

03063

Lij



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

INSTITUTO DE INVESTIGACIONES
EN MATEMATICAS APLICADAS Y EN SISTEMAS

UN PROTOTIPO PARA EL CONTROL DE LOS
DOCUMENTOS DE ADMINISTRACION DE LA
CALIDAD EN EMPRESAS PEQUEÑAS

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACION

P R E S E N T A :

CARLOS JAVIER PEREZ ESCOBAR

DIRECTORA: DRA. HANNA OKTABA

MEXICO, D. F.

SEPTIEMBRE DE

**TESIS CON
FALLA DE ORIGEN**

1999
272866



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

"En cada grano de arena hay un derrumbamiento de montaña".

D. M. Loynaz

A mis padres por enseñarme el camino de la vida y a mis tíos por permitirme llegar al lugar donde estoy.

A la Dra. Hanna Oktaba , mi tutora y guía, por su enorme apoyo y motivación. Llegar hasta aquí es el fruto de su dedicación y consejo.

Al personal de la Maestría en Ciencias de la Computación por su paciencia y apoyo, mi agradecimiento infinito a Lulú y a Juanita.

A mis amigos de la maestría que me apoyaron y enseñaron a entender y asimilar la cultura de este país.

A mis compañeros de Tecnosys, en especial de la Gerencia de Ingeniería de Procesos de Software, por su empeño y presión constante para que concluyera mi trabajo.

Muchas gracias a todos por su "grano de arena",

Carlos Javier Pérez Escobar.

Contenido

INTRODUCCIÓN.....	1
I. LOS MODELOS Y NORMAS DE CALIDAD	5
1.1. Conceptos generales	5
1.2. La norma ISO 9000.....	8
1.2.1. Estructura de la norma	9
1.3. El modelo CMM.....	13
1.3.1. Estructura del modelo.....	15
1.4. Otros modelos de interés.....	19
1.4.1. TickIT.....	20
1.4.2. Bootstrap.....	22
1.4.3. Trillium	23
1.4.4. ISO/SPICE.....	24
II. LA CREACIÓN DE LA DOCUMENTACIÓN PARA LA ADMINISTRACIÓN DEL SISTEMA DE CALIDAD	27
2.1. El manual de administración de la calidad	27
2.1.1. Características del manual de administración de la calidad.....	28
2.1.2. Estructura general del manual de administración de la calidad.....	29
2.1.2.1. Políticas de calidad.....	30
2.1.2.2. Procesos	31
2.1.2.3. Procedimientos.....	32
2.1.2.4. Formas de registro.....	33
2.1.2.5. Roles.....	33
2.2. Herramientas de software para el apoyo a la administración de la calidad.....	34
III. CONTROL DE LOS DOCUMENTOS DEL SISTEMA DE CALIDAD.....	37
3.1. El control de los documentos.....	37
3.2. Los requisitos establecidos para el control de los documentos.....	39
3.2.1. Los requisitos de ISO 9000.....	39
3.2.2. Los requisitos de CMM.....	41

3.3. Elementos a considerar para el control de los documentos.....	43
IV. ESPECIFICACIONES DEL ANÁLISIS Y DISEÑO	47
4.1. Consideraciones generales	47
4.1.1. Requisitos del sistema	48
4.1.2. Actores.....	50
4.2. Resultados del análisis	51
4.2.1. Clases resultantes del análisis.....	52
4.2.2. Casos de uso.....	52
4.2.3. Diagramas de estado	54
4.2.4. Diccionario de clases.....	55
4.3. Resultados del diseño	55
4.3.1. Arquitectura del sistema.....	56
4.3.2. Clases resultantes del diseño	58
4.3.3. Interacción de los actores con el sistema.....	60
4.3.4. Diseño de la base de datos.....	61
CONCLUSIONES Y RECOMENDACIONES	63
BIBLIOGRAFÍA	67
Sitios en Internet.....	69
ANEXOS.....	71
A. Herramientas de software para el apoyo en la administración de la calidad	
B. Documento de análisis	
C. Documento de diseño	

Introducción

Está demostrado que el factor principal para el adecuado funcionamiento de una organización es la calidad o la capacidad de satisfacer las necesidades del cliente en los productos o servicios que oferta. Además, el continuo mejoramiento de éstas capacidades es necesario para obtener y mantener un buen desempeño económico.

En la actualidad existe una creciente tendencia mundial, en diversos sectores productivos, enfocada hacia la solución de los problemas relacionados con la calidad y la forma de mejorar estos, unido a un aumento de las expectativas de los clientes respecto a la calidad de los productos o servicios que adquieren.

Para las empresas dedicadas a la producción de software existen varios modelos o normas que se pueden adoptar para alcanzar un adecuado nivel de calidad. Entre los cuales se destacan los modelos ISO 9000 y CMM como los más reconocidos y utilizados.

La creación del manual de calidad o el conjunto de documentos que sirven de base para el proceso de aseguramiento de la calidad de los productos o servicios, independientemente del modelo que se seleccione, requiere de un gran esfuerzo, en tiempo y personal, además del costo económico por la cantidad de documentos que es necesario generar y poner a disposición de las personas involucradas en la calidad. Al mismo tiempo es necesario que sea el propio personal relacionado con la producción o con los servicios el que cree sus propios manuales y así pueda garantizarse la aceptación y utilización de lo que en ellos se plantea.

No existe fórmula para crear el conjunto de políticas, procesos, procedimientos, guías y estructura organizativa que integran un sistema de calidad. Cada organización debe crear el suyo propio, puesto que un sistema impuesto en lugar de mejorar la eficiencia de la

organización se puede convertir en esquema que fuerza a los que lo aplican a hacer lo que no desean y que carece de todo sentido, además de que una auditoría detectaría inmediatamente las fallas puesto que se dice una cosa y el personal trabaja de otra forma.

El sistema de calidad que se establece en una empresa está influenciado por los objetivos de la organización, por los servicios o productos que oferta y por las características de las prácticas específicas que utiliza lo que le confiere una distinción respecto a las demás organizaciones y es particular a la empresa. Asimismo el modelo que se utilice como norma o base para la creación del sistema de calidad es interpretado de diferentes formas por cada empresa, aunque define puntos que son necesarios cumplir en todas las organizaciones.

Para garantizar que todos tengan acceso a los documentos necesarios y adecuados, la primera actividad que se necesita realizar es establecer un mecanismo para el control de estos y sobre esta tarea se centra el objetivo principal del trabajo.

Las bases para este proyecto surgen de las experiencias adquiridas en el curso de Ingeniería de Software Orientada a Objetos impartido en la maestría por la Dra. Hanna Oktaba y la M.C. Ma. Guadalupe Ibarquengoitia González. En este curso se creó un equipo de trabajo donde se simularon y desempeñaron roles que existen en proyectos reales de desarrollo de software y que en mi caso estuvo relacionado con el aseguramiento de la calidad. Además se estuvieron investigando las peculiaridades que se presentan al realizar esta actividad en una empresa de tamaño pequeño dedicada al desarrollo de software.

El presente trabajo cubre en el Capítulo I los términos utilizados comúnmente y las normas y métodos que actualmente se aplican, para una mejor comprensión del trabajo y una correcta implantación del sistema de calidad. El estudio se centrará en el análisis de ISO 9000, que es el estándar de calidad más difundido, y CMM como uno de los modelos más completos para procesos de desarrollo de software.

El Capítulo II brinda un análisis de la estructura de un sistema de calidad, las características de los documentos que lo componen y la forma en que se organizan. Además se hace un análisis de algunas de las herramientas existentes que ayudan en el proceso de creación y administración del sistema de calidad

El Capítulo III se concentra en los requisitos y principios necesarios para las actividades relacionadas con el control de la documentación, basado en lo que establece ISO 9000 y CMM.

El último capítulo presenta las características del análisis y diseño orientado a objetos de un prototipo que soporte el control de los documentos apoyado por documentos en formato *html* de forma tal que se garantice el acceso a ellos de manera distribuida.

Capítulo I

I. Los modelos y normas de calidad

Esta sección tratará algunos conceptos fundamentales para facilitar la comprensión de los términos empleados y cubre aspectos generales de la norma ISO 9000, el modelo CMM y otros modelos importantes.

1.1. Conceptos generales

El término calidad lo podemos encontrar con diferentes enfoques en distintos contextos, así tenemos:

- **Calidad** es el grado en el que un sistema, componente o proceso cumple con los requisitos establecidos o con las necesidades o expectativas del cliente o usuario. [Pauk93, p.178]
- **Calidad** es el conjunto de características de un producto o servicio que les permiten satisfacer necesidades implícitas o explícitas. [Oskarsson96, p. 85]

En todos el elemento principal es que independientemente de la actividad desarrollada se necesita conocer las necesidades de quien va a utilizar el producto o servicio para poder actuar en función de éstas. Para ello es necesario identificar, definir y documentar las necesidades o requisitos de un producto o servicio y para lograrlo de forma adecuada es imprescindible establecer procesos, procedimientos, guías y planes que permitan desarrollar un producto o servicio con calidad de manera consistente. Los siguientes conceptos pueden ayudar a comprender mejor lo que se ha planteado.

- **Producto** es el resultado de una actividad o proceso que puede ser tangible o no. En el caso del software es el conjunto completo, o parte del conjunto, de programas para

computadoras, procedimientos y documentos y datos asociados diseñados para su entrega al cliente o usuario final. [Paulk 93, p.180]

- **Proyecto** es un compromiso donde se acuerda un esfuerzo, en tiempo y personal, enfocado al desarrollo y/o mantenimiento de un producto. Típicamente tiene asociado un costo, los fondos y el calendario de desarrollo. Para el caso del software un proyecto abarca el análisis, especificación, diseño, implantación, prueba y/o mantenimiento de los componentes del software y documentos asociados del sistema. [Humphrey97, p.48][Paulk93, p. 178, 180]
- **Procedimiento** es una descripción del orden de actividades o tareas que se realizan para alcanzar un objetivo determinado y que indican cómo se hacen las cosas. [Paulk93, p.177][Schmauch95, p.71]
- **Proceso** define una secuencia o flujo de tareas desarrolladas durante la producción de un producto o servicio y que describe qué se hace. El proceso de software es el conjunto de actividades, métodos, prácticas y transformaciones necesarias para desarrollar y mantener el software y los productos asociados. [Paulk93, p. 177, 180] [Schmauch95, p.71]
- **Plan** describe la forma específica en que se realiza un proyecto: cómo, cuándo y a qué costo. [Humphrey97, p. 48]

Asociados al término calidad existen otros muy utilizados y que es necesario identificar y aplicar.

- **Sistema de calidad** es la estructura organizativa, responsabilidades, procedimientos, procesos y recursos para implementar la administración de la calidad de acuerdo a las políticas de la empresa. Es el conjunto de procedimientos, procesos, personal, gerentes, herramientas y facilidades que influyen en la calidad de un producto o servicio. Se usa este término de manera indistinta junto a Sistema de aseguramiento de la calidad y Sistema de administración de la calidad. [Cadena96, p. 28] [Oskarsson96, p. 85] [Schmauch95, p. 72]
- **Política de calidad** es el conjunto de metas y objetivos generales de una organización en cuanto a calidad, expresadas formalmente por la máxima dirección. [Cadena96, p.28]

- **Manual de calidad** o manual de administración de la calidad es el documento o conjunto de documentos que formaliza y describe todas las actividades que realiza la organización para garantizar la calidad. Contiene las políticas y planes de calidad, procesos, procedimientos y registros que forman parte del sistema de calidad. [Cadena96, p.28]
- **Plan de calidad** es el documento que especifica las prácticas, recursos y secuencia de actividades relacionadas con la calidad de un producto, proyecto o contrato. [Cadena96, p.28]

Además existen algunas actividades relacionadas con la calidad y que es igualmente necesario identificar y conocer.

- **Aseguramiento de la calidad** es el conjunto de actividades, desarrolladas dentro del proceso de producción, necesarias para asegurar y garantizar que una organización obtiene un determinado nivel de calidad en el producto o que cumple con los requisitos establecidos. [Cadena96, p.28] [Schmauch95, p.72]
- **Control de la calidad** son las actividades y técnicas usadas al obtener el producto final para garantizar el cumplimiento de los requisitos de calidad del producto que se está desarrollando. [Schmauch95, p.72]
- **Administración de la calidad** es una parte del conjunto de todas las funciones administrativas que determinan e implementan la política de calidad. [Schmauch95, p.72]
- **Auditoría de calidad** es el examen metódico e independiente para determinar si las actividades de calidad son realizadas de acuerdo a lo planeado y si cumplen su objetivo. [Cadena96, p.28]
- **Certificación, Evaluación o Registro** es la evaluación del sistema de calidad, por parte de una entidad que es acreditada por alguna organización normalizadora, para confirmar el cumplimiento y aplicación de lo establecido en los estándares. [Schmauch95, p.69, 73]

Una organización que busca lograr la calidad debe enfocarse en tres objetivos fundamentales que son:

- 1) La organización debe obtener y mantener la calidad de los productos o servicios que produce para cumplir continuamente con las necesidades establecidas o indicadas por el comprador.
- 2) La organización debe dar la confianza a la administración de que la calidad propuesta está siendo obtenida y mantenida.
- 3) La organización debe dar la confianza al comprador de que la calidad propuesta está siendo, o será, obtenida en la entrega del producto o servicio suministrado.

En los puntos siguientes se presentan de forma general algunas de las normas y modelos de calidad existentes.

1.2. La norma ISO 9000

ISO¹ es la Organización Internacional para la Estandarización, fundada el 23 de febrero de 1947 para promover el desarrollo de estándares internacionales. [Cadena96, p.27] ISO está compuesta por representantes de los cuerpos normalizadores de más de 90 países. El trabajo de esta organización cubre todas las áreas excepto las relacionadas con la ingeniería eléctrica y la electrónica, que son cubiertas por la Comisión Electrotécnica Internacional (IEC). Los resultados del trabajo técnico de ISO son publicados en forma de estándares internacionales o guías.

El Comité Técnico ISO/TC 176 para aseguramiento de la calidad fue el encargado de crear el estándar ISO 9000, que es una serie de estándares internacionales para los sistemas de aseguramiento de la calidad. Establece un conjunto básico de requisitos para el sistema de calidad necesario para asegurar que el proceso es capaz de producir de manera consistente productos que cumplen con las expectativas del cliente.

¹ Aunque las letras de ISO corresponden con las iniciales de la organización, no es un acrónimo. ISO es tomado del griego *isos* que significa igualdad y es el prefijo de todas las normas creadas por esta organización. [Cadena96, p. 27]

La serie ISO 9000 de estándares para sistemas de calidad está basada en la premisa que plantea que si la producción y la administración es correcta, el producto o servicio que se produce también lo será. [Schmauch95, p. 4]

El estándar establece guías que describen qué debe hacerse como mínimo para lograr cumplir con los requisitos, pero no especifica cómo se debe hacer. De manera general estos requisitos establecen:

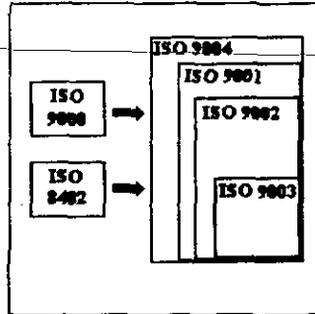
- Escribir un manual de calidad, describiendo el sistema de calidad a un alto nivel.
- Escribir documentos en forma de procedimientos que describan cómo debe hacerse el trabajo en la organización.
- Crear un sistema para controlar la distribución y reedición de documentos.
- Diseñar e implantar un sistema de acciones preventivas y correctivas para prevenir y corregir los problemas que puedan surgir .
- Identificar las necesidades en cuanto a capacitación en la organización.
- Determinar las medidas y equipos para realizar las pruebas.
- Capacitar al personal de la organización en la operación del sistema de calidad.
- Planificar y llevar a cabo auditorías internas de calidad.
- Tener en cuenta los requisitos del estándar con los que no cumple la organización.

1.2.1. Estructura de la norma

La serie ISO 9000 está formado por guías (ISO 9000, ISO 9004) y normas contractuales o modelos (ISO 9001, ISO 9002, ISO 9003) que se aplican a la administración y aseguramiento de la calidad.² Estos modelos son genéricos y no son específicos a un producto en particular. Además existe un estándar ISO 8402 que contiene las definiciones y

² La serie ISO 9000 se adopta en E.U. como ANSI/American Society for Quality Control (ANSI/ASQC) Q 9000. En Europa se asume como la serie European Norm (EN) 29000. Otros países están adoptando su propia nomenclatura para los estándares. [Schmauch95, p. 3]

terminologías empleadas en toda la serie. La estructura de la norma se muestra en la ilustración 1.



• Ilustración 1 Estructura de la norma ISO 9000

La identificación de los estándares principales que constituyen la norma ISO 9000 es la siguiente:

- *ISO 9000, Estándar para el aseguramiento y la administración de la calidad. Guía para la selección y uso.* Explica los conceptos de calidad fundamentales, define términos claves y ofrece una guía para la selección, uso y aplicación de ISO 9001, 9002, 9003.
- *ISO 9001, Sistemas de calidad. Modelo para el aseguramiento de la calidad en el diseño, desarrollo, producción, instalación y servicio.* Es el estándar más amplio de la serie. Esta formado por 20 puntos que incluyen todos los planteados en ISO 9002 y 9003 más los que están enfocados al diseño, desarrollo y servicio.
- *ISO 9002, Sistemas de calidad. Modelo para el aseguramiento de la calidad en la producción e instalación.* Contiene 19 puntos enfocados a la prevención, detección y corrección de problemas durante la producción e instalación. Es más extenso y sofisticado que ISO 9003.
- *ISO 9003, Sistemas de calidad. Modelo para el aseguramiento de la calidad en las inspecciones y pruebas finales.* Es el estándar más pequeño, formado por 16 puntos dirigidos a la detección y control de problemas durante las inspecciones y pruebas finales.

- *ISO 9004, Elementos del sistema de calidad y de administración de la calidad. Guías.*
 Brinda una guía para el desarrollo e implantación de un sistema de calidad y para determinar la extensión a la que es aplicable cada punto del sistema de calidad. Examina cada uno de los puntos del sistema de calidad en detalle y puede ser usado para realizar auditorías internas y externas.

Puntos de los estándares ISO 9000	ISO 9001	ISO 9002	ISO 9003
4.1 Responsabilidad de la dirección	*	*	*
4.2 Sistema de calidad	*	*	*
4.3 Revisión de contratos	*	*	**
4.4 Control de diseño	*		
4.5 Control de documentos y datos	*	*	*
4.6 Control de adquisiciones	*	*	
4.7 Control de productos suministrados por los clientes	*	*	**
4.8 Identificación y seguimiento de productos	*	*	*
4.9 Control de procesos	*	*	
4.10 Inspección y pruebas	*	*	*
4.11 Control del equipo de inspección, medición y pruebas	*	*	*
4.12 Estado del producto en función a inspección y pruebas	*	*	*
4.13 Control de productos no conformes	*	*	*
4.14 Acciones preventivas y correctivas al sistema de calidad	*	*	**
4.15 Manejo, almacenamiento, empaque, conservación y entrega	*	*	*
4.16 Control de registros de calidad	*	*	*
4.17 Auditorías internas de calidad	*	*	**
4.18 Programas de capacitación	*	*	*
4.19 Control del servicio al cliente	*	** ³	
4.20 Aplicación de técnicas estadísticas	*	*	*

Tabla 1 Puntos contenidos en los estándares ISO 9001, 9002, 9003

Una lista completa de las normas vigentes de aseguramiento de la calidad publicadas por ISO se puede encontrar en [Cadena96, p.29]. Los puntos que integran cada modelo se pueden observar en la tabla 1.

ISO 9000 fue en un principio pensado para la industria manufacturera y su aplicación al proceso de desarrollo de software es un caso especial, por este motivo ISO publicó en 1991 *ISO 9000-3, Guías para la aplicación de ISO 9001 al desarrollo, suministro, instalación*

³ Estos puntos fueron incluidos después de la revisión a la norma realizada en 1994.

y mantenimiento del software. ISO 9000-3 es una guía⁴ para la aplicación de ISO 9001 al desarrollo de software y debe ser considerada y analizada en el momento de diseñar un sistema de calidad para software, pero como bien indica es una guía y la certificación y registro de un sistema de calidad es con base en la norma y no en la guía. Esto quiere decir que pueden existir puntos de la guía que no se cumplan o apliquen al sistema de calidad pero los de la norma deben satisfacerse o en su defecto argumentar por qué no se aplican.

ISO 9000-3 está formado por 22 cláusulas, que no se corresponden directamente con los 20 puntos de ISO 9001, agrupadas en tres grandes grupos o secciones como sigue:

- *Sección 4, Sistema de calidad - Estructura*, relacionada con las actividades independientes de un proyecto o etapa pero que existen en todos los proyectos.
- *Sección 5, Sistema de calidad - Actividades del ciclo de vida*, tiene que ver con las actividades que están relacionadas con una fase particular del proceso de desarrollo de un proyecto.
- *Sección 6, Sistema de calidad - Actividades de apoyo*, enfocado a las actividades que no están relacionadas con una etapa en particular pero que se aplican en todo el proceso.

En la tabla 2 se muestran los elementos de ISO 9000 - 3 y su relación con los elementos de ISO 9001. [Schmauch95, p. 124]

Actualmente está en fase de revisión la norma ISO para el año 2000. En esta nueva versión desaparecen las 3 normas contractuales anteriores y quedan integradas en la 9001 que será consistente con cada uno de los puntos de la 9004. La nueva norma estará estructurada en 4 macro procesos que son: Responsabilidad Gerencial, Administración de Proyectos, Recursos y Servicios, y Mejora Continua; los que contendrán los 20 elementos de la norma anterior con algunas mejoras adicionales que se incluyen para facilitar la interpretación de la norma en organizaciones que no están dedicadas a la manufactura como es el caso de las empresas de desarrollo de software.

⁴ ISO 9000 - 3 no es claro en el sentido de si sólo es una guía de interpretación o si en realidad se están agregando

PUNTOS DE ISO 9000-3	PUNTOS DE ISO 9001
4.1 Responsabilidad de la dirección	4.1
4.2 Sistema de calidad	4.2
4.3 Auditorías internas de calidad	4.17
4.4 Acciones correctivas	4.14
5.2 Revisión de contratos	4.3
5.3 Especificación de los requisitos del comprador	4.3, 4.4
5.4 Plan de desarrollo	4.4
5.5 Plan de calidad	4.2, 4.4
5.6 Diseño e implantación	4.4, 4.9, 4.13
5.7 Prueba y validación	4.4, 4.10, 4.11, 4.13
5.8 Aceptación	4.10, 4.15
5.9 Copiado, entrega e instalación	4.10, 4.13, 4.15
5.10 Mantenimiento	4.13, 4.19
6.1 Administración de la configuración	4.4, 4.5, 4.8, 4.12, 4.13
6.2 Control de documentos	4.5
6.3 Registros de calidad	4.16
6.4 Mediciones	4.20
6.5 Reglas, prácticas y convenciones	4.9, 4.11
6.6 Herramientas y técnicas	4.9, 4.11
6.7 Compras	4.6
6.8 Productos de software incluidos	4.7
6.9 Capacitación	4.18

• Tabla 2 Relación de la guía ISO 9000-3 con el modelo ISO 9001

1.3. El modelo CNM

A principios de los años 80 el Departamento de Defensa de los Estados Unidos enfocó sus tareas a la revisión de los problemas del software y a su mejoramiento. Para contribuir a este programa se creó el Instituto de Ingeniería de Software (SEI) a finales de 1984. Como parte de su trabajo, el Instituto se dio a la tarea de desarrollar el Modelo de Madurez del Proceso de Software y para 1986 se comenzó el Proyecto de Evaluación de la Capacidad del Software. Después de varios años de realizar cuestionarios, evaluaciones, consultas e investigaciones, junto a otras organizaciones, en 1991 SEI produce el Modelo de Madurez de la Capacidad del Proceso de Software. [Humphrey93, p. 7]

factores como la administración de la configuración y el control de proyectos, que no aparecen en el modelo ISO 9001, como requisitos adicionales. [Cadena86-2]

El Modelo de Madurez de la Capacidad del Proceso de Software (CMM) permite determinar la capacidad de las organizaciones de desarrollo de software para producir de manera consistente y predecible productos de calidad superior. El modelo brinda guías para seleccionar estrategias de mejoramiento del proceso mediante la determinación de las capacidades actuales del proceso y la identificación de los puntos críticos para mejorar el proceso y la calidad del software. [Paulk93, p. 171]

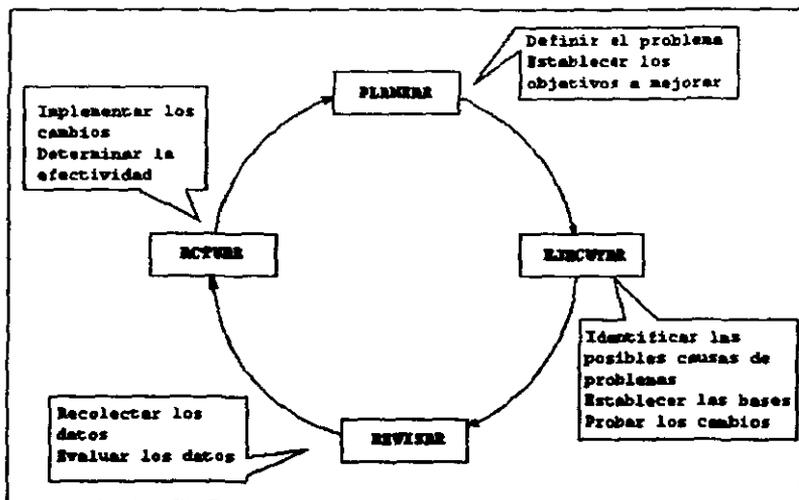
La *capacidad del proceso*⁵ es la habilidad inherente de un proceso para producir los resultados planeados. El principal objetivo de un proceso de software maduro es el de producir productos de calidad que cumplan los requisitos del usuario. Cuando se habla de *madurez del proceso*⁶ se entiende como el crecimiento alcanzado en la capacidad del proceso de software y que se considera como una actividad a largo plazo.

En una organización de software inmadura el proceso de software es generalmente improvisado, no existen planes rigurosos, sus actividades se enfocan en resolver las crisis que se presentan, carecen de bases objetivas para evaluar la calidad de los productos o para resolver los problemas que surgen. Por lo contrario, cuando la organización alcanza cierto grado de madurez posee una gran habilidad para administrar el proceso de desarrollo y mantenimiento del software, se hacen pruebas y análisis de costo-beneficio para mejorar el proceso, el administrador monitorea la calidad del producto y la satisfacción del cliente, se llevan registros y todos los integrantes están involucrados en el proceso de desarrollo.

Las bases para el trabajo de mejoramiento del proceso se proponen en el ciclo Shewhart. Este consta de 4 pasos que se repiten en forma de ciclo hasta que la implantación produce los resultados esperados y los cambios pasan a ser permanentes. [Humphrey93, p. 3] La ilustración 2 muestra los pasos que forman el ciclo.

⁵ La capacidad del proceso de software describe el rango de resultados esperados que se obtienen siguiendo un proceso de software, mientras que el desempeño del proceso de software representa los resultados reales obtenidos.

⁶ La madurez del proceso de software está dada cuando un proceso en específico es explícitamente definido, administrado, medido, controlado y es efectivo.



• Ilustración 2 Pasos del ciclo Shewhart

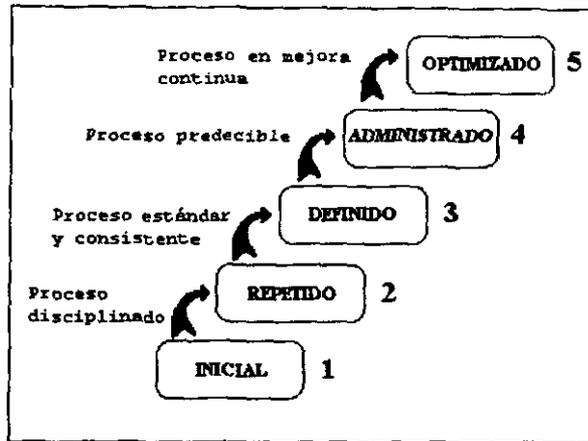
Basado en estos principios se creó el modelo CMM que permite obtener un incremento gradual en la capacidad del proceso. El modelo describe los principios y prácticas relacionadas con la madurez del proceso de software y propone ayudar a las organizaciones dedicadas al desarrollo del software a alcanzar la madurez de su proceso de software en términos del tránsito evolutivo desde un proceso improvisado y caótico a uno maduro con una adecuada disciplina y mayor capacidad.

