

03063
15



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

POSGRADO EN CIENCIA E INGENIERIA DE LA COMPUTACION

UN PROTOTIPO PARA EL CONTROL DE CALIDAD Y
CAMBIOS EN LOS PRODUCTOS DE SOFTWARE

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

P R E S E N T A:

MARTIN DE JESUS JIMENEZ

DIRECTORA DE LA TESIS: DRA. HANNA OKTABA

300195

MEXICO, D. F.

2001



Universidad Nacional
Autónoma de México



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.

A Dios
Quien es la fortaleza
de mi vida

A Miri
Por su paciencia

A mis padres, hermanos y
sobrinos, por su gran apoyo

Agradecimientos:

A la Dra. Hanna Oktaba, mi tutora, por el gran privilegio de contar con su don de enseñanza para hacer posible ésta tesis.

A mis sinodales M. en C. Guadalupe Ibargüengoitia, Dr. Fernando Gamboa Rodríguez, Dr. Manuel Romero Salcedo y Dr. Armando Maldonado Talamantes, por su tiempo y valiosos consejos.

A María de Jesús, César C., Alejandro Talavera, Blanca, Lolita, Sandra, Carmen, Lety, Lulú, Sara, Norma, Henry y Miguel, por su amistad. Y a todo el grupo que formamos en el IIMAS y compartimos momentos gratos.

Martín De Jesús J.

INDICE

Introducción	1
I. Calidad de los productos de software y el control de cambios.	3
1.1. Conceptos generales	3
1.2. El modelo CMM	4
1.3. Administración de Requisitos	9
1.4. Administración de la Configuración del Software	12
1.5. Control y Aseguramiento de la Calidad	18
1.5.1. Revisión entre Pares	20
1.5.2. Ingeniería del Producto de Software.	22
1.5.3. Aseguramiento de la Calidad de Software	25
II. Proceso de producción de software con control de calidad y cambios	27
2.1. Procesos de Software maduro e inmaduro.	27
2.2. Roles en los grupos de producción de software.	30
2.3. Proceso iterativo genérico de producción de software	32
2.3.1. Niveles de control de producción y calidad del software	33
2.3.2. Niveles de control de cambios a productos de software	42
2.3.3. Integración de control de calidad y cambios	49
2.3.4. Resumen de las posibles combinaciones de los diferentes niveles de control	58
2.3.5. Modelo Genérico de Control de Calidad y Cambios a los Productos de Software.	59
2.3.6. Modelo Genérico de Control de Calidad y Cambios a los Productos de Software.	60

III. Herramientas de control de productos de software	63
3.1. Prototipo para el control de los documentos de administración de la calidad en empresas pequeñas..	63
3.2. Características de las herramientas que apoyan el control de cambios a los productos de software.	67
IV. Marco contextual.	73
4.1. Redes de computadoras	73
4.2. World Wide Web (WWW)	74
4.3. Tecnologías orientadas a objetos	77
4.4. Integración de bases de datos en el Web	80
4.5. Resumen de los elementos tecnológicos que se utilizarán para el desarrollo de un prototipo de Control de Calidad y Cambios a los Productos de Software	83
V. Análisis y Diseño de un Prototipo de Control de Calidad y Cambios en los Productos de Software.	85
5.1. Consideraciones generales del prototipo	85
5.2. Requisitos del prototipo	93
5.3. Actores.	96
5.4. Análisis	97
5.4.1. Nivel de Control Simple (NS)	98
5.4.1.1. Clases Principales del Nivel de Control Simple (NS)	98
5.4.1.2. Casos de Uso del Nivel de Control Simple (NS)	100
5.4.1.3. Diagrama de Estado del Nivel de Control Simple (NS)	101
5.4.1.4. Diccionario de Clases del Nivel de Control Simple (NS)	101
5.4.2. Nivel de Control Intermedio (NI)	101
5.4.2.1. Clases Principales del Nivel de Control Intermedio (NI)	101
5.4.2.2. Casos de Uso del Nivel de Control Intermedio (NI)	104
5.4.2.3. Diagrama de Estado del Nivel de Control Intermedio (NI)	106
5.4.2.4. Diccionario de Clases del Nivel de Control Intermedio (NI)	107
5.4.3. Nivel de Control Completo (NC)	107
5.4.3.1. Clases Principales del Nivel de Control Completo (NC)	107
5.4.3.2. Casos de Uso del Nivel de Control Completo (NC)	109
5.4.3.3. Diagrama de Estado del Nivel de Control Completo (NC)	110
5.4.3.4. Diccionario de Clases del Nivel de Control Completo (NC)	111

5.5. Diseño	112
5.5.1. Clases resultantes del diseño	112
5.5.2. Resumen de componentes del prototipo	113
5.5.3. Arquitectura del prototipo	114
5.5.3.1. Arquitectura lógica	114
5.5.3.2. Arquitectura física	115
5.5.3.3. Diseño de la Base de Datos	116
5.6. Aportaciones y limitaciones	121
VI. Propuesta de integración de la herramienta en un proyecto.	
6.1. El proyecto SIPOCIC	123
6.2. Productos de Software del proyecto SIPOCIC y su clasificación con niveles de control.	128
6.3. Limitaciones de la herramienta C3PoS.	139
Conclusiones y trabajos a futuros	141
Bibliografía	145
Anexos	
A. Herramientas que apoyan el control de cambios a los productos de software.	
B. Documento de análisis para productos con Nivel de Control Simple (NS).	
C. Documento de análisis para productos con Nivel de Control Intermedio (NI).	
D. Documento de análisis para productos con Nivel de Control Completo (NC).	
E. Documento de diseño.	

Introducción

La satisfacción del cliente ha llegado a ser el lema de muchas organizaciones que intentan sobrevivir y prosperar en un mundo en que la competitividad aumenta día con día. Para ello las organizaciones desarrolladoras de software deben contar con un proceso de desarrollo de software maduro para obtener productos de calidad que cumplan los requisitos del cliente.

Los requisitos del cliente frecuentemente evolucionan y cambian en cualquier etapa del proyecto y los cambios a los acuerdos existentes deben ser evaluados y negociados con los grupos afectados para que ajusten sus planes de software, productos de trabajo y actividades.

Por lo tanto, los grupos afectados deben llevar a cabo una adecuada administración de requisitos porque son la base para estimar, planear, realizar y darte seguimiento a las actividades del proyecto de software durante todo su ciclo de vida.

Un producto de software puede requerir diferentes controles de calidad y solicitudes de cambio, ya sea porque se necesite de una evaluación independiente de quien lo produce, que podría ser un revisor, quien verificaría el cumplimiento de la especificación de los requisitos. Otros productos podrían necesitar un doble control de calidad, en donde además de un revisor, un aprobador evaluaría el producto para la minimización del impacto en otros productos, debido a su modificación.

Además, para mejorar la calidad de los productos, podrían realizarse revisiones tipo inspección y pruebas tipo "testing", pero siempre garantizando la coordinación de las actividades de comunicación y colaboración entre todos los miembros de los grupos de trabajo, para evitar conflictos y/o realizar tareas repetidas. También, el uso de herramientas para el control de calidad y cambios en los productos de software y que incorporen las mejores prácticas de la ingeniería del software facilitarían su evaluación y agilizarían la comunicación y colaboración entre todos sus miembros.

Hoy en día, las organizaciones en general integran sus sistemas de información tradicionales a sistemas de información basados en tecnologías de redes, particularmente en Internet e intranets. Estas tecnologías también facilitan la integración del trabajo en grupo en las empresas.

El objetivo de la tesis es proponer un modelo genérico para el control de calidad y cambios a los productos de software con tres niveles configurables de control que den la suficiente flexibilidad para ser utilizados de acuerdo a las necesidades de control de un producto y/o recursos de una organización en particular. Con base a este modelo genérico, que incluye conceptos que definen estándares de calidad para la mejora de los procesos obtenidos del Modelo de Madurez de la Capacidad del Proceso de Software [CMM[®]] se desarrolla el prototipo de una herramienta, que es diseñada y documentada utilizando el Proceso Unificado y el Lenguaje de Modelado Unificado (UML) [JBR1999].

Esta tesis cubre en el Capítulo I conceptos fundamentales sobre la mejora al proceso de desarrollo del software, por lo cual se describe en forma general el Modelo de Madurez de la Capacidad del Proceso de Software (CMM) y en particular las áreas clave de Administración de Requisitos, Aseguramiento de la Calidad del Software y Administración de la Configuración del nivel 2 de este modelo, además de las áreas clave de Revisión entre Pares y la Ingeniería del Producto de Software del nivel 3 que constituyen el marco teórico para este trabajo.

En el Capítulo II se propone un modelo genérico para el control de la calidad y cambios en los productos de software con base a lo expuesto en el capítulo I y que sirve de base para el diseño del prototipo de una herramienta.

En el capítulo III se revisan las características de herramientas que apoyan el control de los productos de software, comenzando como antecedente la revisión de una tesis que presenta un prototipo para el control de los documentos de administración de la calidad en empresas pequeñas.

En el capítulo IV se resumen los elementos tecnológicos que se tomarán en cuenta para el diseño e implantación del prototipo de control de calidad y cambios a los productos de software.

El Capítulo V se presenta el análisis y diseño orientado a objetos de un prototipo de control de calidad y cambios en los productos de software (C3PoS), basado en el modelo propuesto en el Capítulo II, utilizando el Lenguaje de Modelado Unificado (UML) e implementado con el lenguaje de programación Java.

El Capítulo VI se presenta una propuesta para la integración y viabilidad del prototipo de control de calidad y cambios en los productos de software (C3PoS) a un proyecto que se realizó durante el estudio de la maestría en el curso de Ingeniería de Software Orientada a Objetos impartido por la Dra. Hanna Oktaba y la M.C. Ma. Guadalupe Ibarguengoitia, en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS). El proyecto se denominó "Sistema de Inscripciones al Posgrado en Ciencia e Ingeniería de la Computación de la UNAM (SIPOCIC)".

Posteriormente se presentan las conclusiones a las que condujo la realización de ésta tesis, así como los trabajos a futuro sugeridos. En la parte final se encuentra la bibliografía y los anexos. En los anexos se proporciona información sobre herramientas que apoyan el control de los productos de software, así como también los documentos de análisis y diseño del prototipo (C3PoS).

Capítulo I

I. Calidad de los productos de software y el control de cambios.

Este capítulo trata sobre conceptos fundamentales sobre la mejora al proceso de desarrollo del software, por lo cual se describe en forma general el Modelo de Madurez de la Capacidad del Proceso de Software (CMM®) y en particular las áreas clave de Administración de Requisitos y Administración de la Configuración del nivel 2 de este modelo, además de la Revisión entre Pares y de la Ingeniería del Producto de Software del nivel 3.

1.1. Conceptos generales.

Un *producto* es un artefacto tangible de salida (o servicio) que es el resultado de un proceso y que será entregado a un cliente o a un usuario final.

Un producto de trabajo de software es cualquier artefacto producido por un proceso. Estos artefactos pueden incluir archivos, documentos, partes del producto, servicios, procesos, especificaciones y facturas.

Un producto de software también es un producto de trabajo que es entregado al cliente.

Utilizaremos el término producto para incluir tanto producto de software como a los productos de trabajo de software.

Un *producto de software* es la suma total de programas de computadora, procedimientos, reglas, documentación y datos asociados para su entrega al cliente o usuario final.

Un *proceso de desarrollo de Software* es el conjunto de actividades con las cuales las necesidades de un usuario se traducen en requisitos de software, que se transforman en diseño, se implementan en código y que es probado, documentado y certificado para su uso operacional.

El *proceso de software* puede ser definido como el conjunto de actividades, métodos, prácticas y transformaciones que la gente utiliza para desarrollar y mantener software y los productos asociados (planes del proyecto, documentos de diseño, código, casos de prueba y manuales de usuario)[PauM1993].

Un *proyecto* es un compromiso donde se acuerda un esfuerzo, en tiempo y personal, enfocado al desarrollo y/o mantenimiento de un producto. Tiene asociado un costo, fondos y

un 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 [Hum1997] [PauM1993].

El *ciclo de vida del software* típicamente incluye una etapa de concepto, una etapa de requisitos, una etapa de diseño, una etapa de implantación, una etapa de prueba una etapa de instalación y liberación, una etapa de mantenimiento y algunas veces una etapa de retiro del producto de software

La *capacidad del proceso* de software describe el rango de resultados esperados que se obtienen siguiendo un proceso de software [PauM1993].

La *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 [PauM1993].

La *madurez del proceso de software* se da cuando un proceso en específico es definido explícitamente, administrado, medido, controlado y es efectivo.

El *desempeño del proceso de software*, representa los resultados reales obtenidos siguiendo un proceso de software.

1.2. El modelo CMM®

El Departamento de Defensa de los Estados Unidos, a principios de los años 80, enfocaba sus actividades a la revisión de los problemas del software y declaró oficialmente *la crisis del software*¹ con la aparición del estudio *Report of the Defense Science Board Task Force on Military Software*.

