usuarios (id_usuario);
```

I.1.1.20.7. Especificaciones de métodos

```
public int ingresar(string usuario, string clave)
```

```
{
    int numfilas=0;
```

```

//establecemos la cadena sql a ejecutar

SqlCommand sql = new SqlCommand("select count(*) from
Usuarios where avatar="" + usuario + "" and
CONVERT(VARCHAR(30),DECRYPTBYPASSPHRASE('$$taller$$',cont
raseña))="" + clave + "" AND activo='s'", conexionbase);

//declaramos instruccion sql

sql.CommandType = CommandType.Text;

try
{
//abriendo la conexion a la base de datos

conexionbase.Open();

//ejecutando la consulta que es del tipo scalar para que devuelva un
valor entero

//y guardando el resultado en la variable numfilas

numfilas = (int)sql.ExecuteScalar();

//cerramos la conexion

conexionbase.Close();

return numfilas;

}

catch

{

Console.WriteLine("error al abrir la conexion");

return numfilas;

}

}

```

```

public string id_usuario(string usuario, string clave)
{
    string id;

    SqlCommand sql = new SqlCommand("select id_usuario from
Usuarios where avatar='" + usuario + "' and
CONVERT(VARCHAR(30),DECRYPTBYPASSPHRASE('$$taller$$',cont
raseña))='" + clave + "'", conexionbase);

    sql.CommandType = CommandType.Text;

    try
    {
        conexionbase.Open();

        id = sql.ExecuteScalar().ToString();

        conexionbase.Close();

        return id;
    }

    catch(System.Exception e)
    {
        return id=e.Message.ToString();
    }
}

public string select_unica(string sql_cadena)
{
    string id;

    SqlCommand sql = new SqlCommand(sql_cadena, conexionbase);

    sql.CommandType = CommandType.Text;

```

```
try
{
    conexionbase.Open();
    id = sql.ExecuteScalar().ToString();
    conexionbase.Close();
    return id;
}
catch (System.Exception e)
{
    return id = e.Message.ToString();
}
}

public bool insertar(string sql_cadena)
{
    SqlCommand comando = new SqlCommand(sql_cadena,
conexionbase);

    try
    {
        conexionbase.Open();
        comando.ExecuteNonQuery();
        conexionbase.Close();
        return true;
    }
    catch (System.Exception e)
    {
```

```
        string error = e.Message.ToString();
        return false;
    }

}

public SqlDataAdapter llenar(string sql_cadena)
{
    SqlDataAdapter sqlDA=new SqlDataAdapter();
    try
    {
        //asignamos la sentencia de seleccion al sqldataadapter
        comando = new SqlCommand(sql_cadena, conexionbase);
        comando.CommandType = CommandType.Text;
        sqlDA = new SqlDataAdapter();
        sqlDA.SelectCommand=comando;
        return sqlDA;
    }
    catch (System.Exception e)
    {
        string error = e.Message.ToString();
        return sqlDA;
    }

    conexionbase.Close();
}
```

```
}
```

```
public void cargar_documento(string sRuta, string sFichero, int tipo, int version, char creacio_estado, char revision_estado, char aprobacion_estado, char historico_estado, string claves, int id_usuario_1, char rol_1, int id_usuario_2, char rol_2, int id_usuario_3, char rol_3, int id_usuario_4, char rol_4, string extension, DateTime fec_aprov, DateTime fec_rev, DateTime fec_hist, string prioridad)
```

```
{
```

```
    string observaciones="Sin observaciones";
```

```
    FileStream fs = new FileStream(sRuta, System.IO.FileMode.Open);//creamos un fileStream para leer el fichero en modo binario
```

```
    Byte[] data = new byte[fs.Length];//creamos un array de bytes para almacenar los datos leidos por el fs
```

```
    fs.Read(data,0, Convert.ToInt32(fs.Length));//guardamos los datos en un array data
```

```
    //SqlConnection cnn = new SqlConnection("Data Source=servidor;Initial Catalog=sigd;Persist Security Info=True;UserID=sa;Password=uajms123");//creamos la cadena de conexion
```

```
    conexionbase.Open();//abrimos la conexion
```

```
    SqlCommand cmd = new SqlCommand("SubirDoc", conexionbase);
```

```
    cmd.CommandType = CommandType.StoredProcedure;
```

```
    //añadimos los parametros especiales
```

```
    cmd.Parameters.Add("@doc",data);
```

```
    cmd.Parameters.Add("@nombre",sFichero);
```

```
    cmd.Parameters.Add("@tipo",tipo);
```

```
//cmd.Parameters.Add("@creacion",creacion);

cmd.Parameters.Add("@version",version);

cmd.Parameters.Add("@creacion_estado",creacio_estado);

cmd.Parameters.Add("@revision_estado",revision_estado);

cmd.Parameters.Add("@aprobacion_estado",aprobacion_estado);

cmd.Parameters.Add("@historico_estado",historico_estado);

cmd.Parameters.Add("@claves",claves);

cmd.Parameters.Add("@id_usuario_1",id_usuario_1);

cmd.Parameters.Add("@rol_1",rol_1);

cmd.Parameters.Add("@id_usuario_2",id_usuario_2);

cmd.Parameters.Add("@rol_2",rol_2);

cmd.Parameters.Add("@id_usuario_3",id_usuario_3);

cmd.Parameters.Add("@rol_3",rol_3);

cmd.Parameters.Add("@id_usuario_4",id_usuario_4);

cmd.Parameters.Add("@rol_4",rol_4);

cmd.Parameters.Add("@extension", extension);

cmd.Parameters.Add("@observaciones",observaciones);

cmd.Parameters.Add("@fec_aprov", fec_aprov);

cmd.Parameters.Add("@fec_rev",fec_rev);

cmd.Parameters.Add("@fec_hist",fec_hist);

cmd.Parameters.Add("@prioridad",prioridad);