CMM es un modelo descriptivo en el sentido que describe los atributos esenciales que se esperan caractericen una organización dentro de un nivel de madurez en particular. Es un modelo normativo ya que las prácticas detalladas caracterizan el tipo normal de comportamiento que se espera de una organización que realiza proyectos a gran escala. No es prescriptivo ya que no dice a la organización cómo mejorar.

1.3.1. Estructura del modelo.

El modelo consta de 5 niveles, diseñados de manera que los niveles inferiores proveen las bases para que de forma progresiva se alcancen los superiores. Estas 5 etapas de desarrollo son referidas como niveles de madurez y en cada uno la organización alcanza

una capacidad superior del proceso. En la ilustración 3 se muestra la estructura de los niveles del modelo.



• Ilustración 3 Niveles de madurez del proceso de software [Pauk93,p.5]

- 1) **Inicial:** El proceso de software es un proceso improvisado y caótico. Pocos procesos están definidos y el éxito que se pueda obtener depende de las habilidades, conocimientos y motivaciones del personal. No existen calendarios ni estimados de costos y la funcionalidad y calidad del producto es impredecible. No existe un ambiente estable para el desarrollo y mantenimiento del software. El proceso del software es impredecible por el continuo cambio o modificación a medida que avanza el trabajo.
- 2) **Repetido:** Se establecen procedimientos de administración del proceso que son básicos para determinar costos, calendarios y funcionalidad. Se establecen las políticas para la administración del proceso y los procedimientos de implantación. El proceso se basa en repetir éxitos anteriores en proyectos de similares características, por lo que los mayores riesgos se presentan cuando se enfrentan a nuevos proyectos. Existen problemas de calidad y no hay una adecuada estructura para mejorarla.
- 3) **Definido:** El proceso de software para las actividades administrativas y técnicas está documentado, estandarizado e integrado en un proceso de software estándar dentro de la organización que ayudará a obtener un desempeño más efectivo. El grupo que

trabaja en el proceso enfoca y guía sus esfuerzos al mejoramiento del proceso, facilita la introducción de técnicas y métodos e informa a la administración del estado del proceso. La capacidad del proceso está basada en una amplia comprensión común dentro de la organización de las actividades, roles y responsabilidades definidas en el proceso de software.

- 4) **Administrado:** Se recolectan medidas detalladas del proceso de software y de la calidad del producto. Ambos son cuantitativamente entendidos y controlados. El ciclo de Shewhart es constantemente utilizado para planear, implementar y registrar las mejoras al proceso. Este nivel de capacidad del proceso permite a la organización predecir las tendencias en el proceso y en la calidad del producto dentro de los límites establecidos y además tomar las acciones necesarias en caso que sean excedidos. Los productos son predeciblemente de alta calidad.
- 5) **Optimizado:** El mejoramiento continuo del proceso es garantizado por la retroalimentación cuantitativa desde el proceso y desde las pruebas de técnicas y herramientas innovadoras. La organización tiene los medios para identificar los puntos débiles del proceso y conocer cómo fortalecerlos. Su actividad clave es el análisis de las causas de defectos y su modo de prevención.

Cada nivel sirve de base para que los siguiente establezcan una implantación del proceso eficiente y efectiva. La organización puede, sin embargo, de forma provechosa usar procesos descritos en otros niveles. Saltar niveles es contraproducente debido a que cada uno es básico para obtener el siguiente y la capacidad de poder implementar procesos superiores de madurez no implica que se pueda saltar un nivel.

Cada nivel de madurez está compuesto de varias áreas claves del proceso. Cada una es organizada en 5 secciones definidas como características comunes. Éstas especifican las prácticas claves para el cumplimiento de las metas en el área correspondiente. Las características comunes son:

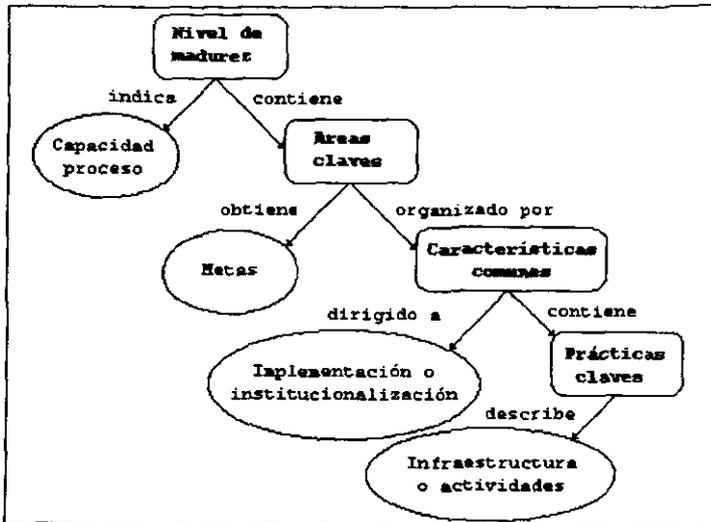
- **Compromiso a desarrollar.** Describe las acciones que la organización debe tomar para establecer el proceso y que pueda ser soportado. Esta característica está asociada con el establecimiento de políticas y con la responsabilidad de la alta dirección.

- **Capacidad a desarrollar.** Describe las precondiciones que deben existir en el proyecto u organización para implementar un proceso de software de manera competente. Involucra los recursos, estructura de la organización y capacitación requerida
- **Actividades desarrolladas.** Describe los roles y procedimientos necesarios para implementar las metas de un área clave del proceso. Considera los planes, procedimientos, actividades, revisiones y acciones correctivas que se requieren.
- **Mediciones y Análisis.** Describe las necesidades de medir el proceso y analizar los resultados.
- **Verificación de la implantación.** Describe los pasos para asegurar que las actividades se desarrollan de acuerdo con lo establecido en el proceso. Generalmente abarca las revisiones y auditorías de la dirección y de los aseguradores de la calidad.

Las prácticas claves describen la infraestructura y actividades que más contribuyen a la efectiva implantación e *institucionalización*⁷ del proceso. La ilustración 4 muestra la arquitectura completa del modelo.

Excepto para el nivel 1, cada nivel de madurez es dividido en varias áreas claves que indican el área en la organización hacia la cual debe enfocarse el mejoramiento del proceso de software, identifican las políticas que se deben seguir para obtener un nivel de madurez y describen como la organización puede madurar. Cada área identifica un grupo de actividades relacionadas que, cuando se desarrollan de forma colectiva, permiten lograr una serie de objetivos considerados importantes para ampliar la capacidad del proceso. Cuando las metas que propone el área son cumplidas, la organización puede afirmar que se ha institucionalizado la capacidad del proceso caracterizada por ésta. Las metas indican el alcance, las fronteras y la intención para cada una. La estructura de las áreas claves por nivel se muestra en la tabla 3.

⁷ Institucionalización es la creación de una infraestructura y cultura corporativa que soporte métodos, prácticas y procedimientos de manera que definan una forma continua de hacer todas las actividades y que ésta sea adoptada para siempre. [Paulk93, p.174]



• Ilustración 4 Arquitectura del modelo CMM [Paulk93, p. 15]

1.4 . Otros modelos de interés

Además de los anteriores existen diversos modelos enfocados a mejorar la calidad de los procesos y productos de software, entre ellos tenemos:

- TickIT (Inglaterra)
- Bootstrap (Comunidad Europea)
- Trillium (Canadá)
- ISO/SPICE

Está dirigido a establecer cómo implementar un sistema de calidad basado en ISO 9000 e interpretado al ambiente de desarrollo de software y no en crear un estándar o decir qué se debe hacer. Facilita la interpretación de la norma ISO 9001 al agregar prácticas específicas de desarrollo de software y requisitos especiales para los auditores.

El esquema de TickIT incluye los siguientes puntos:

- 1) Un capítulo de conceptos de calidad.
- 2) La norma ISO 9000-3, que se considera un requisito para este esquema.
- 3) Una serie de guías para proveedores y compradores donde se explica cuáles deben ser las expectativas en cuanto al sistema de calidad, un proceso para su introducción y un modelo de sistema de calidad.
- 4) Una guía para la auditoría del sistema de calidad.
- 5) Proceso de certificación.
- 6) Una serie de guías complementarias anteriores a la guía TickIT que definían la aplicación de ISO 9001, 9002 ó 9003 a diferentes situaciones contractuales. [Cadena96-2]

La aplicación del esquema está limitado al Reino Unido, aunque muchos países están considerando la introducción de esquemas similares para el desarrollo del software. Hasta 1995 había cerca de 850 organizaciones certificadas bajo TickIT.

A pesar de que existen criterios diversos en cuanto a la aplicación del esquema, ha servido como base para el estudio de otros modelos como son ImproveIT y Quantum. [Ould92]

1.4.2. Bootstrap

Bootstrap es una metodología europea que mediante prácticas, herramientas y estándares de calidad internacional mide, evalúa y propone mejoras al proceso de desarrollo de software.

El objetivo de la metodología es desarrollar un método para la evaluación de procesos de desarrollo de software. Inicialmente se basó en el modelo de madurez de CMM añadiendo conceptos de ISO 9000 y el modelo de proceso ESA (European Space Agency). Actualmente el *Bootstrap Institute*⁹ es quien dirige los destinos de la metodología y está constituido por organismos de diferentes países de Europa. [Duran96]

Es un método para analizar, rediseñar y mejorar los procesos de desarrollo de software. Está compuesto por:

- modelo,
- proceso de evaluación, está dividido en cuatro etapas principales que son: preparación, ejecución de la evaluación, determinación del nivel de madurez y capacidades, y la presentación de resultados de la evaluación,
- base de datos de soporte y que es una de las características fundamentales del método que permite hacer análisis contra el estado de la industria,
- proceso de mejora, que parte de la evaluación inicial, planeación del cambio y evaluación de los resultados, e
- instrumentos de evaluación que forman un conjunto de herramientas que permiten realizar cuestionarios, registrar y presentar los resultados, y brindar guías para los asesores.

Bootstrap genera una buena foto de la organización en corto tiempo mostrando fortalezas y debilidades, hace evaluaciones a varios niveles, presenta bases concretas para las

⁹ Bootstrap Institute surge en 1993 como un nuevo Grupo de Interés Económico Europeo fundado por organizaciones de Finlandia, Italia, Austria y Alemania al término del proyecto inicial de ESPIRIT 5441 que se convirtió en Bootstrap.

mejoras, hace recomendaciones y las monitorea, permite comparar distintas unidades de producción de software contra las mejores de la industria, utiliza y propone métodos probados para implementar prácticas de punta en ingeniería de software.

1.4.3. Trilium

El objetivo del modelo es brindar un medio para iniciar y guiar de manera continua el programa de mejoramiento mediante prácticas claves que pueden ser usadas para mejorar el proceso o ciclo de vida existente.

Sus características principales son:

- está orientado hacia la industria de las telecomunicaciones,
- brinda un enfoque desde el punto de vista del cliente,
- brinda una perspectiva del producto,
- cubre los estándares ISO, IEEE y IEC principalmente, e
- incluye adicionalmente prácticas específicas del modelo.

Está basado fundamentalmente en el modelo CMM desarrollado por SEI y debe ser usado como parte del programa de la organización para el continuo mejoramiento de la capacidad del proceso de desarrollo y soporte del producto. El modelo es aplicado en 4 formas diferentes.

- 1) **Evaluación de la capacidad**, el objetivo es que el cliente pueda evaluar la capacidad de desarrollo y soporte del producto de un proveedor utilizando el modelo como marco de referencia.
- 2) **Estimación en equipo de la capacidad**, es una evaluación de la capacidad realizada por un equipo combinado de cliente-suministrador .

- 3) Estimación interna de la capacidad, se hace énfasis en la efectividad de los procesos, técnicas y herramientas con que cuenta la organización para obtener sus objetivos.
- 4) Programa de mejoramiento continuo, generalmente es complementaria de la anterior puesto que luego de identificar los puntos a mejorar se debe desarrollar un mecanismo de gestión que permita constantemente identificar oportunidades de mejora y actuar con base en eso.

Su aplicación permitirá que:

- la organización de desarrollo sea más responsable con el cliente y con las demandas del mercado,
- se minimice el costo del ciclo de vida de los productos,
- se logre la máxima satisfacción del cliente,
- disminuyan los costos de mantenimiento y desarrollo,
- aumente la efectividad en la estimación de los esfuerzos y el análisis de los riesgos del proyecto, y
- aumente la capacidad para cumplir de manera cuantificable con el diseño y los objetivos de calidad en todas las etapas de desarrollo. [WEB8]

1.4.4. ISO/SPICE

*ISO/SPICE*¹⁰ es un proyecto cuyo objetivo es proponer a la comunidad internacional un estándar para la evaluación de procesos de desarrollo de software y cuya definición recayó sobre el grupo de trabajo Work Group 10 (WG10). Debido al interés en crear un estándar que integre los métodos y modelos existentes se tuvo especial cuidado en lograr la participación de los principales desarrolladores de éstos.

¹⁰ Software Process Improvement and Capability Determination

El modelo, que en la actualidad se le ha llamado modelo internacional emergente, provee un marco para la evaluación de procesos de software. Este marco puede ser usado por organizaciones en la planeación, administración, monitoreo, control y mejora de la adquisición, oferta, desarrollo, operación, evolución y soporte del software. [Díaz96, p. 37]

Brinda un enfoque estructurado para la evaluación del proceso de software de una organización, además de permitir mejorar el proceso y determinar su capacidad, con el objetivo de:

- 1) entender el estado de sus propios procesos para efectos de mejora de procesos,
- 2) determinar la adecuación de sus propios procesos para cumplir un requisito o tipo de requisito particular, y
- 3) determinar la adecuación de los procesos de otra organización para cumplir un contrato o tipo de contrato particulares.

Este proyecto necesitó de un período de pruebas prácticas antes de ser publicado como estándar. Así de pasar por las etapas de evaluación de decisiones críticas e integridad y valor de los productos completos, el modelo debe estar siendo validado para comprobar el cumplimiento de los objetivos y requisitos iniciales. Actualmente está por aprobarse como guía ISO 15504.

La aplicación del modelo permitirá que:

- los proveedores de software puedan guiarse por un esquema único para la estimación del proceso,
- las organizaciones de desarrollo de software cuenten con una herramienta para iniciar y mantener un mejoramiento continuo del proceso,
- los jefes de programadores tengan un medio para asegurar que el desarrollo del software cumple y apoya las necesidades de la organización, y

- los clientes sean capaces de determinar la capacidad de los proveedores de software y estimar el riesgo que implica el seleccionar a un proveedor u otro.
[WEB7]
-

Capítulo II

II. La creación de la documentación para la administración del sistema de calidad

En esta sección se abordará el tema de la creación de los documentos de administración del sistema de calidad y se analizarán las características generales de algunas herramientas disponibles en el mercado para este propósito.

2.1. El manual de administración de la calidad

Una pieza clave para la administración de la calidad en una organización es la creación precisamente del manual de administración de la calidad. Esta tarea compleja por naturaleza y costosa en tiempo y recursos, resulta de vital importancia para que el sistema de calidad sea implantado correctamente ya que define las bases que servirán de guía al proceso de administración y aseguramiento de la calidad.

El sistema de calidad debe cumplir determinadas premisas para que funcione correctamente. Primero deben estar definidos los objetivos y metas de calidad y la forma en que se alcanzarán éstos. Luego es necesario definir el compromiso y actitud hacia la calidad. Finalmente es necesario controlar todas las actividades, garantizar la efectividad del sistema, realizar auditorías para demostrar qué se hace, dónde y cómo, e impulsar el mejoramiento continuo. Todo lo anterior debe estar reflejado en documentos donde se incluyan además los procesos y procedimientos que se llevan a cabo en la organización. De forma general el conjunto de documentos anterior es lo que se conoce como el manual de administración de la calidad.

2.1.1. Características del manual de administración de la calidad

El manual de administración de la calidad es un resumen escrito de todas las actividades de administración de la calidad en la organización. En este manual debe aparecer claramente definido la manera en que se hacen las cosas y con base en este documento se revisará que se realizan como está establecido. La premisa fundamental que se debe tener en mente en el momento de crear la documentación es: *"Di lo que haces, haz lo que dices"* [Paulk94, p.19] De esta manera debemos documentar todo aquello que de no estarlo puede afectar la calidad de los productos o servicios, se acostumbra plantear que se debe documentar todo lo que sería necesario conocer en caso de que cambiara todo el personal de la empresa y se desee mantener el proceso establecido.

El éxito del manual de calidad depende en gran medida de su redacción; debe ser sencillo, convincente, escrito en un lenguaje claro y preciso, y lo más breve posible para garantizar su rápida asimilación. Además después se necesita una correcta capacitación que asegure que es entendido y aceptado por todos y de esta manera pueda aplicarse correctamente. Como complemento es necesario seguir un mantenimiento sistemático del manual para lo cual se pueden realizar diversas actividades, como son:

- 1) auditorías internas de calidad, con el objetivo de revisar que se esté cumpliendo con lo que se establece en los documentos, revisando los problemas que existan y buscando las oportunidades de mejoras,
- 2) mejoramiento continuo, con base en la información recolectada y al posterior análisis se deben tomar decisiones y determinar los posibles cambios,
- 3) aceptar las quejas y sugerencias del personal sobre el sistema de calidad, facilitando la forma en que éstas son entregadas y asegurando que todos conozcan cómo y dónde deben ser entregadas, y
- 4) análisis *post-mortem* inmediatamente después de finalizado un proyecto para determinar los éxitos y fracasos obtenidos y la forma en que se incorporarán estas experiencias a los demás proyectos.

El manual de administración de la calidad debe definir claramente los objetivos relacionados con la calidad, establecer el compromiso con la calidad y la forma en que se

involucra y el papel que juega cada quien para obtener esos objetivos. Contiene además de manera general:

- procedimientos de administración de proyectos,
- procedimientos de control,
- procedimientos de revisión y pruebas,
- procedimientos para determinar las necesidades de los clientes,
- procedimientos para efectuar auditorías internas,
- planes de pruebas,
- planes de desarrollo de proyectos,
- planes de elaboración de los productos, y
- guías para la medición de las actividades y productos.

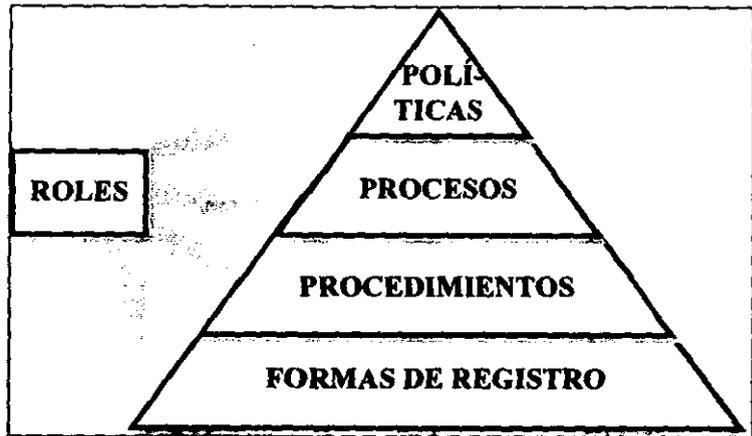
2.1.2. Estructura general del manual de administración de la calidad

La estructura del manual puede variar en dependencia de las características de la organización pero básicamente contendrá los siguientes puntos:

- 1) Políticas de calidad
- 2) Procesos
- 3) Procedimientos
- 4) Formas de registro

Unido a lo anterior deben ser definidos y documentados todos los roles existentes en la organización para que sean reconocidos e identificados quiénes realizan cada una de las actividades que se establecen en cada uno de los documentos que se originen.

La estructura funciona en forma de pirámide donde cada nivel sirve de base a los demás como se muestra en la ilustración 5.



• Ilustración 5 Los niveles del sistema de administración de la calidad

2.1.2.1. Políticas de calidad

Para que funcione el sistema de administración de la calidad es necesario que la organización tenga bien definida su misión o por lo menos una comprensión clara y común de su propósito y de la forma en que los productos o servicios cumplirán los requisitos de calidad de los clientes.

Esta sección deberá contener las políticas de calidad de la organización donde se incluye: los objetivos y las metas en cuanto a calidad, las responsabilidades y la autoridad, así como las vías y canales de comunicación. Define la estrategia de la organización en función de sus políticas, objetivos y estructura. Aquí se debe considerar el alcance de la administración de la calidad con el objetivo de cubrir los procesos básicos y luego extenderse a todos, con lo que podrían existir varios manuales de políticas de calidad vinculados entre sí. Es conveniente, independientemente de la norma o modelo que se escoja para introducir el sistema de calidad, tener identificado claramente la forma en que el sistema de calidad cumple con lo establecido en la norma o modelo de calidad.

La política de calidad debe estar enfocada fundamentalmente hacia la completa satisfacción del cliente, elaborada en forma breve y sencilla. Los objetivos de calidad deben ser complementados de manera que puedan ser medidos y cuantificados y de esta forma sirvan de base para conocer los logros que se van alcanzando. Cualquiera que lea esta sección debe obtener una visión interior de las funciones primarias de la organización y de la manera en que éstas se dirigen y controlan. [Waller97, p.53]

La documentación a este nivel debe ser creada, establecida y mantenida por la alta dirección de la organización que es la que puede establecer las metas, determinar las responsabilidades, otorgar la autoridad y definir las interfaces, además decide la forma en que se implantará el sistema de calidad. De esta manera se garantiza el funcionamiento del sistema ya que los directores a diferentes niveles, por el compromiso que asumen en su redacción, sienten que éste es importante y adecuado. La firma de la declaración por el director general garantiza que todos vean claramente que el impulso hacia la calidad tiene la comprensión y el apoyo del nivel más alto de la organización.

2.1.2.2. Procesos

Este nivel contiene los procesos generales o claves que se dan en la organización. Define qué se hace en la organización por lo que al revisar su contenido debemos obtener una idea general de las actividades que se realizan, nos permite apreciar la organización como un todo.

La descripción de los procesos debe ser breve, concisa, sin ambigüedades y apoyada, siempre que sea posible, en representaciones gráficas de lo que se debe hacer. Los responsables en la redacción de estos procesos deben tener un amplio conocimiento de los procesos que están establecidos o se realizan en la organización, además de tener una visión global de como es que funciona todo.

2.1.2.3. Procedimientos

Aquí encontramos los procedimientos e instrucciones de trabajo, los métodos, reglas y estándares a seguir para garantizar un producto o servicio de calidad. Define el nivel táctico, práctico u operacional del sistema de calidad ya que establece un modelo detallado de la manera en que la organización debería operar, en otras palabras, define cómo se hacen las cosas.

Esta documentación deberá ser elaborada por el personal con experiencia y que esté involucrado en la actividad que se va a regular. Tienen la gran responsabilidad de interpretar las políticas de la empresa y enfocarla hacia el trabajo que se realiza. Definen los procedimientos que dirigirán el trabajo de toda la organización por lo que se requiere de un trabajo serio y consciente, ya que los procedimientos adaptados o impuestos no funcionarían o serían finalmente rechazados. Una buena técnica para comprobar que son realmente efectivos es probar y verificar éstos, en la medida que se redactan, con las personas que los usarán.

Se debe crear un procedimiento donde se establezca la forma en que se elaboran y el contenido mínimo de los documentos. De manera general un procedimiento debe contener una identificación, su objetivo claramente definido, el alcance (ya sea por áreas o individuos) y la descripción de las actividades (quién hace qué, con qué, cuándo, dónde se registra).

Se busca que los procedimientos definidos sean fáciles de usar y para ello se necesita que puedan ser fácilmente localizados y que exista una claridad absoluta en su significado.

En realidad sólo se requiere documentar los procedimientos establecidos y ajustarlos un poco a los requisitos del nuevo sistema. Por lo general se necesitará definir nuevos procedimientos para las actividades de control y registro que no estaban establecidas previamente. El objetivo es asegurar que todos los integrantes de la organización hagan las cosas de la misma manera de modo que encajen con la forma en que las demás personas trabajan sin sustituir la experiencia, el sentido común o la capacitación. [Waller97, p.92]

2.1.2.4. Formas de registro

A este nivel se encuentran todas las formas en donde registramos todo lo establecido en los procesos y procedimientos. Los registros generados permiten verificar que se cumple con lo que está establecido, permiten controlar el proceso de desarrollo y sirven de referencia para el mejoramiento posterior del sistema de calidad.

Se debe procurar que el personal esté bien informado de la forma en que se llevan éstos, cómo se procede, por qué se necesitan y que se mantengan actualizados si ocurriera algún cambio. Todo esto debe ser muy bien controlado pero a la vez debe fluir de manera natural sin entorpecer su trabajo y caer en un proceso burocrático que lo complique más en lugar de facilitarlo.

Estos registros son la base más sólida del sistema de calidad ya que son la evidencia de que las tareas se cumplen de acuerdo a lo establecido o que pueden existir fallas en el sistema. Cualquier fallo en estos registros afectaría al sistema, de aquí que es realmente importante garantizar un adecuado control de los registros y reportes de calidad.

2.1.2.5. Roles

Es necesario definir y documentar los roles que se desempeñan en la organización, de manera que queden establecidas las tareas que debe cumplir cada uno y a su vez se pueda identificar quién realiza qué actividad.

Para asociar una determinada tarea con la persona que la realiza es aconsejable solamente referirse a los nombres de los roles y no los de las personas que los desempeñan, para garantizar que no sea necesario volver a escribir los documentos cada vez que se produzca un cambio en las personas que desempeñan un determinado papel. Por ésta razón es necesario tener claramente definidos y documentados cada uno de los roles que existan.

2.2. Herramientas de software para el apoyo a la administración de la calidad.

Existen numerosas herramientas de software de apoyo a la administración de la calidad que abarcan diversas áreas desde la evaluación del estado de la calidad en la organización hasta el proceso de certificación, enfocadas a un área en específico o a varias áreas.

En Internet podemos encontrar un gran número de estas herramientas, de las que seleccionamos algunas como bases para el modelo que proponemos. [Ver Anexo A] El análisis que se hizo de las mismas está basado en la información que brindan las empresas desarrolladoras así como en los programas demostrativos y tutoriales que están disponibles en algunos casos. En ningún caso se pretende apoyar algún producto o comparar su funcionalidad respecto a otro, sólo han sido evaluados para conocer el estado general que existe en este tipo de herramientas.

De forma general podemos señalar algunas características que se distinguen en estas herramientas:

- 1) La mayoría está enfocada hacia el cumplimiento de lo establecido por las normas de ISO 9000.
- 2) Muy pocas están enfocadas hacia los procesos de creación y desarrollo del software.
- 3) Están ligadas a un tipo de plataforma, principalmente para computadoras personales.
- 4) Establecen un esquema muy rígido para la documentación.
- 5) No están pensadas para el trabajo en equipo, donde varios individuos pueden en un momento dado requerir la misma información.
- 6) El costo en muchos casos es muy alto, aunque los beneficios que puedan reportar sean mayores.
- 7) Están ligadas con procesadores de texto como Microsoft Word o sistemas operativos como Windows.

- 8) Aunque hay herramientas que cubren diversas áreas, en su mayoría están enfocadas a un área en particular de la administración de la calidad.
- 9) Se caracterizan por tener una interfaz agradable y fácil de utilizar, sin necesidad de requerir mucho aprendizaje.
- 10) Utilizan términos en inglés que pueden dificultar su utilización.

Con base en lo anterior el prototipo a desarrollar debe conservar algunas de estas características y mejorar otras para lograr un control más eficiente. Así tenemos que la propuesta debe tener en cuenta:

- 1) Desarrollar una herramienta de propósito general para el control de documentos sin depender de un modelo o norma específica.
- 2) Enfocarse hacia las actividades de desarrollo de software.
- 3) Pensar en la posibilidad de que sea utilizada independientemente de la plataforma que se emplee.
- 4) Elaborar un prototipo flexible y que se adapte a las peculiaridades de cada organización.
- 5) Utilizar algún soporte para el trabajo en redes.
- 6) Crear un modelo sencillo pero que brinde grandes beneficios.
- 7) Utilizar formatos de texto que puedan prescindir de un procesador de textos en específico, en este caso el formato de los documentos HTML puede ser de gran utilidad.
- 8) Enfocarse al área de control de documentos principalmente.
- 9) Elaborar un prototipo agradable y fácil de utilizar, con un aprendizaje mínimo.
- 10) Utilizar términos conocidos en español.

Capítulo III

III. Control de los documentos del sistema de calidad

Este capítulo se centrará en el estudio de los requisitos necesarios para realizar el control de los documentos del sistema de calidad de acuerdo a lo que se establece en ISO 9000 y CMM y los puntos que se deben tener en cuenta para diseñar un prototipo de sistema para cumplir con este fin.

3.1. El control de los documentos

El control de los documentos del sistema de calidad es uno de los elementos claves para su correcto funcionamiento. Es una tarea aparentemente sencilla pero que resulta difícil y propensa a errores que pueden afectar a toda la organización. Surge como una necesidad imperiosa desde el mismo momento en que se desea mejorar el proceso y se vuelve indispensable en la medida que se avanza.