A finales de 1984, el Departamento de Defensa de los Estados Unidos de América creó el Instituto de Ingeniería del Software (SEI) para mejorar las prácticas de la Ingeniería del Software debido a que la calidad del mismo es un elemento crítico en los sistemas de defensa.

La misión del SEI es proporcionar dirección en la mejora de la prácticas de la ingeniería del software para mejorar la calidad de los sistemas que dependen del software.

El SEI cumple esta misión al promover la evolución de la ingeniería del software de una actividad muy personal a toda una disciplina que es bien administrada y apoyada por la tecnología.

En 1991 el Instituto de Ingeniería del Software (SEI) produce *el Modelo de Madurez de la Capacidad del Proceso de Software (CMM®)*.

¹ En los inicios de 1968 surge la crisis del software pues el proceso de desarrollo de software sobrepasa los costos y el tiempo estimados y tiene demasiadas fallas.

El modelo CMM[®] se enfoca en el proceso, como una forma para capacitar a la gente como hacer su trabajo. En una organización madura de desarrollo de software, su proceso es mejor definido y consistentemente implementado por toda la organización.

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 inmediatas, carecen de bases objetivas para evaluar la calidad de los productos o para resolver los problemas que surgen.

Las organizaciones maduras, poseen una gran habilidad para administrar el proceso de desarrollo del software y mantenerlo, se hacen pruebas y análisis de costo-beneficio para mejorar el proceso, el administrador supervisa la calidad del producto de software y la satisfacción del cliente, se llevan registros y todos los integrantes están involucrados en el proceso de desarrollo llevando a cabo un proceso disciplinado porque comprenden el valor que esto representa.

Con base a los principios anteriores, se creó el Modelo de Madurez de la Capacidad del Proceso de Software (CMM[®]) que permite obtener un incremento gradual en la capacidad del proceso de software para producir de manera consistente y predecible productos de calidad superior. En el modelo se describen 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 una forma evolutiva, que puede ser desde un proceso improvisado y caótico a un proceso maduro con una mayor capacidad y disciplina.

CMM[®] proporciona guías para seleccionar estrategias de mejoramiento del proceso mediante la determinación de las capacidades actuales del proceso y la identificación de los puntos críticos y la calidad del software.

El principal objetivo de un proceso de software maduro es el de producir productos de calidad que cumplan los requisitos del usuario. La satisfacción del cliente ha llegado a ser el lema de muchas organizaciones que intentan sobrevivir y prosperar en un mundo en que la competitividad aumenta día con día.

El Modelo de Madurez de la Capacidad del Proceso de Software (CMM[®]) está formado de 5 niveles de madurez, diseñados de tal manera que los niveles inferiores proporcionan las bases para que de forma progresiva se alcancen los niveles superiores. Estos niveles son vistos como niveles de madurez con la cual una organización alcanza una mayor capacidad del proceso.

La figura 1.1 muestra los 5 niveles de madurez del proceso de software que son:

1. **Inicial.** El proceso de software es un proceso improvisado y caótico.
2. **Repetible.** Se establecen procedimientos de administración del proceso que son básicos para determinar costos, calendarios y funcionalidad.
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.
4. **Administrado.** Se recolectan medidas detalladas del proceso de software y de la calidad del producto. Ambos son cuantitativamente entendidos y controlados.
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.

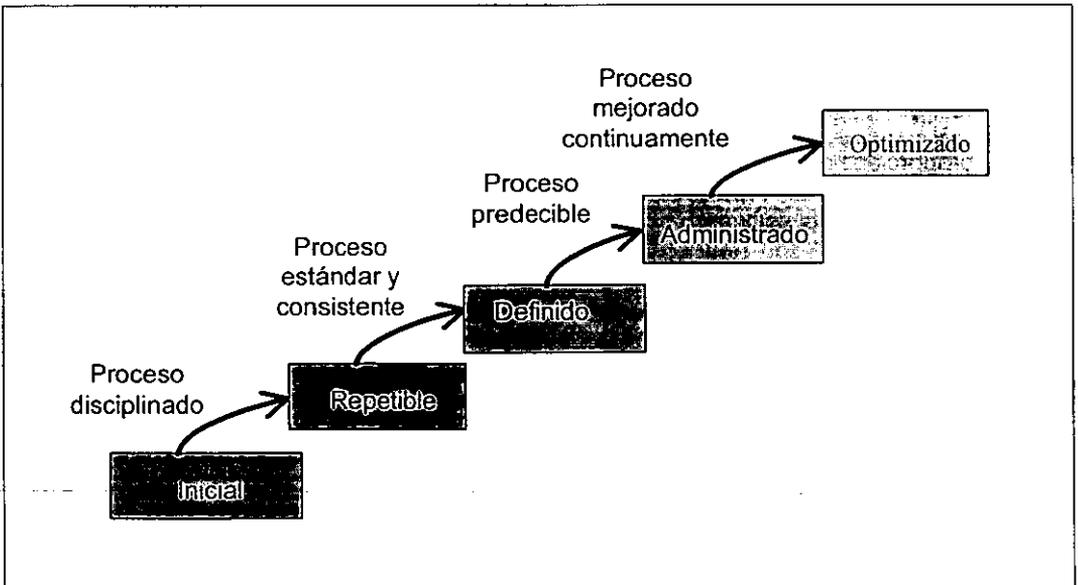


Figura 1.1 Niveles de madurez del CMM®

Cada nivel de madurez está compuesto de varias *áreas claves del proceso*², que indica a la organización donde debe enfocarse para mejorar su proceso de software.

El *objetivo*³ de cada área clave del proceso resume las prácticas clave y pueden usarse para determinar, ya sea si una organización o un proyecto ha implementado su área clave del proceso.

Para el caso del nivel 2, Repetible, de CMM® las áreas clave son:

Áreas clave

1. Administración de Requisitos.
2. Planeación del Proyecto de Software.
3. Supervisión y Seguimiento del Proyecto de Software.
4. Administración de Subcontratistas de Software.
5. Aseguramiento de la Calidad del Software.
6. Administración de la Configuración del Software.

En el nivel 3 de CMM®, Definido, las áreas clave son:

Áreas clave

1. Atención a los Procesos de la Organización.
2. Definición de los Procesos de la Organización.
3. Programa de Entrenamiento.
4. Administración de la Integración del Software.
5. Ingeniería del Producto de Software.
6. Coordinación entre Grupos.
7. Revisión entre Pares.

² Una *área clave* del proceso identifica un conjunto de actividades relacionadas que al realizarse colectivamente, alcanzan un conjunto de objetivos considerados importantes para mejorar el proceso de capacidad del software.

³ El *objetivo* significa el alcance, los límites y la intención de cada área clave del proceso.

Entre las características generales que se pueden mencionar en el nivel repetible, se encuentran:

- Existen en la organización políticas y procedimientos para la administración del proyecto de software.
- Los proyectos implementan procesos efectivos, los cuales se encuentran definidos y documentados, además estos procesos realmente se practican, se enseñan, se miden y se mejoran.
- Existen controles básicos de administración del software en cada proyecto.
- Los requisitos del software y el desarrollo de los productos de trabajo que lo satisfacen, constituyen las líneas base⁴ controladas.
- Las organizaciones son disciplinadas puesto que son estables tanto en la planeación como el seguimiento del proyecto, y pueden repetirse éxitos anteriores.

A continuación se presentarán las áreas clave de Administración de Requisitos, la Administración de la Configuración de Software y Aseguramiento de la Calidad del Software del nivel repetible de CMM[®], además, la Revisión entre Pares y la Ingeniería del Producto de Software del nivel 3. Dichas áreas servirán de base al desarrollo de esta tesis.

⁴ 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 solo puede ser modificada mediante el procedimiento de control de cambios establecido.

1.3. Administración de Requisitos

Los cambios a los requisitos obliga también a realizar cambios a los productos de software que deberán ser evaluados y negociados con los grupos afectados y si se cuenta con procedimientos recomendados por estas dos áreas clave del nivel repetible de CMM® en la que el requisito puede cambiar y la administración del producto lo realiza la administración de la configuración que se llevan a cabo durante todo el proceso de desarrollo, se podrían predecir los impactos al desarrollo del proyecto. Además no solo se lograría mantener la integridad de los productos de software sino también se obtendría un entendimiento común entre el cliente y el grupo del proyecto de software.

Lo anterior son algunos de los elementos necesarios para contribuir a establecer una disciplina de administración del proceso con lo cual se mejoraría la capacidad del proceso de desarrollo del software (figura 1.2) [JBRu1999].

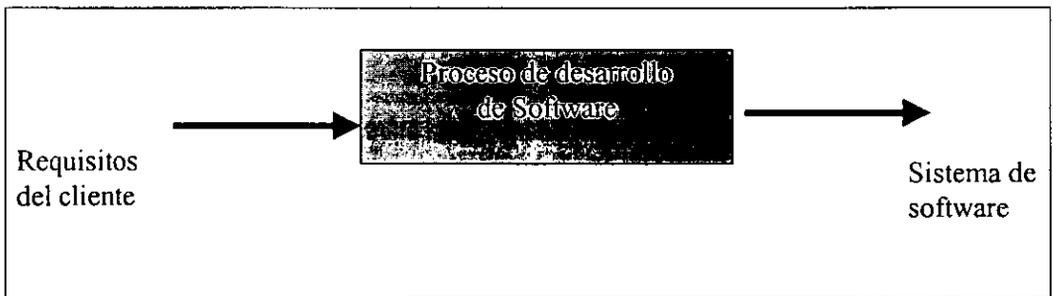


Figura 1.2 Proceso de desarrollo de software

El software llega a cambiar constantemente, debido a necesidades para repararlo (eliminar errores que no fueron detectados antes de liberar la aplicación), a necesidades de soporte por la evolución de la aplicación, por nuevos requisitos o a causa de cambios de los requisitos iniciales.

El propósito del área clave de proceso de la Administración de Requisitos, según el Modelo de Madurez de la Capacidad del Proceso de Software (CMM®) nivel 2, es

establecer un entendimiento común sobre los requisitos del proyecto entre el *cliente*⁵ y el *grupo del proyecto de software*.

Lo anterior, es con el fin de monitorear la calidad del producto de software y la satisfacción del cliente.

⁵Por "cliente" se entiende el grupo de ingeniería de sistemas, grupo de ventas, otra organización interna o el cliente externo.

De acuerdo a CMM® el área clave de Administración de Requisitos debe cumplir con:

Objetivos:

1. Controlar los requisitos del sistema para establecer las líneas base de Ingeniería de Software y de uso administrativo.
2. Mantener la consistencia entre los planes del software, los productos y las actividades, con los requisitos del sistema.

Compromisos:

1. El proyecto seguirá una política escrita y propia de la organización, para administrar los requisitos del sistema.
2. Los requisitos se deben documentar.
3. Deben existir recursos y fondos necesarios para administrar los requisitos.
4. Los elementos del grupo de ingeniería de software y otros grupos relacionados, deberán estar debidamente entrenados para realizar las actividades de la administración de requisitos.

Actividades:

1. Los requisitos serán revisados por el grupo de ingeniería de software antes de que sean incorporados al Proyecto de software.
2. El grupo de ingeniería de software usará los requisitos como base para los planes, productos de trabajo y actividades.
3. Deben revisarse e incorporarse al proyecto, todos los cambios que se hayan hecho a los requisitos.

Métricas:

Deben tomarse métricas que determinen el estado de las actividades de administración de requisitos.

Verificaciones:

1. Revisar periódicamente las actividades de administración de requisitos junto con el Administrador Principal.
2. Revisar periódicamente o según los eventos que se presenten, las actividades de administración de requisitos junto con el Administrador del Proyecto.
3. El grupo de aseguramiento de la calidad del software revisará o auditará y reportará, las actividades y los productos de trabajo de la administración de los requisitos.

Por lo anterior podemos decir que los miembros del *grupo de ingeniería de software y otros grupos relacionados deben* realizar sus actividades de administración de requisitos con el fin de que sean documentados y controlados debido a que son la base para estimar, planear, realizar y darle seguimiento a las actividades de proyecto de software durante todo el ciclo de vida del software⁶.

Los requisitos son revisados por los probadores, ingenieros de software, aseguradores de la calidad, administradores de la configuración y soporte de la documentación antes de que sean incorporados al proyecto de software

Los requisitos del cliente frecuentemente evolucionan y cambian y los cambios a los acuerdos existentes son evaluados y negociados con los grupos afectados.

Cada vez que cambien los requisitos del sistema asociados al software se ajustan los planes del software, productos de trabajo y actividades para mantenerlos consistentes con los requisitos modificados.

Además es de suma importancia que los cambios a los requisitos se deben:

- Identificar
- Evaluar riesgo
- Documentar
- Planear
- Comunicarlos a los grupos e individuos afectados
- Dar seguimiento para completarse

El administrador general necesita determinar y revisar el estado de las actividades de administrar los requisitos, por ejemplo, el número de cambios acumulados (tanto propuestas como aprobaciones) y el *grupo de aseguramiento de calidad* los audita.