//Ejecutamos el procedimiento que contiene los datos del nuevo
documento

cmd.ExecuteNonQuery();
```

```

        //Cerramos la conexion
        conexionbase.Close();
        fs.Close();

    }

    public void NuevaVersion(string sRuta,int version,string estado,string
extension,string id_documento)
    {
        string observaciones = "Sin observaciones";

        FileStream fs = new FileStream(sRuta,
System.IO.FileMode.Open);//creamos un fileStream para leer el fichero en
modo binario

        Byte[] data = new byte[fs.Length];//creamos un array de bytes para
almacenar los datos leidos por el fs

        fs.Read(data, 0, Convert.ToInt32(fs.Length));//guardamos los datos
en un array data

        //SqlConnection cnn = new SqlConnection("Data
Source=servidor;Initial Catalog=sigd;Persist Security Info=True;User
ID=sa;Password=uajms123");//creamos la cadena de conexion

        conexionbase.Open();//abrimos la conexion

        SqlCommand cmd = new SqlCommand("NuevaVersion",
conexionbase);

        cmd.CommandType = CommandType.StoredProcedure;

        //añadimos los parametros especiales
        cmd.Parameters.Add("@doc", data);

        //cmd.Parameters.Add("@creacion",creacion);

```

```

cmd.Parameters.Add("@version", version);

cmd.Parameters.Add("@estado", estado);

cmd.Parameters.Add("@extension", extension);

cmd.Parameters.Add("@observaciones", observaciones);

cmd.Parameters.Add("@id_documento", id_documento);

//Ejecutamos el procedimiento que contiene los datos del nuevo
documento

cmd.ExecuteNonQuery();

//Cerramos la conexion

conexionbase.Close();

fs.Close();

}

public void AprobarDocumento(string observaciones, char estado,char
estado_1, int id_documento, char bandera,char bandera_1, char creacion,
char revision, char aprobacion, char historico)

{

conexionbase.Open();//abrimos la conexion

SqlCommand cmd = new SqlCommand("AprobarDocumento",
conexionbase);//llamamos al procedimiento AprobarDocumento

cmd.CommandType = CommandType.StoredProcedure;

//añadimos parametros al procedimiento

cmd.Parameters.Add("@observaciones", observaciones);

cmd.Parameters.Add("@estado", estado);

cmd.Parameters.Add("@estado_1", estado_1);

```

```

        cmd.Parameters.Add("@id_documento", id_documento);

        cmd.Parameters.Add("@bandera", bandera);

        cmd.Parameters.Add("@bandera_1", bandera_1);

        cmd.Parameters.Add("@creacion", creacion);

        cmd.Parameters.Add("@revision", revision);

        cmd.Parameters.Add("@aprobacion", aprobacion);

        cmd.Parameters.Add("@historico", historico);

        cmd.ExecuteNonQuery();//ejecutamos el procedimiento
        conexionbase.Close();
    }

    public void RechazarDocumento(string observaciones, char estado, int
id_documento)
    {
        conexionbase.Open();//abrimos la conexion

        SqlCommand cmd = new SqlCommand("ReprobarDocumento",
conexionbase);//llamamos al procedimiento AprobarDocumento

        cmd.CommandType = CommandType.StoredProcedure;

        //añadimos parametros al procedimiento

        cmd.Parameters.Add("@observaciones", observaciones);

        cmd.Parameters.Add("@estado", estado);

        cmd.Parameters.Add("@id_documento", id_documento);

        cmd.ExecuteNonQuery();//ejecutamos el procedimiento
        conexionbase.Close();
    }

```

```

public void cargar_documento_corto(string sRuta, string sFichero, int
tipo, int version, char creacio_estado, char revision_estado, char
aprobacion_estado, char historico_estado, string claves, int id_usuario_1,
char rol_1, int id_usuario_5, char rol_5, string extension, DateTime fec_pub,
string prioridad)
{
    string observaciones = "Sin Observaciones";

    FileStream fs = new FileStream(sRuta,
System.IO.FileMode.Open);//creamos un fileStream para leer el fichero en
modo binario

    Byte[] data = new byte[fs.Length];//creamos un array de bytes para
almacenar los datos leidos por el fs

    fs.Read(data, 0, Convert.ToInt32(fs.Length));//guardamos los datos
en un array data

    //SqlConnection cnn = new SqlConnection("Data
Source=servidor;Initial Catalog=sigd;Persist Security Info=True;User
ID=sa;Password=uajms123");//creamos la cadena de conexion

    conexionbase.Open();//abrimos la conexion

    SqlCommand cmd = new SqlCommand("SubirDoc_1",
conexionbase);

    cmd.CommandType = CommandType.StoredProcedure;

    //añadimos los parametros especiales

    cmd.Parameters.Add("@doc", data);

    cmd.Parameters.Add("@nombre", sFichero);

    cmd.Parameters.Add("@tipo", tipo);

    //cmd.Parameters.Add("@creacion", creacion);

    cmd.Parameters.Add("@version", version);

```

```

cmd.Parameters.Add("@creacion_estado", creacio_estado);
cmd.Parameters.Add("@revisión_estado", revisión_estado);
cmd.Parameters.Add("@aprobación_estado", aprobación_estado);
cmd.Parameters.Add("@historico_estado", historico_estado);
cmd.Parameters.Add("@claves", claves);
cmd.Parameters.Add("@id_usuario_1", id_usuario_1);
cmd.Parameters.Add("@rol_1", rol_1);
cmd.Parameters.Add("@id_usuario_5", id_usuario_5);
cmd.Parameters.Add("@rol_5", rol_5);
cmd.Parameters.Add("@extension", extension);
cmd.Parameters.Add("@observaciones", observaciones);
cmd.Parameters.Add("@fec_hist", fec_pub);
cmd.Parameters.Add("@prioridad", prioridad);

//Ejecutamos el procedimiento que contiene los datos del nuevo
documento

cmd.ExecuteNonQuery();