Cuando no existe un eficiente control de los documentos se pueden presentar diversos problemas que afectan en gran medida el trabajo adecuado del sistema de calidad, entre los que podemos citar:

- utilización de documentos que no han sido autorizados,
- empleo de versiones incorrectas,
- aparecen documentos con igual identificación y versión pero con contenidos diferentes,
- no hay seguridad de que un documento pertenezca a la versión correcta,

- no hay acceso a los documentos necesarios, o
 - no existen los documentos.
-

El sistema de calidad requiere que todo lo que se haga en la organización esté establecido en algún documento o quede registrado de alguna forma, de aquí que el control que se tenga sobre éstos determinará la eficiencia del mismo. Este control abarca desde la identificación, edición y distribución de los documentos hasta las revisiones y modificaciones que se hagan a éstos, todo lo cual debe realizarse de manera organizada y planificada.

El control de los documentos abarca todos aquellos que influyen en la calidad del producto o servicio, así tenemos entre otros:

- especificación de requisitos del cliente,
- documentos de diseño,
- planes,
- códigos fuentes,
- procedimientos de trabajo, y
- estándares internos y externos.

Todos éstos deben ser debidamente identificados y luego distribuidos de forma controlada a los roles de la organización, que lo requieran. El control que se tenga sobre la distribución de los documentos garantizará una adecuada utilización de los mismos además de facilitar las tareas de recolección y renovación de los materiales que han caducado.

También es necesario establecer los roles responsables de autorizar el uso de los documentos, así como su revisión y modificación. La persona o grupo de personas responsable de esta tarea garantizarán el empleo de los documentos autorizados y el

control de los cambios que se realicen a éstos. Es necesario considerar que las personas que revisan o aprueban los documentos deben ser independientes de las personas que los elaboran.

En el caso de los productos de software la documentación que se tenga es un factor esencial en el éxito de la actividad de mantenimiento, pero por lo general la elaboración de la misma es una tarea que se deja de lado por la mala planificación además de estar considerada como la menos interesante en todo el proceso de desarrollo del software. Aquí es necesario señalar la importancia que tiene también en el desarrollo de productos de software la administración de las configuraciones y que debe garantizar que exista un adecuado control de las versiones que se tengan de cada uno de los productos de software.

3.2. Los requisitos establecidos para el control de los documentos

Para conocer los requisitos que se necesitan tener en cuenta para realizar el control de los documentos nos basaremos en los requisitos que establecen ISO 9000 y CMM.

3.2.1. Los requisitos de ISO 9000

ISO 9001 establece los requisitos para el control de los documentos en el punto 4.5. [ISO/TC95, p. 31]

4.5 Control de los datos y documentos

4.5.1 Resumen

El *suministrador*¹ debe establecer y mantener documentados los procedimientos para el control de todos los documentos y datos relacionados con los requisitos de este Estándar Internacional incluyendo, de ser necesario, documentos externos tales como estándares y diseños del cliente.

NOTA: Los documentos y datos pueden estar en cualquier tipo de soporte ya sea en papel o en algún medio electrónico.

4.5.2 Aprobación y liberación de los documentos y datos

Los documentos y datos deben ser revisados y aprobados para ser utilizados por el personal autorizado antes de ser liberado. Se debe establecer una lista maestra u otro procedimiento equivalente para el control de documentos identificando el estado de revisión actual de los documentos y debe estar disponible para evitar el uso de documentos inválidos y/o obsoletos.

El control debe asegurar que:

- a) están disponibles en todos los lugares, donde se desarrollan operaciones esenciales para el funcionamiento efectivo del sistema de calidad, las ediciones necesarias de los documentos apropiados
- b) retirar inmediatamente de todos los puntos donde se utilicen los documentos inválidos y/o obsoletos o de otra forma asegurar que no sean empleados
- c) identificar adecuadamente cualquier documento obsoleto retenido para propósitos legales y/o de conservación

4.5.3 Cambios a los documentos y datos

Los cambios a los documentos y datos deben ser revisados y aprobados por quienes realizaron la revisión y aprobación original, a menos que se designe específicamente otra cosa. Los designados para esta función deben tener acceso a la información de respaldo necesaria sobre la cual se basa su actividad.

Donde sea posible, la naturaleza o tipo de cambio debe ser identificado en el documento o en los anexos apropiados.

La norma permite cierta flexibilidad en la forma que se controlan los documentos pero establece que deben existir procedimientos para revisar, aprobar, modificar y editar los documentos y que aseguren que éstos lleguen a la persona adecuada en el momento oportuno y que los documentos obsoletos no sean utilizados. El propósito fundamental de este requisito es brindar al personal la información que realmente necesita y que la distribución y modificación de los documentos sea controlada, además de garantizar que todos los que trabajen en un proyecto utilicen el mismo grupo de documentos.

En empresas pequeñas el control de los documentos se facilitaría en gran medida puesto que se puede establecer una copia central de los documentos autorizados a la que todos

¹¹ En términos de un contrato, es la parte contratada para proveer un producto o servicio. Para ISO 9000 es la organización cuyo sistema de calidad está siendo revisado. En términos del software es el desarrollador del producto [Schmauch96, p. 73]

tendrían acceso, con lo que se evita el problema de establecer las listas de distribución y se simplifica el control de los cambios.

3.2.2. Los requisitos de CMM

CMM, por su parte, se enfoca más en los procesos de software y establece el control de los documentos a través del área clave del proceso Administración de la configuración del software (SCM)¹² definida para el Nivel 2 (Repetido) [Paulk 93, p. 73].

El propósito de SCM es establecer y mantener la integridad de los productos del proyecto de software a lo largo de todo su ciclo de vida.

SCM comprende la identificación de la configuración del software en determinados puntos en el tiempo, el control sistemático de los cambios, el mantenimiento de la integridad y el seguimiento de ésta a través del ciclo de vida del software. Los productos de software que están bajo el control de SCM incluyen los productos de software que son entregados al cliente (por ejemplo documentos de requisitos del software y el código) y los productos relacionados o requeridos para crear éstos.

Establecer una biblioteca con las *líneas base*¹³ del software en la forma que han ido evolucionando. Los cambios y la liberación de los productos elaborados a partir de éstas son controlados sistemáticamente por medio de las actividades de control del cambio y auditorías a las configuraciones de SCM.

Metas

1. Planificar las actividades de SCM.
2. Identificar, controlar y distribuir los *productos de trabajo*¹⁴.
3. Controlar los cambios identificados en los productos de trabajo.
4. Informar del estado y contenido de las líneas base del software a los grupos y personas afectadas.

Compromiso a desarrollar

¹² Software Configuration Management

¹³ Línea base (*baseline*) es una especificación o producto que ha sido formalmente revisado y aprobado para que sirva después como base para desarrollos posteriores y que sólo puede ser modificada mediante el procedimiento de control de cambios establecido. [Paulk93, p. 171]

¹⁴ Los productos de trabajo (*software work products*) son: documentos relacionados con el proceso (planes, estándares, procedimientos y demás), requisitos del software, documentos de diseño, códigos fuente, procedimientos de pruebas, software para realizar las pruebas, software que se entrega al cliente, compiladores y demás herramientas de apoyo. [Paulk93, p. 76]

1. El proyecto sigue lo establecido en la política de la organización para la implantación de SCM

Habilidades a desarrollar

1. Existe o es establecido un consejo con la autoridad para administrar las líneas base del software.
2. Existe un grupo responsable por la coordinación e implantación de SCM en el proyecto.
3. Se suministran los fondos y recursos necesarios para el desempeño de las actividades de SCM.
4. Los miembros del grupo de SCM son capacitados en los objetivos, procedimientos y métodos para el desempeño de sus actividades.
5. Los miembros de los grupos relacionados con el proceso de software son capacitados para desempeñar las actividades de SCM.

Actividades desarrolladas

1. Se prepara un plan de SCM para cada proyecto de software según lo establecido en el procedimiento correspondiente.
2. Se utiliza el plan de SCM, documentado y aprobado, como base para realizar las actividades de SCM.
3. Se establece un sistema para la biblioteca de administración de configuraciones como depósito para las líneas base del software.
4. Se identifican los productos de trabajo que serán controlados por la administración de configuraciones.
5. Las solicitudes de cambio y problemas reportados para todos los elementos de configuración son iniciadas, registradas, revisadas, aprobadas y rastreadas según lo establecido en el procedimiento correspondiente.
6. Los cambios en las líneas base son controlados de acuerdo al procedimiento existente.
7. La creación y liberación de los productos de la biblioteca de líneas base del software son controladas según lo establecido en el procedimiento correspondiente.
8. El estado de los elementos de configuración es registrado según establece el procedimiento correspondiente.
9. Los reportes estándar de las actividades de SCM y el contenido de las líneas base del software son desarrollados y puestos a disposición de los grupos afectados.
10. Las auditorías a las líneas base del software son realizadas de acuerdo al procedimiento correspondiente.

Mediciones y análisis

1. Se hacen y utilizan las mediciones para determinar el estado de las actividades de SCM.

Verificación de la implantación

1. Las actividades de SCM son revisadas periódicamente por el gerente principal.
2. Las actividades de SCM son revisadas eventual y periódicamente por el gerente del proyecto.
3. El grupo de SCM periódicamente audita las líneas base del software para verificar la conformidad con los documentos definidos.
4. El grupo de aseguramiento de la calidad revisa y/o audita las actividades y productos del grupo de SCM y reporta los resultados.

Lo anterior es un resumen de lo que se establece en esta área, debido a que es muy amplio todo lo que abarca.

El modelo CMM es mucho más específico en los requisitos para el control de los documentos ya que establece claramente quiénes lo realizan, qué se controla, cómo, cuándo y dónde se registra. El problema aquí radica en el gran número de funciones o roles que se requieren para llevar a cabo las actividades de SCM las cuales no encontramos en empresas pequeñas que cuentan con un personal reducido.

3.3. Elementos a considerar para el control de los documentos

Con lo que hemos analizado hasta el momento podemos identificar algunas tareas o funciones básicas para garantizar un adecuado control de los documentos, así tenemos que es necesario:

- 1) Controlar los documentos que están activos o están siendo utilizados.
- 2) Controlar el acceso y distribución de los documentos.
- 3) Controlar los cambios y modificaciones a los documentos.
- 4) Controlar los documentos obsoletos.

Para llevar a cabo el control de los documentos primeramente debemos establecer la identificación y el estado de los mismos. Principalmente necesitamos:

- título
- número o identificador del documento
- versión
- fecha de elaboración
- identificación del responsable de su elaboración
- tipo de documento
- estado del documento (borrador, en revisión o aprobado)
- identificación de los que realizan la revisión
- fecha de revisión
- identificación del responsable de aprobar el documento
- fecha de aprobación

Una vez que los documentos estén debidamente identificados deben ser registrados de alguna forma para tener un control de cuáles son los documentos que actualmente se tienen y que sería el *registro de control de documentos* que contendría la información de identificación y alguna descripción general del contenido de cada documento.

En el caso que se necesite distribuir copias de los documentos se necesita crear un *registro de distribución de los documentos* que contenga en principio:

- identificación del documento
- persona o función a la que se le entrega la copia

Con lo anterior tendríamos control sobre las existencia y la distribución de los documentos pero no nos permitiría conocer el estado de las revisiones y modificaciones a éstos. Para ello debemos tener un *registro de control de cambios* en el que se debe registrar la información de las *solicitudes de cambios* que se hayan emitido para cada uno de los documentos y que deben contener:

- identificación de la solicitud de cambio
- título de la solicitud de cambio
- identificación del documento al que se solicita el cambio
- fecha de la solicitud de cambio
- estado de la solicitud de cambio (aprobada, rechazada, implantada)
- identificación del que aprueba la solicitud de cambio
- fecha de aprobación de la solicitud de cambio
- identificación del que implanta el cambio
- fecha de implantación del cambio
- descripción de la solicitud de cambio y de los resultados de las evaluaciones e implantación

Sólo nos falta por establecer la forma en que se controlen los documentos que caducan o son obsoletos y para ello con base en el *registro de distribución de los documentos* debemos garantizar que se retiren todas las copias existentes de un documento y que de ser el caso se reemplacen por las nuevas. Los documentos obsoletos, que se deseen conservar, deben ser adecuadamente identificados y distinguidos de los que no lo son. Tal vez se podría pensar en llevar un *registro de control de documentos obsoletos* para tal propósito.

Tomando en cuenta todos estos puntos podemos garantizar un correcto y adecuado control de los documentos, pero para facilitar esta tarea se requiere de una herramienta o software de apoyo. En el siguiente capítulo se cubren las especificaciones del análisis y diseño de el prototipo que proponemos para apoyar esta actividad.

Capítulo IV

IV. Especificaciones del análisis y diseño

En esta sección se exponen los resultados obtenidos en las fases de análisis y diseño del prototipo para el control de los documentos.

4.1. Consideraciones generales

En el capítulo anterior se presentaron los requisitos generales que hay que considerar para realizar el control de los documentos. Con base en esos puntos se propone el desarrollo del prototipo que es el objetivo de este trabajo.

Parte de las experiencias de este proyecto surgieron durante el curso de Ingeniería de Software Orientada a Objetos que ya se había mencionado en la Introducción. También de este curso es retomada la metodología de desarrollo que fue elaborada para el proyecto que se desarrolló como parte del curso. La misma está basada en varios modelos orientados a objeto existentes y ampliamente reconocidos, principalmente desarrollados por *Grady Booch*, *Ivar Jacobson*, *James Rumbaugh* y *Peter Coed*. De cada modelo se retomaron los puntos que eran aplicables a las necesidades del proyecto y fueron utilizados por el equipo de trabajo que para esa materia se creó.

Las bases de toda la "metodología" puede ser tema para otras tesis y como quedan fuera del alcance de este trabajo sólo se indican los puntos principales que es necesario tomar en cuenta. Hay que señalar que parte de esa información se puede encontrar en [Ibargüengoitia] y puede servir de apoyo durante la fase de análisis.

UML (*Unified Modeling Language*) se escogió como base para el modelado de casos de uso, escenarios, diagramas de clases y de estado, principalmente, ya que permite integrar varios modelos que reflejan las mejores prácticas en la industria, además que facilita el proceso de modelado de sistemas de software. La herramienta *Rational Rose*¹⁵ resultó de gran ayuda para elaborar todos los diagramas.

4.1.1. Requisitos del sistema

La primera actividad que se debe realizar para el desarrollo de un sistema es definir el problema que se quiere resolver. En este caso consiste en crear un sistema que permita realizar el control de los documentos de administración de la calidad de acuerdo a los requisitos que se deben cumplir para desempeñar adecuadamente esta actividad y que se detallan más adelante.

Requisitos no funcionales

Partiendo de las características estudiadas anteriormente en varias herramientas afines se deben establecer como requisitos no funcionales del prototipo, los siguientes:

- 1) Permitir que pueda ser implantado fácilmente sin depender de la plataforma para la cual sea desarrollado, para lograr esto se propone utilizar como lenguaje de programación Java por las facilidades que brinda al respecto.
- 2) Permitir el acceso de forma distribuida a los documentos de manera que se puedan leer desde cualquier equipo conectado al servidor que contendrá físicamente los documentos para lo cual se deben crear applets en Java que puedan operar desde un servidor HTTP.
- 3) Desvincular el formato de los documentos de un procesador de textos en específico y para ello la utilización de documentos html puede ser de gran ayuda.

¹⁵ *Rational Rose* es desarrollado por *Rational Software Corporation*, para mayor información sobre este producto puede consultar la página en <http://www.rational.com/>

- 4) Establecer un esquema abierto donde el usuario pueda crear los parámetros que considere y en este caso se desea poder definir los grupos de documentos que se considere, los estados por los que pasarán los documentos y su contenido.
- 5) Minimizar el uso de pantallas y utilizar términos conocidos en español para facilitar el aprendizaje y permitir una rápida aplicación del sistema.
- 6) Garantizar que sólo el personal autorizado tenga acceso a los documentos y evitar que se efectúen cambios o modificaciones que no hayan sido aprobadas previamente.

Requisitos funcionales

En cuanto a las funciones que debe desempeñar el prototipo es necesario considerar principalmente los siguientes requisitos:

- 1) Crear y editar los documentos, estableciendo la definición de cada una de las características que se necesitan para llevar el control y que fueron presentadas en el capítulo anterior.
- 2) Consultar los documentos para leer su contenido.
- 3) Cambiar el estado de los documentos.
- 4) Regresar documentos que entran al ciclo de evaluación y que no son aceptados.
- 5) Generar nuevas versiones de los documentos aprobados.
- 6) Crear y editar solicitudes de cambio y definir las características que se necesitan para llevar el control de las mismas según se planteó en el capítulo anterior.
- 7) Cambiar el estado de las solicitudes de cambio.
- 8) Rechazar solicitudes de cambio.
- 9) Avisar a cada uno de los responsables de elaboración, revisión o aprobación cuando se requiera su participación.

4.1.2. Actores

Todo lo anterior son los requisitos que necesita cumplir el prototipo para dar solución a la problemática planteada inicialmente pero el acceso a estas funciones está en dependencia del nivel de acceso que tenga el usuario, el cual está relacionado con el grupo de usuarios al que pertenece. Los usuarios o grupos de usuarios que interactúan con el sistema se conocen como "actores". Un usuario puede pertenecer a varios grupos en dependencia de la función que quiera ejecutar. Los diferentes usuarios del sistema de acuerdo a los derechos que tienen se pueden clasificar en las siguientes categorías:

- **Administrador**, es el único que puede crear nuevos usuarios del sistema, puede establecer las características de los grupos y estados de los documentos y en principio es el propietario de todos los documentos.
- **Propietario**, es el que crea y edita los documentos o solicitudes de cambio, puede enviar a evaluar los documentos, establece revisores y aprobadores para los documentos y lo modifica cuando existen solicitudes de cambio.
- **Revisor**, es el que establece el propietario de un documento para que revise el documento y puede cambiar su estado a revisado o rechazado, en cuyo caso elabora una solicitud de cambio justificando el rechazo.
- **Aprobador**, es el que establece el propietario de un documento para que apruebe el documento y puede cambiar el estado a aprobado o rechazado, en cuyo caso elabora una solicitud de cambio justificando el rechazo. El aprobador de una solicitud de cambio a un documento es el propietario del documento.
- **Lector**, tiene acceso al sistema para leer los documentos y solicitudes de cambio y puede crear solicitudes de cambio a los documentos.

Una vez establecidos los requisitos e identificados los actores se puede continuar con las siguientes actividades de la fase de análisis.

4.2. Resultados del análisis

En función de las características establecidas en la sección anterior son identificadas las clases principales que serán refinadas en etapas sucesivas del análisis hasta que cumplan con los requisitos que fueron establecidos. Las clases resultantes son agrupadas en 4 componentes que son:

- Dominio del problema (PD), formado por las clases que son significativas para el sistema.
- Interfaz humana (HI), se forma a partir de las clases del dominio del problema con las que el usuario requiere interactuar por medio de ventanas.
- Manejo de datos (DM), se obtienen de cada una de las clases del dominio del problema que se desea sea persistente.
- Interacción con otros sistemas (SI), son clases que se agregan, cuando se requiera, para cada uno de los sistemas o dispositivos con los que se necesite interactuar.

COMPONENTE	CLASES PRELIMINARES
Dominio del problema	Document ChangeRequest User GroupDoc StateDoc DocManager
Interfaz humana	CondorWindow LogonWindow UserConfigWindow EditDocumentWindow DefineDocumentWindow DefineChangeWindow OptionConfigWindow ErrorWindow
Manejo de datos	DocumentDB ChangeRequestDB UserDB GroupDocDB StateDocDB

Tabla 4 Clases del análisis

4.2.1. Clases resultantes del análisis

A modo de estándar y para un mejor entendimiento se establece que las clases se nombran usando términos en inglés representativos de la función que realiza cada una. Se facilita la identificación empleando los mismos nombres, siempre que sea posible, para las clases relacionadas en los distintos grupos y sólo se agrega la terminación *Window* en las clases de HI y DB en las de DM. En la tabla 4 se muestran las clases resultantes del análisis y en el Anexo B "Documento de Análisis" se muestra con más detalle las características y relaciones que se dan entre cada una.

4.2.2. Casos de uso

Otra actividad que se realiza durante esta fase es la identificación de los casos de uso que definen la parte dinámica del análisis. Adicionalmente se crean los escenarios correspondientes a cada uno donde se modela la interrelación que se da entre los objetos. Los casos de uso surgen del análisis de las funciones del sistema y se van refinando en la medida que se logra un mayor entendimiento del problema. Igual sucede con los escenarios de los que pueden surgir nuevas clases, se refinan las existentes o se amplían con nuevos métodos y atributos.

Los casos de uso identificados se relacionan en la tabla 5 y los detalles de los escenarios se incluyen en el Anexo B, mencionado anteriormente.

CASO DE USO	DESCRIPCIÓN
Entrar al sistema	Leer la identificación y clave del usuario y verificar si tiene acceso al sistema. Si es usuario del sistema puede entrar. De lo contrario se le permite volver a intentar y si ha llegado al número máximo de intentos debe abandonar y no se le permite entrar al sistema.
Crear un documento	Mostrar la ventana de edición para capturar la información del documento así como sus características principales para poder establecer el control sobre el mismo. Adicionalmente se deben establecer los responsables de revisión/aprobación según sea el caso.

CASO DE USO	DESCRIPCIÓN
Editar un documento existente	Seleccionar el documento a editar de acuerdo al grupo indicado y mostrar la ventana de edición para modificar los datos que se consideren. Se pueden modificar los responsables de revisión/aprobación según sea el caso. Esto se permite si el usuario es el propietario y está en borrador el documento.
Leer el contenido de un documento	Seleccionar el documento a visualizar de acuerdo al grupo indicado y mostrar su contenido.
Enviar a evaluación un documento	Seleccionar el documento a evaluar. Si el usuario es propietario lo puede enviar a evaluación notificando por correo a los responsables de realizar dicha acción.
Crear solicitud de cambio	Seleccionar el documento de acuerdo al grupo e indicar que se hará una solicitud de cambio. En este caso se mostrará la ventana de edición para definir las características de la solicitud.
Editar solicitud de cambio existente	Seleccionar el documento y luego seleccionar la solicitud de cambio a editar. Entonces se muestra la ventana de edición para modificar los datos que se consideren. Esto se permite si el usuario es el propietario y está en borrador la solicitud.
Leer una solicitud de cambio existente	Seleccionar el documento de acuerdo al grupo y luego seleccionar la solicitud de cambio a visualizar. Como resultado se debe mostrar el contenido de la misma.
Revisar un documento	Seleccionar un documento y luego indicar que se va a cambiar su estado. Para ello es necesario que el usuario sea el que está propuesto como revisor y que el documento esté en evaluación. El documento puede cambiar el estado a revisado.
Aprobar un documento	Seleccionar un documento y luego indicar que se va a cambiar su estado. Es necesario que el usuario sea el que está propuesto como aprobador y que el documento esté en evaluación. El documento puede cambiar el estado a aprobado en cuyo caso cambia a una nueva versión y si existe un documento aprobado anterior éste pasa a obsoleto.
Rechazar un documento en revisión o aprobación	Seleccionar un documento y luego indicar que se va a cambiar de estado. Se indica que se rechaza la revisión o aprobación. El documento pasa a borrador con una nueva versión y se emite una solicitud de cambio donde se describe por qué se rechaza.
Enviar a modificar un documento aprobado	Seleccionar un documento aprobado para generar una nueva versión. Si el usuario es el propietario y existen solicitudes de cambio se puede enviar a borrador con una nueva versión.

CASO DE USO	DESCRIPCIÓN
Modificar el estado de una solicitud de cambio	Seleccionar un documento y luego seleccionar la solicitud de cambio a la que se desea modificar el estado. Los estados de la solicitud de cambio pueden ser aprobado, rechazado o implantado. Para modificar el estado se debe tener la autorización correspondiente.
Crear un nuevo usuario del sistema	El administrador del sistema llena los datos del nuevo usuario.
Establecer grupos y estados	El administrador del sistema establece los grupos de documentos que se tendrán y los estados con los nombres que existirán para los documentos.
Seleccionar un documento correspondiente a un grupo	Es utilizado por otros casos de uso y en esencia consiste en mostrar los documentos pertenecientes a un grupo y leer la selección del documento con el que se va a trabajar.
Llenar datos de control del documento	Es utilizado por otros casos de uso y permite capturar los datos donde se definen las características del documento.
Seleccionar una solicitud de cambio de un documento	Es utilizado por otros casos de uso y en esencia consiste en que con base en el documento seleccionado se muestran las solicitudes de cambio existentes y se lee la selección de la solicitud de cambio con que se va a trabajar.
Llenar datos de control de la solicitud de cambio	Es utilizado por otros casos de uso y permite capturar los datos donde se definen las características de la solicitud de cambio.

Tabla 5 Casos de uso

4.2.3. Diagramas de estado

Para los objetos que presentan un comportamiento dinámico debido a que cambian su estado producto a un estímulo se crea un diagrama de estado donde se refleje el paso por los distintos estados, partiendo del estado inicial hasta llegar al estado final, así como los estímulos que provocan ese cambio. En este caso los objetos con un comportamiento dinámico son obtenidos de las clases *Document* y *ChangeRequest*. Los diagramas correspondientes se muestran en el Anexo B.

Un documento al ser creado está en el estado de borrador y en su versión inicial. Se puede modificar mientras no se envíe a evaluación. El documento en evaluación no puede ser editado y cambia su estado a revisado o aprobado según establezca el responsable de su evaluación. Una vez aprobado el documento se genera una nueva versión aprobada y el

que estaba vigente pasa a obsoleto. En todos los casos si el documento que se está evaluando es rechazado se elabora una solicitud de cambio y el documento pasa nuevamente al estado de borrador con una nueva versión.

La solicitud de cambio se crea en el estado borrador. Una vez que pasa a evaluación no puede ser modificada y cambia su estado a aprobada o rechazada según establezca el responsable de evaluación. La solicitud aprobada debe cambiarse a implantada una vez que los cambios son incorporados en el documento.

4.2.4. Diccionario de clases

Finalmente es necesario definir el diccionario de clases donde queda plasmado todo el conocimiento adquirido en la fase y que será utilizado posteriormente para conocer lo que se debe hacer. En este caso esa información también aparece reflejada en el Anexo B.

Los productos obtenidos durante esta fase deben ser refinados y ampliados para reflejar las características del lenguaje de programación que se seleccione en la fase de diseño.

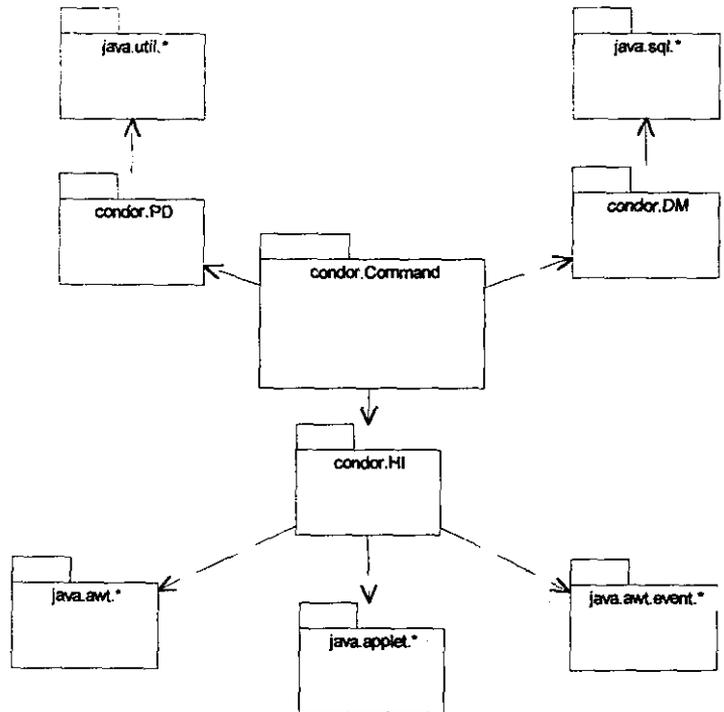
4.3. Resultados del diseño

Durante la fase de diseño se refinan los modelos del análisis, se identifican las clases que serán reutilizadas y se adapta el modelo al ambiente de implementación.

El diseño que se propone está pensado para ser implementado en el lenguaje de programación Java utilizando las clases que forman los diferentes paquetes de JDK 1.1 o superior. El sistema funcionaría como un applet insertado en una página html que corre desde un servidor HTTP. Funciona de manera interactiva recibiendo ordenes del usuario a través del menú y enviando las respuestas por medio de las pantallas de visualización. El manejo de los datos persistentes se soportaría con las clases de JDBC y utilizando ODBC como manejador.