En la Administración de Requisitos se establece una línea base para el control de requisitos para mantener la consistencia con los planes, productos y actividades de software

⁶Por ciclo de vida del software es el período de tiempo que inicia cuando un producto de software es concebido y finaliza cuando el software ya no está disponible para su uso.

Se debe de contar con herramientas que soporten la actividad de Administración de Requisitos como herramientas de administración de la configuración, para darle seguimiento y administración a los productos.

Las recomendaciones que se tomarán en cuenta, con respecto a esta área clave, para el desarrollo de la tesis son:

1. La documentación de los requisitos.
2. Identificar los controles necesarios para mantener la consistencia entre los requisitos y los productos de software del sistema.
3. Facilitar la revisión de los requisitos por parte de los grupos involucrados en el proyecto así como darle seguimiento hasta su implantación.
4. Control de cambios a los requisitos.

Por lo tanto se hace necesario contar con herramientas que permitan almacenar, visualizar, revisar y darle seguimiento a los requisitos en el ciclo de vida del software

1.4. Administración de la Configuración del Software

Durante el desarrollo del proyecto de software se pueden dar cambios en cualquier etapa del proyecto y mediante procedimientos que controlen los cambios se pueden predecir los impactos en el sistema.

Los cambios se puede dar con el cliente cuando desea modificar sus requisitos, con el equipo de desarrolladores que modifican cuestiones técnicas o con los administradores todo con el fin de que funcione mejor el software,

El *propósito* del área clave de proceso de Administración de la Configuración del Software (ACS)⁷ según el Modelo de Madurez de la Capacidad del Proceso de Software (CMM[®]) nivel 2, es

establecer y mantener la integridad de los productos⁸ del proyecto de software a lo largo de todo su ciclo de vida.

⁷ Sus siglas en ingles son SCM que significa Software Configuration Management

⁸ Los productos de trabajo (software work products) son documentos relacionados con el proceso (planes, estándares, procedimientos y demás), requisitos de software, documentos de diseño, códigos fuente, procedimientos de pruebas, software para realizar pruebas, software que se entrega al cliente, compiladores y demás herramientas de apoyo.

Los productos de software que están bajo el control de ACS incluyen los productos de software que son entregados al cliente y los productos relacionados o requeridos para crear éstos.

De acuerdo a CMM® el área clave de Administración de la Configuración debe cumplir con:

Metas:

1. Planificar las actividades de la ACS.
2. Identificar, controlar y distribuir los productos de trabajo
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:

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

Condiciones:

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 la ACS en el proyecto.
3. Se suministran los fondos y recursos necesarios para el desempeño de las actividades de ACS.
4. Los miembros de los grupo de ACS 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 ACS.

Actividades:

1. Se prepara un plan de ACS para cada proyecto de software según lo establecido en el procedimiento correspondiente.
2. Se utiliza el plan de ACS, documentado y aprobado, como base para realizar las actividades de ACS.
3. Se establece un sistema para la biblioteca de administración de configuración 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.

Métricas y análisis:

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

Verificación de la implantación:

1. Las actividades de ACS son revisadas periódicamente por el gerente principal.
2. Las actividades de ACS son revisadas eventual y periódicamente por el gerente del proyecto.
3. El grupo de ACS 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 ACS y reporta los resultados.

Líneas Base

Una línea base es un conjunto de elementos de software generados en una fase del ciclo de vida de un proyecto. Pueden ser una especificación o un producto revisado, que posteriormente servirán como base en el desarrollo y mantenimiento del producto de software. Una línea base sólo puede ser modificada a través de un procedimiento de control formal de cambios. En forma general, existen cinco líneas base:

1. Análisis de Requisitos. Se logra cuando se tiene la definición de los requisitos del producto de software.
2. Diseño. Se logra cuando se ha determinado el cómo construir o codificar y satisfacer los requisitos del sistema.
3. Codificación y pruebas de unidad. Se logra cuando se termina la programación o la codificación de las unidades, y éstas son probadas para verificar la satisfacción de los requisitos del diseño, las cuales a su vez, deben satisfacer los requisitos del software.
4. Integración y liberación. Se logra cuando las pruebas se han efectuado y se demuestra que el producto reúne los requisitos solicitados, y está listo para entregarse al cliente o usuario.

5. Soporte del producto. Se logra cuando, después de entregar el producto al cliente, el producto puede ser adaptado, modificado, corregido y mejorado.

En general, se piensa que la administración de la configuración debe hacerse en la fase de soporte del producto, pero debe realizarse a lo largo de todo el proceso de desarrollo como se muestra en la figura 1.3 [BMT1999].

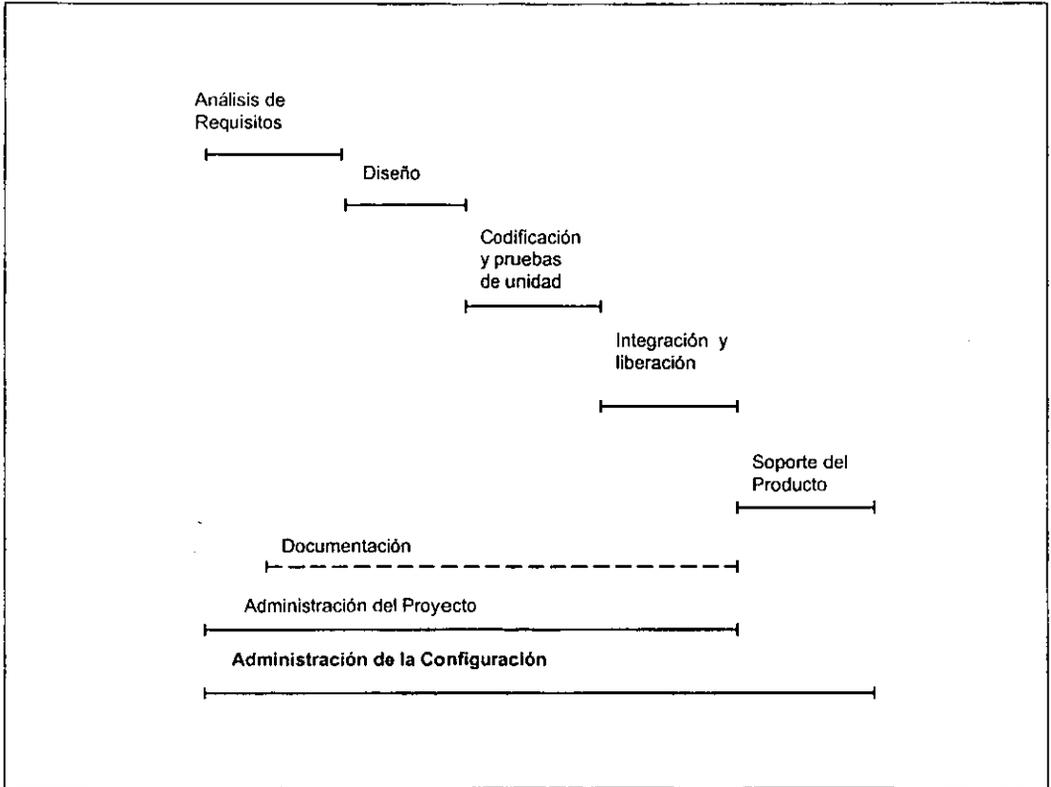


Figura 1.3 La Administración de la Configuración en el proceso de desarrollo

ACS debe de identificar, controlar y distribuir los productos de trabajo además de controlar sus cambios, informando del estado y contenido de las líneas base del software a los grupos y personas afectadas.

De acuerdo a la interpretación en [Ibar1999] el Administrador de la Configuración define las líneas de base en conjunto con el administrador del proyecto y el proceso para cambiarlas, planearlas, organiza y además informa de las liberaciones del producto.

Se establece un sistema para la biblioteca de administración de configuraciones como repositorio para las líneas base del software. Esto permiten asegurar que cada componente en un instante de tiempo, es revisado y controlado. Los componentes y sus identificadores se guardan en bibliotecas del sistema para cada línea base.

Un componente puede requerir un cambio por diferentes motivos: corrección de errores, mejora, aumento de funcionalidad, adaptación a nuevo ambiente. Cada cambio debe pasar por un proceso de control que recibe la solicitud de cambio, lo evalúa y si se aprueba, solicita que se efectúe. Esto genera un nuevo componente en una nueva versión que se guarda en las bibliotecas.

Existe una Mesa de Control de Cambios (MCC) que evalúa y autoriza la implantación del cambio (figura 1.4) cuyas funciones son:

- Administra las configuraciones del software.
- Recibe una propuesta de cambio a un componente en alguna línea base.
- Evalúa el cambio, el alcance (los productos de software que afecta), su costo, ventajas y desventajas, su importancia, urgencia y los impactos que causará.
- Si se aprueba, pide al programador que efectúe el cambio y al probador que lo verifique.
- Recibe el componente con el cambio aprobado y conforma la nueva configuración.

Para el control de cambios se debe de contar con:

1. Un medio mediante el cual solicitar un cambio
2. Un procedimiento para evaluar el cambio que tome en cuenta costos y efectos en otros productos de software.
3. Un responsable para aprobar o rechazar el cambio
4. Un procedimiento de seguimiento al cambio (desde su solicitud hasta su implantación)

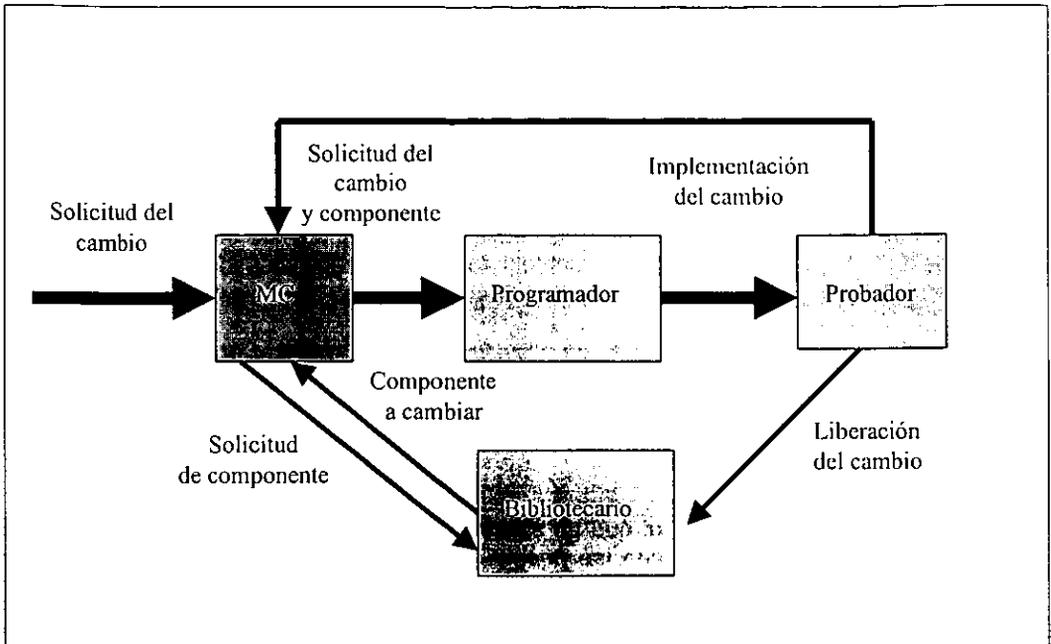


Figura 1.4 Control de cambios a los productos de software

Las recomendaciones que se tomarán en cuenta del área clave de la Administración de la Configuración, para el desarrollo de la tesis son:

1. La identificación, control y distribución de los productos de software
2. Controlar las solicitudes de cambio
3. Informar a los grupos involucrados, del estado y contenido de los productos de software
4. Definición de los responsables de realizar las revisiones y aprobaciones tanto de los productos de software así como de las solicitudes de cambio.

1.5. Control y Aseguramiento de la Calidad.

El principal objetivo de un proceso de software maduro es el de producir productos de calidad que cumplan los requisitos del cliente.

Cada vez que cambien los requisitos del sistema asociados al software se ajustan los planes, productos de trabajo y actividades para mantenerlos con los requisitos modificados.

Se deben de realizar todo un conjunto de actividades para garantizar que una organización obtiene un nivel de calidad en los productos y determinar si cumple con los requisitos establecidos. Esto implica definir con claridad otros conceptos relacionados con la calidad misma y que servirán de base para el desarrollo de esta tesis:

- **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 [PauM1993].
- **Control de calidad** se enfoca básicamente en las actividades de encontrar y eliminar defectos [Jal2000].
- **Defecto** es la diferencia entre el resultado esperado y el real [CM1999].
- **Aseguramiento de la calidad** es el conjunto de actividades, desarrolladas dentro de un 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. [Cad1996][Sch1995]

CMM[®] en diferentes áreas clave (Revisión entre Pares y la Ingeniería del Producto de Software) hace énfasis en la necesidad de controlar la calidad, desde el momento mismo en que se comienza a desarrollar el producto. Existen diferentes revisiones tipo inspección o tipo pruebas ("testing"), que representa la forma más efectiva para mejorar la calidad.