//Cerramos la conexión
conexionbase.Close();

fs.Close();

}

public void PublicarDocumento(string observaciones, char estado, int
id_documento, char bandera)
{

```

```

        conexionbase.Open();//abrimos la conexion

        SqlCommand cmd = new SqlCommand("PublicarDocumento",
conexionbase);//llamamos al procedimiento AprobarDocumento

        cmd.CommandType = CommandType.StoredProcedure;

        //añadimos parametros al procedimiento

        cmd.Parameters.Add("@observaciones", observaciones);

        cmd.Parameters.Add("@estado", estado);

        cmd.Parameters.Add("@id_documento", id_documento);

        cmd.Parameters.Add("@bandera", bandera);

        cmd.ExecuteNonQuery();//ejecutamos el procedimiento

        conexionbase.Close();

    }

    public SqlDataAdapter DevolverDataAdapter(string consulta)

    {

        SqlDataAdapter adaptador = new SqlDataAdapter();

        adaptador.SelectCommand= new SqlCommand();

        adaptador.SelectCommand.Connection = conexionbase;

        adaptador.SelectCommand.CommandType = CommandType.Text;

        adaptador.SelectCommand.CommandText = consulta;

        conexionbase.Close();

        return adaptador;

    }

    public void modificarusuario(string nombre, string appaterno, string
apmaterno, string avatar, string contraseña, string dir, string id_usuario,
string mail)

```

```

    {

        conexionbase.Open();

        SqlCommand cmd = new SqlCommand("ModificarUsuario",
conexionbase);

        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("@nombre",nombre);
        cmd.Parameters.Add("@appaterno",appaterno);
        cmd.Parameters.Add("@apmaterno",apmaterno);
        cmd.Parameters.Add("@avatar",avatar);
        cmd.Parameters.Add("@contraseña", contraseña);
        cmd.Parameters.Add("@dir", dir);
        cmd.Parameters.Add("@id_usuario",id_usuario);
        cmd.Parameters.Add("@mail", mail);

        cmd.ExecuteNonQuery();

        conexionbase.Close();

    }

    //llamar al procedimiento almacenado para asignar un nuevo documento
a una persona

    public void AsignarDocumento(string nombre, DateTime fecha, string
tipo_doc, string usuario_creador, string usuario_revisor, string
observaciones, string prioridad)

```

```

    {
        conexionbase.Open();

        SqlCommand cmd = new SqlCommand("AsignarDocumento",
conexionbase);

        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("@nombre",nombre);

        cmd.Parameters.Add("@tipo",tipo_doc);

        cmd.Parameters.Add("@id_usuario_1",usuario_creador);

        cmd.Parameters.Add("@rol_1",'C');

        cmd.Parameters.Add("@id_usuario_5",usuario_revisor);

        cmd.Parameters.Add("@rol_5",'R');

        cmd.Parameters.Add("@rol_5_1",'P');

        cmd.Parameters.Add("@observaciones",observaciones);

        cmd.Parameters.Add("@fec_lim",fecha);

        cmd.Parameters.Add("@prioridad", prioridad);

        cmd.ExecuteNonQuery();

        conexionbase.Close();
    }

//metodo para dar de baja a un usuario
public string DarBaja(string usuario)
{
    try
    {

```

```

        conexionbase.Open();

        SqlCommand cmd = new SqlCommand("DarBaja",
conexionbase);

        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("@usuario",int.Parse(usuario));

        cmd.ExecuteNonQuery();

        conexionbase.Close();

        return "El usuario ha sido dado de baja correctamente";
    }catch(System.Exception e)
    {
        e.Message.ToString();

        return "Ha ocurrido un error y no se a podido realizar la
operacion";
    }
}

//metodo para agregar un nuevo tipo de documento
public string NuevoTipo(string tipo)
{
    SqlCommand sql = new SqlCommand("INSERT INTO Tipo
(Nombre) VALUES ('"+ tipo +"")", conexionbase);

    sql.CommandType = CommandType.Text;

    try

```

```

        {
            conexionbase.Open();
            sql.ExecuteScalar();
            conexionbase.Close();

            return "El nuevo tipo de documento ha sido guardado
correctamente";
        }
        catch (System.Exception e)
        {
            e.Message.ToString();

            return "Ha ocurrido un error y el documento no se ha cargado a la
base de datos";
        }
    }

    public void NuevoAsignado(string sRuta, int version, string extension,
int id_documento, string claves)
    {
        try
        {
            FileStream fs = new FileStream(sRuta,
System.IO.FileMode.Open);//creamos un fileStream para leer el fichero en
modo binario

            Byte[] data = new byte[fs.Length];//creamos un array de bytes para
almacenar los datos leidos por el fs

            fs.Read(data, 0, Convert.ToInt32(fs.Length));//guardamos los
datos en un array data

```

```
        conexionbase.Open();

        SqlCommand cmd = new SqlCommand("NuevoAsignado",
conexionbase);

        cmd.CommandType = CommandType.StoredProcedure;

        cmd.Parameters.Add("@doc",data );
        cmd.Parameters.Add("@version",version );
        cmd.Parameters.Add("@extension", extension);
        cmd.Parameters.Add("@id_documento", id_documento);
        cmd.Parameters.Add("@creacion_estado", 'N');
        cmd.Parameters.Add("@revision_estado", 'S');
        cmd.Parameters.Add("@aprobacion_estado", 'N');
        cmd.Parameters.Add("@historico_estado", 'N');
        cmd.Parameters.Add("@claves ", claves);

        cmd.ExecuteNonQuery();

        conexionbase.Close();

    }

    catch (System.Exception e)
    {
        e.Message.ToString();
    }

}
```

```

public void mandarmail(string maildestino, string titulo, string cuerpo)
{
    MailMessage mail = new MailMessage();

    SmtpClient SmtpServer = new SmtpClient();