4.3.1. Arquitectura del sistema

Para la descripción de la arquitectura del sistema se establece la arquitectura lógica y física. La arquitectura lógica del sistema muestra la agrupación de las clases en categorías mientras que la arquitectura física muestra el agrupamiento de los módulos en subsistemas.



• Ilustración 6 Arquitectura lógica

Las categorías que integran el sistema están asociadas a los componentes que presentamos en la fase de análisis y que son: Dominio del problema (PD), Interfaz humana (HI) y Manejo de datos (DM). Adicionalmente se incorpora un nuevo grupo de

componentes Ejecutantes (Command). La interacción entre los distintos componentes se muestra en la Ilustración 6.

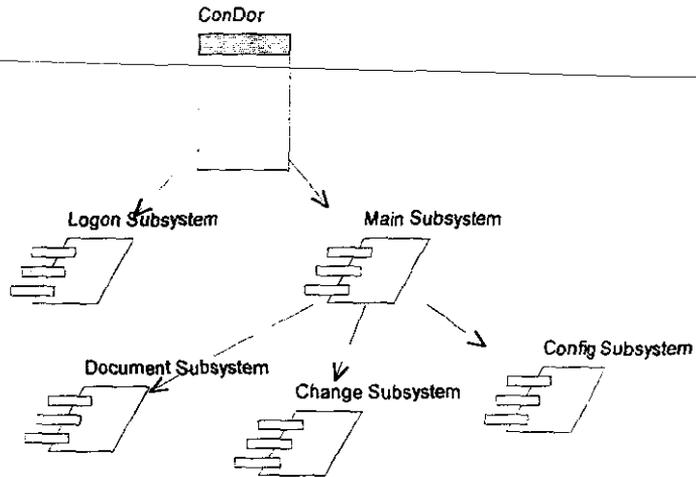
El nuevo grupo de componentes que se crea es con el fin de separar las funcionalidades entre los demás grupos de manera que sólo las clases ejecutantes conozcan la forma en que interactúan las clases entre sí en el momento de realizar una acción, con lo que se logra una mayor independencia, se facilita el mantenimiento y se minimizan las interfaces. Las clases ejecutantes tienen todo el control del sistema y las demás clases son pasivas en el sentido que sólo pueden hacer lo que les corresponde y si requieren interactuar con otras necesitan invocar el método correspondiente en la clase ejecutante.

Los subsistemas en que se agrupan los módulos anteriores se muestran en la Ilustración 7 y son los que se describen a continuación:

- *Subsistema de inicio*, donde se verifica que el usuario que intenta entrar al sistema tiene derecho de acceso.
- *Subsistema principal*, aquí se realizan todas las funciones correspondientes al control de los documentos.

El *Subsistema principal* a su vez se divide en otros subsistemas de acuerdo a las funciones que se establecieron durante el análisis, obteniéndose los siguientes grupos:

- *Subsistema de control de documentos*, donde se crean y editan los documentos, se muestra su contenido y se cambia su estado.
- *Subsistema de control de cambios*, donde se crean y editan las solicitudes de cambio a los documentos, se muestra su contenido y se cambia su estado.
- *Subsistema para la configuración del sistema*, donde se establecen las opciones del sistema y los usuarios con acceso al sistema.



• Ilustración 7 Arquitectura física

Cada subsistema de la arquitectura física contiene componentes de la arquitectura lógica. De esta forma podemos encontrar componentes del Dominio del problema, Manejo de datos, Interfaz humana y Ejecutantes en cada uno de los subsistemas. Principalmente los que difieren en cada subsistema son los dos últimos por ser más específicos a cada funcionalidad mientras que los del PD y DM pueden estar presentes en varios subsistemas.

4.3.2. Clases resultantes del diseño

Para efectos del prototipo que se propone como parte de este trabajo se restringe el alcance del diseño para cubrir los elementos asociados con el control de los documentos y el mecanismo que se debe implantar para garantizar el mismo. Incluyendo en esta parte el ciclo de aprobación de los documentos y la emisión de solicitudes de cambio. Lo relacionado con el manejo de los documentos de forma distribuida y el envío de mensajes a los usuarios no se cubre por las características que requiere un desarrollo de ese tipo y las consideraciones que es necesario hacer por cuestiones de seguridad lo cual haría muy extenso el alcance del trabajo y se saldría del propósito de establecer un esquema para el control de los documentos.

Partiendo de los anterior se realizó un refinamiento de las clases obtenidas en la fase de análisis del que se obtuvieron nuevas clases producto de introducir las propiedades de herencia.

En el grupo PD aparece la clase *GeneralDocument* como una generalización de los tipos de documentos existentes, pensando que se tienen documentos y solicitudes de cambio. Las clases preliminares *StateDoc* y *GroupDoc* por su simplicidad no se consideran como tales y se incluyen como Vectores dentro de *DocManager*. Por otra parte en HI las clases *DefineChangeWindow* y *DefineDocumentWindow* heredan de la clase *DefineAbstractWindow* que generaliza el funcionamiento de ambas. Las clases del DM no han sido modificadas, sólo se agrega una clase *DataBase* que se encarga de la conexión y de los detalles del manejo de la base de datos en general. Las nuevas clases que surgen son del grupo *Command* que se relacionan con cada uno de los subsistemas de la arquitectura física, estas clases heredan de la clase *AbstractCommand*.

La información general sobre las clases resultantes del diseño se indican en la tabla 6 así como los paquetes de java que se utilizan en cada grupo de componentes. La descripción detallada sobre las clases se encuentra en el Anexo C.

COMPONENTE	CLASES RESULTANTES DEL DISEÑO	PAQUETES DE JAVA QUE SE REQUIEREN
Dominio del problema	Document ChangeRequest User DocManager GeneralDocument (Generalización de Document y ChangeRequest)	java.util
Interfaz humana	ConDorWindow LogonWindow UserConfigWindow OptionConfigWindow EditDocumentWindow DefineDocumentWindow DefineChangeWindow	java.awt java.awt.event java.util

COMPONENTE	CLASES RESULTANTES DEL DISEÑO	PAQUETES DE JAVA QUE SE REQUIEREN
	ErrorWindow DefineAbstractWindow (Generalización de DefineChangeWindow y DefineDocumentWindow)	
Manejo de datos	DocumentDB ChangeRequestDB UserDB GroupDocDB StateDocDB DataBase	java.sql java.net java.util
Ejecutantes	AbstractCommand CondorCommand EditCommand	java.util

• Tabla 6 Clases resultantes del diseño

4.3.3. Interacción de los actores con el sistema

En la sección anterior se presentaron las distintas categorías de usuarios de acuerdo a sus derechos dentro del sistema. Con base en esto en la tabla 7 se presentan las opciones a las que tienen derecho cada uno, dependiendo de la categoría y pensando en que existen 3 estados en los que puede cambiar su función y que son: inicio de sesión, documento seleccionado y solicitud de cambio seleccionada. Las opciones disponibles para el lector son aplicables para todos los usuarios dentro del mismo estado del sistema, por lo que no se repiten en cada uno. En todos los casos es posible salir del sistema.

ESTADO	USUARIO	OPCIONES DISPONIBLES
Inicio de sesión	Lector	Crear un documento Seleccionar un grupo de documentos Editar los datos de su usuario
	Administrador	Crear/editar los datos de los usuarios Establecer la configuración del sistema
Documento seleccionado	Lector	Crear un documento Mostrar el contenido del documento seleccionado Crear una solicitud de cambio Seleccionar un grupo de documentos Mostrar las solicitudes de cambio del documento seleccionado

ESTADO	USUARIO	OPCIONES DISPONIBLES
	Propietario	Editar el documento seleccionado Enviar el documento seleccionado a evaluación
	Revisor (Aprobador)	Cambiar el estado del documento seleccionado (Revisado / aprobado / rechazado)
Solicitud de cambio seleccionada	Lector	Crear una solicitud de cambio Mostrar el contenido de la solicitud de cambio Regresar al grupo de documentos seleccionado
	Propietario	Editar la solicitud de cambio seleccionada
	Aprobador	Editar la solicitud de cambio seleccionada Cambiar el estado de la solicitud de cambio (Aprobada/ rechazada)
	Implantador	Editar la solicitud de cambio seleccionada Cambiar el estado de la solicitud de cambio (Implantada)

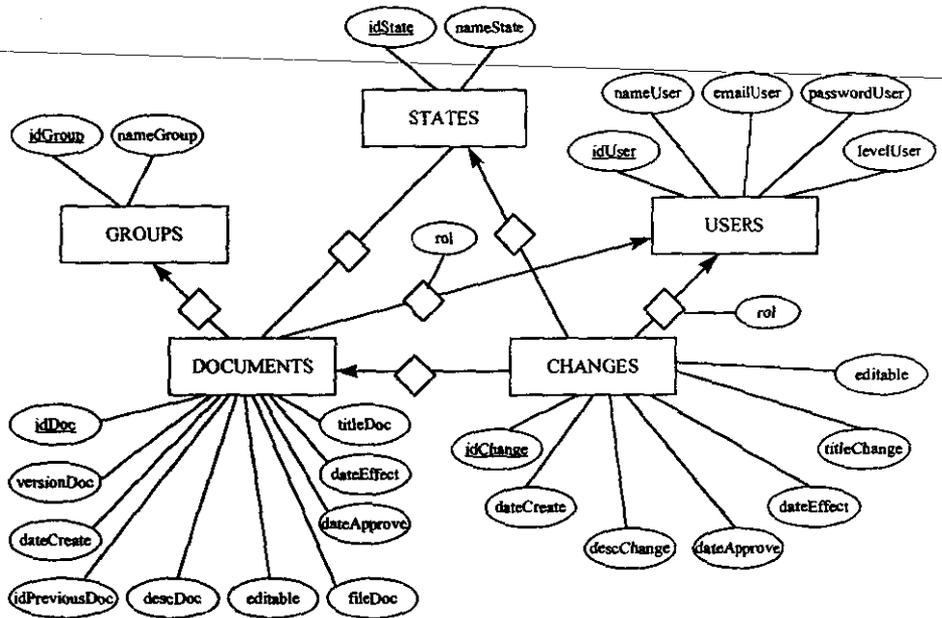
• Tabla 7 Opciones disponibles por usuario según el estado del sistema

4.3.4. Diseño de la base de datos

El manejo de los datos de la herramienta debe funcionar de forma distribuida pero tomando en consideración el alcance del diseño se hará una conexión directa a una base de datos que reside en la misma PC en que está instalado el prototipo. En revisiones posteriores se debe considerar esta restricción y ampliar el acceso a otras máquinas y a diferentes plataformas.

Para crear la base de datos es necesario considerar las clases que son persistentes y que en esencia son las que pertenecen al grupo del Manejo de datos. Como se utilizan clases del paquete JDBC que realizan un puente con ODBC se establece un sistema de bases de datos relacional. Para ello es necesario crear por cada una de las clases DM una relación.

Las relaciones que se obtienen son: *Documents*, *Changes*, *Groups*, *States* y *Users*. Para definir la estructura de cada una podemos basarnos en el mapeo que se realiza de las clases a relaciones que se encuentra en el Anexo C y que se apoya en el diagrama entidad relación que se muestra en la Ilustración 8.



• Ilustración 8 Diagrama entidad relación

Las relaciones se crean con el apoyo de Microsoft Access y se debe configurar el nombre de la fuente de datos para lograr la conexión y el acceso a la información desde el prototipo.

Considerando todo lo establecido en el diseño se puede elaborar el prototipo que cubra el alcance propuesto en esta fase. En revisiones posteriores se puede mejorar esta primera aproximación para lograr cubrir los requisitos iniciales del sistema.

Conclusiones y recomendaciones

El creciente interés por mejorar los procesos de desarrollo de software ha tenido como consecuencia un mayor interés por mejorar la calidad de los procesos y productos que se obtienen. Para lograr esto es necesario establecer la forma en que debe operar la organización de manera tal que la calidad se obtenga de forma consistente en todos los proyectos que se desarrollen.

Hemos visto que un factor imprescindible para lograr la calidad de los productos que se desarrollan es contar con un adecuado sistema para el control de los documentos que rigen todo el proceso de administración y desarrollo de los proyectos. Por otra parte esta tarea se dificulta al no contar con las herramientas adecuadas.

El trabajo presenta una propuesta de solución aplicable a pequeñas empresas que requieren establecer control sobre sus documentos. Establece los principios que deben regir el control de los mismos y de los cambios que se realizan a estos, así como la descripción del ciclo de evaluación que debe seguir todo documento o solicitud de cambio.

Las características del ciclo de evaluación propuesto en este trabajo no sólo se adapta al control de documentos sino que también puede ser aplicado a cualquier producto que requiera estar bajo control y que cumpla determinados parámetros de calidad, los que serían evaluados por el personal responsable de velar por ellos.

La información recopilada como parte del fundamento del proyecto puede ser de gran utilidad para numerosas organizaciones interesadas en introducir conceptos y elementos relacionados con la calidad, así como los principios y objetivos de algunas normas y modelos existentes para la mejora de los procesos de desarrollo de la organización. De

igual forma el análisis de las herramientas que fueron objeto de estudio puede servir como referencia para la evaluación de otras herramientas que existen actualmente en el mercado.

La implantación del prototipo propuesto facilitaría en gran medida la tarea de controlar y poner a disposición del personal involucrado los documentos que regulan su trabajo. Añadiendo algunas de las funcionalidades que se plantearon inicialmente la herramienta sería una gran ayuda desde el punto de vista funcional y en cuanto a presupuesto.

La utilización de las técnicas de modelado Orientado a Objetos y en específico de las facilidades que brinda el Lenguaje de Modelado Unificado (UML) resultaron de gran ayuda para la concepción del prototipo. A través de ellas se puede realizar un análisis de los requisitos planteados inicialmente e ir creando un modelo que en fases sucesivas se va refinando hasta alcanzar el nivel de detalle requerido para el diseño. Permite manejar diferentes niveles de abstracción, partiendo de elementos básicos que se van enriqueciendo hasta lograr elementos más complejos que permitan cubrir las funcionalidades que se requieren. Los diferentes diagramas que plantea UML permiten describir con gran detalle las características del prototipo lo que facilita el proceso de codificación.

El empleo del lenguaje de programación *Java* es fundamental para el desarrollo del prototipo. Esto se fundamenta principalmente por ser: orientado a objetos, independiente de la plataforma de desarrollo, seguro y sencillo, y porque establece un mecanismo para el manejo de excepciones que es fundamental para el desarrollo de aplicaciones distribuidas.

El prototipo obtenido permite garantizar el control sobre los documentos de manera tal que exista control sobre las versiones y sobre quienes pueden emitirlos. De igual forma establece un esquema para controlar los cambios a las versiones ya aprobadas. Aún existen varias funcionalidades que no han sido cubiertas y que se proponen como mejoras para trabajos posteriores.

Trabajos a futuro

Como recomendaciones para proyectos de mejora futuros sería necesario considerar algunos elementos que no se incluyeron para esta primera versión.

Para lograr que el sistema cumpla con todas sus funcionalidades es necesario ampliar el alcance del diseño para cubrir áreas que son importantes y que no fueron cubiertas en esta primera aproximación. Básicamente las sugerencias serían:

- Considerar la posibilidad de usar la herramienta de forma distribuida, para lo cual es necesario examinar las posibilidades de Internet y las características del manejo y seguridad de la información a través de este medio .
- Posibilitar el acceso al correo electrónico con lo que se completa la posibilidad de informar de las acciones que se requieren a los usuarios del sistema.
- Evaluar si es viable que puedan participar al mismo tiempo varios usuarios en la evaluación de los documentos y solicitudes de cambio. Así como las características que tendría el proceso en estos casos.
- Estudiar otras opciones que permitan que el sistema se adapte a las necesidades de los diferentes usuarios, como podrían ser: incluir nuevas vistas para los documentos, permitir facilidades de búsqueda, establecer otros parámetros por el usuario además de los grupos y estados, brindar posibilidades de impresión, crear mecanismos de control sobre las copias impresas y mejorar las funciones actuales.
- Tomar en cuenta lo que sucede con los documentos que han caducado.
- Mejorar la edición y visualización de los documentos basándose en programas existentes para cubrir esta funcionalidad.
- Establecer un adecuado manejo de los errores que pueden ocurrir durante la operación de la herramienta.
- Garantizar que la herramienta sea independiente de la plataforma de desarrollo para que pueda operar en cualquier máquina, esto principalmente está dado por la parte de la base de datos.

Independientemente de las funcionalidades que faltan por considerar el prototipo propuesto puede ser una herramienta de gran ayuda para garantizar el control de los documentos que regulan la administración y desarrollo de los proyectos en las empresas. Puede ser la base para establecer las prácticas de documentación y el punto de partida para una herramienta más potente e integradora.

Bibliografía

- 1) [Baena97] Baena, Guillermina: "Instrumentos de investigación", Editores Mexicanos Unidos S.A., 21ª Reimpresión, México, 1997.
- 2) [Cadena96] Cadena Gómez, Eduardo: "ISO 9000, una visión general", Soluciones Avanzadas, Abril/1996, Año 4 No. 32, pp. 27-34.
- 3) [Cadena96-2] Cadena Gómez, Eduardo: "Tick IT, certificación en calidad para desarrolladores de software", Soluciones Avanzadas, Septiembre/1996, Año 4 No. 37.
- 4) [Deutsch88] Deutsch, Michael S. y Willis, Ronald R.: "Software Quality Engineering. A total Technical and Management Approach", Prentice Hall, 1988.
- 5) [Díaz96] Díaz Olaverrieta, Arnoldo: "Un estándar Internacional para la evaluación del Proceso de Desarrollo de Software ISO / SPICE", Soluciones Avanzadas, Marzo/1996, Año 4 No. 31, pp. 34-42.
- 6) [Duran96] Durán Rubio, Sergio E.: "BOOTSTRAP el Estándar Europeo para Evaluación y Mejoras de Procesos de Desarrollo de Software", Soluciones Avanzadas, Octubre/1996, Año 4 No. 38.
- 7) [Humphrey93] Humphrey, Watts S.: "Introduction to Software Process Improvement", SEI, CMU/SEI-92-TR-7, 1993.
- 8) [Humphrey97] Humphrey, Watts S.: "Introduction to the Personal Software Process", SEI Series, Addison Wesley, E.U. 1997.
- 9) [Ibergüengoitia] Ibergüengoitia, Guadalupe y Oktaba, Hanna: "El Rol del Analista para Desarrollos con Tecnología Orientada a Objetos", Soluciones Avanzadas, Abril/1998, Año 6 No. 56, pp. 50-56.
- 10) [Macfarlane] Macfarlane, Malcolm L.: "Eating the elephant one bite at a time" (<http://www.exit109.com/~leebee/case01.htm>)

- 11) [Morrison] Morrison Michael, et al.: "*Java 1.1 Unleashed*" Third Edition, Sams Net, E.U., 1997.
- 12) [Oskarsson96] Oskarsson, Östen y Glass, Robert L.: "*An ISO Approach to Building Quality Software*", Prentice Hall, E.U., 1996.
- 13) [Ould92] Ould, Martin y Miller, Chris: "*The evolution of quality standards: ISO 9001, Tickit, Improveit and Quantum*", SQM, issue 15, 1992.

(<http://www.avnet.co.uk/tesseract/QiC/articles/Miller/15.html>)
- 14) [Paulk92] Paulk, Mark C., et. al. : "*The Capability Maturity Model for Software.*", SEI Technical review, 1992.
- 15) [Paulk93] Paulk, Mark C., et. al. : "*The Capability Maturity Model for Software. Version 1.1*", SEI, CMU/SEI-93-TR-24, 1993.
- 16) [Paulk94] Paulk, Mark C.: "*A comparison of ISO 9001 and the Capability Maturity Model for Software*", Technical Report, SEI, CMU/SEI-94-TR-12, Julio 1994.
- 17) [ISO/TC95] "*Quality Systems Explained*" A guide to the interpretation and application of the ISO 9000 standars for small business. ISO/TC 176/SC 2 SBTG N 18, July 1995
- 18) [Schmauch95] Schmauch, Charles H.: "*ISO 9000 for Software Developers*", Revised Edition, ASQC Quality Press, E.U., 1995.
- 19) [Silberschatz91] Silberschatz, Abraham y Korth, Henry F. : "*Database System Concepts*", Second Edition, McGraw-Hill, E.U., 1991
- 20) [Von90] Von Mayrhauser, Anneliese: "*Software Engineering. Methods & Management*", Academic Press, 1990.
- 21) [Waller97] Waller, Jenny, et. al.: "*El Manual de Administración de la Calidad, ISO 9000*", Panorama Editorial S.A. de C.V., 3ª reimpresión, México, 1997.
- 22) [Wilkinson95] Wilkinson, Peter : "*Software quality: the basic issues*", SQM, issue 24, 1995. (<http://www.avnet.co.uk/tesseract/QiC/articles/Wilkinson/24.html>)

Sitios en Internet.

- 1) [WEB1] <http://www.hciqc.com/hcifre~1.htm> (THE ISO 9000 STANDARDS, 1997)
- 2) [WEB2] <http://www.logpars.army.mil/reflib/inter~1/International%20Standard%209000.txt> (INTERNATIONAL STANDARD ISO 9000, 1997)
- 3) [WEB3] <http://www.logpars.army.mil/reflib/inter~1/International%20Standard%209003.txt> (INTERNATIONAL STANDARD ISO 9003, 1997)
- 4) [WEB4] <http://www2.umassd.edu/SWPI/iso/isoques.html> (QUESTIONS AND ANSWERS ON QUALITY, 1997)
- 5) [WEB5] <http://sunflower.singnet.com.sg/~makhdoom/isobib.txt> (QUALITY AND ISO 9000 BIBLIOGRAPHY, 1997)
- 6) [WEB6] <http://sunflower.singnet.com.sg/~makhdoom/isosteps.txt> (THE BASIC STEPS IN GAINING ISO REGISTRATION, 1997)
- 7) [WEB7] <http://www.sgi.cit.gu.edu.au/spice> (THE SPICE PROJECT, 1997)
- 8) [WEB8] <http://www.sgi.cit.gu.edu.au/trillium> (THE TRILLIUM MODEL, 1998)
- 9) [WEB9] <http://www.sei.cmu.edu/technology/cmm/cmm.sun.html> (THE CMM FOR SOFTWARE, 1997)
- 10) [WEB10] <http://www.swquality.com/users/pustaver/> (SOFTWARE QUALITY PAGE, 1997)
- 11) [WEB11] <http://home.okstate.edu/homepages.nsf/toc/level2.index.html> (CMM LEVEL 2, 1997)
- 12) [WEB12] <http://www.sei.cmu.edu/> (SOFTWARE ENGINEERING INSTITUTE, 1997)
- 13) [WEB13] <http://www.ace.co.uk/isopro/> (ISO-PRO FOR WINDOWS, 1998)
- 14) [WEB14] <http://www.magi.com/~optimum/> (QS3 PRODUCTS, 1998)
- 15) [WEB15] <http://www.systemcorp.com/> (STEP-BY-STEP ISO 9000, 1998)

- 16) [WEB16] <http://www.iso9000usa.com/> (1..2..3 ISO 9000, 1998)
- 17) [WEB17] <http://www.dlogic.com/> (ISO LOGIC FORMS, 1998)
- 18) [WEB18] <http://www.r-m.com/> (R & M ENGINEERING CONSULTANTS, 1998)
- 19) [WEB19] <http://www.easternsystems.com/> (REQUISITEPRO, 1998)
- 20) [WEB20] <http://www.citoolkit.com/> (CI - TOOLKIT, 1998)
- 21) [WEB21] <http://www.crs-web.com:80/business/ams/index.html> (POWERWAY TOOLS, 1998)
- 22) [WEB22] <http://www.spc.ca/index.html> (SOFTWARE PRODUCTIVITY CENTRE, 1998)
- 23) [WEB23] <http://www.airware.com/> (AIMWARE, 1998)

Anexos

A. Herramientas de software para el apoyo en la administración de la calidad

B. Documento de análisis

C. Documento de diseño

Anexo A

Herramientas de software para el apoyo en la administración de la calidad

ISO-PRO

ISO-PRO es un sistema experto para el aseguramiento de la calidad desarrollado por Business Challenge Inc. [WEB13] Es una herramienta de software completa que ayuda a empresas de manufactura, de pequeño y mediano tamaño, en el proceso de registro a ISO 9000. Permite conocer los conceptos de ISO 9000, compara la posición actual de la empresa contra lo que establece ISO 9000, brinda una gran variedad de herramientas para el seguimiento del proceso y garantiza el desarrollo de un material que cumpla con los requisitos que establece la norma.

Consta de 5 módulos que facilitan la interacción y utilización del sistema y que son:

- **Seminario**, que incluye un amplio tutorial y ayuda en línea sobre ISO 9000, sugerencias, un glosario de términos y consultas.
- **Estimación**, es el centro del sistema y permite realizar una estimación del estado actual de la organización mediante una serie de preguntas de evaluación, con base en la cual, y con el apoyo de las herramientas que brinda el sistema, se obtiene un plan de acción, un calendario y reportes relacionados con el proceso de calidad.
- **Implantación**, registra eventos necesarios para la exitosa implantación de ISO 9000 como auditorías y registros.
- **Configuración**, permite establecer las características básicas de los módulos restantes

e incluye además un registro de todo el personal involucrado en la calidad.

- Taller ("Workshop"), brinda modelos y consejos para la redacción y edición de los materiales necesarios para cumplir los requisitos de la norma.

El sistema opera en máquinas compatibles con IBM (PC 386 o superior, con 4MB en RAM, 20 MB disponibles en disco duro, DOS 5.0 o superior o Windows 3.1 o superior, VGA) y su costo es de alrededor de \$795.00

QS3

QS3¹ es un conjunto completo de herramientas de software para el apoyo en el desarrollo, implantación, estimación o actualización del sistema de calidad para que cumpla con los requisitos de ISO 9000 y que ha sido elaborado por OPTIMUM Technology. [WEB14] El paquete de herramientas permite, además, prevenir o corregir las no conformidades y aumentar la efectividad de los cambios producidos en el sistema de calidad de la organización.

Las herramientas que integra son:

- QSAK - Paquete para la estimación del sistema de calidad, que permite de manera fácil y rápida completar, analizar, observar, rastrear y calendarizar las estimaciones y auditorías internas y externas.
- QSBK - Paquete de referencia para el sistema de calidad, donde se establecen definiciones, se establecen las referencias y se realizan las actividades que establece la norma para el desempeño de la calidad.
- QSDK - Paquete de desarrollo del sistema de calidad, que contiene herramientas necesarias para la administración, desarrollo y soporte al sistema de calidad
- QS3 Seminar - Es un paquete de apoyo a la capacitación del personal involucrado en la calidad.

¹ Quality System Solution Software

El paquete requiere para su operación de una PC 386 o superior, con 4MB en RAM, 25 MB para programas y 50 MB para datos disponibles en disco duro, Microsoft Word 6.0, Windows 95, VGA o superior.

Step-by-Step ISO 9000

Step-by-Step ISO 9000 de James Harrington y Systemcorp es una herramienta de apoyo al trabajo de análisis de defectos, capacitación y control y desarrollo de documentos iniciales. [WEB15]

Contiene instrucciones, modelos y ejemplos con casos de estudio de gran ayuda en las fases iniciales del proceso de implantación y capacitación de ISO 9000. Brinda la estructura y herramientas necesarias para el control de la documentación y de los cambios, y la administración de proyectos. Además soporta un manual en línea que pone a disposición de los usuarios todos los documentos autorizados y permite el acceso por Internet a consultores que pueden dar asesoría en el proyecto.

Los requisitos del sistema son una PC 486 a 33 Mhz, con 8MB en RAM, 6MB disponibles en disco duro y Windows 3.1 o superior.

1...2...3 ISO 9000

1...2...3 ISO 9000 es un paquete desarrollado por Global QA que permite a las empresas desarrollar su sistema de calidad basado en las normas ISO 9001, 9002 o 9003 sin la necesidad de realizar gastos en consultores externos. [WEB16]

Permite utilizar diagramas de flujo para lograr una mejor comprensión de los procedimientos, brinda información de consulta sobre los estándares, la forma de alcanzar la certificación y toda la información es accesible a través de la computadora. Muchas compañías lo han usado para su certificación. El costo del paquete es de \$1995.00

ISO'Logic Forms

ISO'Logic Forms ha sido desarrollado por O'Logic Inc y consiste en un conjunto de plantillas integradas a un procesador de texto (Microsoft Word o WordPerfect) que facilitan la creación y el acceso al manual de calidad. Directamente desde el procesador de texto y con la ayuda de diálogos interactivos y cajas de texto se reduce el tiempo y esfuerzo que implica documentar el sistema de calidad. Soporta además información de consulta sobre los estándares. [WEB17]

Para su funcionamiento requiere de por lo menos un procesador 386 compatible con IBM, con al menos 4MB de RAM, 2MB de espacio disponible en disco duro, Windows 3.1 o superior, Microsoft Word 6.0 o superior o WordPerfect 6.1. El costo de una copia del paquete es de \$99.95 y puede llegar a \$10 000.00 según la cantidad de copias.