Así mismo, CMM[®] en otra área clave (Aseguramiento de la Calidad del Software) prescribe las prácticas necesarias para asegurar que los controles de calidad a los productos de software se cumplan (requisitos, estándares y procedimientos).

El control de calidad se realiza principalmente mediante dos tipos de revisiones:

1. Por humanos mediante revisiones tipo inspección.

- Estas revisiones se llevan a cabo entre elementos del mismo equipo de trabajo con el conocimiento necesario y suficiente para poder corregir y mejorar el trabajo de un compañero. Las personas involucradas en este tipo de revisiones pueden estar o no trabajando conjuntamente en el mismo producto o en productos muy relacionados.
- La ventaja de incorporar a varias personas es de tener diferentes perspectivas y experiencias que enriquecen la inspección.

En una inspección, dos o más ingenieros revisan el producto de otro ingeniero para encontrar defectos y problemas. El objetivo no es solucionar los problemas pero si encontrarlos para que el desarrollador pueda resolverlos. El momento de realizar una inspección es cuando el ingeniero ha terminado de desarrollar un producto, lo ha revisado personalmente y ha corregido todos los problemas obvios. En ese momento el ingeniero usualmente necesita ayuda para encontrar el resto de los defectos. [Hum1999].

2. Utilizando un apoyo automatizado, mediante revisiones tipo pruebas (“testing”).

- Prueba (“testing”) es un proceso de planificación, preparación, realización y medición dirigido a determinar las características de un sistema de software y mostrar la diferencia entre el estatus actual y el requerido.
- Con “testing” se reduce el nivel de incertidumbre en la calidad del producto de software [KP1999]
- Estas revisiones se llevan a cabo por un grupo especialmente asignado para esta tarea (pueden ser elementos del mismo equipo de trabajo), siguiendo una planeación establecida. Para este tipo de revisiones existe el material previo y elementos auxiliares (casos de prueba), así como un proceso bien establecido para registrar los defectos, tiempos de revisión y corrección, clasificación de los defectos, perfil y capacidad del equipo de trabajo.

En general existen dos clases de pruebas a nivel de unidades:

- Las pruebas de caja blanca se realizan en los componentes de código, que pueden ser unidades de software. Estas pruebas consideran la estructura y los caminos lógicos del programa.
- Las pruebas de caja negra toman en cuenta los requisitos enfocados a los aspectos externos del sistema. Validan que el software cumpla con los requisitos sin tomar en cuenta los caminos de ejecución llevados a cabo para cumplir cada uno de ellos. Este es el tipo de prueba conducida de software, esto es, la integración de unidades de código.

Las pruebas y las inspecciones proporcionan el método más efectivo para mejorar la calidad de los productos de software durante el desarrollo.

En CMM[®], el primer tipo de revisiones (tipo inspección) está contenido en las áreas clave de Revisión entre Pares del nivel 3 y en Aseguramiento de la Calidad del Software del nivel 2. Para el segundo tipo de revisiones (tipo pruebas) se enuncian en la actividades 5, 6 y 7 del área clave de Ingeniería del Producto de Software nivel 3.

A continuación se resumirán estas áreas.

1.5.1. Revisión entre Pares

El propósito de la revisión entre pares es eliminar defectos de los productos de trabajo de software desde un principio y en forma eficiente. Con el fin de desarrollar un mejor entendimiento de los productos de trabajo de software y prevenirlos de los defectos.

La revisión entre pares involucra un examen metódico de los productos de trabajo de software por pares de productores para identificar los defectos y las áreas donde los cambios sean necesarios. Los productos específicos que deban ser sometidos a un par de revisores son identificados en el proceso de software definido del proyecto y planearse como parte de las actividades de planeación del proyecto de software.

De acuerdo a CMM® el área clave de Revisión entre Pares debe cumplir con:

Objetivos:

1. Se planean las actividades de revisión entre pares.
2. Se identifican y eliminan los defectos en los productos de software.

Compromisos:

1. El proyecto sigue una política organizacional escrita para realizar revisión entre pares.
 - a. La organización identifica un conjunto estándar de productos de trabajo de software que deberá ser sometido a la revisión entre pares
 - b. Cada proyecto identifica los productos de trabajo de software que serán sometidos a la revisión entre pares.
 - c. La revisión entre pares es conducida por los líderes entrenados en la revisión entre pares
 - d. La revisión entre pares se enfoca en los productos de trabajo de software que serán revisados y no en el productor.
 - e. Los resultados de la revisión entre pares no son utilizados por el administrador para evaluar el rendimiento de los individuos.

Condiciones:

1. Se deberán proporcionar los recursos y fondos para realizar la revisión entre pares de cada uno de los productos de trabajo de software que sea revisado.
2. Se requiere entrenamiento para los líderes de la revisión en como conducir la revisión entre pares.
3. Los revisores quienes participan en la revisión entre pares reciben entrenamiento en los objetivos, principios y métodos de la revisión entre pares.

Los productos que se revisan entre pares pueden ser:

- Los requisitos del software
- El diseño del software
- El código
- Los procedimientos de prueba del software

Actividades:

1. Las revisiones son planeadas y los planes son documentados.
2. Las revisiones entre pares son realizadas de acuerdo a un procedimiento documentado.
 - a. Los materiales a revisar se distribuyen a los revisores con anticipación para que preparen anticipadamente la revisión entre pares.

- b. Los materiales a revisar incluyen las entradas importantes para el desarrollo de los productos de trabajo de software que serán sometidos a la revisión entre pares, por ejemplo:
- Los objetivos del producto de trabajo de software.
 - Los estándares aplicados.
 - Los requisitos para el diseño.
 - Los detalles importantes para un código.
- c. Se utilizan las listas de verificación para identificar criterios para la revisión de productos de trabajo de software de forma consistente (incluyen conceptos como cumplimiento con los estándares o procedimientos, completos, correctéz, reglas de construcción y mantenimiento).
3. Los datos y los resultados de las revisiones entre pares se registran.

Esto datos incluyen los tipos y número de defectos encontrados y arreglados.

Métricas:

1. Deben tomarse medidas que determinen el estatus de las actividades de la revisión entre pares.

Verificaciones:

1. El grupo de aseguramiento de la calidad del software revisa y/o audita las actividades y las revisiones entre pares de los productos de trabajo y reporta los resultados.

1.5.2. Ingeniería del producto de software.

El propósito de la ingeniería del producto de software es para realizar en forma consistente un proceso de ingeniería bien definido que integre todas las actividades de ingeniería de software para producir en forma efectiva y eficiente productos de software correctos y consistentes.

La Ingeniería del Producto de Software incluye realizar las tarea de ingeniería para construir y mantener el software utilizando los procesos de software definidos en el proyecto y los métodos y herramientas apropiadas.

Las tareas de la ingeniería del producto de software incluyen *analizar los requisitos* del software (estos requisitos son descritos en el área clave de Administración de Requisitos), desarrollar los requisitos del software, desarrollar la arquitectura del software, diseñar el

software, implementar el software en el código, integrar los componentes de software y probar el software *para verificar que satisface los requisitos especificados*.

La documentación necesaria para realizar las tareas de ingeniería de software (documentos de requisitos de software, documentos de diseño de software, plan de pruebas y procedimientos de prueba) es desarrollado y revisado para asegurar que cada tarea obtiene los resultados de las tareas anteriores y los resultados producidos sean los correctos para las tareas subsecuentes (incluyendo las tareas de operar y mantener el software).

Cuando los cambios son aprobados, los productos de trabajo de software, los planes, los compromisos, los procesos y las actividades son revisados para reflejar que los cambios hayan sido efectuados.

A continuación se describirán las actividades 5, 6 y 7 del área clave de Ingeniería del Producto de Software de acuerdo a CMM[®], las cuales se refieren a pruebas.

Actividades:

5. La prueba de software es realizada de acuerdo a los procesos de software definidos en el proyecto.
 - a. Los criterios de prueba son realizados y revisados con el cliente y los usuarios finales.
 - b. Se utilizan métodos eficaces para probar el software
 - c. La adecuación de las pruebas está basada en:
 - i. El nivel de las pruebas realizadas (unidad, integración, sistema, aceptación)
 - ii. La estrategia de la prueba seleccionada (funcional, estructural, estática).
 - iii. El alcance de la prueba a ser realizada.
 - d. Por cada nivel de prueba de software, se establecen y utilizan los criterios de prueba disponibles (ejemplo, las unidades de software deben concluir con éxito las pruebas de revisión entre pares y las pruebas de unidad, antes de introducirlas a la prueba de integración).
 - e. La pruebas de regresión se realizan, cuando es apropiado, en cada nivel de prueba si cambia el software a ser probado o su ambiente cambia.
 - f. El plan de prueba, los procedimientos de prueba y los casos de prueba se someten a la revisión entre pares antes de que sean considerados listos para usarse.
 - g. Los planes de prueba, los procedimientos de prueba, y los casos de prueba se administran y controlan.

Administrado y controlado implica que la versión del producto de trabajo en uso en un tiempo dado (pasado o presente) debe ser conocido y los cambios deberán incorporarse de forma controlada.

6. Las pruebas de integración del software se planean y realizan de acuerdo al proceso de software definido en el proyecto.
- Los planes para la prueba de integración se documentan y se basan en el plan de desarrollo del software.
 - Los casos de prueba de integración y los procedimientos de prueba se revisan con los responsables individuales de los requisitos, diseño del software y pruebas del sistema y aceptación.
 - Las pruebas de integración del software se realizan contra la versión del documento de requisitos del software y el documento del diseño de software.
7. Las pruebas del sistema y de aceptación del software se planean y realizan para demostrar que el software *satisface sus requisitos*.

La prueba del sistema se realiza para asegurar que el software satisface los requisitos del software. La prueba de aceptación se realiza para demostrar al cliente y a los usuarios finales que el software satisface los requisitos asignados.

- Los recursos para probar el software se asignan de manera oportuna para permitir una preparación adecuada de la prueba (documentación de la prueba, planear los recursos, desarrollar los conductores de la prueba y desarrollar simuladores)
- Las pruebas de aceptación y del sistema se documentan en el plan de pruebas, el cual es revisado con y aprobado por el cliente y los usuarios finales, cuando es apropiado. El plan de prueba cubre:
 - i. El enfoque general de prueba y verificación.
 - ii. Las responsabilidades de la organización de desarrollo, subcontratistas, clientes y usuarios finales.
- Los casos de prueba y los procedimientos de prueba se planean y preparan por un grupo de prueba que es independiente de los desarrolladores de software.
- Los casos de prueba se documentan, se revisan y son aprobados por el cliente y el usuario final, si es apropiado, antes de comenzar a probar.
- Las pruebas de software se aplican al software de las líneas base y a la documentación de líneas base contra los requisitos asignados y los requisitos del software.
- Los problemas identificados durante las pruebas son documentados y seguidos hasta su cierre.
- Los resultados de las pruebas se documentan y utilizan como la base para determinar si el software satisface sus requisitos.
- Los resultados de las pruebas se administran y controlan.

1.5.3 Aseguramiento de la Calidad del Software

- El objetivo del Aseguramiento de la Calidad del Software es administrar con la debida supervisión, los procesos que se usan en el Proyecto de Software (PS) y los productos construidos.
- La supervisión se logra con la revisión y auditoria de las actividades y de los productos de software, para verificar que cumplen con los estándares y procedimientos adecuados.
- Se revisan en primer lugar, los elementos acordados, los cuales se corrigen si así es necesario.
- El Grupo de Aseguramiento de la Calidad del Software (GAQS), deberá escalar apropiadamente la solución de los elementos que aún no han sido corregidos dentro del proyecto de software.

De acuerdo a CMM® el área clave de Aseguramiento de la Calidad de Software debe cumplir con:

Objetivos:

1. Planear las actividades de aseguramiento de calidad del software (AQS)
2. Verificar objetivamente que las actividades y los productos de software sigan los requisitos, estándares y procedimientos.
3. Informar las actividades y resultados de las acciones tomadas en el aseguramiento de la calidad.
4. El administrador general resolverá los conflictos del PS⁹ que no hayan sido corregidos.

Compromisos:

1. Seguir una politica estricta de la organización para la implantación del AQS¹⁰

Condiciones:

2. Debe existir un grupo responsable de coordinar e implantar el AQS en el proyecto.
3. Deben existir recursos y fondos adecuados para realizar las actividades del AQS
4. Los miembros del GAQS¹¹ deberán estar debidamente entrenados para realizar las actividades del AQS.
5. Los miembros del PS deberán tener una orientación en sus roles, responsabilidades, autoridad e importancia en el GAQS.

⁹PS Proyecto de Software.

¹⁰AQS Aseguramiento de la Calidad del Software.

¹¹GAQS Grupo de Aseguramiento de la Calidad del Software.