    SmtpServer.Credentials = new
System.Net.NetworkCredential("cortedeptaj@gmail.com", "corte2010");

    SmtpServer.Port = 587;

    SmtpServer.Host = "smtp.gmail.com";

    SmtpServer.EnableSsl = true;

    mail = new MailMessage();

    try
    {
        mail.To.Add(new MailAddress(maildestino));

        mail.From = new MailAddress("cortedeptaj@gmail.com", "Taller
3", System.Text.Encoding.UTF8);

        mail.Subject = titulo;

        mail.Body = cuerpo;

        mail.DeliveryNotificationOptions =
DeliveryNotificationOptions.OnFailure;

        mail.ReplyTo = new MailAddress("cortedeptaj@gmail.com");

        SmtpServer.Send(mail);

    }

    catch (System.Exception ex)
    {

```

```
        ex.Message.ToString();
    }
}
public void eliminarDocumento(string id_documento)
{
    SqlCommand cmd = new SqlCommand("BorrarDoc",
conexionbase);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.Add("@id_documento", id_documento);

    cmd.ExecuteNonQuery();

    conexionbase.Close();
}
public void eliminarVersion(string id_ver)
{
    SqlCommand cmd = new SqlCommand("BorrarVer", conexionbase);

    cmd.CommandType = CommandType.StoredProcedure;

    cmd.Parameters.Add("id_ver", id_ver);

    cmd.ExecuteNonQuery();

    conexionbase.Close();
}
public string GenerarNombreFichero()
```

```

{
    int ultimoTick = 0;
    while (ultimoTick == Environment.TickCount)
    {
        System.Threading.Thread.Sleep(1);
    }
    ultimoTick = Environment.TickCount;
    return DateTime.Now.ToString("yyyyMMddhhmmss") + "." +
        ultimoTick.ToString();
} }

```

I.1.1.20.8. Procedimientos almacenados

Aprobar documento

GO

CREATE PROCEDURE [dbo].[AprobarDocumento]

-- Add the parameters for the stored procedure here

(@observaciones AS Text, @estado AS Char(1), @id_documento
AS int, @bandera AS Char(1), @bandera_1 AS Char(1), @creacion AS
Char(1), @revision AS Char(1), @aprobacion AS Char(1), @historico AS
Char(1), @estado_1 AS Char(1))

AS

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,
bandera) values (@observaciones,GetDate(),@estado_1, @id_documento,
@bandera_1)
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,
bandera) values (@observaciones,GetDate(),@estado, @id_documento,
@bandera)
```

```
UPDATE Traza set creacion=@creacion, revision=@Revision,
aprobacion=@aprobacion, historico=@historico, estado=@estado,
bandera=@bandera WHERE id_documento=@id_documento
```

AsignarDocumento

```
GO
```

```
CREATE PROCEDURE [dbo].[AsignarDocumento]( @nombre AS
VarChar(100), @tipo AS int, @id_usuario_1 AS int, @rol_1 AS char(1),
@id_usuario_5 AS int, @rol_5 AS char(1),@rol_5_1 AS char,
@observaciones AS Text, @fec_lim AS DATETIME, @prioridad AS
CHAR(1))
```

```
AS
```

```
INSERT INTO Documentos (nombre, id_tipo, creacion, prioridad) values
(@nombre, @tipo, Getdate(), @prioridad)
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,
bandera)values (@observaciones, Getdate(),
'C',IDENT_CURRENT('Documentos'), 'P')
```

```
INSERT INTO doc_plazos(id_documento, revision, aprobacion,
publicacion, creacion) values (IDENT_CURRENT('Documentos'),
Getdate(), Getdate(), Getdate(), CONVERT(DATETIME, @fec_lim,103))
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_1, IDENT_CURRENT('Documentos'),
@rol_1)
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_5, IDENT_CURRENT('Documentos'),
@rol_5)
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_5, IDENT_CURRENT('Documentos'),
@rol_5_1)
```

BorrarDoc

```
GO
```

```
CREATE PROCEDURE [dbo].[BorrarDoc](@id_documento AS int)
```

```
AS
```

```
DELETE FROM Documentos WHERE (id_documento=@id_documento)
```

```
DELETE FROM version WHERE (id_documento=@id_documento)
```

```
DELETE FROM Traza WHERE (id_documento=@id_documento)
```

```
DELETE FROM Historial_doc WHERE (id_documento=@id_documento)
```

```
DELETE FROM notas WHERE (id_documento=@id_documento)
```

```
DELETE FROM doc_plazos WHERE (id_documento=@id_documento)
```

```
DELETE FROM Usuarios_documentos WHERE
(id_documento=@id_documento)
```

BorrarVer

GO

```
CREATE PROCEDURE [dbo].[BorrarVer](@id_ver AS int)
```

AS

```
DELETE FROM version WHERE (id_version=@id_ver)
```

DarBaja

GO

```
CREATE PROCEDURE [dbo].[DarBaja]
```

```
(@usuario AS INT)
```

AS

```
UPDATE Usuarios set activo='n' WHERE id_usuario=@usuario
```

ModificarUsuario

GO

```
ALTER PROCEDURE [dbo].[ModificarUsuario]
```

```
-- Add the parameters for the stored procedure here
```

```
(@nombre AS text,@appaterno AS text, @apmaterno AS text,  
@avatar AS text, @contraseña AS varchar(30), @dir AS text, @id_usuario  
as int, @mail as text)
```

AS

```
UPDATE Usuarios set nombre=@nombre ,  
apellido_paterno=@appaterno, apellido_materno=@apmaterno,  
avatar=@avatar,  
contraseña=ENCRYPTBYPASSPHRASE('$$taller$$',@contraseña),direcci  
on=@dir, email=@mail WHERE id_usuario=@id_usuario
```