Doc-Pro 9000

Doc-Pro 9000 (junto a los 4 paquetes que continúan) está disponible a través de R & M Engineering Consultant. [WEB18] Garantiza el control de la documentación, según establecen los requisitos en ISO 9000, desde su creación hasta su eliminación. Contiene 10 módulos para realizar diversas tareas relacionadas con la documentación. Lleva un registro con datos sobre el autor, el grupo del documento, la aprobación de liberación y las revisiones que se han realizado. Incluye 50 reportes predefinidos, permite importar y exportar datos de una base de datos Access y garantiza la protección de la información por claves.

Audit-Pro/Training 9000

Este paquete contiene 6 módulos integrados que permiten llevar el control de las auditorías internas y externas, describir las tareas del personal, controlar la capacitación, tomar acciones correctivas, documentar el manual y los planes de calidad, así como los procesos, procedimientos e instrucciones de trabajo de la organización, además permite registrar las quejas de los clientes. El sistema garantiza la protección de la información mediante claves, igualmente incluye reportes predefinidos y la interacción con datos de Access.

ISO Training Accelerator

Este programa permite controlar los requisitos de capacitación y la asignación de tareas al personal. Con base en la información sobre departamentos, tareas y empleados permite definir los requisitos de capacitación para cada tarea, determinar los cursos de capacitación, el calendario de clases y llevar el registro de los resultados. Incluye más de 150 reportes predefinidos.

ISO Flow Pro II

Es un programa que de manera fácil permite crear diagramas de flujo y de organización que facilitan la documentación sistemática de los procesos y procedimientos necesarios para implantar y mantener el sistema de calidad basado en ISO 9000. Contiene numerosos símbolos predefinidos que pueden ser personalizados para una mejor descripción de la información.

COQ - PRO 9000

COQ - PRO 9000 es un programa para el control de los costos del sistema de calidad según establece la norma ISO 9000. Clasifica los costos de acuerdo a su tipo ya sea por: prevención, apreciación, interno o externo; y por su origen en: departamento, tarea o servicio. Incluye calendarios para la recolección de datos y facilita la administración de proyectos mediante la asignación de tareas y el monitoreo de los resultados. Contiene reportes y gráficas predefinidas.

RequisitePro

Esta herramienta ha sido desarrollada por Eastern Systems y está enfocada a la administración de requisitos en equipos de desarrollo de software. [WEB19] Integra Microsoft Word con una base de datos multiusuario.

El sistema asigna atributos a cada tipo de requerimiento para poder establecer prioridades, ordenarlos y asignarlos. Analiza las relaciones entre los requisitos de un proyecto o entre diferentes proyectos para de esta manera controlar y garantizar un software de calidad. Indica el impacto de un cambio de requisitos en los demás y controla el cambio en los mismos mediante un registro histórico de cada uno. Integra herramientas de desarrollo de

Rational y otras industrias importantes.

Para su operación necesita contar con un procesador Pentium a 133 Mhz, 32 MB en RAM, 20 MB de espacio en disco duro como mínimo, Windows 95, Microsoft Word 95 o Word 97 y una red soportada por Windows.

CI - Toolkit

CI - Toolkit en su versión 5 es un paquete de herramientas para el mejoramiento continuo desarrollado por miembros del AFCI². [WEB20]

Cada una de las herramientas puede ser usada para un objetivo en particular e incluyen: planificación de estrategias, administración y control de equipos, evaluación y mejoramiento de la organización, reducción de los tiempos y costos de los proyectos, administración y mejoramiento del proceso, solución de los problemas de equipos, mejoramiento del proceso de administración del proyecto, aumento de la productividad, mejoramiento de la calidad de los productos y servicios, reducción del costo de capacitación con el aumento de su efectividad, mejoramiento de la comunicación y el flujo de trabajo en los equipos, autoevaluaciones y otras funcionalidades que se integran en documentos HTML montados sobre una INTRANET.

El paquete es compatible con los sistemas Windows 3.1 y superior y con numerosas redes existentes en el mercado.

Powerway Tools

Powerway Tools es un conjunto de herramientas de software para la administración y cumplimiento de los requisitos de ISO 9000 desarrolladas por AMS³[WEB21]. Las herramientas que lo integran son:

- **Assessor 9000** es una herramienta de auto evaluación soportada sobre Windows que permite dar los primeros pasos hacia ISO 9000. Parte de un extenso conjunto de preguntas detalladas que permiten obtener una correcta evaluación. Desarrolla una

² Alliance for Continuous Improvement

³ Advanced Manufacturing Services

guía de implantación de acuerdo a los estimados obtenidos y compara el estado de la organización con respecto a otras.

- **Policy Writer 9000** permite realizar el primer paso hacia la obtención de la certificación ISO 9000 con el comienzo de la creación de la documentación del sistema de calidad en un formato adecuado. Con esta herramienta se obtiene un manual de políticas de calidad.
- **Procedures Writer 9000** permite crear, editar y vincular los procedimientos que se aplican en la organización apoyado en 42 procedimientos preelaborados.
- **Document Manager 9000** facilita la administración de todos los documentos independientemente de su ubicación o tipo, crea un catálogo completo, establece listas de distribución y rutas de aprobación, y además garantiza la protección de la información contra accesos no permitidos.

SOFTGUIDE

SOFTGUIDE ha sido desarrollado por SPC⁴ y es un programa de software interactivo y fácil de usar que facilita el diagnóstico y solución de los problemas que existen en los procesos de software y la planeación del proceso de mejora. [WEB22]

Está basado en el modelo CMM versión 1.1 elaborado por SEI y está diseñado para compañías que buscan mejorar la productividad y calidad de sus productos de software, así como para realizar un estimado del estado de los procesos de software en la compañía. Puede ser usado tanto por ingenieros de software como por líderes de proyecto, gerentes de calidad y directores generales.

Identifica las áreas problemáticas con base en las respuestas a un cuestionario que presenta y propone un plan de mejoramiento. Para ello se deben realizar 4 actividades fundamentales que son:

- 1) Diagnosticar el problema, para ésto cuenta con un cuestionario (Process Questionnaire) que permite determinar los puntos débiles que existen y con base en el

⁴ Software Productivity Centre es un centro de recursos técnicos para compañías desarrolladoras de software que suministra capacitación, servicio técnico, consultoría y productos acordes con las necesidades de la industria.

cual se crea un perfil de las capacidades del proyecto (Capability Profile) que puede ser utilizado para evaluaciones posteriores.

- 2) Priorizar las áreas a mejorar, para ello crea una lista por áreas priorizadas de acuerdo al diagnóstico anterior y que puede ser nuevamente estructurada de acuerdo al criterio del usuario.
- 3) Preparar el plan de acción, con base en la lista de tareas obtenida y a las metas que se tracen se usan los modelos (Action Plan) para planear y documentar los cambios a los procesos.
- 4) Ejecutar el plan de acción, para lo cual establece listas de chequeo (Implementation Checklist) para llevar un registro de los elementos a considerar o completar en esta etapa.

AIMfirst

AIMfirst es una herramienta desarrollada por AIMWARE que permite la administración del proceso de software. AIMWARE es una compañía con sede en Irlanda y Estados Unidos que tiene como objetivo proveer herramientas para apoyar los esfuerzos de las organizaciones interesadas en mejorar sus procesos de negocio y están enfocadas especialmente a las empresas de desarrollo de software. Sus principales productos son *AIMfirst* y *AIMstep*. [WEB23]

Sus herramientas constituyen un esquema completo que permite la implantación de las mejores prácticas establecidas para el desarrollo de software de acuerdo con modelos y estándares internacionales como ISO 9001/Tick IT y SW-CMM.

AIMfirst mantiene los activos del proceso de la organización bajo control y permite utilizarlos y apoyarse en estos para responder a las necesidades de administración de los proyectos. Ayuda a documentar y planear la forma en que opera la organización. Por su parte al equipo de trabajo le permite desarrollar su trabajo de acuerdo con la forma establecida. Garantiza que el proceso de desarrollo de software sea visible, controlado y predecible.

Está compuesto por 6 módulos que son:

- 1) *Processware*, es un depósito para los documentos estándar de la organización (Políticas, Procesos, Procedimientos, Roles y Formatos). Además permite tener control de las versiones de los documentos y del ciclo de aprobación.
- 2) *Projectware*, permite controlar información específica del desarrollo de los proyectos.
- 3) *Improveware*, brinda una estructura que permite establecer el proceso de mejora en toda la organización.
- 4) *Docware*, permite crear, controlar y darle seguimiento a los documentos creados con base en los formatos definidos en *Processware* o en *Docware*.
- 5) *Standardware*, base de datos con información sobre los estándares de calidad y modelos; tanto externos como internos.
- 6) *Peoploware*, depósito de información sobre la organización, su estructura, datos de los empleados y cursos de capacitación.

AIMstep por su parte facilita el trabajo de los ingenieros y equipo de pruebas permitiendo la implantación de la ingeniería y las pruebas al software. Se enfoca en las pruebas de software ayudando a los equipos de trabajo a administrar y desarrollar las mismas, así como a controlar, revisar y asegurar la calidad.

Ambas herramientas están soportadas sobre Lotus Notes y requieren de la versión 4 o superior del software y/o un *browser* de Internet con Lotus Domino.

Anexo B

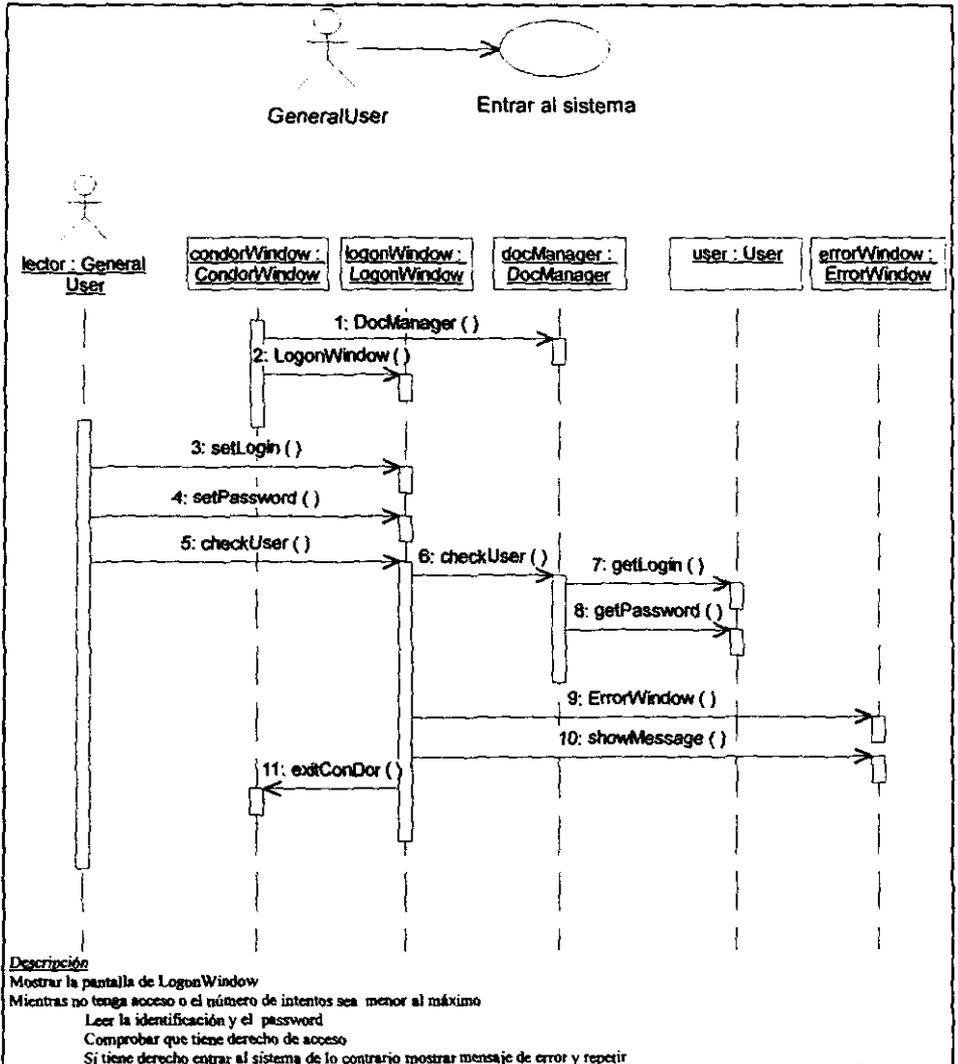
Documento de análisis

El documento contiene la información obtenida durante la fase de análisis del prototipo para el control de documentos del sistema de calidad. Está formado por las siguientes secciones:

- 1) Casos de uso y escenarios.
- 2) Diagramas de clases.
- 3) Diagramas de estado.
- 4) Diccionario de clases.

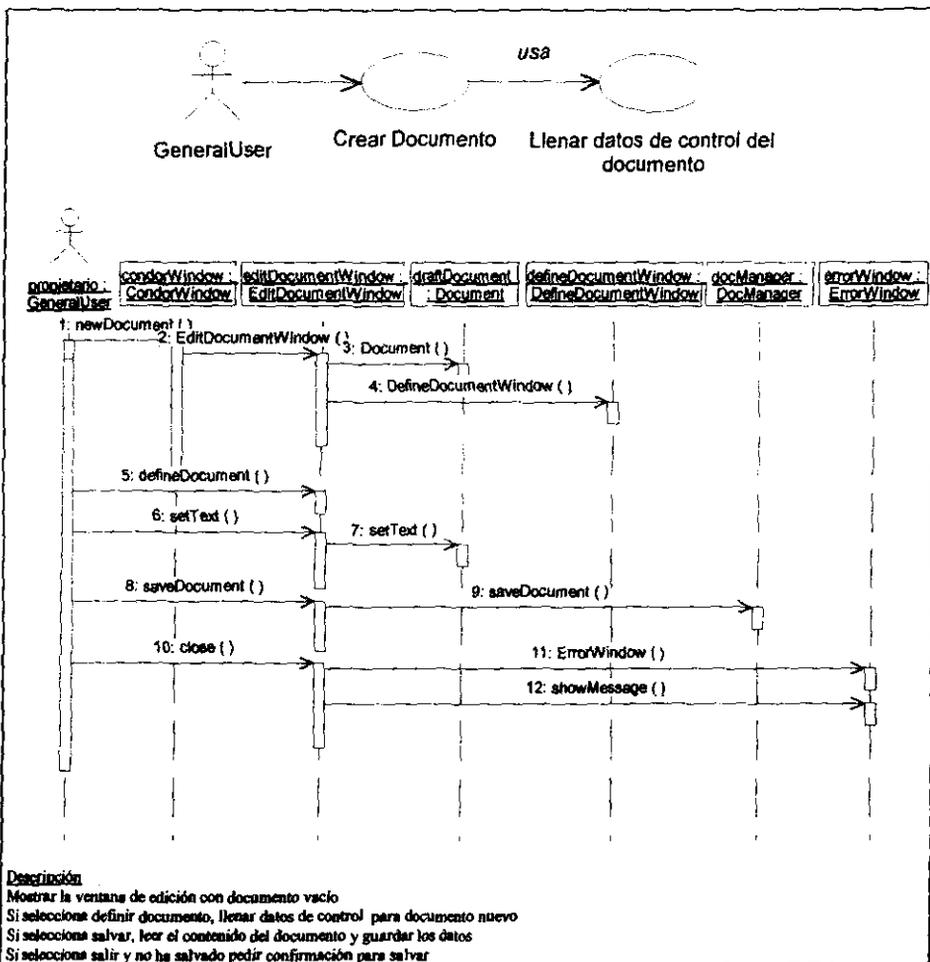
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 ENTRAR AL SISTEMA



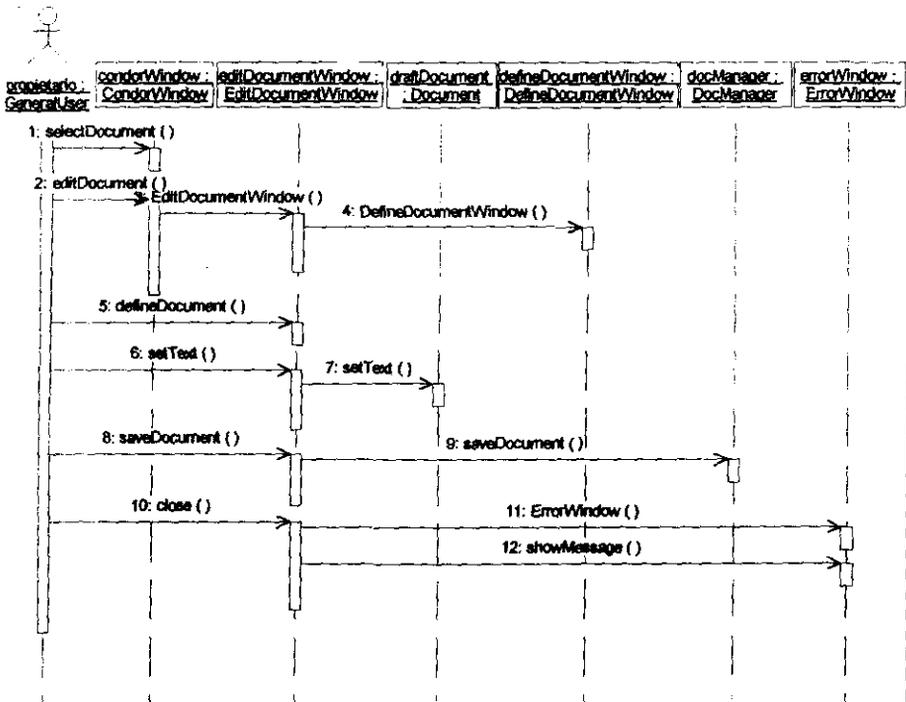
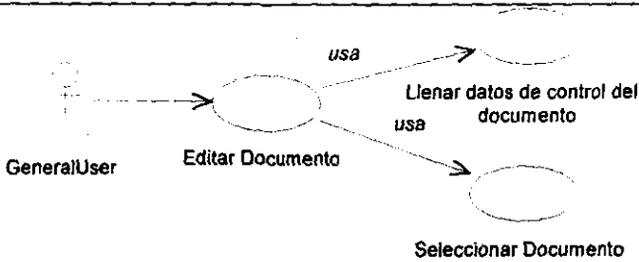
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 CREAR UN DOCUMENTO



Con.Dor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 EDITAR UN DOCUMENTO EXISTENTE

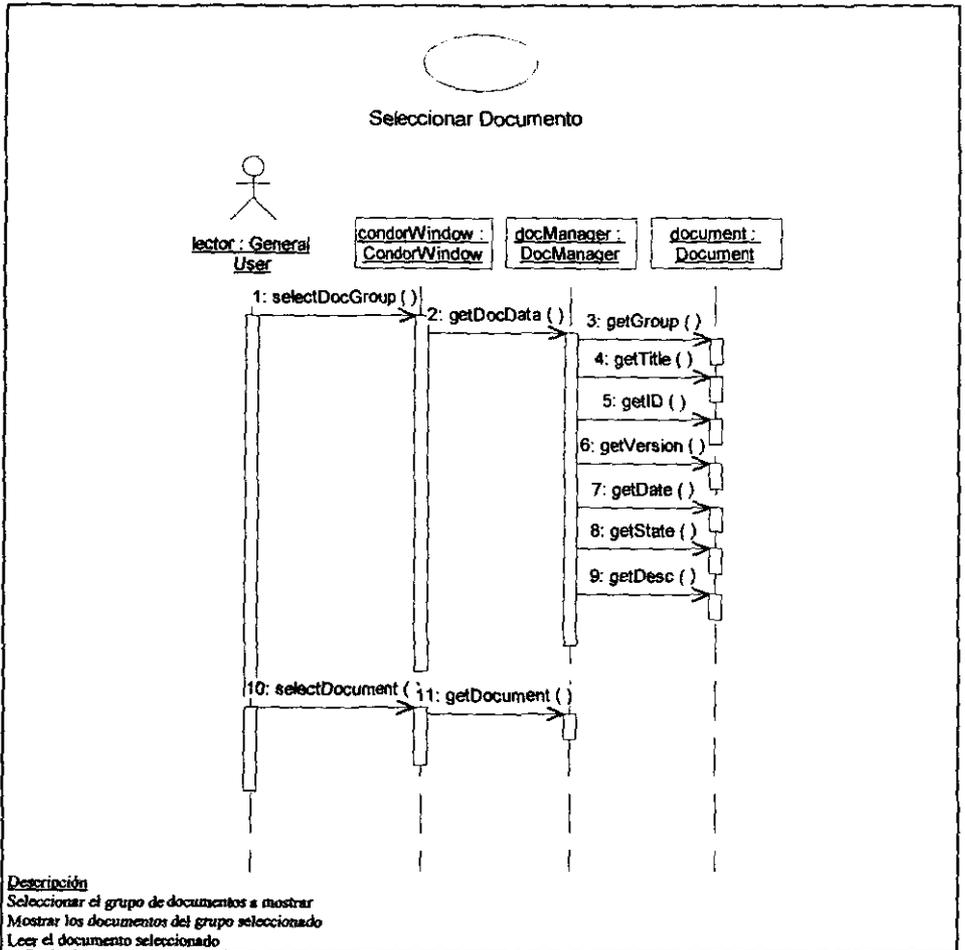


Descripción

Seleccionar el documento a editar
 Si es el propietario, mostrar la ventana de edición con el contenido del documento seleccionado
 Si selecciona definir documento, editar datos de control del documento nuevo
 Si selecciona salvar, leer el contenido del documento y guardar los datos
 Si selecciona salir y no ha salvado pedir confirmación para salvar

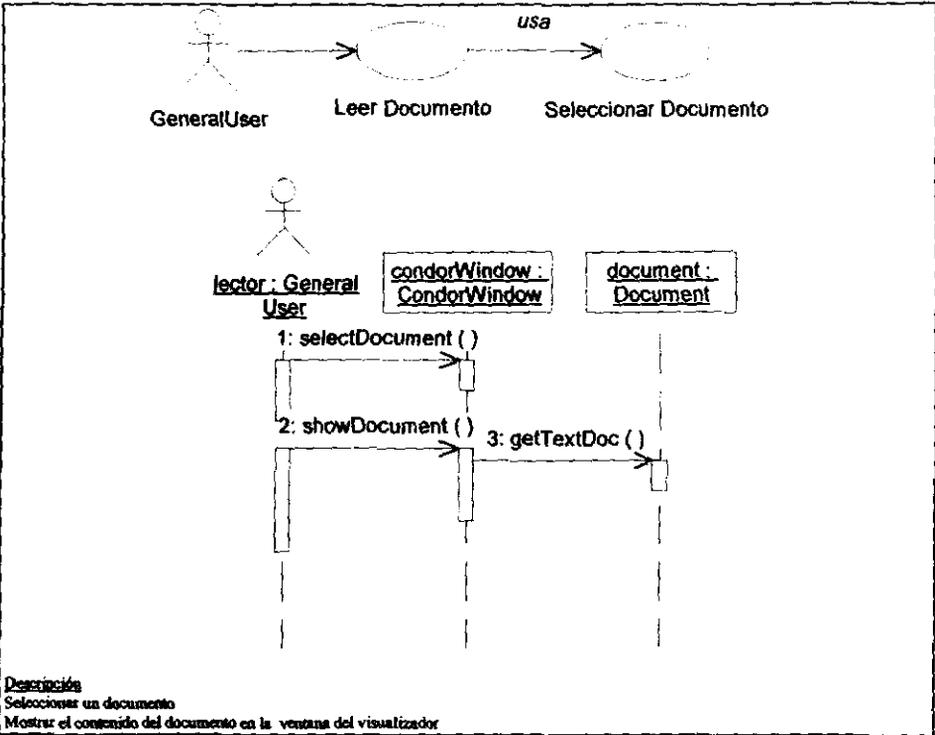
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez-Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 SELECCIONAR UN DOCUMENTO CORRESPONDIENTE A UN
 GRUPO



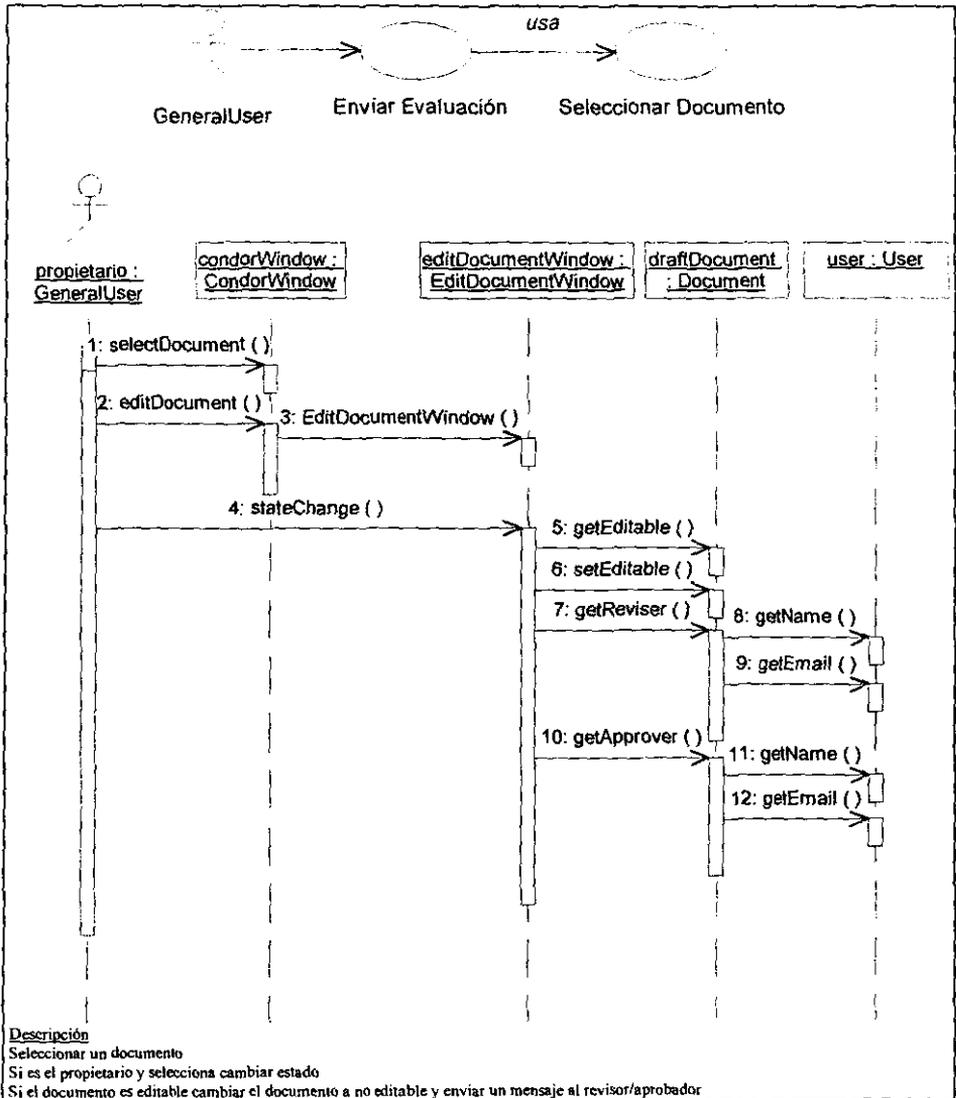
ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
CASOS DE USO Y ESCENARIOS (VERSION 4.0)
LEER EL CONTENIDO DE UN DOCUMENTO