Actividades:

1. Preparar un plan de AQS (PAQS) para el PS, según los procedimientos documentados.
2. EL GAQS¹² debe realizar sus actividades según el PAQS¹².
3. EL GAQS debe participar en la preparación y revisión del plan de desarrollo del PS, de los estándares y procedimientos.
4. EL GAQS debe revisar que se realicen las actividades de IS.
5. El GAQS debe auditar los productos de trabajo del software diseñado.
6. El GAQS debe reportar periódicamente los resultados de sus actividades al GIS.
7. Documentar y manejar, según los procedimientos escritos, las desviaciones identificadas en las actividades y en los productos de software.
8. El GAQS debe realizar periódicamente revisiones de sus auditorías y resultados con el personal de AQS del cliente.

Métricas:

1. Deben tomarse medidas que determinen el costo y estado del calendario de las actividades de AQS.

Verificaciones:

1. Revisar periódicamente, las actividades de AQS junto con el Administrador principal.
2. Revisar periódicamente o según los eventos que se presenten, las actividades de AQS junto con el Administrador del Proyecto.
3. Periódicamente, un grupo independiente de expertos en AQS revisará las actividades y los productos de trabajo del GAQS del proyecto.

Las recomendaciones que se tomarán en cuenta, de estas áreas clave, para el apoyo del control de la calidad en el desarrollo de la tesis son:

1. La documentación de los defectos encontrados a un producto de software.
2. El control de los defectos encontrados a los productos de software para reportarlos y darle seguimiento hasta su eliminación.

¹²PAQS Plan de Aseguramiento de la Calidad del Software.

Capítulo II

II. Proceso de producción de software con control de calidad y cambios.

Con base a lo expuesto en el capítulo I se mostrará un proceso genérico para la producción de software donde los productos son controlados con respecto a la calidad y a los cambios.

El principal objetivo de un *proceso de software maduro* es el de producir productos de calidad que cumplan las expectativas de quien los solicita.

Los *requisitos* del producto frecuentemente evolucionan y pueden cambiar en cualquier etapa del proyecto. Estos cambios que pudieran darse deberán ser evaluados y negociados con los grupos afectados e incorporados de manera controlada en los productos de software.

2.1. Procesos de software maduro e inmaduro.

Existen diferentes grados de madurez en el *proceso de producción de software* relacionados con el control de calidad de los productos y el control de cambios y por lo tanto se darán diferentes grados de capacidad en la predicción de impactos presentes o futuros en un proyecto.

En el proceso de software las necesidades de un solicitante se transforman en requisitos y en diseño que es transformado en código que será documentado y certificado para su uso operacional.

Recordemos que un *producto* es un artefacto tangible de salida (o servicio) que es el resultado de un proceso y que intenta ser entregado a un cliente o a un usuario final.

Un *producto de trabajo de software* es cualquier artefacto creado como parte de la definición, mantenimiento, o uso de un proceso de software. Estos pueden incluir descripción de procesos, planes, procedimientos, programas de computadoras y documentación relacionada que puede o no ser entregado a un cliente o usuario final [CMM[®]].

Llamaremos *producto de software* o simplemente producto a cualquier producto de trabajo de software creado durante un proceso de software.

Los procesos de software que los grupos involucrados utilizan para desarrollar y mantener software pueden diferir significativamente sobre todo entre organizaciones inmaduras y maduras. Esto es más evidente si observamos la *capacidad del proceso de software* que describe el rango de resultados esperados que obtiene una organización siguiendo cada uno de sus procesos [CMM®].

Organizaciones inmaduras

- El Proceso de Software generalmente es improvisado por practicantes y administradores durante el curso del proyecto.
- Aún si existe un Proceso de Software especificado, este no es apoyado ni se toma en cuenta rigurosamente.
- Las organizaciones son reactivas, donde los administradores generalmente están enfocados en resolver crisis inmediatas.
- Estas organizaciones generalmente sobrepasan los presupuestos, los calendarios y tiempos de entrega, puesto que no están basados en estimaciones reales.
- En muchos de los casos, comprometen la funcionalidad y la calidad del producto en aras de cumplir con los calendarios y tiempos cuando se aproximan.
- No existen bases objetivas para juzgar la calidad del producto ni para resolver los problemas del proceso y aún menos los problemas del producto.
- No se toma en cuenta el cómo los pasos del Proceso de Software afectan a la calidad dificultando su predicción.
- Cuando un proyecto se retrasa, generalmente se recortan o incluso se eliminan las tareas de revisión y pruebas.
- Los clientes o usuarios finales tienen muy poca participación en el proceso hasta que el producto es liberado.

Organizaciones Maduras.

- La organización posee la habilidad para manejar el desarrollo de software y procesos de mantenimiento.
- Puede comunicar el Proceso de Software tanto al equipo actual como a los nuevos empleados.
- Realiza las tareas y actividades de acuerdo con el proceso planeado.
- El Proceso de Software está documentado, es utilizable y consistente con el trabajo que en realidad se realiza.
- La definición de los procesos se actualiza cada vez que es necesario y las mejoras se realizan en forma controlada mediante pilotos de prueba o con análisis de costo – beneficio.
- Existe en la organización una participación activa durante las actividades de mejora.
- Los roles y las responsabilidades dentro del proceso son claros durante el proyecto y dentro de la organización.
- Los administradores supervisan la calidad de los productos de software y los procesos que producen dichos productos.
- Existen bases objetivas y cuantitativas para juzgar la calidad del producto y para analizar los problemas del producto y del proceso.
- Los presupuestos, calendarios y tiempos están basados en información histórica y son realistas.
- Generalmente se alcanzan los resultados esperados en costos, calendarios, tiempos, funcionalidad y calidad del producto.
- En forma general, una organización madura sigue procesos disciplinados consistentemente, puesto que cada participante entiende el valor de los que se hace, además que existe la infraestructura necesaria.

En términos generales se puede decir que en un *proceso de software inmaduro*, como característica principal, domina la improvisación y no se evalúa la calidad de los productos.

En un *proceso de software maduro* se administran, miden y controlan los productos de software.

2.2. Roles en los grupos de producción de software.

En la literatura de la Ingeniería del Software [CMM®] podemos encontrar descripciones de diversos roles que intervienen en el proceso de producción de software.

Para fines de este trabajo nos concentraremos en los siguientes roles:

- Solicitante
- Productor
- Revisor
- Aprobador
- Bibliotecario

Estos roles están involucrados en diferentes etapas del "ciclo de vida" de un producto de software que abarca su creación a causa de una solicitud, su constante mejora mediante sus revisiones, su aprobación, su uso y hasta que se retire.

Así mismo estos roles pueden ser desempeñados por una o más personas, así será posible tener a un grupo de Productores, a un grupo de Revisores o a un grupo de Aprobadores, etc.

A continuación se describen brevemente las responsabilidades de cada rol con respecto a un producto de trabajo de software.

Solicitante

- El solicitante es el que realiza una solicitud de creación de un producto de software o la solicitud de cambio al mismo. Un solicitante puede ser ya sea el cliente o cualquier miembro del grupo del proyecto.

Productor

- El *Productor* será el responsable de producir un producto de software.
- Es responsable de corregir defectos y/o introducir cambios.

Revisor

- El *Revisor* verifica y/o valida el producto para asegurar la calidad de su contenido. En caso de encontrar defectos regresa el producto al productor con los defectos documentados. Las revisiones pueden realizarse con técnicas tipo inspecciones, revisiones entre colegas, etc., o pueden referirse a pruebas automatizadas tipo "testing".
- En caso de no encontrar defectos acepta el producto.

Aprobador

- El Aprobador determina si el producto cumple con los criterios de entrada a la biblioteca de las líneas base.
- En caso de encontrar defectos regresa el producto al productor con los defectos documentados.
- Si el Aprobador determina que el producto es aprobado éste podrá ingresar a la biblioteca de las línea base.

Bibliotecario

- El Bibliotecario organiza las diferentes bibliotecas del proyecto para cada línea base, guarda los componentes y los proporciona cuando se les hace un cambio y en su caso actualiza las bibliotecas.

2.3. Proceso iterativo genérico de producción de Software

La calidad de los productos de software y el control de cambios a éstos son elementos fundamentales que reflejan la madurez de un proceso de producción.

En la práctica de la industria del software y en los modelos de referencia tipo CMM® encontramos diferentes niveles de tratarlos.

A continuación se presentarán dos modelos con diferente nivel de control sobre:

- Control de producción y calidad del software
- Control de cambios a productos de software

Con estos modelos se plantearán diferentes contextos en los que podrían utilizarse.

Posteriormente se muestran diversas combinaciones que podrían darse entre los diferentes niveles de control de producción y calidad con control de cambios, siguiendo cada uno de los modelos propuestos.

Al final se propone un Modelo Genérico para el Control de Calidad y Cambios en los Productos de Software incluyendo a todos los roles descritos en la sección anterior.

Los niveles de procesos de producción de software que se muestran a continuación representan una forma gradual de cómo los grupos de producción de software pueden controlar la producción, la calidad y los cambios:

- **El Control de Producción** tiene que ver con la identificación de diferentes atributos del producto, tales como: el nombre, quien es el productor, la fecha de producción, su versión, la relación con otros productos y control de acceso del mismo que puede estar en un estado editable o no editable.
- **El Control de Calidad** está relacionado a las actividades de revisión y aprobación del producto con el fin de cumplir con los requisitos del solicitante y evitar los cambios no autorizados. Esta actividad se enfoca a básicamente en las actividades de encontrar y eliminar defectos. Tiene que ver con los atributos del producto como su estado, sus defectos y control de acceso.
- **El Control de Cambios** a los requisitos del producto de software se controlan mediante solicitudes, donde se detalla el porqué y quién desea el cambio, lo que permitirá apoyar el análisis de impacto y su seguimiento así como determinar si el cambio fue solicitado, aceptado, rechazado, implantado o verificado.

2.3.1. Niveles de control de producción y calidad del software

En esta sección se presentan tres niveles de control de producción y calidad de los productos de software, que podemos encontrar comúnmente en las prácticas de la industria del software. El nivel 1 es el que ofrece menos control y deja toda la responsabilidad en manos del Productor. El nivel 2 incluye revisiones que ayudan a asegurar mayor calidad del producto y finalmente el nivel 3, incluye las prácticas que apoyan la administración de configuración de productos de software adecuada [CMM®].

La tabla 2.1. muestra el grado de participación de los roles en cada uno de los niveles:

Roles	Nivel 1	Nivel 2	Nivel 3
Productor	√	√	√
Revisor		√	√
Aprobador			√
Bibliotecario			√

Tabla 2.1 Participación de los roles en los niveles de producción y calidad del software

Nivel 1. Control personal de producción y calidad

Característica general:

La calidad del producto depende totalmente del juicio del productor

En este nivel de control que representa seguramente el caso más familiar vivido; todas las actividades que tienen que ver con la producción del producto tales como: creación, revisión, finalización, resguardo, repositorio compartido, etc., están a cargo de una sola persona.

Este nivel podría darse por diferentes razones, justificables o no, que tiene que ver con las características mismas de la organización como son su tamaño en cuanto al número de personas disponibles para el apoyo a la producción, o por otras razones que tendrían que ver con el producto mismo que podría hacer que los administradores determinen que toda la responsabilidad este bajo un solo rol.

No siempre una organización tiene todos los recursos para contar con personal especializado o podría considerar que involucrar a más personas le signifique introducir un esquema burocrático.

La integridad del producto depende de la buena fe de cada uno de los miembros del equipo.

El concepto de calidad que vimos al inicio del capítulo podría estar en riesgo porque no se cuenta con la especificación de los requisitos para el producto y no existe una revisión diferente al del Productor:

Roles involucrados:

- El *Solicitante* quien solicita un producto informalmente.
- El *Productor* es el responsable de producir un producto de software que cumpla con los requisitos implícitos y una vez terminado lo ingresará a un repositorio compartido, con una versión n (definida por él) y el estado del producto sea "editable".

La figura 2.1 y la tabla 2.2 resumen las actividades de los roles en este nivel:

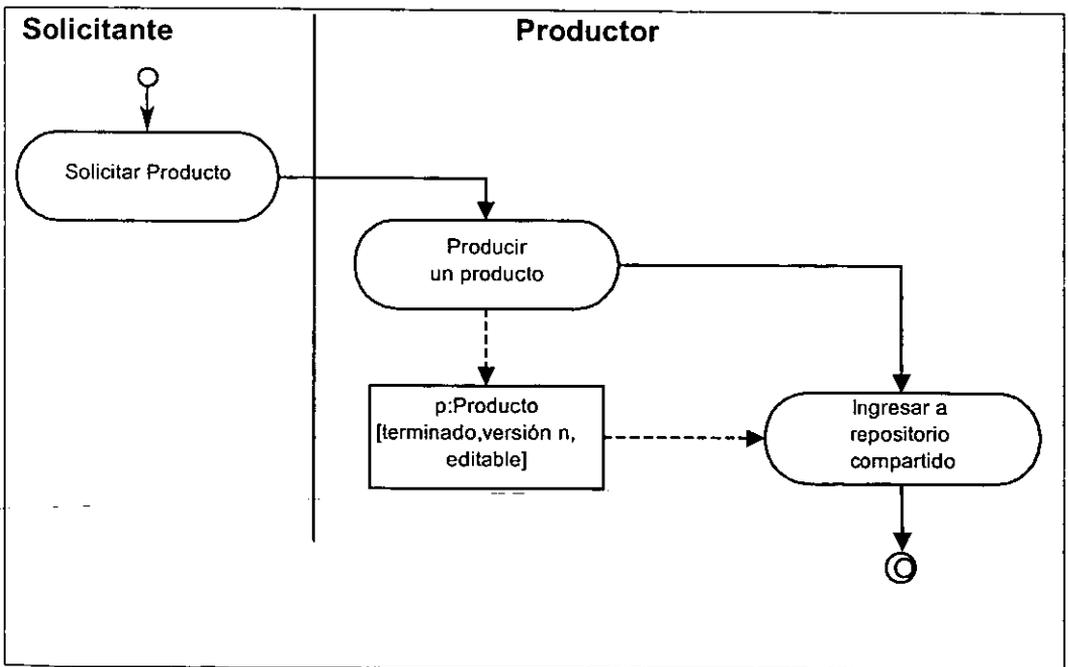


Figura 2.1 Nivel 1. Control personal de producción y calidad.