Nueva Version

GO

```
CREATE PROCEDURE [dbo].[NuevaVersion](@doc AS Image, @version
AS int,@extension AS varchar(6),@observaciones AS Text,
@id_documento AS int, @estado AS char(1))
```

AS

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,
bandera)values (@observaciones, Getdate(), @estado ,@id_documento, 'P')
```

```
INSERT INTO version (version, documento, id_documento, extencion)
values (@version, @doc, @id_documento, @extension)
```

```
UPDATE Traza set id_version=IDENT_CURRENT('version') WHERE
id_documento=@id_documento
```

NuevoAsignado

GO

```
CREATE PROCEDURE [dbo].[NuevoAsignado](@doc AS Image,
@version AS int,@extension AS varchar(6), @id_documento AS INT,
@creacion_estado AS char(1),@revision_estado AS
char(1),@aprobacion_estado AS char(1), @historico_estado AS char(1),
@claves AS Text)
```

AS

```
INSERT INTO version (version, documento, id_documento, extencion)
values (@version, @doc, @id_documento, @extension)
```

```
INSERT INTO Traza (creacion, revision, aprobacion, historico,
id_documento, id_version, estado, bandera) values (@creacion_estado,
@revision_estado, @aprobacion_estado, @historico_estado,
@id_documento, IDENT_CURRENT('version'), 'R', 'P')
```

```
INSERT INTO notas (claves, id_documento) values (@claves,  
@id_documento)
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,  
bandera) VALUES ('sin observaciones en creacion', getDate(), 'R',  
@id_documento, 'P')
```

PublicarDocumento.

```
GO
```

```
CREATE PROCEDURE [dbo].[PublicarDocumento]
```

```
-- Add the parameters for the stored procedure here
```

```
(@observaciones AS Text, @estado AS Char(1), @id_documento  
AS int, @bandera AS Char(1))
```

```
AS
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,  
bandera) values (@observaciones,getDate(),@estado, @id_documento,  
@bandera)
```

```
UPDATE Traza SET bandera='A' WHERE id_documento=@id_documento
```

ReprobarDocumento.

```
GO
```

```
CREATE PROCEDURE [dbo].[ReprobarDocumento]
```

```
-- Add the parameters for the stored procedure here
```

```
(@observaciones AS Text, @estado AS Char(1), @id_documento  
AS int)
```

```
AS
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,  
bandera) values (@observaciones,getDate(),@estado, @id_documento, 'R')
```

```
UPDATE Traza SET bandera='R' WHERE id_documento=@id_documento
```

SubirDoc.

```
GO
```

```
ALTER PROCEDURE [dbo].[SubirDoc](@doc AS Image, @nombre AS  
VarChar(100), @tipo AS int, /*@creacion AS datetime,*/ @version AS int,  
@creacion_estado AS char(1),@revision_estado AS  
char(1),@aprobacion_estado AS char(1), @historico_estado AS char(1),  
@claves AS Text, @id_usuario_1 AS int, @rol_1 AS  
char(1),@id_usuario_2 AS int, @rol_2 AS char(1), @id_usuario_3 AS int,  
@rol_3 AS char(1),@id_usuario_4 AS int, @rol_4 AS char(1), @extension  
AS varchar(6),@observaciones AS Text, @fec_aprov AS datetime,  
@fec_rev AS datetime, @fec_hist AS datetime, @prioridad AS CHAR(1))  
AS
```

```
INSERT INTO Documentos (nombre, id_tipo, creacion, prioridad) values  
(@nombre, @tipo, Getdate(), @prioridad)
```

```
INSERT INTO version (version, documento, id_documento, extension)  
values (@version, @doc, IDENT_CURRENT('Documentos'), @extension)
```

```
INSERT INTO Traza (creacion, revision, aprobacion, historico,  
id_documento, id_version, estado, bandera) values (@creacion_estado,  
@revision_estado, @aprobacion_estado, @historico_estado,  
IDENT_CURRENT('Documentos'), IDENT_CURRENT('version'), 'R', 'P')
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,  
bandera)values (@observaciones, Getdate(),  
'R',IDENT_CURRENT('Documentos'), 'P')
```

```
INSERT INTO notas (claves, id_documento) values (@claves,  
IDENT_CURRENT('Documentos'))
```

```
INSERT INTO doc_plazos(id_documento, revision, aprobacion,
publicacion) values
(IDENT_CURRENT('Documentos'),CONVERT(DATETIME,
@fec_rev,103),CONVERT(DATETIME, @fec_aprov,103),
CONVERT(DATETIME, @fec_hist,103))
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_1, IDENT_CURRENT('Documentos'),
@rol_1)
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_2, IDENT_CURRENT('Documentos'),
@rol_2)
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_3, IDENT_CURRENT('Documentos'),
@rol_3)
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_4, IDENT_CURRENT('Documentos'),
@rol_4)
```

SubirDoc1

```
GO
```

```
ALTER PROCEDURE [dbo].[SubirDoc_1](@doc AS Image, @nombre AS
VarChar(100), @tipo AS int, /*@creacion AS datetime,*/ @version AS int,
@creacion_estado AS char(1),@revision_estado AS
char(1),@aprobacion_estado AS char(1), @historico_estado AS char(1),
@claves AS Text, @id_usuario_1 AS int, @rol_1 AS
char(1),@id_usuario_5 AS int, @rol_5 AS char(1), @extension AS
varchar(6),@observaciones AS Text, @fec_hist AS datetime, @prioridad
AS CHAR(1))
```

```
AS
```

```
INSERT INTO Documentos (nombre, id_tipo, creacion, prioridad) values
(@nombre, @tipo, Getdate(), @prioridad)
```

```
INSERT INTO version (version, documento, id_documento, extencion)
values (@version, @doc, IDENT_CURRENT('Documentos'), @extension)
```

```
INSERT INTO Traza (creacion, revision, aprobacion, historico,
id_documento, id_version, estado, bandera) values (@creacion_estado,
@revision_estado, @aprobacion_estado, @historico_estado,
IDENT_CURRENT('Documentos'),IDENT_CURRENT('version'), 'R', 'P')
```