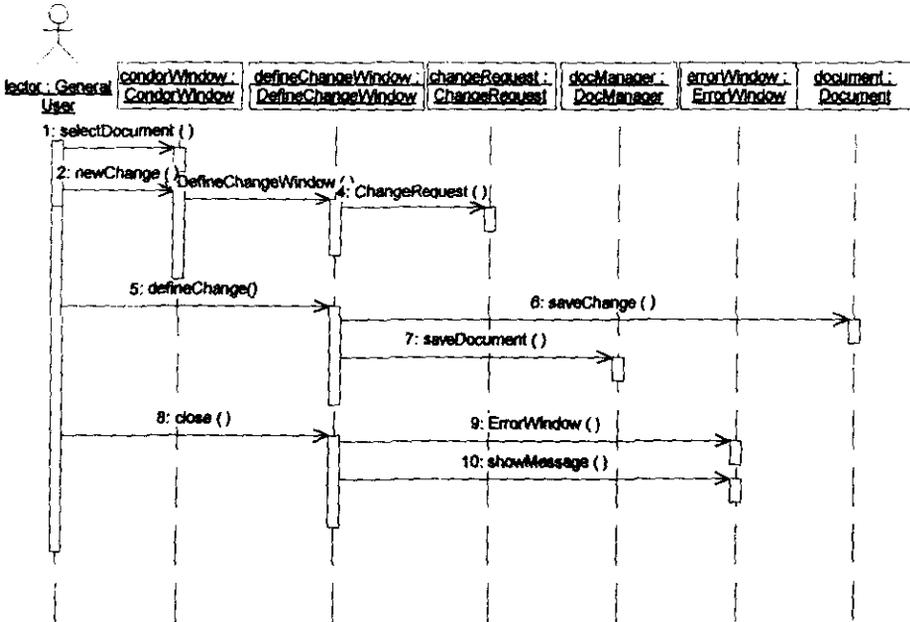
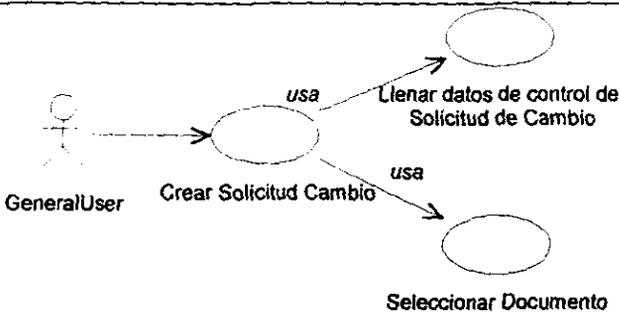
ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez-Escobar

DOCUMENTO DE ANÁLISIS
CASOS DE USO Y ESCENARIOS (VERSION 4.0)
ENVIAR A EVALUACIÓN UN DOCUMENTO



ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 CREAR SOLICITUD DE CAMBIO



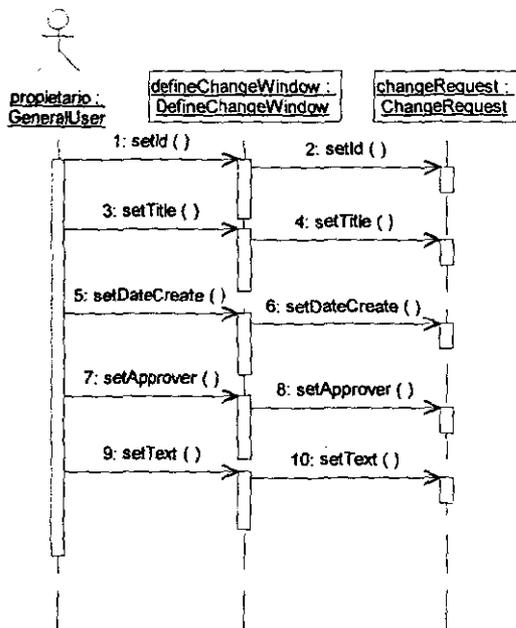
Descripción

Seleccionar el documento
 Si selecciona nueva solicitud de cambio
 Mostrar la ventana de definición de la solicitud de cambio
 Llenar datos de control para solicitud de cambio nueva
 Si selecciona salvar, leer el contenido de la solicitud de cambio y guardar los datos
 Si selecciona salir y no ha salvado pedir confirmación para salvar

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 LLENAR DATOS DE CONTROL DE LA SOLICITUD DE
 CAMBIO

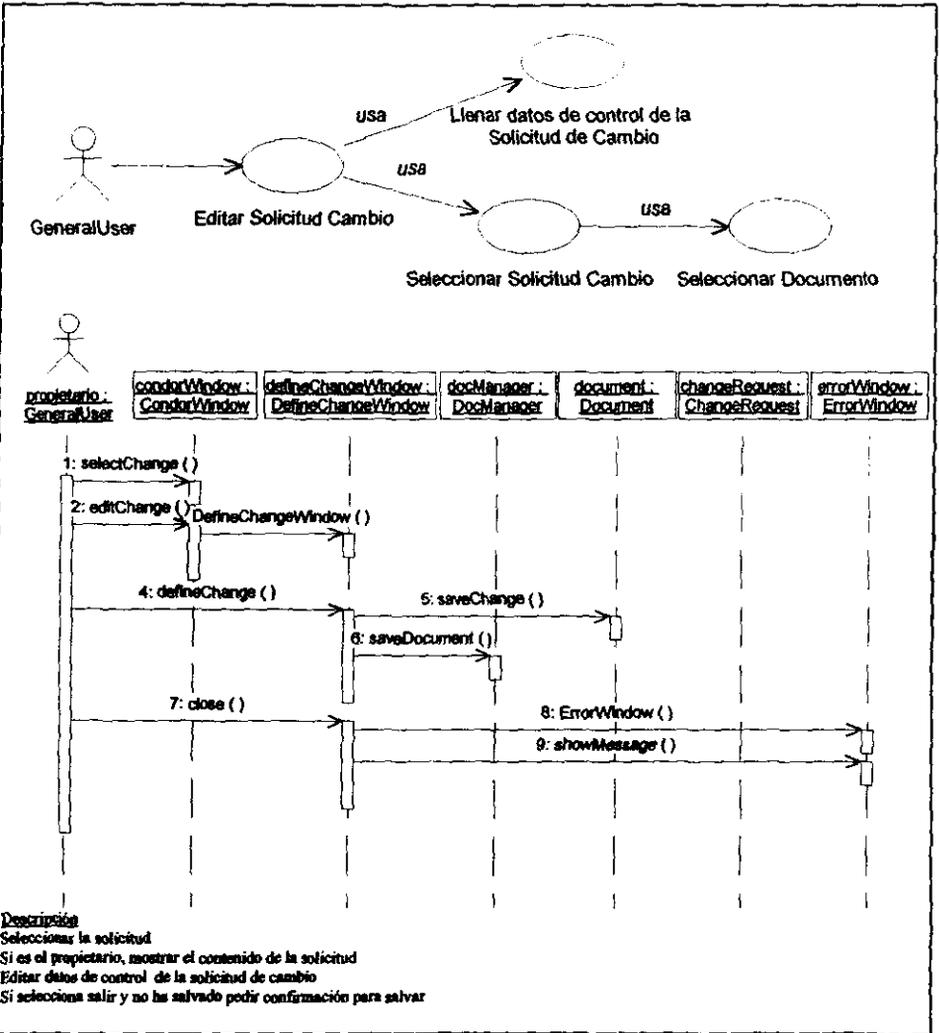
Llenar datos de control de
 Solicitud de Cambio



Descripción
 Leer los datos tecleados

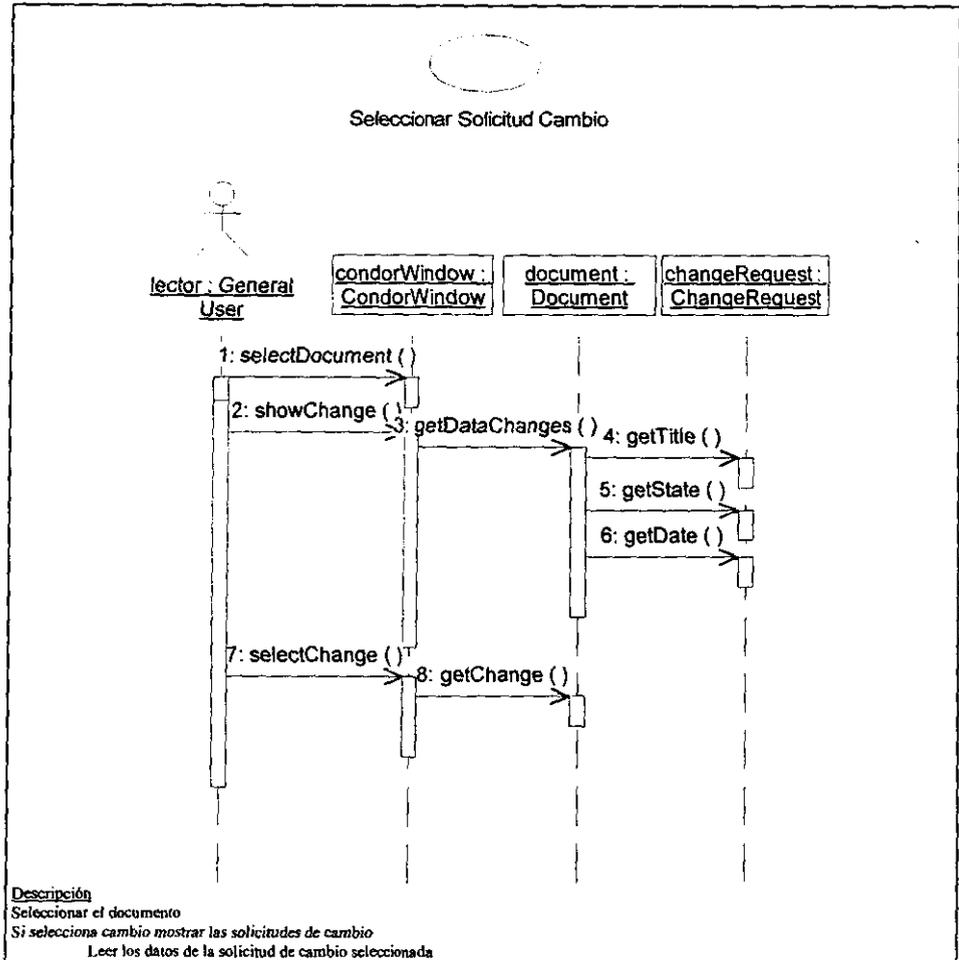
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANALISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 EDITAR SOLICITUD DE CAMBIO EXISTENTE



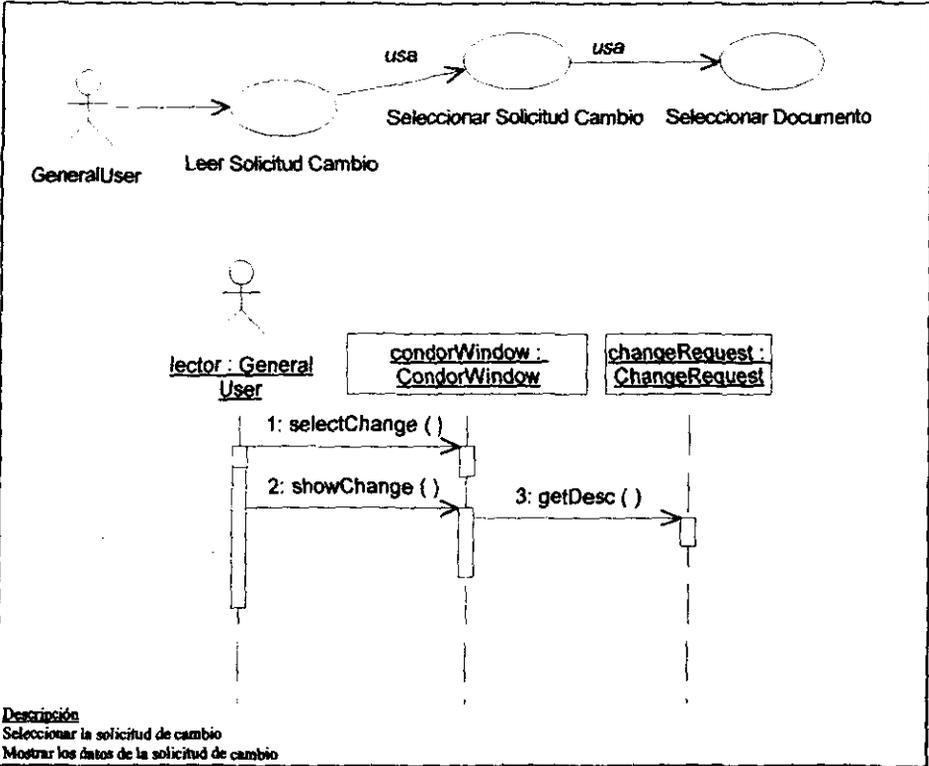
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
**SELECCIONAR UNA SOLICITUD DE CAMBIO DE UN
 DOCUMENTO**



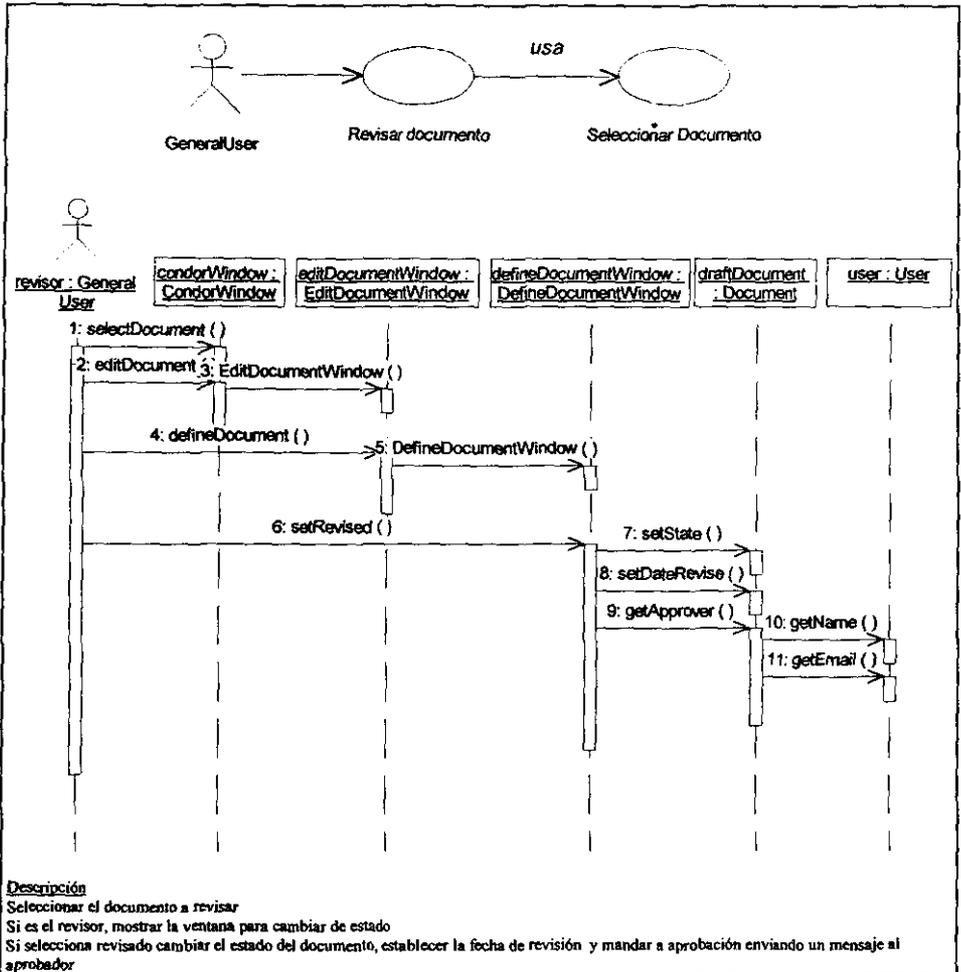
ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
CASOS DE USO Y ESCENARIOS (VERSION 4.0)
LEER UNA SOLICITUD DE CAMBIO EXISTENTE



ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

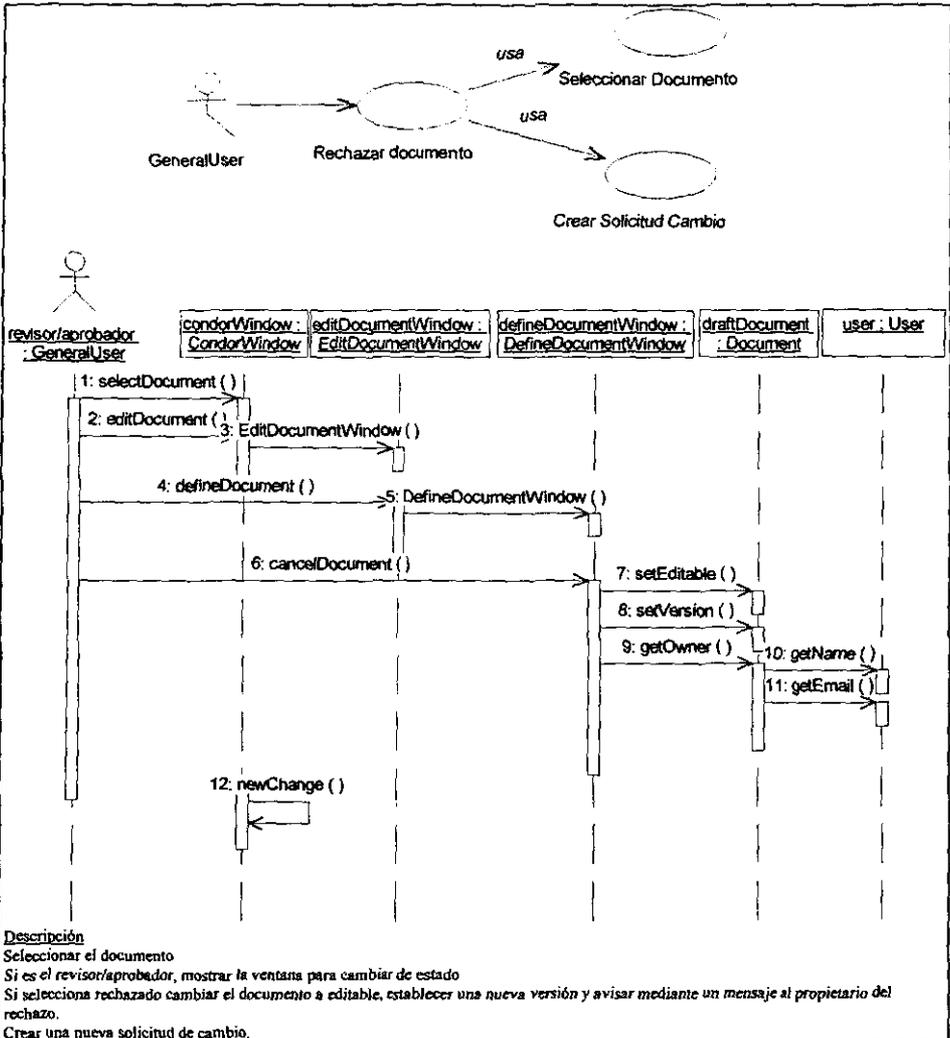
DOCUMENTO DE ANÁLISIS
CASOS DE USO Y ESCENARIOS (VERSION 4.0)
REVISAR UN DOCUMENTO



ConDor Versión 0.1 (1999)

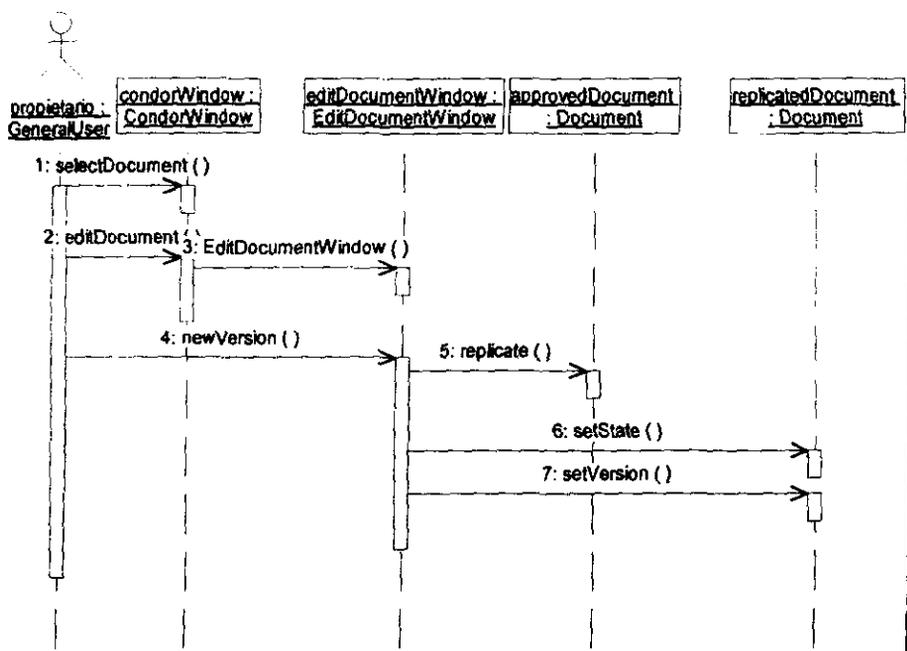
Sistema para el control de documentos del sistema de calidad

Carlos Javier Pérez Escobar

DOCUMENTO DE ANALISIS
CASOS DE USO Y ESCENARIOS (VERSION 4.0)
RECHAZAR UN DOCUMENTO EN EVALUACION

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 ENVIAR A MODIFICAR UN DOCUMENTO APROBADO

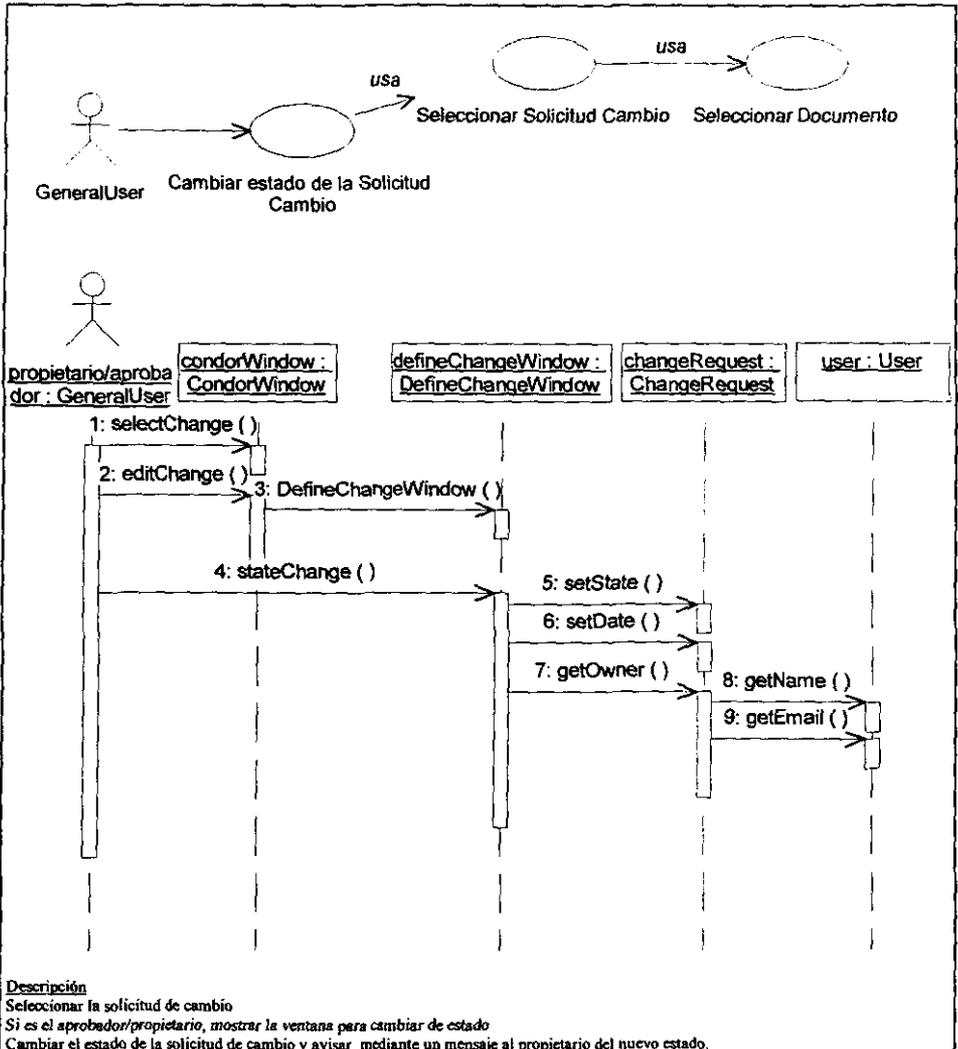


Descripción

Seleccionar el documento
 Si es el propietario y selecciona modificar
 Hacer una copia del documento con estado borrador, nueva versión y con las fechas de revisión, aprobación y efectiva vacías.

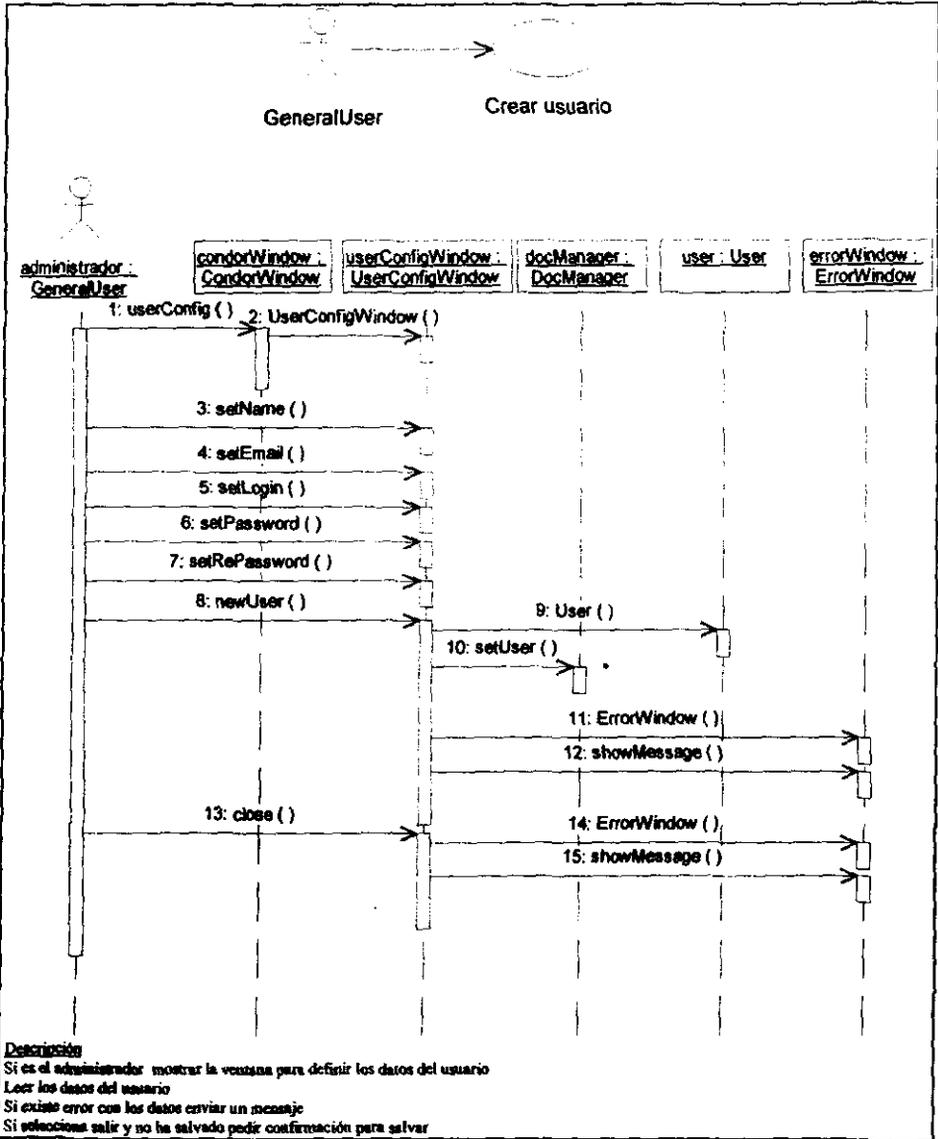
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 APROBAR/RECHAZAR/APLICAR UNA SOLICITUD DE CAMBIO