Responsable	Entrada	Actividad	Salida
Productor		Producir un producto	Producto: - terminado - versión n - estado editable

Tabla 2.2 Responsabilidades del Nivel 1 del control personal de producción y calidad del software.

Características:

1. No existe ningún control de calidad del producto independiente del Productor.

La calidad depende de:

- La habilidad del productor para:
 - Comprender correctamente la solicitud del producto.
 - Cumplir con los requisitos durante el desarrollo del producto.
 - Del buen juicio para la finalización del producto.
 - Ausencia de modificaciones no controladas del producto terminado.
2. No existe ningún control de consistencia de versiones porque el producto es editable por el mismo Productor o por cualquier miembro del equipo.
 3. No hay control de modificaciones a causa de defectos.

Este nivel de control es recomendado para solicitudes y productos simples y de bajo riesgo.

Nivel 2. Control de producción con revisión de calidad.

Característica general:

La calidad del producto depende de una revisión independiente al productor, el cual utiliza una especificación documentada de los requisitos.

En éste nivel se cuenta con unos requisitos documentados para el desarrollo del producto. Los requisitos pueden ser cualquier producto de trabajo, tales como: requisitos del cliente, documentos de análisis, diseño, el código, etc., que se usan para construir el producto en cuestión, y con respecto a ellos dicho producto tiene que estar correcto y consistente.

Se introduce el rol del Revisor cuya razón podría originarse desde la incipiente necesidad del Productor cuando acepta al decir: "terminé mi producto pero necesito que alguien me lo revise" o hasta una inspección formal.

Es de gran utilidad revisar si el producto cumple con los requisitos del solicitante por alguien diferente al que produjo el software.

Las revisiones pueden abarcar diferentes técnicas de verificación y validación desde las revisiones entre colegas, las inspecciones formales o las pruebas unitarias, de integración, de sistema o de aceptación. Un punto importante es el registro de defectos encontrados para facilitar el trabajo de corrección.

Roles Involucrados:

- El *Solicitante* quien solicita el producto por medio de la especificación de requisitos documentados.
- El *Productor* es el responsable de producir un producto de software de acuerdo con los requisitos especificados y documentados. Corregirá los defectos encontrados por el Revisor(es).

Cuando el Productor considere que el producto ya es correcto lo envía a revisión con el estado de borrador, versión de trabajo y no editable.

- *Revisor(es)* quien revisa si el producto es correcto y consistente y si tiene defectos, utilizando técnicas de validación/verificación. Si fuera el caso el producto será regresado al productor del producto con los *defectos* documentados para que sean corregidos.

Si el *Revisor* determina que el producto es aceptado cambia su estado a "revisado" con la versión n, será no editable e ingresa al repositorio compartido.

La figura 2.2 y la tabla 2.3 resumen las actividades de los roles en este nivel:

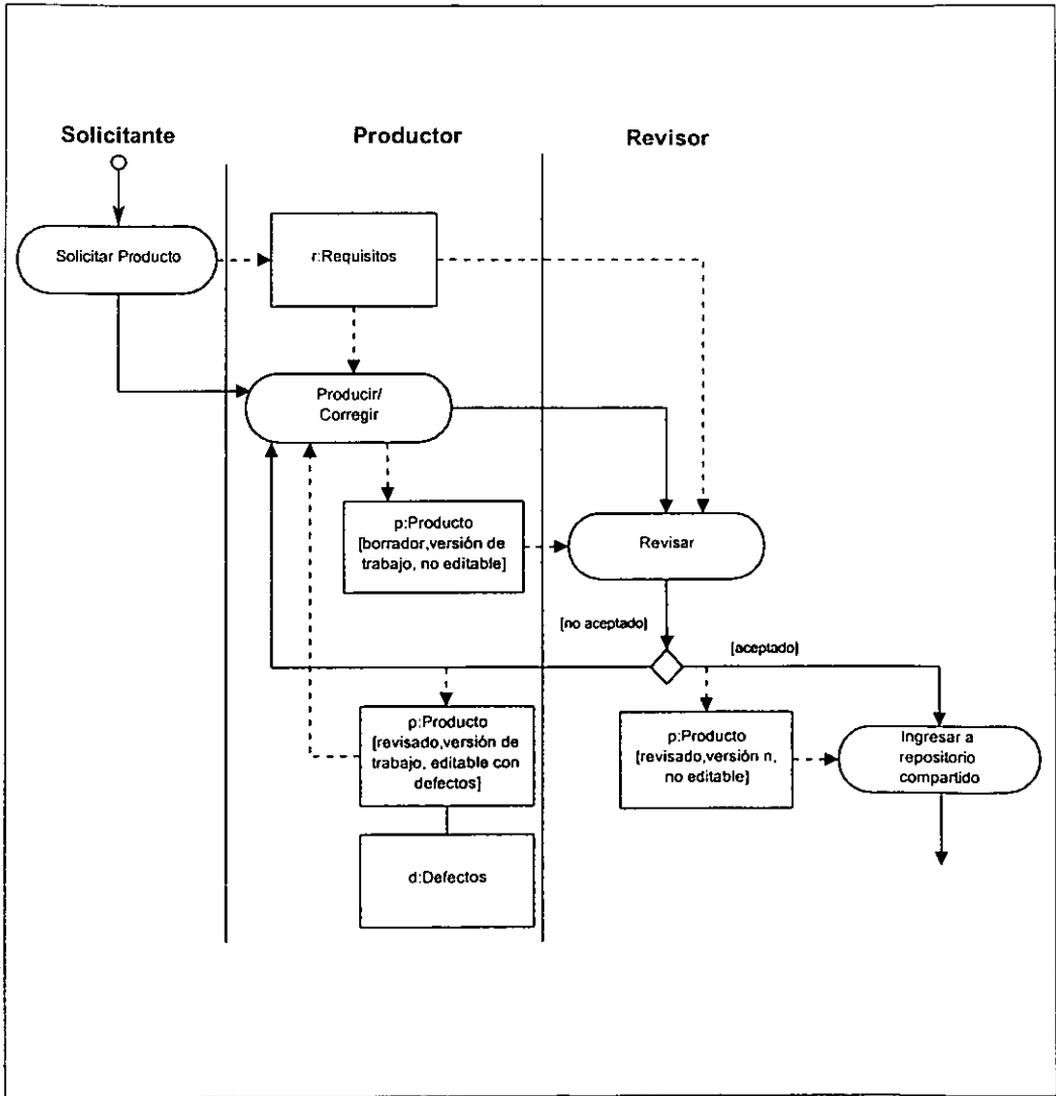


Figura 2.2 Nivel 2. Control de producción con revisión de calidad del software.

Responsable	Entrada	Actividad	Salida
Productor	Requisitos	Producir o corregir el producto	<ul style="list-style-type: none"> Producto: - borrador - versión de trabajo - no editable
Revisor	<ul style="list-style-type: none"> Producto: - borrador - versión de trabajo - no editable 	Revisar el producto sin encontrar defectos (aceptado)	<ul style="list-style-type: none"> Producto: - revisado - versión n - no editable
	<ul style="list-style-type: none"> Producto: - borrador - versión de trabajo - no editable 	Revisar el producto encontrando defectos (no aceptado)	<ul style="list-style-type: none"> Producto - borrador - versión de trabajo - editable Defectos

Tabla 2.3 Responsabilidades del nivel 2 de control de producción con revisión de calidad del software.

Características:

- El control de calidad depende de:
 - La especificación documentada de los requisitos y su comprensión por parte del Productor y del Revisor(es).
 - Las técnicas de revisión (verificación/validación).
 - La claridad de la documentación de los defectos para el productor.
- Existe control de versiones intermedias.
- Se añade el control de versiones del producto en estado terminado.
- Existe documentación y control de defectos del contenido del producto.

Este nivel de control es recomendado para toda clase de productos, especialmente para los entregados al cliente.

Nivel 3. Control de producción con doble control de calidad

Característica general:

La calidad del producto depende no solo del cumplimiento de la especificación de los requisitos sino también de la minimización del impacto en otros productos de la línea base.

Bajo este nivel se introducen dos roles más, el del Aprobador y el Bibliotecario, por obvias razones esto implica un mayor grado de especialización y de división de tareas que podría considerarse hasta cierto punto burocrático ya sea porque no se reconoce su utilidad o no se cuentan con los recursos necesarios.

En este nivel, un Aprobador revisa las características deseadas del producto, apoyándose en una lista de verificación y un Bibliotecario mantiene un esquema de administración que permite un almacenamiento y recuperación ágil de los diferentes productos de software.

Roles Involucrados:

- El *Solicitante* quien solicita el producto por medio de la especificación de requisitos documentados.
- El *Productor* quien crea un producto de software de acuerdo con los requisitos especificados y documentados. Corrige los defectos encontrados por el Revisor(es) y el Aprobador(es).

Cuando el Productor considere que el producto ya es correcto lo envía a revisión con el estado de borrador, versión de trabajo y no editable.

- *Revisor(es)* quien revisa si el producto es correcto y consistente y si tiene defectos, utilizando técnicas de validación/verificación. Si fuera el caso el producto será regresado al productor del producto con los *defectos* documentados para que sean corregidos.

Si el *Revisor* determina que el producto es aceptado cambia su estado a "revisado" con la versión n, será no editable e ingresa al repositorio compartido y será enviado al *Aprobador*

- *Aprobador(es)* determina si el producto tiene algún defecto que pueda impactar a otros productos existentes o futuros en las líneas base, apoyándose en una lista de verificación y si fuera el caso el producto será regresado al productor con los *defectos* documentados para que sean corregidos.

Si el *Aprobador* determina que el producto es aprobado podrá ingresar a la biblioteca de la línea base con una versión n y será no editable.

- El *Bibliotecario* organiza las diferentes bibliotecas, guarda los productos y los proporciona cuando se les hace un cambio y en su caso actualiza las bibliotecas.

Los productos se guardan en bibliotecas del sistema para cada línea base, de ahí se tomarán para generar nuevos productos o hacerles cambios.

Un producto no puede ser modificado mientras esté en la biblioteca.

La figura 2.3 y la tabla 2.4. resumen las actividades de los roles en este nivel:

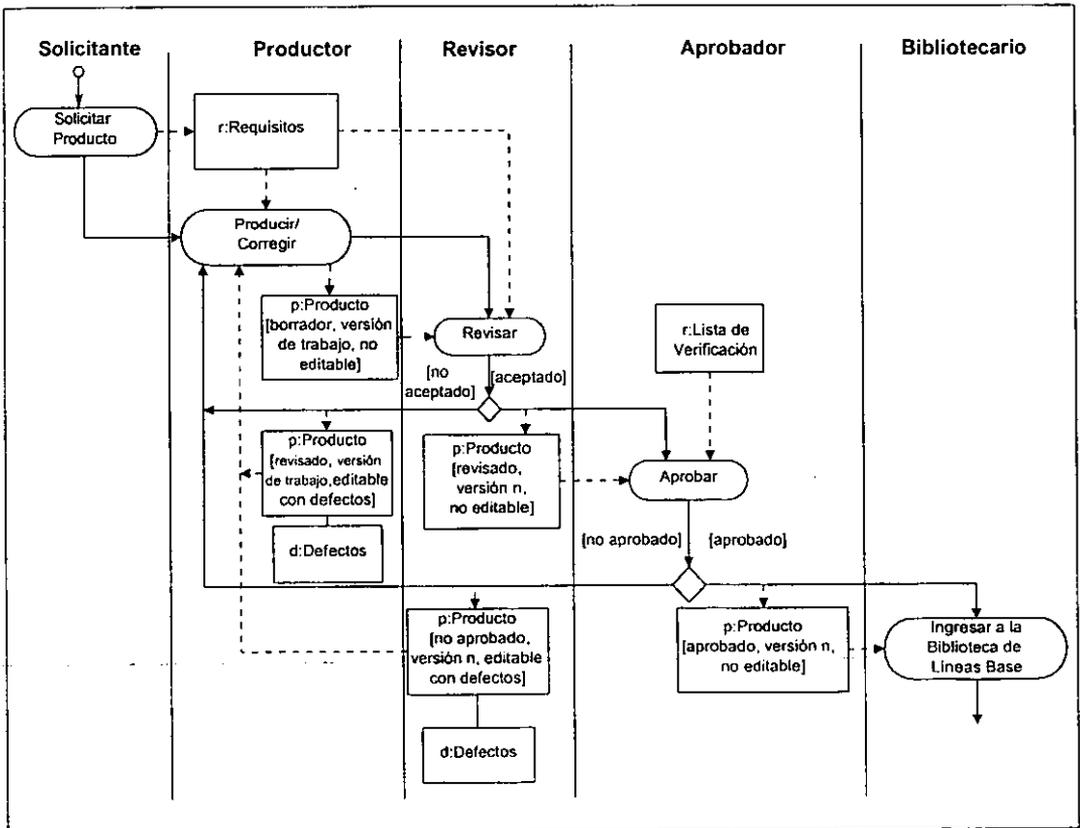


Figura 2.3 Nivel 3. Control de producción con doble control de calidad

II. Proceso de producción de software con control de calidad y cambios

Responsable	Entrada	Actividad	Salida
Productor	Requisitos	Producir o corregir el producto	<ul style="list-style-type: none"> Producto: <ul style="list-style-type: none"> - borrador - versión de trabajo - no editable
Revisor	<ul style="list-style-type: none"> Producto: <ul style="list-style-type: none"> - borrador - versión de trabajo - no editable 	Revisar el producto sin encontrar defectos (aceptado)	<ul style="list-style-type: none"> Producto: <ul style="list-style-type: none"> - revisado - versión n - no editable
	<ul style="list-style-type: none"> Producto: <ul style="list-style-type: none"> - borrador - versión de trabajo - no editable 	Revisar el producto encontrando defectos (no aceptado)	<ul style="list-style-type: none"> Producto <ul style="list-style-type: none"> - borrador - versión de trabajo - editable Defectos
Aprobador	Producto: <ul style="list-style-type: none"> - revisado - versión n - no editable 	Aprobar ingreso como línea base con defectos. (no aprobado)	<ul style="list-style-type: none"> Producto con sus defectos <ul style="list-style-type: none"> - no aprobado - versión n - no editable Defectos
	Producto: <ul style="list-style-type: none"> - revisado - versión n - no editable 	Aprobar ingreso como línea base sin defectos (aprobado)	<ul style="list-style-type: none"> Producto: <ul style="list-style-type: none"> - aprobado - versión n - no editable
Bibliotecario	Producto aprobado	Ingreso a la Biblioteca de línea base	

Tabla 2.4 Responsabilidades del nivel 3 de control de producción con doble control de calidad del software.

Características:

- La calidad del producto depende de:
 - Lo mismo que en el nivel 2 de control de producción con revisión de calidad.
 - Listas de verificación de elementos, para evaluar el producto antes de su inserción a la biblioteca de la línea base. Estos elementos deberán expresar los puntos de riesgo relacionados con la dependencia del producto con respecto a otros productos.
- Control de versiones intermedias.

3. Se añade el control de versiones de producto en estado terminado.
4. Control sobre el impacto a otros productos.
5. Control de defectos sobre el impacto a otros productos.

Este nivel de control es recomendado para productos de gran importancia, con tiempo de vida largo y necesidades de mantenimiento.

2.3.2. Niveles de control de cambios a productos de software

Los requisitos de un producto de software pueden evolucionar y cambiar en cualquier etapa del proyecto.

Los cambios a los requisitos obligan a realizar cambios a los productos de software. Estos cambios deberán ser evaluados y negociados con los grupos afectados con el fin de predecir los impactos al desarrollo del proyecto. Esta evaluación permite mantener la integridad de los productos de software y también un entendimiento común entre el solicitante y el grupo del proyecto de software.

La tabla 2.5 muestra el grado de participación de los roles en cada uno de los niveles de control de cambios:

Roles	Nivel 1	Nivel 2	Nivel 3
Productor	√	√	√
Revisor			√
Aprobador			√
Bibliotecario			√

Tabla 2.5 Participación de los roles en cada uno de los niveles de control de cambios

Nivel 1. Control verbal de cambios a productos de software.

Característica general:

La solicitud de cambio se realiza mediante una comunicación verbal.

Este sería el caso más común en donde las sugerencias de cambio hechas al productor suelen ser acompañadas del:

“Si el producto también incluyera podría estar mejor”

donde la solicitud se realiza en forma verbal, y en muchos casos improvisada, sin que medie una solicitud de cambio formal para ello, por lo tanto el productor no siempre se compromete con realizar el cambio.

El solicitante también podría creer que basta con mencionar su solicitud de cambio para que Productor “eche manos a la obra” y los conflictos que suelen derivarse de las posibles confusiones no se hacen esperar.

Rol:

- El *Solicitante* expresa verbalmente sus necesidades de cambio al producto de software.

El diagrama de la figura 2.4 resume la responsabilidad del rol bajo este nivel:

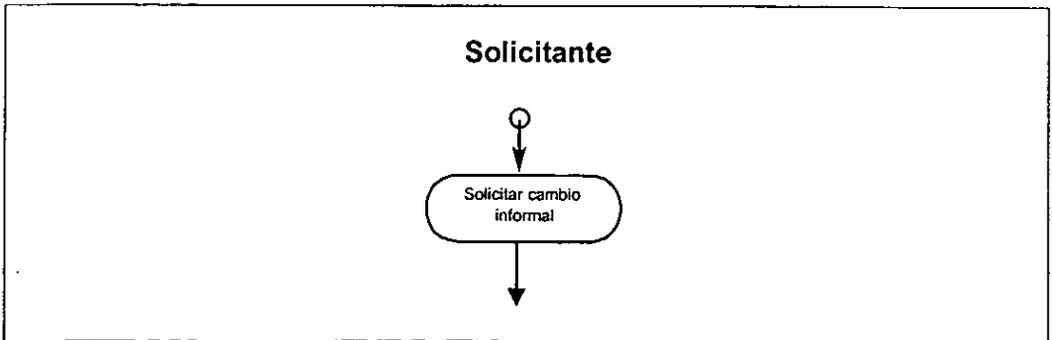


Figura 2.4 Nivel 1. Control verbal de cambios a productos de software

Características:

1. Se realiza una solicitud informal de cambio a un producto de software, por lo regular en forma verbal.
2. El control del cambio depende de una buena comprensión de la solicitud y una buena memoria del productor.
3. Falta de documentación.
4. Falta de análisis de impacto a otros productos.

Este nivel de control de cambios es recomendado para pequeños cambios que no son importantes para el contenido general del producto.

Nivel 2. Control documentado de cambios a productos de software.

Característica general:

La solicitud de cambio es documentada.

Bajo este nivel, se introduce la formalidad mediante una solicitud documentada del cambio a un producto de software que ya cumplió con un ciclo de desarrollo.

El objetivo es "no dejar nada en el aire" y que posteriormente implique diversas confusiones provocando que el producto no funcione adecuadamente o que impacte a otros productos.

Rol:

- El Solicitante quien expresa sus necesidades de cambio al producto de una forma documentada.

Las figuras 2.5 resume la responsabilidad del rol bajo este nivel:

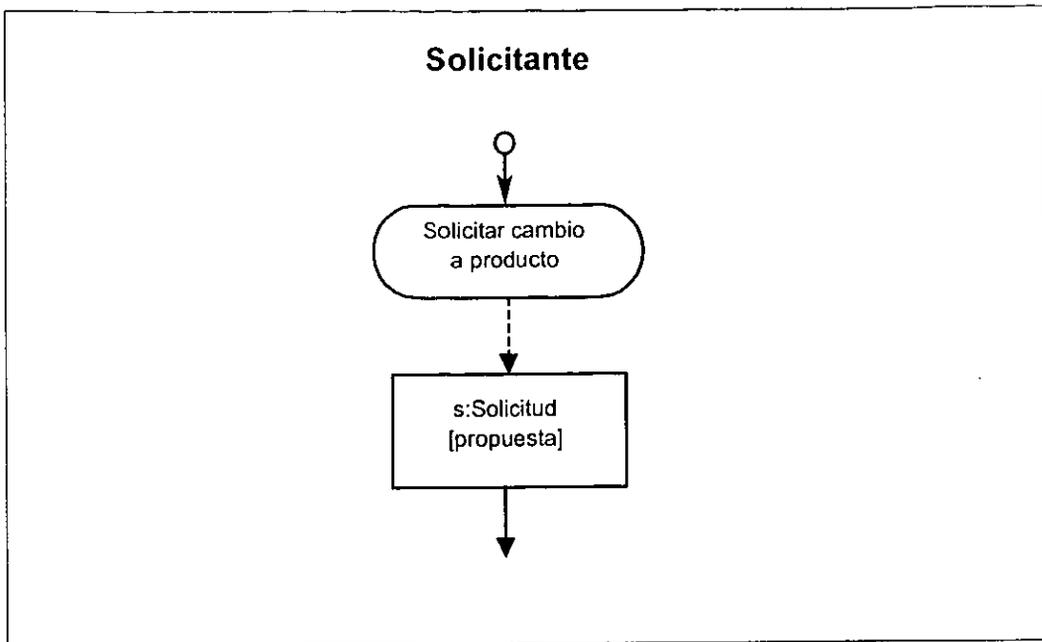


Figura 2.5 Nivel 2 . Control documentado de cambios a productos de software

Características:

1. Se realiza una solicitud documentada de un cambio a un producto de software.
2. El control del cambio depende de la buena comprensión de la solicitud documentada.
3. Falta de análisis de impacto a otros productos.

Este nivel de control es recomendado para cambios que no consumen tiempo y no impactan a otros productos.

Nivel 3. Control de cambios evaluados a productos de software

Característica general:

Son evaluados el impacto y el costo de la solicitud de cambio.

En este nivel de control de cambios, además del solicitante, están involucrados un Aprobador y un Bibliotecario. El primero tendrá que realizar el análisis de impacto del cambio solicitado al sistema y, el segundo se encargará de la administración del producto mismo, todo bajo un esquema que permita tener un fuerte control del producto, sobre todo considerando que el producto ya ha sido liberado y entregado al cliente o usuario final.

Roles:

- El *Solicitante* quien expresa sus necesidades del cambio al producto en una forma documentada.

Los cambios a los productos de software se controlarán una vez que hayan sido ingresados a la línea base.

Una vez que el producto ingresó a la línea base, si el *solicitante* desea hacer un cambio al producto, el solicitante deberá crear una *propuesta de cambio* a un producto aprobado.

- *Aprobador(es)* quien evalúa el impacto y el costo de la solicitud de cambio y decide si es o no aprobada.

El Aprobador tendrá que evaluar la *propuesta de cambio* para determinar el impacto al producto y al proyecto. Si la solicitud de cambio es aprobada el producto tendrá que someterse al proceso de cambio con el Productor.

- El *Bibliotecario* quien extrae el producto de la biblioteca de línea base a ser accesada por el Solicitante y el Aprobador(es).

La tabla 2.6 y la figura 2.6 resumen la responsabilidad del rol bajo este nivel:

Responsable	Entrada	Actividad	Salida
Solicitante	<ul style="list-style-type: none"> • Producto: <ul style="list-style-type: none"> - aprobado - versión n - no editable 	Solicitar cambio a producto	<ul style="list-style-type: none"> • Solicitud <ul style="list-style-type: none"> - propuesta
Aprobador	Solicitud propuesta	Evaluar solicitud de cambio (no aprobada)	Solicitud rechazada
	Solicitud propuesta	Evaluar solicitud de cambio (aprobada)	<ul style="list-style-type: none"> • Producto: <ul style="list-style-type: none"> - aprobado - versión n - editable • Solicitud aprobada

Figura 2.6 Responsabilidades del nivel 3 de control de cambios evaluados a productos de software

Características:

1. Existe documentación de la solicitud.
2. El control del cambio depende de:
 - La buena comprensión de la solicitud documentada.
 - La evaluación del impacto sobre el producto y otros productos y la estimación de los costos del cambio.

Recomendado para solicitudes de cambio las cuales consumen tiempo/costo y tienen impacto sobre otros productos que se encuentran en las línea base.