```
INSERT INTO Historial_doc(observaciones, fecha, estado, id_documento,
bandera)values (@observaciones, Getdate(),
'R',IDENT_CURRENT('Documentos'), 'P')
```

```
INSERT INTO notas (claves, id_documento) values (@claves,
IDENT_CURRENT('Documentos'))
```

```
INSERT INTO doc_plazos(id_documento, revision, aprovacion,
publicacion) values (IDENT_CURRENT('Documentos'), Getdate(),
Getdate(),CONVERT(DATETIME, @fec_hist,103))
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_1, IDENT_CURRENT('Documentos'),
@rol_1)
```

```
INSERT INTO Usuarios_documentos(id_usuario, id_documento,
rol_usuario) values (@id_usuario_5, IDENT_CURRENT('Documentos'),
@rol_5)
```

I.1.1.21. Modelo de implementación

I.1.1.21.1. Diagrama de componentes

Gestionar Usuarios

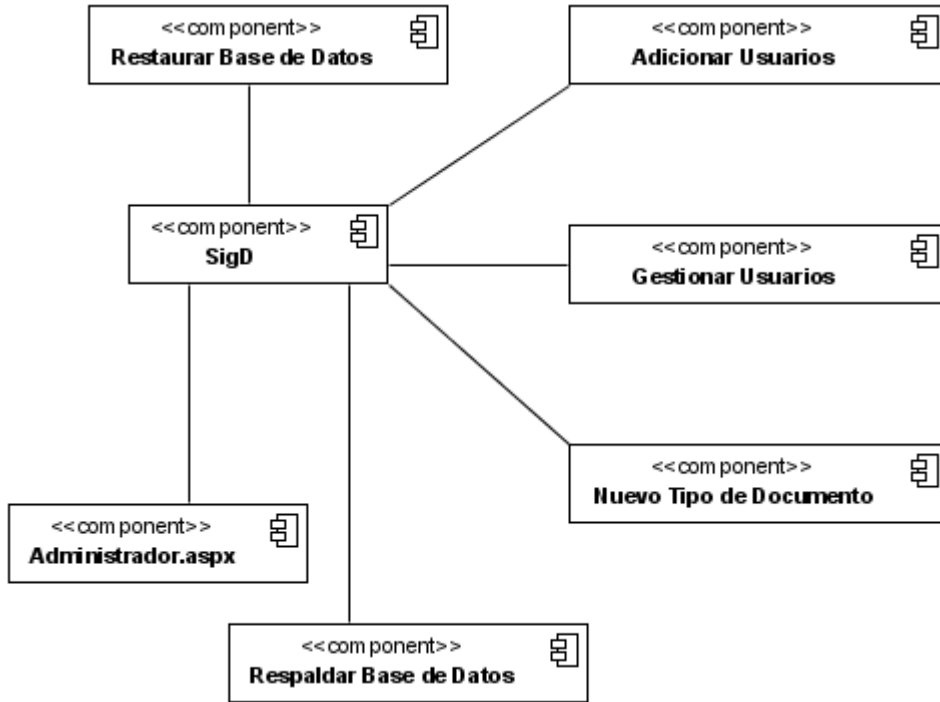


Figura 71: Diagrama de Componentes Gestionar usuarios

Gestionar Documentos

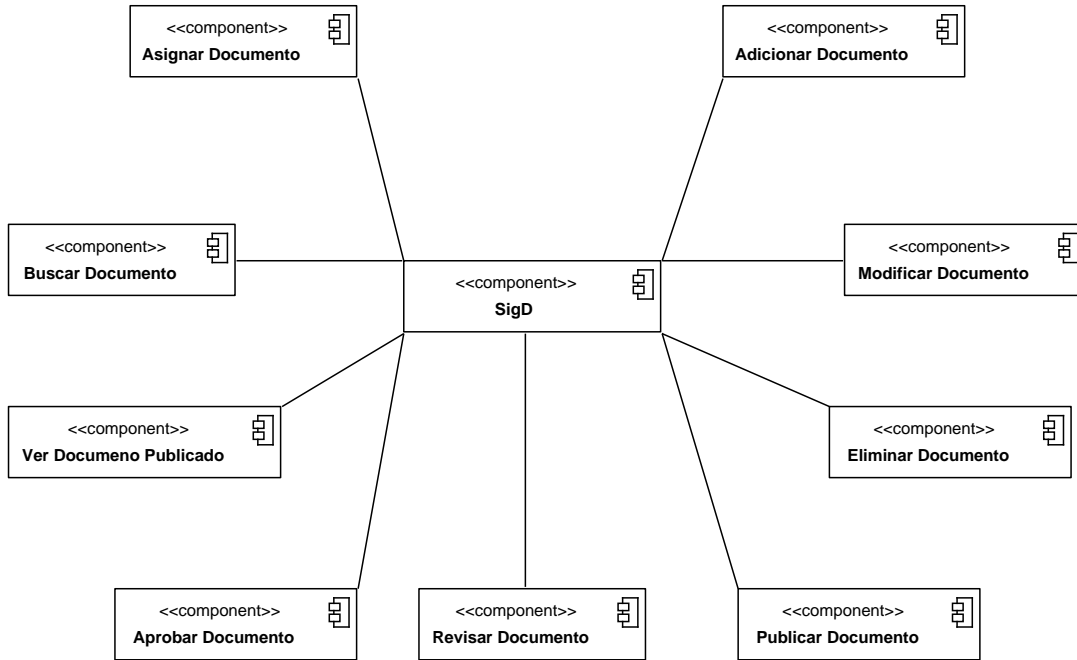


Figura 72: Diagrama de Componentes Gestionar documentos

I.1.1.22. Modelo de despliegue

I.1.1.22.1. Diagrama de despliegue

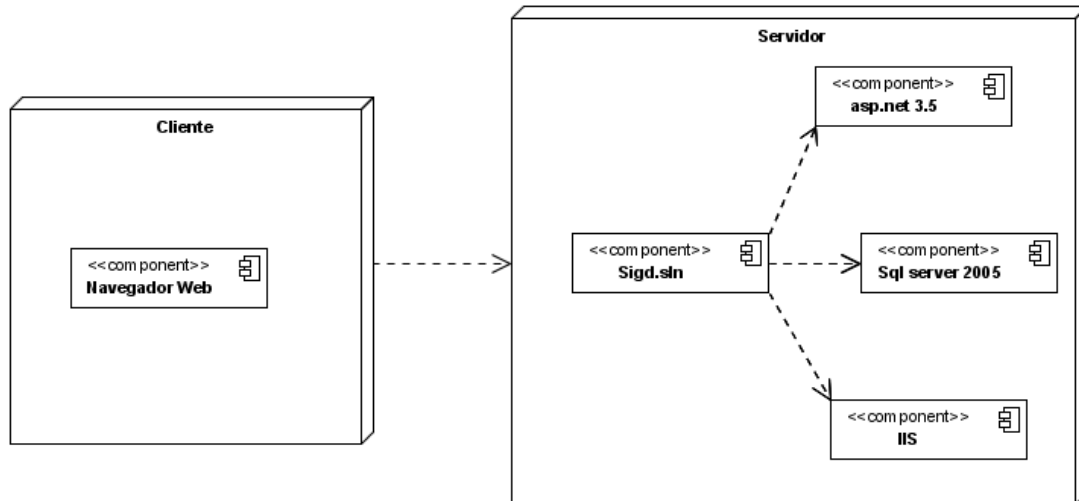


Figura 73: Diagrama de despliegue

I.2. Componente 2: Capacitación en el uso correcto del sistema de gestión documental (SigD).

I.2.1. Introducción

La “capacitación” es la educación profesional que busca adaptar al hombre para determinada empresa, a todos los niveles constituye una de las mejores inversiones en recursos humanos y una de las principales fuentes de bienestar para el personal de toda organización.

En el caso del sistema SigD, la capacitación es de vital importancia para el buen funcionamiento del mismo en la Corte Departamental Electoral de

Tarija. Esta se impartirá solamente al personal de que pertenece a la dirección de Informática, para que a partir de esta dirección se extienda al resto del personal de la CDE Tarija.

Aunque la capacitación auxilia a los miembros de la CDE Tarija a desempeñar su trabajo actual con el sistema, sus beneficios pueden prolongarse a toda su vida laboral y pueden auxiliar en el desarrollo de esa persona para cumplir futuras responsabilidades.

I.2.2. Propósito

Preparación al personal de la dirección de Informática para el uso del sistema SigD y el correcto uso del mismo.

I.2.3. Alcance

Esta capacitación abarcará los módulos de gestión de documentación, administrar el sistema, gestión de usuarios, y de datos personales, dejando una idea clara y concisa del uso correcto del sistema SigD, para que así dicho sistema logre cubrir las falencias en el tratamiento documental que tiene la Corte de Tarija.

I.2.4. Metodologías

Estudio de casos.- mediante el estudio de una situación específica o simulada la persona en capacitación aprende sobre las acciones que es deseable emprender en situaciones análogas. Para ello, cuenta con las sugerencias de otras personas, así como las propias. Además de aprender gracias al caso que se estudia, la persona puede desarrollar habilidades de toma de decisiones. Cuando los casos están bien seleccionados, poseen relevancia y semejanza con las circunstancias diarias, también hay cierre de transferencia. Existe también la ventaja de la participación mediante la discusión del caso. No es frecuente encontrar elementos de retroalimentación y de repetición.

Lectura, estudios individuales, introducción programada.- los materiales de instrucción para el aprendizaje individual resultan de gran utilidad en circunstancias de dispersión geográfica o de gran dificultad para reunir a un grupo de asistentes a un programa de capacitación.

Se emplean en casos en que el aprendizaje requiere poca integración (cursos basados en lecturas, grabaciones, fascículos de instrucción programada y ciertos programas de computadora).

Los materiales programados proporcionan elementos de participación, repetición, relevancia y retroalimentación. La transferencia tiende a ser baja.