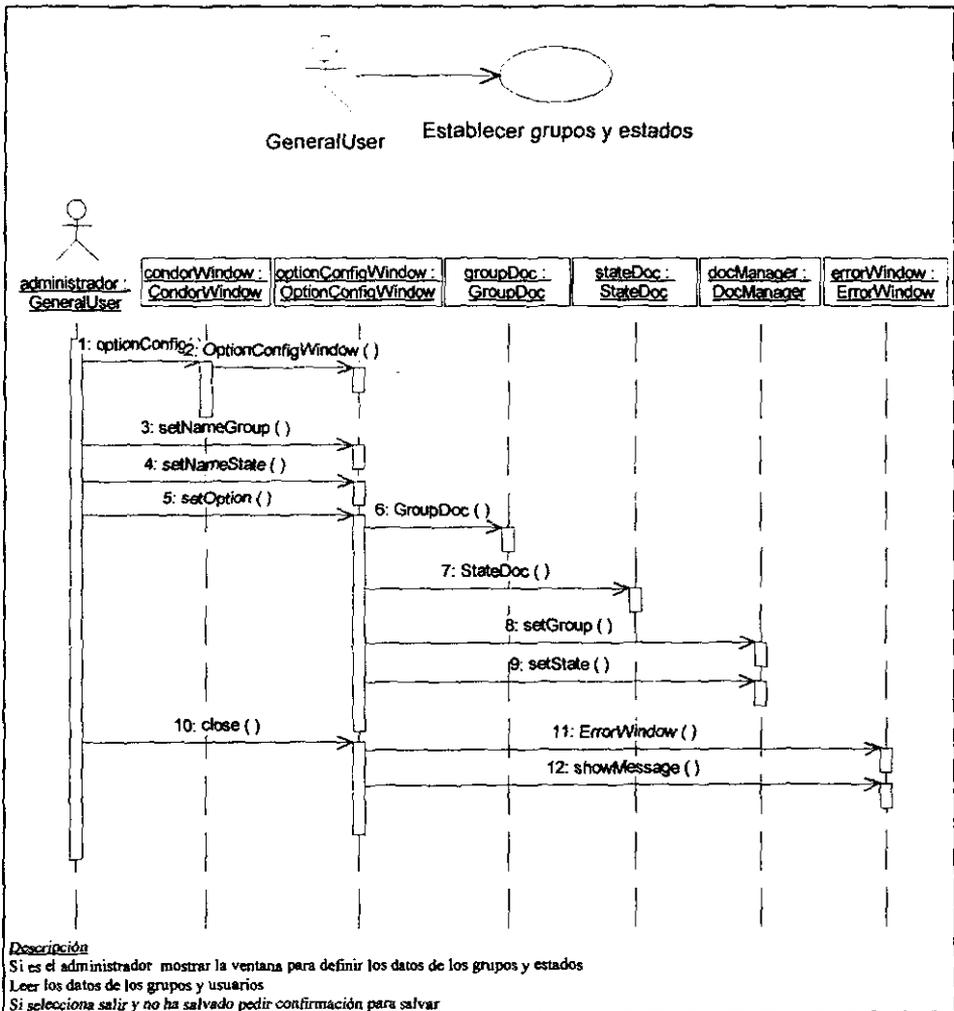
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 CREAR UN NUEVO USUARIO DEL SISTEMA



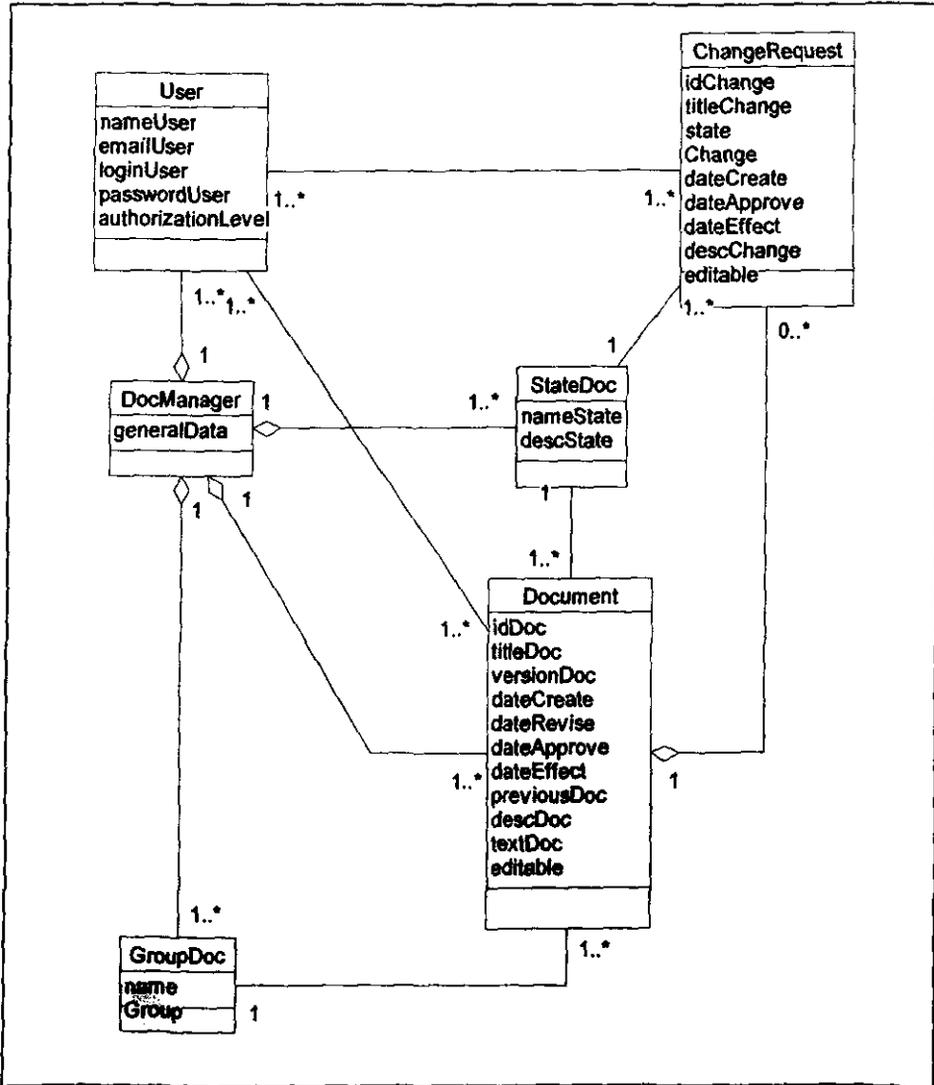
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANALISIS
 CASOS DE USO Y ESCENARIOS (VERSION 4.0)
 ESTABLECER GRUPOS Y ESTADOS



ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de calidad
 Carlos Javier Pérez Escobar

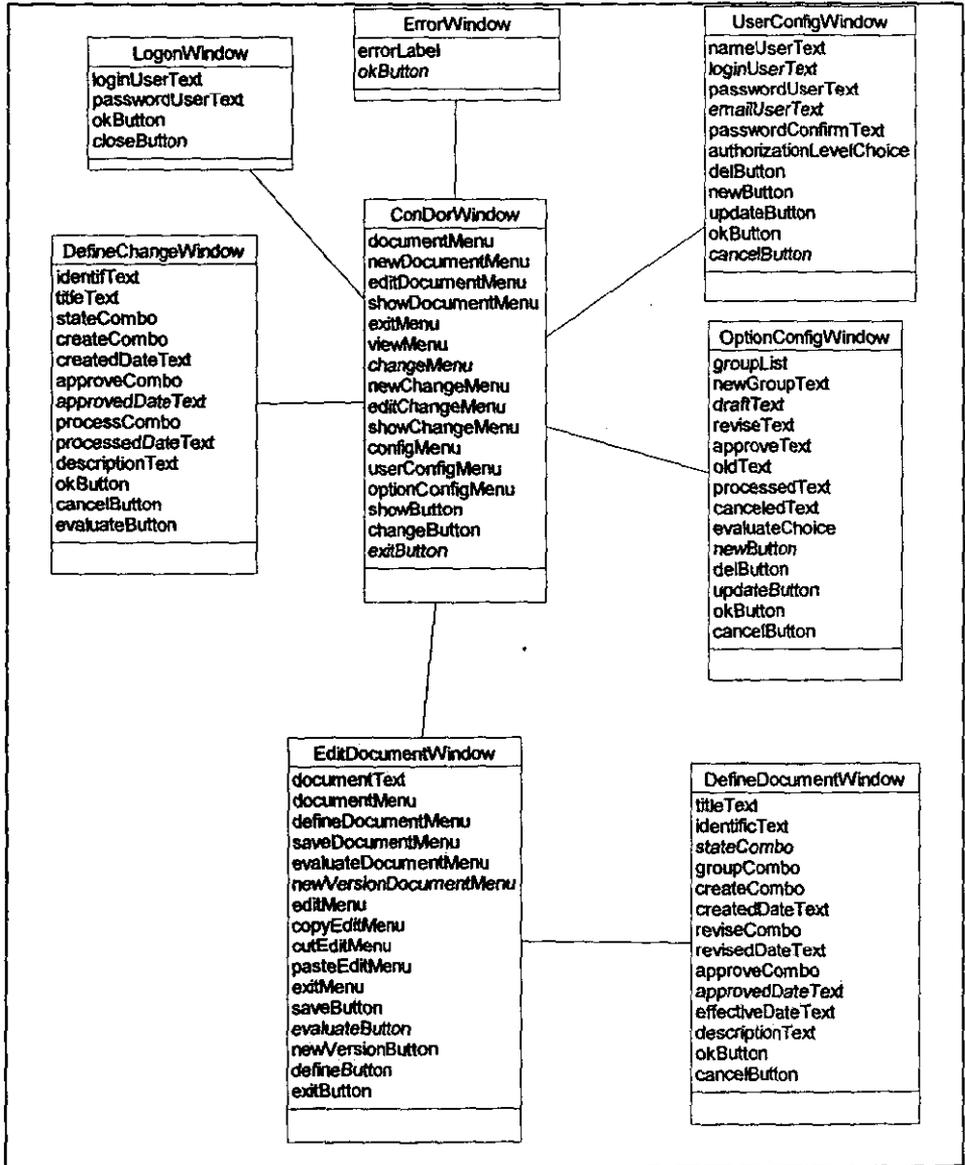
DOCUMENTO DE ANÁLISIS
 DIAGRAMAS DE CLASES (VERSION 4.0)
 DOMINIO DEL PROBLEMA



ConDor Versión 0.1 (1999)

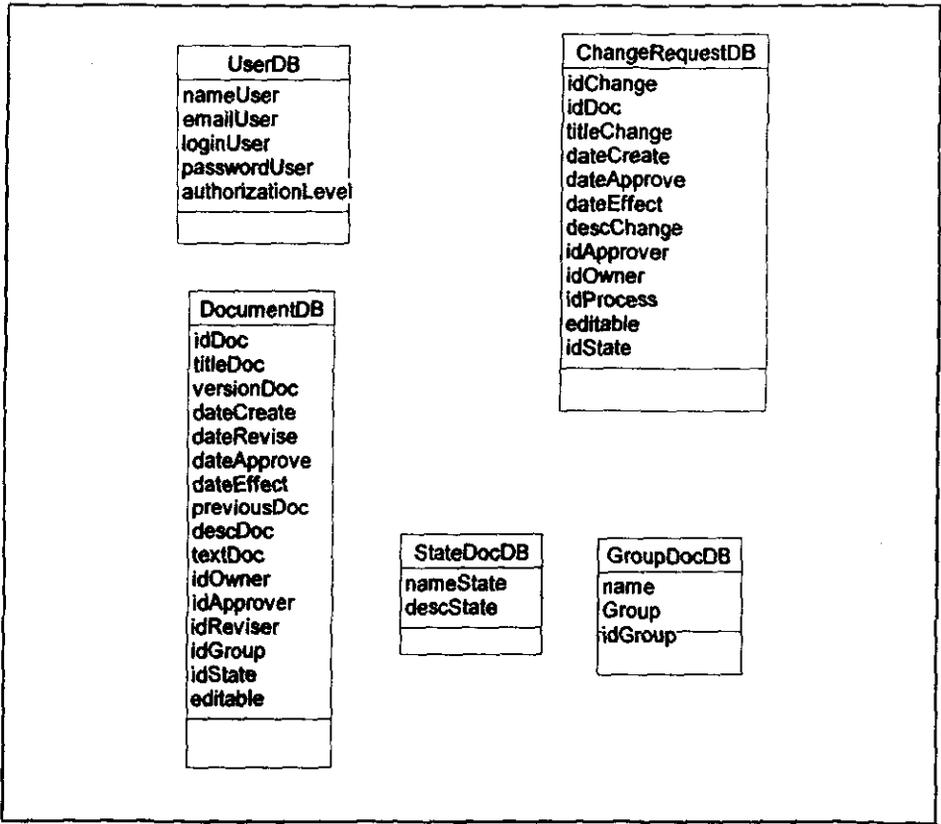
Sistema para el control de documentos del sistema de calidad

Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DIAGRAMAS DE CLASES (VERSION 4.0)
INTERFAZ HUMANA

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

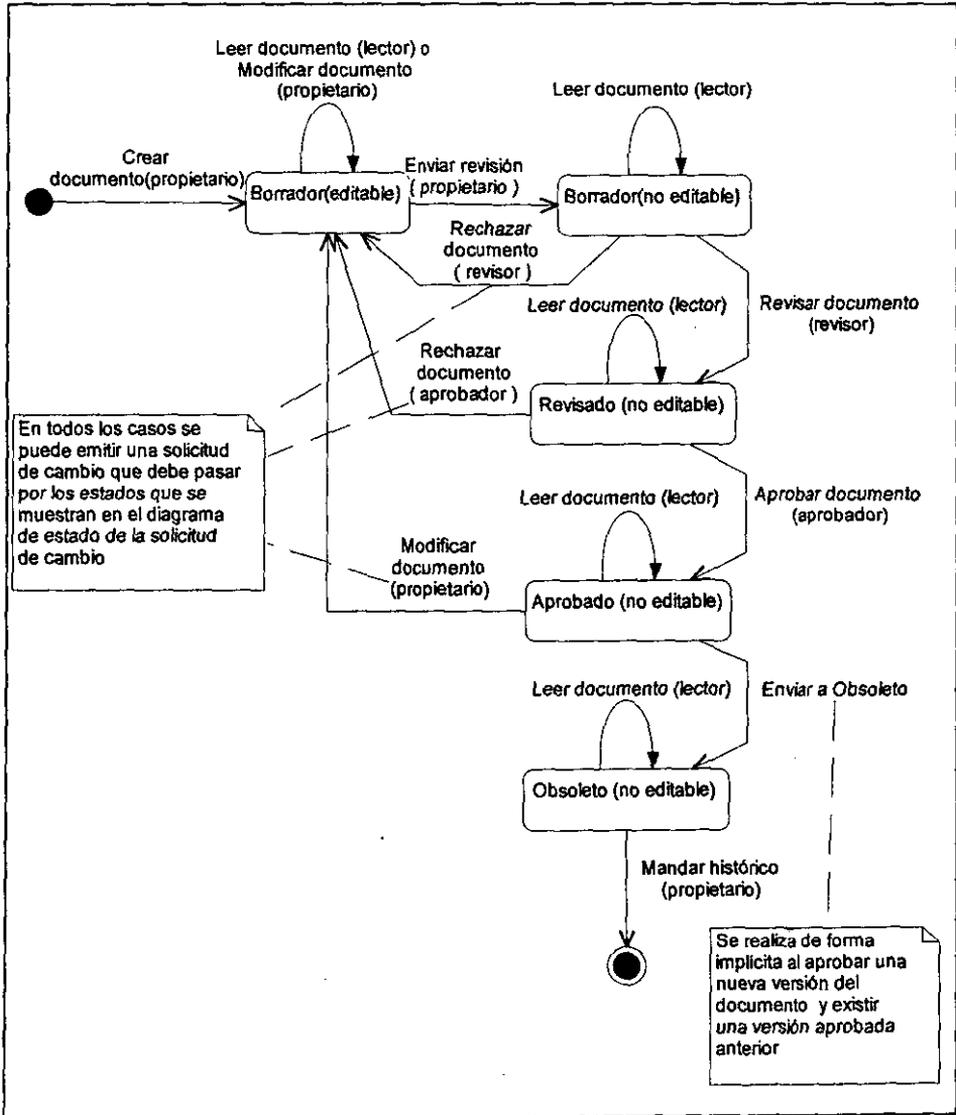
DOCUMENTO DE ANÁLISIS
DIAGRAMAS DE CLASES (VERSION 4.0)
MANEJO DE DATOS



ConDor Versión 0.1 (1999)

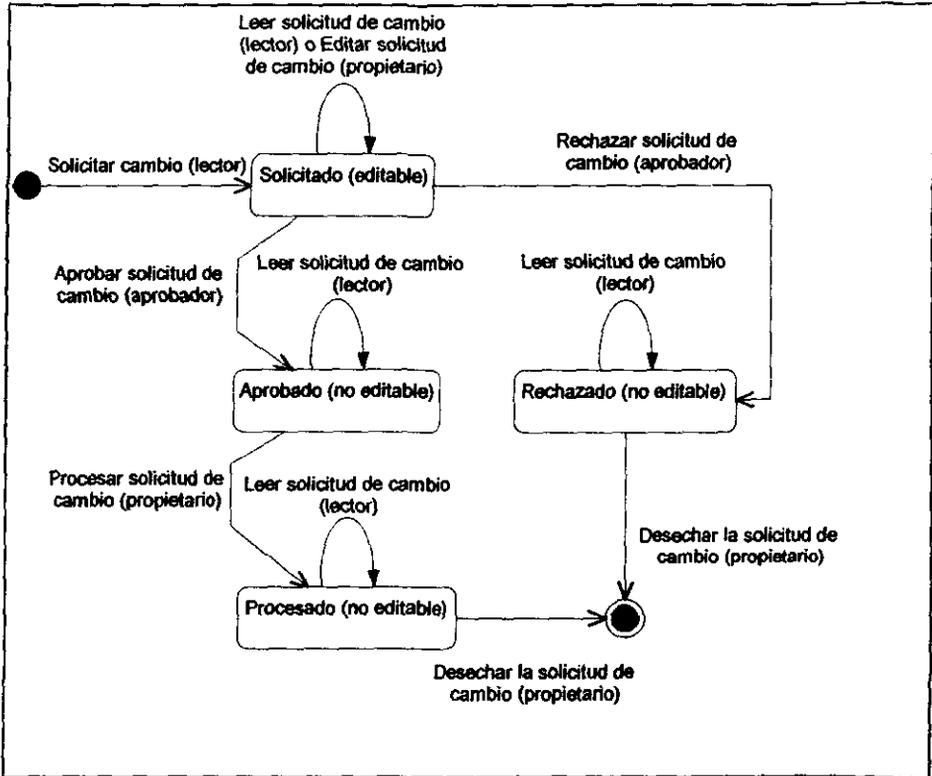
Sistema para el control de documentos del sistema de calidad

Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DIAGRAMAS DE ESTADOS (VERSION 4.0)
DOCUMENTO

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DIAGRAMAS DE ESTADOS (VERSION 4.0)
SOLICITUD DE CAMBIO

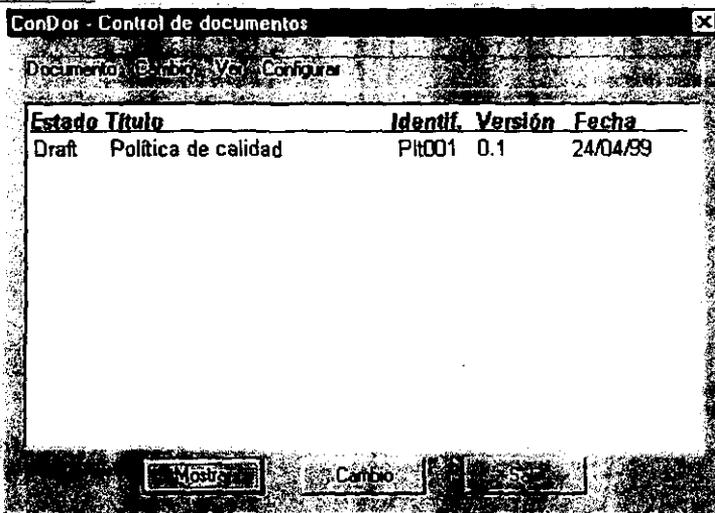


ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

ConDorWindow -- Es la ventana principal del sistema y desde la misma se tiene acceso a las principales funciones del sistema. Se muestra una vez que el usuario entró sus datos de identificación y se autorizó su acceso al sistema.

Prototipo de la ventana



Atributos Cada uno de los componentes de la ventana

documentMenu
changeMenu
viewMenu
configMenu
newDocumentMenu
editDocumentMenu
showDocumentMenu
exitMenu
newChangeMenu
editChangeMenu
showChangeMenu
userConfigMenu
optionConfigMenu

showButton - Permite mostrar el documento o la solicitud de cambio seleccionada

changeButton - Permite mostrar las solicitudes de cambio del documento seleccionado (con la etiqueta Cambio) o los documentos del grupo seleccionado (con la etiqueta Documento)

exitButton - La misma función de *exitMenu*

Métodos

ConDorWindow() - Crear la ventana y mostrarla

newDocument() - Crear un nuevo documento

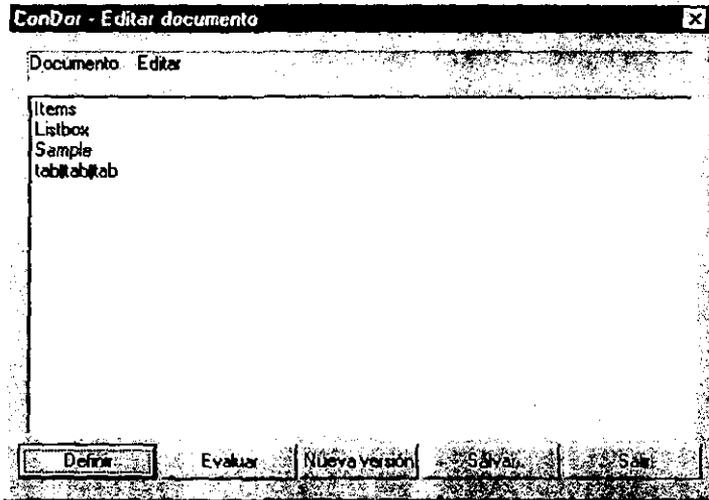
editDocument() - Editar un documento que fue previamente seleccionado
newChange() - Crear una solicitud de cambio para un documento seleccionado
editChange() - Editar la solicitud de cambio seleccionada de un documento
groupView() - Mostrar los documentos del grupo seleccionado
userConfig() - Editar los datos de un usuario
optionConfig() - Editar la configuración del sistema
exitConDor() - Cerrar la ventana y salir del sistema
showDocument() - Mostrar el documento seleccionado
changeView() - Mostrar las solicitudes de cambio del documento seleccionado o mostrar los documentos según el grupo seleccionado
showChange() - Mostrar la solicitud de cambio seleccionada

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

EditDocumentWindow - Es la ventana para editar el contenido de un documento. Se muestra cuando el usuario selecciona crear o editar un documento.

Prototipo de la ventana



Atributos Son cada una de los componentes de la ventana

documentMenu
editMenu
defineDocumentMenu
evaluateDocumentMenu
newVersionDocumentMenu
saveDocumentMenu
exitMenu
copyEditMenu
cutEditMenu
pasteEditMenu
documentText

defineButton - Permite definir las características del documento
evaluateButton - Permite enviar el documento a revisión/aprobación
newVersionButton - Permite crear una nueva versión del documento
saveButton - La misma función de *saveDocumentMenu*
exitButton - La misma función de *exitMenu*

Métodos

EditDocumentWindow() - Crear la ventana y mostrar, si recibe un documento como parámetro, el contenido del documento para editarlo
defineDocument() - Definir las características de un documento
saveDocument() - Salvar la información asociada al documento

evaluateDocument() - Enviar el documento a evaluación: avisa a los evaluadores y establece no editable
newVersionDocument() - Crea una versión en borrador del documento
copyText() - Copiar el texto seleccionado
cutText() - Copiar y borrar el texto seleccionado
pasteText() - Pegar el texto copiado en la posición indicada
close() - Cerrar la ventana y regresar a la ventana principal

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

DefineDocumentWindow - Es la ventana para definir las características del documento. Se muestra cuando el usuario selecciona definir características del documento que se está creando o editando.

Prototipo de la ventana

ConDor - Definir Documento

Identificador	PR001	Version	01
Título	Política de calidad	Estado	Borrador
Grupo por	CJPE	con fecha	24/04/99
Revisado por	ADP	con fecha	
Aprobado por	HJG	con fecha	
Grupo	Políticas	Inicio vigencia	
Descripción	Define la política que rige el trabajo de la empresa		

Aceptar Cancelar

Atributos Son cada una de los componentes de la ventana

identificText

titleText

versionLabel

stateCombo

groupCombo

createCombo

createdDateText

reviseCombo

revisedDateText

approveCombo

approvedDateText

effectiveDateText

descriptionText

okButton - Permite salvar la información y cerrar la ventana

cancelButton - Permite cerrar la ventana sin salvar la información

Métodos

DefineDocumentWindow() - Crear la ventana y mostrar, si recibe un documento como parámetro, las características del documento para editarlas

defineDocument() - Definir las características de un documento

close() - Cerrar la ventana y regresar a la ventana de edición, sin definir el documento

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

DefineChangeWindow - Es la ventana para definir las características de la solicitud de cambio. Se muestra cuando el usuario selecciona crear o editar una solicitud de cambio.

Prototipo de la ventana

ConDor - Definir solicitud de cambio

Identificador	SC001	Estado	Borrador
Título	Modificar elementos	con fecha	28/04/99
Creado por	GHT	con fecha	
Evaluado por	ADP	con fecha	
Implementado por		con fecha	
Descripción	Modificar los elementos de la política para ajustarlos a		

Evaluar Aceptar Cancelar

Atributos Son cada una de los componentes de la ventana

identificText

titleText

stateCombo

createCombo

createdDateText

approveCombo

approvedDateText

processCombo

processedDateText

descriptionText

okButton - Permite salvar la información y cerrar la ventana

cancelButton - Permite cerrar la ventana sin salvar la información

Métodos

DefineChange() - Crear la ventana y mostrar, si recibe una solicitud de cambio como parámetro, las características de la solicitud para editarlas

defineChange() - Definir las características de la solicitud de cambio

close() - Cerrar la ventana y regresar a la ventana principal, sin definir la solicitud de cambio.

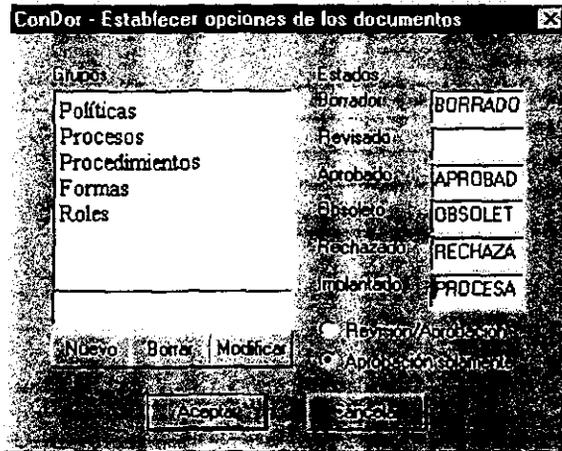
evaluateChange() - Enviar la solicitud de cambio a evaluación: avisa a los evaluadores y establece no editable

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

OptionConfigWindow - Es la ventana para establecer los grupos y estados del sistema. Se muestra cuando el usuario selecciona configurar opciones del sistema.

Prototipo de la ventana



Atributos Son cada una de los componentes de la ventana

groupList

newGroupText

draftText

reviseText

approveText

oldText

processedText

canceledText

evaluateChoice

newButton - Permite crear un nuevo grupo de documentos

delButton - Permite borrar el grupo seleccionado

updateButton - Permite modificar el grupo seleccionado

okButton - Permite salvar la información y cerrar la ventana

cancelButton - Permite cerrar la ventana sin salvar la información

Métodos

OptionConfigWindow() - Crear la ventana y mostrar las opciones establecidas previamente

newGroup() - Crear un nuevo grupo de documentos

delGroup() - Borrar el grupo de documentos seleccionados

updateGroup() - Actualizar el grupo de documentos seleccionado

setOption() - Establecer las nuevas opciones y cerrar la ventana

close() - Cerrar la ventana sin establecer las opciones y regresar a la ventana principal

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

UserConfigWindow - Es la ventana para editar las características de un usuario. Se muestra cuando el usuario selecciona configurar usuarios del sistema.

Prototipo de la ventana

ConDor Establecer usuarios del sistema

Nombre: Carlos J. Pérez Escobar

Correo: carlosj@email.mx

Identificador: carlosj

Contraseña: _____

Confirmación: _____

Administrador

Nuevo Borrar Actualizar Aceptar Cancelar

Atributos Son cada una de los componentes de la ventana

nameUserText

emailUserText

loginUserText

passwordUserText

passwordConfirmText

authorizationLevelChoice

newButton - Permite crear un nuevo usuario del sistema

delButton - Permite borrar la información correspondiente al usuario del sistema

updateButton - Permite actualizar la información del usuario

okButton - Permite salvar la información y cerrar la ventana

cancelButton - Permite cerrar la ventana sin salvar la información

Métodos

UserConfigWindow() - Crear la ventana y mostrar las características del usuario actual del sistema

newUser() - Crear un nuevo usuario del sistema

delUser() - Borrar el usuario actual del sistema

updateUser() - Actualizar la información del usuario

setUser() - Establecer las nuevas características del usuario y cerrar la ventana

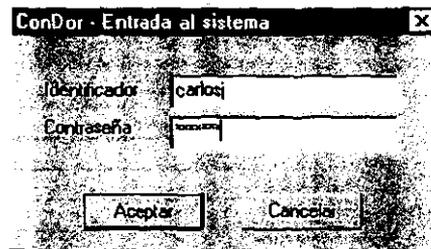
close() - Cerrar la ventana sin establecer las características y regresar a la ventana principal

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 INTERFAZ HUMANA

LogonWindow - Es la ventana donde se escribe la identificación y la clave del acceso del usuario para comprobar si tiene derecho de acceso al sistema. Se muestra al iniciar el sistema.