Capacitación en laboratorio (sensibilización).- constituye una modalidad de la capacitación en grupo. Se emplea para desarrollar las habilidades interpersonales. Se puede utilizar también para el desarrollo de conocimientos, habilidades y conductas adecuadas para futuras responsabilidades laborales. Los participantes se postulan como objetivo en mejoramientos de sus habilidades en relaciones humanas mediante la comprensión de sí mismos y de las otras personas.

Instrucción directa sobre el puesto.- se imparte durante las horas de trabajo. Se emplea básicamente para asignar a obreros y empleados a desempeñar un puesto actual. La instrucción es impartida por un capacitador, supervisor o compañero de trabajo. En la mayoría de los casos el interés del capacitador se centra en obtener un determinado producto y en una buena técnica de capacitación.

Se distinguen varias etapas:

Se brinda a la persona que va a recibir la capacitación, una descripción general del puesto, su objetivo, y los resultados que se esperan de él.

El capacitador efectúa el trabajo a fin de proporcionar un modelo que se pueda copiar.

Se pide al individuo que imite el ejemplo. Las demostraciones y las prácticas se repiten hasta que la persona domine la técnica.

Se pide a la persona que lleve a cabo el ejercicio sin supervisión.

I.2.5. Aplicación

La capacitación del sistema SigD se realizó a través de las siguientes etapas:

- Etapa 1.- La planificación del plan de capacitación en coordinación con el personal de la dirección de informática, estableciendo fechas y horas para realizar la misma.
- Etapa 2.- En la primera etapa de la capacitación se proporciono manuales de instalación y funcionamiento del sistema SigD, a todo el personal de la dirección de informática de la Corte Departamental Electoral de Tarija.
- Etapa 3.- En esta etapa se dio una introducción a la gestión documental, beneficios y desventajas y la importancia de un sistema informático para controlar esta.
- Etapa 4.-Se procedió a explicar el sistema SigD, su instalación sobre una plataforma de Windows Server 2003, y los programas necesarios para el funcionamiento del mismo, considerando que el sistema es web, solo es necesaria la instalación en un servidor.
- Etapa 5.- Se procedió con la explicación del sistema mostrando a través de un proyector el funcionamiento del mismo, donde los participantes pudieron ver con distintos ejemplos como crear usuario y documentos, como respaldar el sistema, imprimir reportes, etc.
- Etapa 6.- En esta última etapa se procedió a la prueba del sistema, el personal de la dirección de informática a través de sus propios ordenadores hicieron uso el sistema SigD, donde los capacitadores realizaron una instrucción personalizada, persona por persona cubriendo las dudas generadas con el uso del sistema SigD.

I.3. Componente 3: Implementación del Sistema de Gestión Documental (SigD).

Introducción.

La implementación del proyecto es una etapa importante ya que depende de esta el buen funcionamiento del proyecto.

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

Proposito.

- Elaborar un diseño adecuado y eficaz para la implementación.
- Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.
- Integrar los componentes en el sistema siguiendo un enfoque incremental.

Software Complementario.

- Crystal Reports.
- Internet Information Services.
- Sql Server 2005.
- Windows Server 2003.
- Ajax Tool Kit.

Resumen de Actividades

Para ejecutar esta tarea se deben tener en cuenta las características propias de la CDE Tarija y los recursos existentes.

La implementación del sistema SigD dentro de la granja de servidores de la Corte Departamental Electoral, se realizo preparando el servidor web de la misma, con los programas necesarios para la ejecución del Sistema Informático de Gestión

Documental (SigD), como ser la instalación del framework 3.5, Crystal Reports para la generación de reportes, y finalmente complementos ajax para la ejecución correcta del SigD.