Prototipo de la ventana



Atributos

loginText - Región de la ventana donde se escribe la identificación del usuario
passwordText - Región de la ventana donde se escribe la clave de acceso del usuario
okButton - Permite comprobar si el usuario tiene acceso al sistema
cancelButton - Permite abandonar la ventana sin entrar al sistema

Métodos

LogonWindow() - Crear la ventana
checkUser() - Comprobar los datos del usuario para asegurar que tiene derecho de acceso al sistema
exitWindow() - Abandonar el sistema sin revisar los datos del usuario

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DICCIONARIO DE CLASES (VERSIÓN 2.0)
INTERFAZ HUMANA

ErrorWindow - Es la ventana donde se muestran los mensajes del sistema que requieren atención del usuario, ya sea por una falla ocurrida o para confirmar una acción.

Prototipo de la ventana



Atributos

messageLabel - Región donde se muestra el mensaje de atención

okButton - Permite aceptar lo que se plantea en el mensaje

Métodos

ErrorWindow() - Crear la ventana

showMessage() - Mostrar el mensaje indicado en la región correspondiente de la ventana

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad	DOCUMENTO DE ANÁLISIS DICCIONARIO DE CLASES (VERSIÓN 2.0) DOMINIO DEL PROBLEMA
Carlos Javier Pérez Escobar	

Document - Es el documento con todos los elementos que lo definen. Un documento puede tener asociadas una o más solicitudes de cambio. Puede tener uno de los siguientes estados: borrador, revisado, aprobado y obsoleto. Es editable mientras no entre en el ciclo de revisión/aprobación. El usuario que crea el documento es su propietario y define quienes lo revisan y/o aprueban. Inicialmente se crea con la versión 0.1 y se incrementa con cada ciclo de revisión/aprobación hasta que es aprobado y pasa a ser la versión 1.0, cuando se crea un borrador del documento aprobado cambia a la versión 1.1 y así se repite sucesivamente.

Atributos

idDoc - Texto con el identificador único del documento, establecido por el propietario

titleDoc - Texto con el título del documento, establecido por el propietario

versionDoc - Número de versión del documento, establecida por el sistema

dateCreate - Fecha en que se crea el documento, establecida por el propietario

dateRevise - Fecha en que se revisa el documento, establecida por el revisor

dateApprove - Fecha en que se aprueba el documento, establecida por el aprobador

dateEffect - Fecha a partir de la cual comienza la vigencia del documento, establecida por el propietario

previousDoc - Objeto documento aprobado anterior, establecida por el sistema cuando existe una versión aprobada previa y que se mantiene hasta que el documento actual es aprobado y la versión previa pasa a obsoleto. No es visible para el usuario

descDoc - Texto con la descripción del contenido del documento con carácter de información general establecida por el usuario

textDoc - Texto con la referencia al archivo que contiene el texto del documento, establecida por el sistema

stateDoc - Texto con el identificador del estado del documento, establecido por el propietario/revisor/aprobador

groupDoc - Texto con el identificador del grupo de documentos al que pertenece el documento, establecido por el propietario

reviserDoc - Objeto usuario que revisa el documento, establecido por el propietario

approverDoc - Objeto usuario que aprueba el documento, establecido por el propietario

ownerDoc - Objeto usuario que crea el documento, establecido por el sistema

changesDoc - Objetos solicitudes de cambio creadas para el documento y que se mantienen hasta que se aprueba el documento, establecidos por el sistema

editable - 0 si el documento es modificable y 1 si sólo se permite lectura

Métodos

Document() - Crear el objeto documento

setText() - Escribir la referencia al archivo con el contenido del documento

setId() - Escribir el identificador del documento

setTitle() - Escribir el título del documento

setDateCreate() - Escribir la fecha de creación

setGroup() - Escribir el grupo a que pertenece el documento

setReviser() - Escribir el usuario que revisa el documento

setApprover() - Escribir el usuario que aprueba el documento

setDesc() - Escribir la descripción del documento

setEditable() - Escribir si el documento es modificable

saveChange() - Incorporar la solicitud de cambio al documento

getDataChanges() - Leer las solicitudes de cambio del documento

getChange() - Leer la solicitud de cambio del documento

setState() - Escribir el estado del documento

setDateRevised() - Escribir la fecha en que se revisa el documento
setDateApproved() - Escribir la fecha en que se aprueba el documento
setDateEffect() - Escribir la fecha en que comienza la vigencia del documento
setVersion() - Escribir la versión del documento
replicate() - Replicar el documento para crear un borrador con nueva versión
getGroup() - Leer el grupo del documento
getTitle() - Leer el título del documento
getId() - Leer el identificador del documento
getVersion() - Leer la versión del documento
getDate() - Leer las fechas del documento
getState() - Leer el estado del documento
getDesc() - Leer la descripción del documento
getEditable() - Leer si el documento es modificable
getApprover() - Leer el usuario que aprueba el documento
getReviser() - Leer el usuario que revisa el documento
getPrevious() - Leer el documento aprobado anterior
getOwner() - Leer el usuario que creó el documento

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez-Escobar	DOCUMENTO DE ANÁLISIS DICCIONARIO DE CLASES (VERSIÓN 2.0) DOMINIO DEL PROBLEMA
---	--

ChangeRequest - Es la solicitud de cambio con todos los elementos que la definen. El propietario de la solicitud de cambio es el usuario que la crea y es aprobada por el propietario del documento al que se le solicita el cambio. Una solicitud de cambio puede tener uno de los siguientes estados: borrador, aprobado, rechazado, procesado.

Atributos

idChange - Texto con el identificador de la solicitud de cambio, establecido por el sistema

titleChange - Texto con el título de la solicitud de cambio

stateChange - Texto con el identificador del estado de la solicitud de cambio, establecido por el propietario/ aprobador

dateCreate - Fecha de creación de la solicitud de cambio, establecida por el propietario

dateApprove - Fecha en que se aprueba la solicitud de cambio, establecida por el aprobador

dateEffect - Fecha en que se incorpora la solicitud de cambio, establecida por el propietario

descChange - Texto con la descripción detallada de la solicitud de cambio, establecida por el propietario

ownerChange - Objeto usuario que crea la solicitud de cambio, establecido por el sistema

approverChange - Objeto usuario que aprueba la solicitud de cambio, establecido por el usuario

processChange - Objeto usuario que implanta la solicitud de cambio, establecido por el usuario

editable - 0 si la solicitud de cambio es modificable y 1 si sólo se permite lectura

Métodos

ChangeRequest() - Crear el objeto solicitud de cambio

setId() - Escribir el identificador de la solicitud de cambio

setTitle() - Escribir el título de la solicitud de cambio

setDateCreate() - Escribir la fecha en que se crea la solicitud de cambio

setApprover() - Escribir el usuario que aprueba la solicitud de cambio

getTitle() - Leer el título de la solicitud de cambio

getState() - Leer el estado de la solicitud de cambio

getDate() - Leer la fecha de la solicitud de cambio

getDesc() - Leer la descripción de la solicitud de cambio

getOwner() - Leer el usuario que creó la solicitud de cambio

setState() - Escribir el estado de la solicitud de cambio

setDate() - Escribir la fecha de la solicitud de cambio

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DICCIONARIO DE CLASES (VERSIÓN 2.0)
DOMINIO DEL PROBLEMA

User - Define la información de los usuarios que tienen acceso al sistema.

Atributos

nameUser - Texto con el nombre del usuario, establecido por el administrador
emailUser - Texto con la dirección de correo electrónico del usuario, establecido por el administrador
loginUser - Texto con el identificador del usuario, establecido por el administrador
passwordUser - Texto con la clave de acceso del usuario, establecida por el administrador
authorizationLevel - 0 si el usuario es el administrador y 1 si es otro usuario

Métodos

User() - Crear el objeto usuario
setLogin() - Escribir el identificador del usuario
setPassword() - Escribir la clave de acceso del usuario
setName() - Escribir el nombre del usuario
setEmail() - Escribir la dirección de correo electrónico del usuario
setAuthorizationLevel() - Escribir el nivel de acceso del usuario
getLogin() - Leer el identificador del usuario
checkPassword() - Revisar la clave de acceso del usuario contra un texto
getName() - Leer el nombre del usuario
getEmail() - Leer la dirección de correo electrónico del usuario
getAuthorizationLevel() - Leer el nivel de acceso del usuario

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DICCIONARIO DE CLASES (VERSIÓN 2.0)
DOMINIO DEL PROBLEMA

GroupDoc - Contiene la información de los grupos que pueden existir para contener los documentos. Un documento sólo puede pertenecer a un grupo.

Atributos

nameGroup - Texto con el nombre del grupo, establecido por el administrador

Métodos

GroupDoc() - Crear el objeto grupo de documentos

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez Escobar	DOCUMENTO DE ANÁLISIS DICCIONARIO DE CLASES (VERSIÓN 2.0) DOMINIO DEL PROBLEMA
---	--

StateDoc - Contiene la información de los estados que puede tener un documento o solicitud de cambio.

Atributos
nameState - Texto con el nombre del estado, establecido por el sistema
desc.State - Texto con el nombre del estado para el usuario, establecido por el administrador

Métodos
StateDoc() - Crear el objeto estado de los documentos

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez Escobar	DOCUMENTO DE ANÁLISIS DICCIONARIO DE CLASES (VERSIÓN 2.0) DOMINIO DEL PROBLEMA
---	--

DocManager - Contiene todos los documentos del sistema, los grupos establecidos, los estados posibles y los usuarios con acceso al sistema.

Atributos

documents - Objetos documento creados en el sistema

users - Objetos usuario con acceso al sistema

groupsDoc - Objetos grupo establecidos en el sistema

statesDoc - Objetos estado establecidos en el sistema

generalData - Texto con información general que se pueda utilizar

Métodos

DocManager() - Crear el objeto

checkUser() - Revisar si existe el usuario

saveDocument() - Guardar la información del documento

getDocData() - Leer la información de los documentos del sistema

getDocument() - Leer el documento indicado

setUser() - Escribir la información del usuario

setGroupDoc() - Escribir la información del grupo de documentos

setStateDoc() - Escribir la información del estado de los documentos

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez Escobar	DOCUMENTO DE ANÁLISIS DICCIONARIO DE CLASES (VERSIÓN 2.0) MANEJO DE DATOS
---	---

DocumentDB- Contiene la información que se necesita guardar de cada documento.

Atributos

idDoc
titleDoc
versionDoc
dateCreate
dateRevise
dateApprove
dateEffect
idPreviousDoc - Identificador del documento aprobado anterior
descDoc
textDoc
idOwner - Identificador del usuario que es propietario del documento
idApprover - Identificador del usuario que aprueba el documento
idReviser - Identificador del usuario que revisa el documento
idGroup - Identificador del grupo al que pertenece el documento
idState - Identificador del estado en que se encuentra el documento
editable

Métodos

DocumentDB()
saveDocument()
loadDocument()
deleteDocument()
updateDocument()

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
 DICCIONARIO DE CLASES (VERSIÓN 2.0)
 MANEJO DE DATOS

ChangeRequestDB - Contiene la información de las solicitudes de cambio

Atributos

idChange

idDoc - Identificador del documento al que pertenece la solicitud de cambio

titleChange

idState - Identificador del estado en que se encuentra la solicitud de cambio

dateCreate

dateApprove

dateEffect

descChange

idOwner - Identificador del usuario que es propietario de la solicitud de cambio

idApprover - Identificador del usuario que aprueba la solicitud de cambio

idProcess - Identificador del usuario que implanta la solicitud de cambio

editable

Métodos

ChangeRequestDB()

saveChange()

loadChange()

deleteChange()

updateChange()

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DICCIONARIO DE CLASES (VERSIÓN 2.0)
MANEJO DE DATOS

UserDB - Contiene la información de los usuarios del sistema

Atributos

nameUser
emailUser
loginUser
passwordUser
authorizationLevel

Métodos

UserDB()
saveUser()
loadUser()
deleteUser()
updateUser()

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez Escobar	DOCUMENTO DE ANÁLISIS DICCIONARIO DE CLASES (VERSIÓN 2.0) MANEJO DE DATOS
---	---

GroupDocDB - Contiene la información de los grupos establecidos para los documentos

Atributos

idGroup - Identificador único del grupo, establecido por el sistema

nameGroup

Métodos

GroupDocDB()

saveGroup()

loadGroup()

deleteGroup()

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE ANÁLISIS
DICCIONARIO DE CLASES (VERSIÓN 2.0)
MANEJO DE DATOS

StateDocDB - Contiene la información de los estados establecidos para los documentos y solicitudes de cambio

Atributos

nameState

descState

Métodos

StateDocDB()

saveState()

loadState()

deleteState()

updateState()

Anexo C

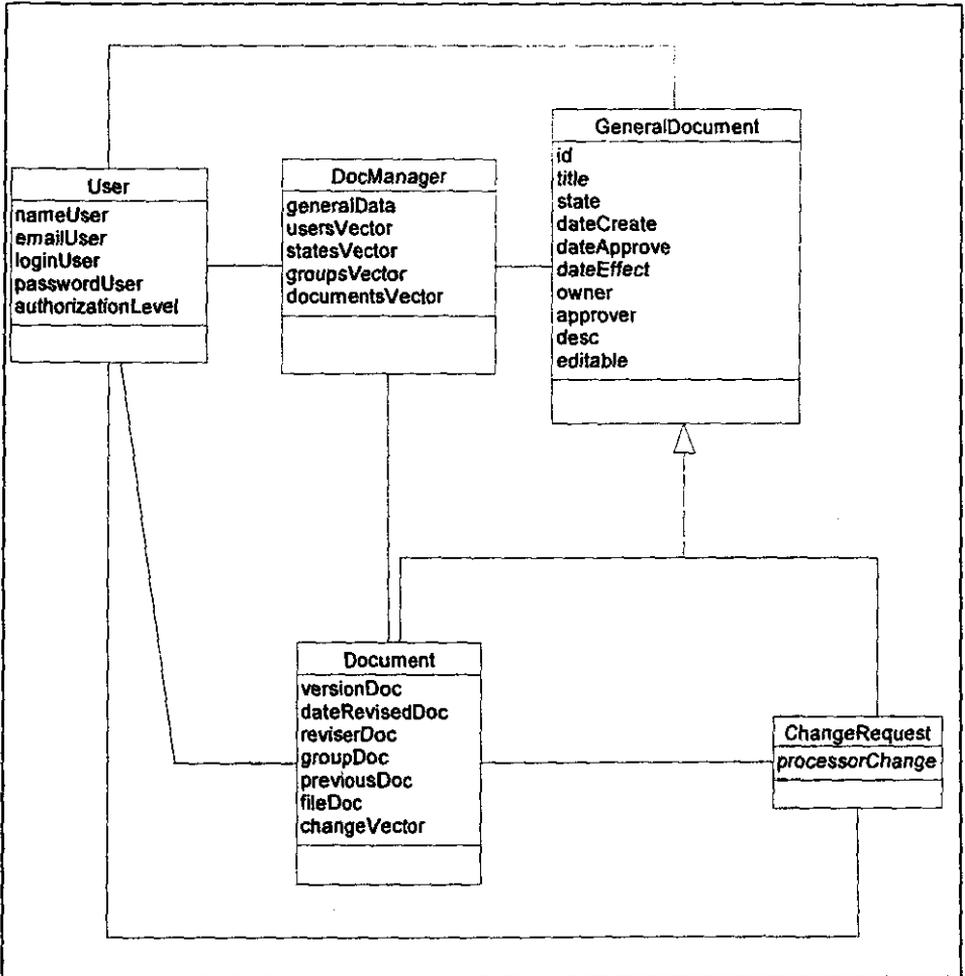
Documento de diseño

El documento contiene la información obtenida durante la fase de diseño del prototipo para el control de documentos del sistema de calidad. Está formado por las siguientes secciones:

- 1) Diagramas de clases del diseño.
- 2) Diseño de la base de datos.
- 3) Diccionario de clases del diseño.

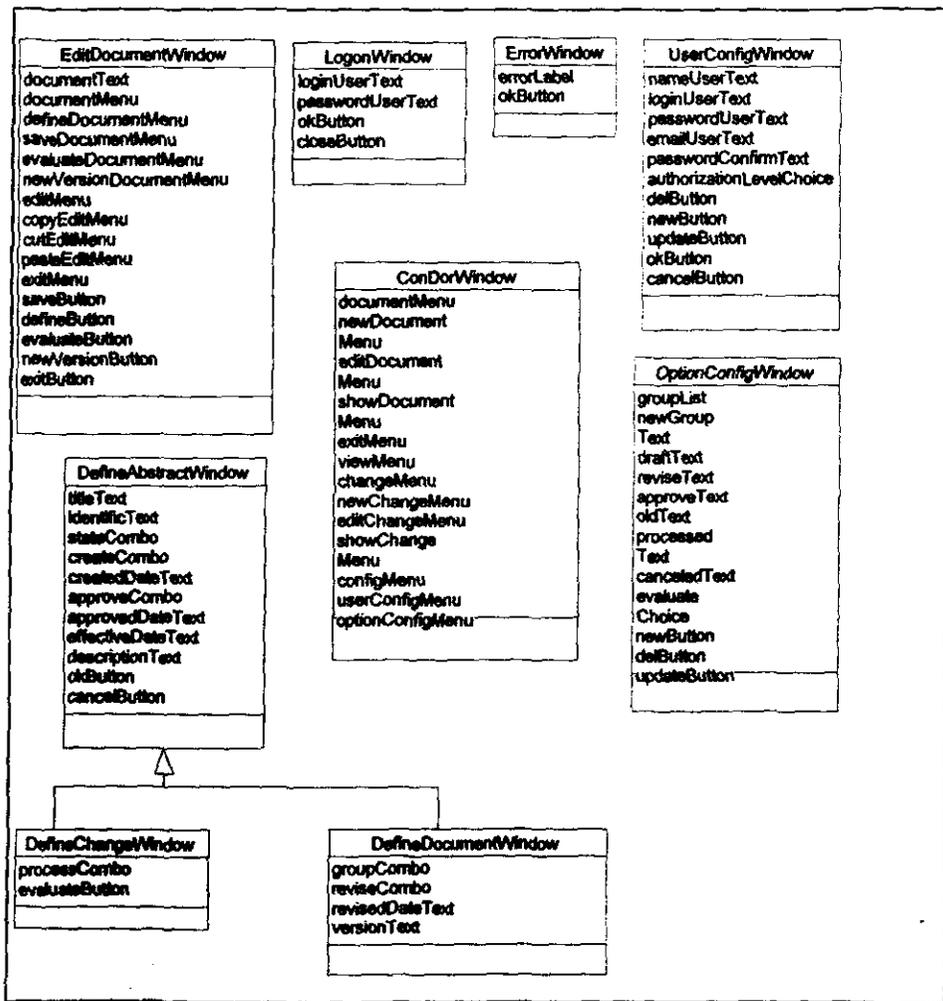
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE DISEÑO
 DIAGRAMAS DE CLASES DEL DISEÑO (VERSION 2.0)
 DOMINIO DEL PROBLEMA



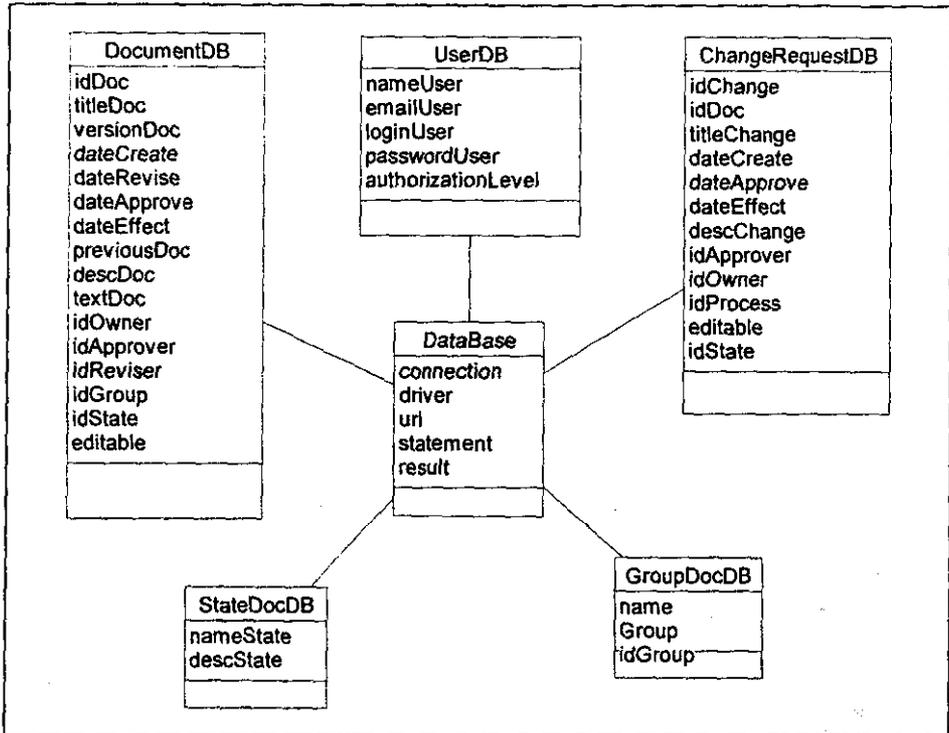
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE DISEÑO
 DIAGRAMAS DE CLASES DEL DISEÑO (VERSION 2.0)
 INTERFAZ HUMANA



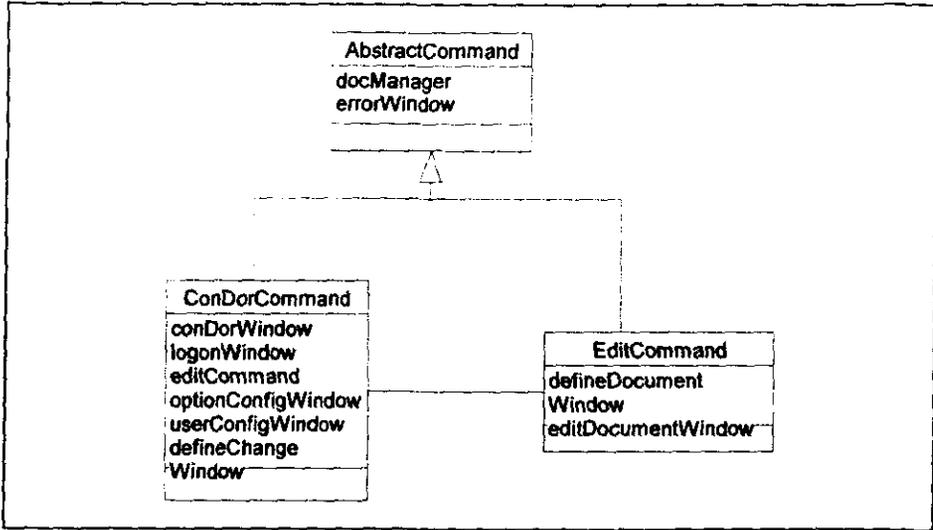
ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE DISEÑO
 DIAGRAMAS DE CLASES DEL DISEÑO (VERSION 2.0)
 MANEJO DE DATOS



ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE DISEÑO
DIAGRAMAS DE CLASES DEL DISEÑO (VERSION 2.0)
EJECUTANTES



ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez Escobar	DOCUMENTO DE DISEÑO DISEÑO DE LA BASE DE DATOS DEL DISEÑO(VERSION 2.0)
---	---

CLASE DP	RELACION	NULO	DOMINIO
Document id state title versionDoc groupDoc owner dateCreate approve approvedDate effectDate previousDoc changeVector editable desc fileDoc	DOCUMENTS { idDoc } { idState } titleDoc versionDoc idGroup idOwner dateCreate idApprover dateApprove dateEffect idPreviousDoc - editable descDoc fileDoc	No No No No No No No Si Si Si Si - No Si No	Numérico Texto Texto Texto Numérico Texto Texto Texto Texto Numérico - S/N Texto Texto
ChangeRequest id title state owner dateCreate approve approvedDate effectDate processorChange editable desc	CHANGES { idChange } titleChange idState idOwner dateCreate idApprover dateApprove dateEffect idImplement editable descChange idDoc	No No No No No Si Si Si Si No Si No	Numérico Texto Texto Texto Texto Texto Texto Texto Texto S/N Texto Numérico
User loginUser nameUser passwordUser emailUser authorizationLevel	USERS { idUser } nameUser passwordUser emailUser levelUser	No No No Si No	Texto Texto Texto Texto S/N
DocManager groupsVector statesVector	GROUPS { idGroup } nameGroup STATES { idState } nameState	No No No No	Numérico Texto Texto Texto

ConDor Versión 0.1 (1999)
 Sistema para el control de documentos del sistema de
 calidad
 Carlos Javier Pérez Escobar

DOCUMENTO DE DISEÑO
 DICCIONARIO DE CLASES DEL DISEÑO(VERSION 2.0)
 EJECUTANTES

ConDorCommand - Clase principal de la aplicación que controla la mayor parte de sus acciones. Sus métodos y atributos son estáticos por lo que no se requiere crear una instancia de la clase para invocar a los mismos. Define los valores constantes que se utilizan en el sistema como son los estados de los documentos y los parámetros de conexión a la base de datos. Las funciones principales son crear y editar un documento o solicitud de cambio, mostrar los documentos o las solicitudes de cambio del documento seleccionado, mostrar los documentos del grupo seleccionado, modificar las opciones y usuarios del sistema, abandonar el sistema.

Métodos

main()

DataBase -> Verificar la conexión con la base de datos.
 Crear y actualizar DocManager
 Si no hay problemas con la conexión
 Crear LogonWindow y mostrarla
 Si el usuario tiene acceso al sistema
 Crear ConDorWindow y mostrarla

newDocument()

Crear EditDocumentCommand y ejecutar

editDocument()

Leer el documento seleccionado
 Crear EditDocumentCommand con el documento seleccionado y ejecutar

newChange()

Crear DefineChangeWindow y mostrarla

editChange()

Leer la solicitud de cambio seleccionada
 Crear DefineChangeWindow con la solicitud de cambio seleccionada y mostrarla

userConfig()

Crear UserConfigWindow y mostrarla

optionConfig()

Crear OptionConfigWindow y mostrarla

stateChange()

Si hay documentos
 Mostrar solicitudes de cambio del documento en ConDorWindow
 de lo contrario
 viewGroup()

viewgroup()

Mostrar documentos del grupo seleccionado en ConDorWindow

exitCondor()

DataBase -> Cerrar la conexión con la base de datos
 Abandonar la aplicación

ConDor Versión 0.1 (1999) Sistema para el control de documentos del sistema de calidad Carlos Javier Pérez Escobar	DOCUMENTO DE DISEÑO DICCIONARIO DE CLASES DEL DISEÑO (VERSION 2.0) EJECUTANTES
---	--

EditDocumentCommand - Clase que controla las acciones de edición de un documento. Las funciones principales son mostrar la ventana de edición y definición de las características de los documentos, indicar que el documento está en evaluación, generar una nueva versión del documento.

editDocument()

Crear EditDocumentWindow y mostrarla

defineDocument()

Crear DefineDocumentWindow y mostrarla

defineDocument()

Recibe los datos del documento

DocManager -> Actualizar características del documento

evaluateDocument()

DocManager -> Cambiar el documento a no editable

newVersionDocument()

DocManager -> Copiar el documento con versión 0.1 y estado borrador

saveDocument()

DocManager -> Salvar la información del documento

ConDor Versión 0.1 (1999)
Sistema para el control de documentos del sistema de
calidad
Carlos Javier Pérez Escobar

DOCUMENTO DE DISEÑO
DICCIONARIO DE CLASES DEL DISEÑO (VERSION 2.0)
MANEJO DE DATOS

DataBase - Es la clase que se utiliza para la gestión del acceso a la base de datos mediante el puente JDBC-ODBC. Esta clase debe iniciar y cerrar la conexión con la base de datos. Permitir a las clases del manejo de datos realizar consultas y actualizaciones de la información. Para realizar las operaciones en la base de datos las clases del manejo de datos deben crear la expresión en SQL requerida y *DataBase* se encarga de procesarla y enviar los resultados.

openConnection()

iniciar la conexión de acuerdo con el driver y url establecido
verificar que no existan errores al iniciar la conexión

closeConnection()

terminar la conexión
verificar que no existan errores al terminar la conexión

update(String expresion)

ejecutar la operación de actualización establecida por la expresión SQL
verificar que no existan errores en la actualización

openCursor(String expresion)

ejecutar una operación de consulta establecida por la expresión SQL
verificar que no existan errores en la consulta