Posteriormente se procedió a cargar los archivos necesarios del sistema SigD a la carpeta wwwroot carpeta utilizada por defecto por el Internet Information Services(IIS) servidor de paginas web que utiliza por defecto el Windows server 2003. Para crear un sitio web se necesita la herramienta de configuración del IIS, para que así, dentro del dominio de la CDE Tarija, puedan acceder al sistema solo poniendo la dirección en la barra del navegador.

Para restaurar la base de datos utilizamos el Sql Server Management Studio, herramienta de gestión de bases de datos utilizada por el Sql Server en su versión 2005, siendo esta la versión que utiliza la CDE Tarija para alojar las bases de datos de sus diferentes sistemas, desde ahí escogimos la base de datos “sigd” y la restauramos.

A partir de ese momento el sistema ya es utilizable entrando con una cuenta de administrador, para gestionar los diferentes usuarios, y se pueda empezar a utilizar el sistema SigD.

I. Conclusiones y Recomendaciones

I.1. Conclusiones

La gestión documental sin duda alguna, permite aumentar drásticamente la productividad organizacional mediante la implantación de un Sistema de Gestión Documental Informático.

Entre las mejoras más claras que supone hacer disponible la Gestión Documental mediante un sistema de este tipo, se cuenta lo siguiente:

- Reducción del tiempo de consulta en documentos electrónicos, al estar estos perfectamente categorizados e indexados, mediante atributos o descriptores que los identifican y evitan su pérdida, a la vez que optimizan su búsqueda.
- Reducción de los costos de archivado: en el caso de documentos electrónicos solo hace falta categorizarlos para que estos se archiven de forma transparente para que estos se archiven de forma transparente para el usuario. En el caso de documentos que tienen su versión original en papel, y que deben archivarse en un archivo físico por cuestiones legales, en el momento en que son digitalizados se produce una reducción de costos en varios sentidos:
 1. No se precisa su manipulación física (ya que están escaneados), con lo que el archivado y acceso al documento es mucho más rápido.
 2. No se deterioran ni pierden documentos, al no manipularse manualmente.
- Acceso concurrente a un documento: disponer de los documentos dentro de un Sistema de Gestión Documental nos permite que

varios empleados puedan acceder simultáneamente a un mismo documento, lo cual es imposible con el documento físico (en este caso hay que esperar a que el usuario A termine de utilizar un documento para que B pueda hacerlo también).

- Incremento en la satisfacción de los usuarios internos (miembros de la organización) por agilizar sus procesos de trabajo.

I.2. Recomendaciones

Es indudable que un sistema de este tipo aporta innegables ventajas a una organización, aumentando la calidad y la eficacia de su gestión, pero es igualmente indudable que supone un impacto que hay que saber “gestionar” correctamente.

La implantación de un Sistema Informático de Gestión Documental implica un cambio en la forma de trabajar. Como todo cambio cultural, generara inicialmente una resistencia y requerirá de un tiempo y un esfuerzo para adaptarse. Para minimizar esta resistencia hay tres factores fundamentales a tener en cuenta:

- Las diferentes direcciones deben comunicar a sus funcionarios, su decidido apoyo al nuevo sistema como elemento clave de mejora estratégica y de mejora de la competitividad.
- Que el sistema sea sencillo de manejar. El tiempo de formación y adaptación a la herramienta debe ser el mínimo posible.
- El funcionario debe percibir claramente que la implantación del nuevo sistema le aporta ventajas tangibles en su trabajo diario.

A cambio de esta resistencia inicial, la organización percibirá un impacto positivo que en grandes líneas viene determinado por:

- Aumento de la productividad.
- Utilización de una metodología eficiente y homogénea en el tratamiento de la documentación.
- Seguridad y protección de la documentación.
- Posibilidad de extraer “conocimiento” de la documentación.
- Ahorro en costos de archivo.