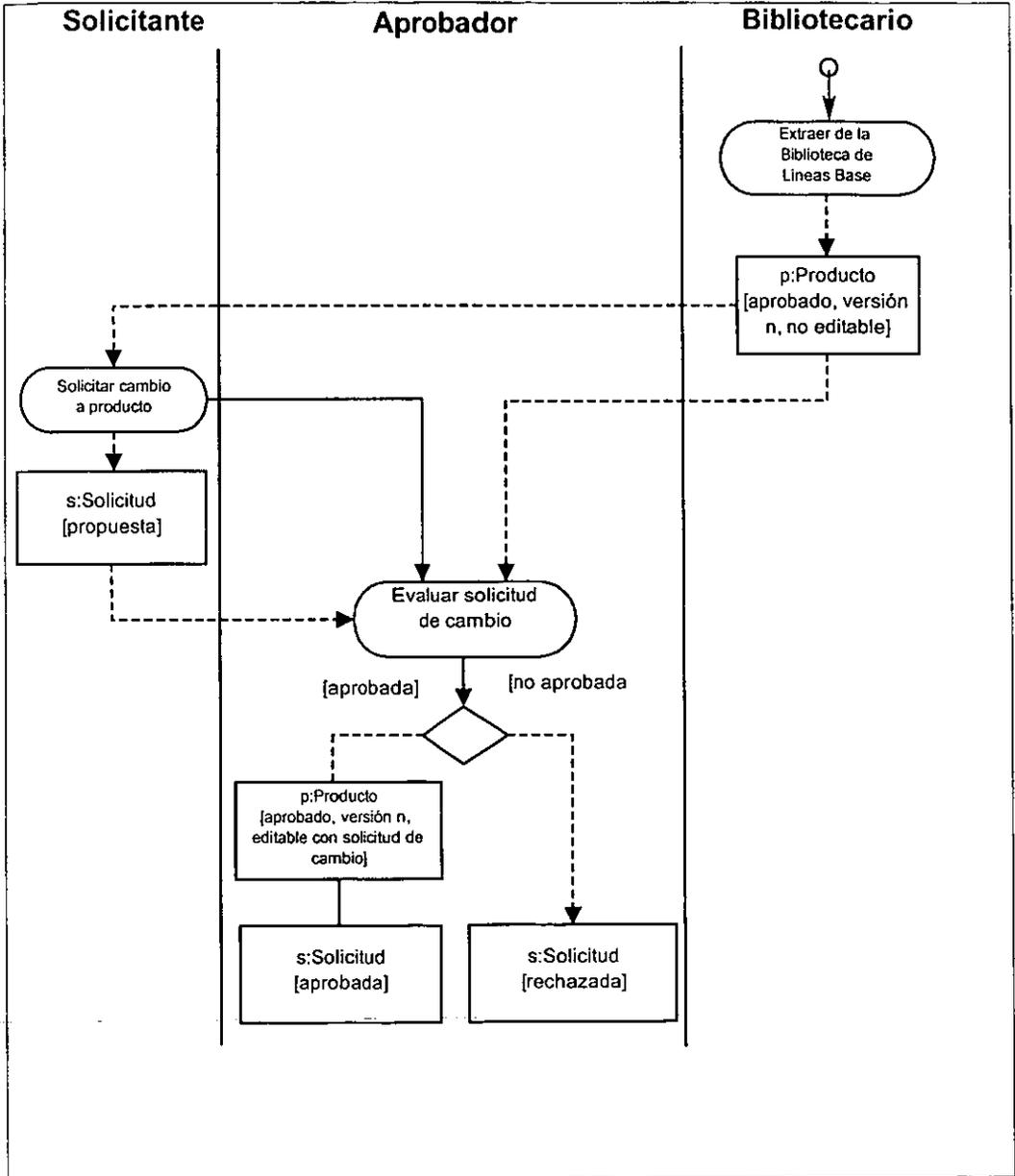


Figura 2.6 Nivel 3. Control de cambios evaluados a productos de software

2.3.3. Integración de control de calidad y cambios

Se expone un análisis del modelo de control de producción y calidad de software, con sus diferentes niveles, mediante su combinación con los niveles de control de cambios.

1. Control personal de producción y calidad (nivel 1) y control verbal de cambios a productos de software (nivel 1).

Este es el caso más común donde el control de la calidad depende del Productor y no existe control de cambios sobre el producto de software (figura 2.7) todo se reduce a la informalidad en donde los posibles defectos que el Productor encuentre al producto no serían documentados, como tampoco los cambios que le sean solicitados.

Aunque a simple vista pudiera parecer no muy recomendable, si lo es en casos donde una persona es dueña de todo y ese "todo" es de tamaño "controlable" y las solicitudes de cambio y los productos son simples.

Característica general:

- La calidad del producto depende totalmente del juicio del Productor
- La solicitud del cambio se realiza mediante una comunicación verbal

Roles involucrados

- El *Solicitante* es quien solicita un producto (informalmente) y expresa verbalmente sus necesidades de cambio del producto de software.
- El *Productor* es el responsable de producir un producto de software, realizar los cambios e introducirlo a un repositorio compartido cuando termine.

La Calidad depende de:

- La habilidad del productor para:
 - Comprender correctamente la solicitud del producto y la solicitud del cambio.
 - Cumplir con la solicitud durante el desarrollo del producto.
 - Del buen juicio para la finalización del producto.
- Ausencia de modificaciones controladas del producto terminado.

El control de los cambios depende de:

- La buena comprensión de la solicitud.
- Una buena memoria del productor.

No hay control de modificaciones a causa de defectos.

Este nivel de control es recomendado para solicitudes y productos simples y de bajo riesgo.

En cuanto al control de cambios, es recomendado para pequeños cambios que no son importantes para el contenido general del producto.

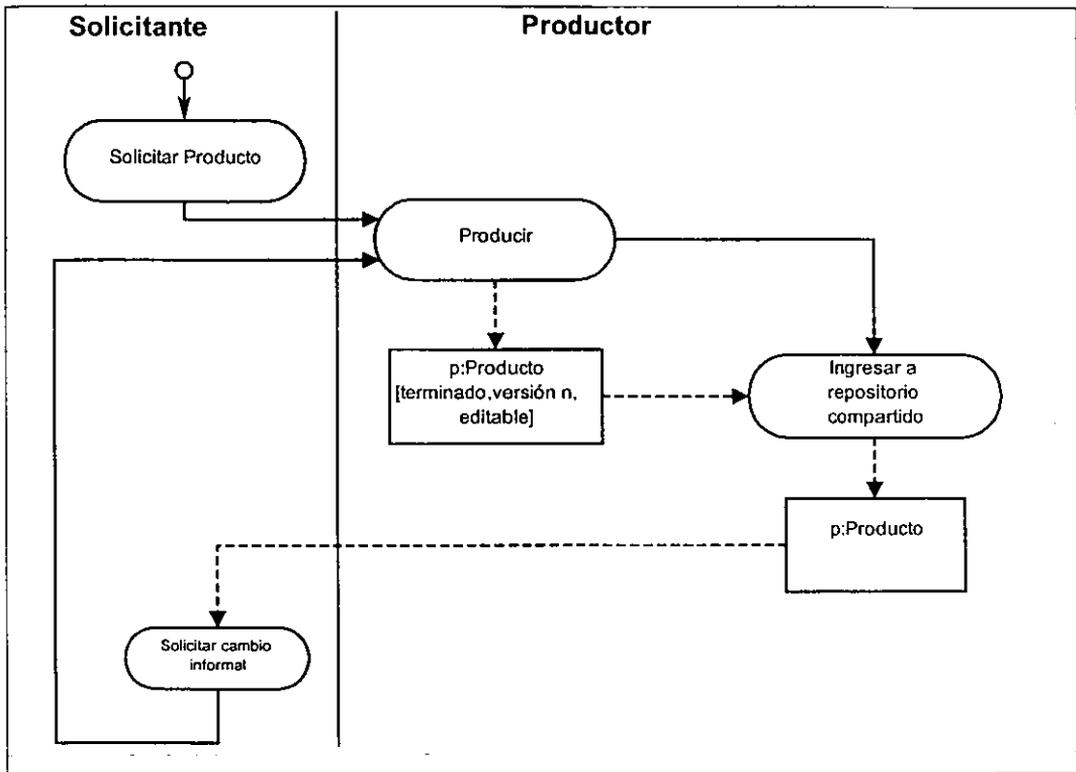


Figura 2.7 Control personal de producción y calidad (nivel 1) y control verbal de cambios a productos de software (nivel 1).

2. Control de producción con revisión de calidad (nivel 2) y control verbal de cambios a productos de software (nivel 1).

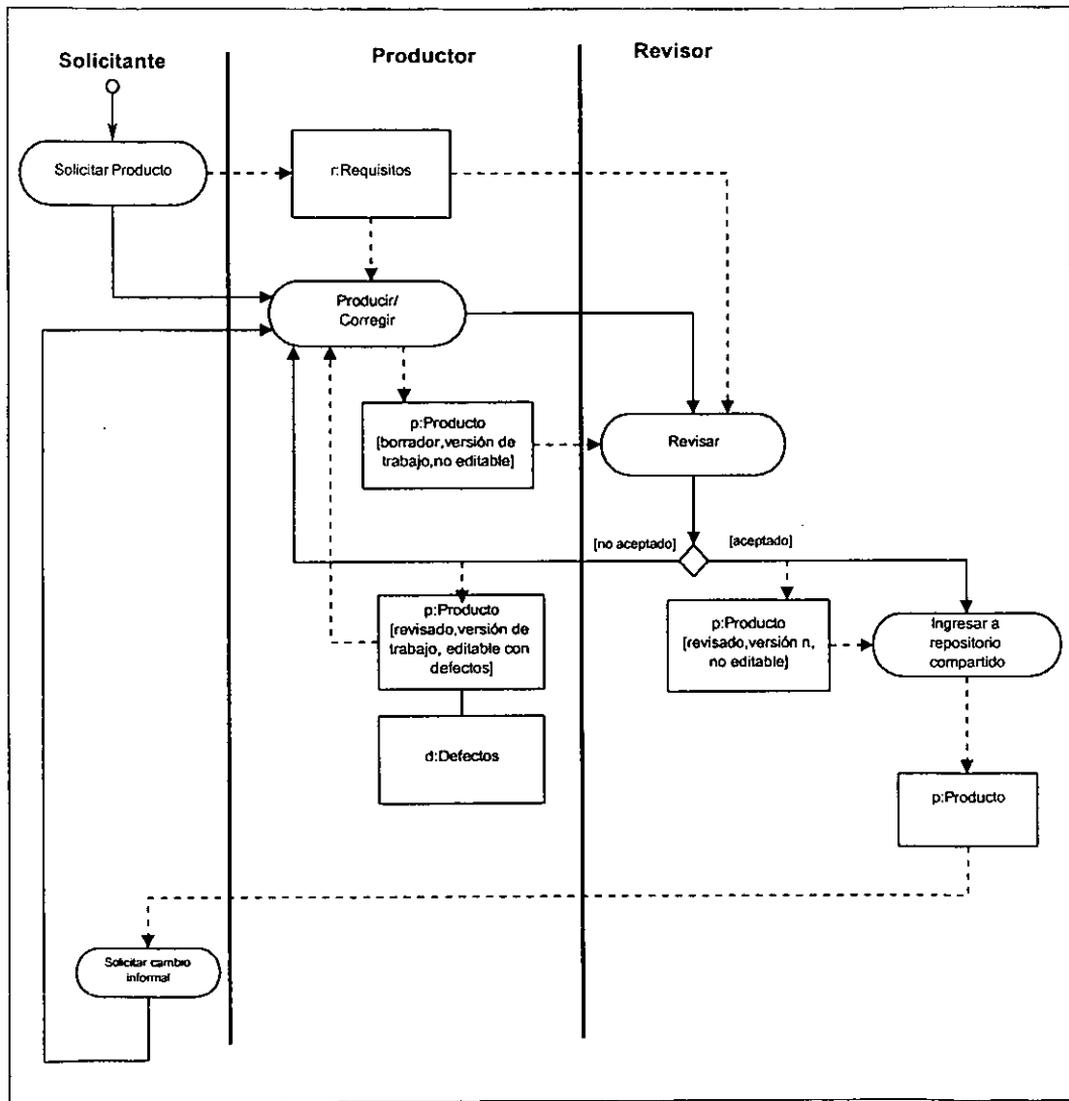


Figura 2.8 Control de producción con revisión de calidad (nivel 2) y control verbal de cambios a productos de software (nivel 1).

En el caso del control de producción y calidad nivel 2 con nivel 1 de control de cambios, (figura 2.8), se incluye el rol del Revisor, que proporciona un control formal de calidad mucho mejor del que pudiera ofrecer el Productor.

Debido a la naturaleza informal de las solicitudes de cambio el producto terminado podría no corresponder con las expectativas del solicitante a pesar que existe un Revisor.

Al Productor le solicitan directamente los cambios y el papel del Revisor podría quedar relegado o confuso ya que no sabría cuales son las expectativas del solicitante, "no sabría contra que revisar".

Por otro lado, dadas las características del rol del Productor, el análisis del impacto del producto sobre otros estaría limitado, salvo en caso de proyectos pequeños.

El control de cambios sobre el producto no existe, lo que puede dificultar la consistencia de versiones del producto.

Este esquema aunque factible, estaría limitado.

3. Control de producción con doble control de calidad (nivel 3) y control verbal de cambios a productos de software (nivel 1).

En este esquema sería muy riesgoso que un solicitante realice una solicitud de cambio informal, si en el modelo de producción y calidad existen los roles de revisor, aprobador y bibliotecario que controlan la producción del software.

4. Control personal de producción y calidad (nivel 1) y control documentado de cambios a productos de software (nivel 2).

Similar a 1, solo que existe una solicitud formal del cambio, (figura 2.9).

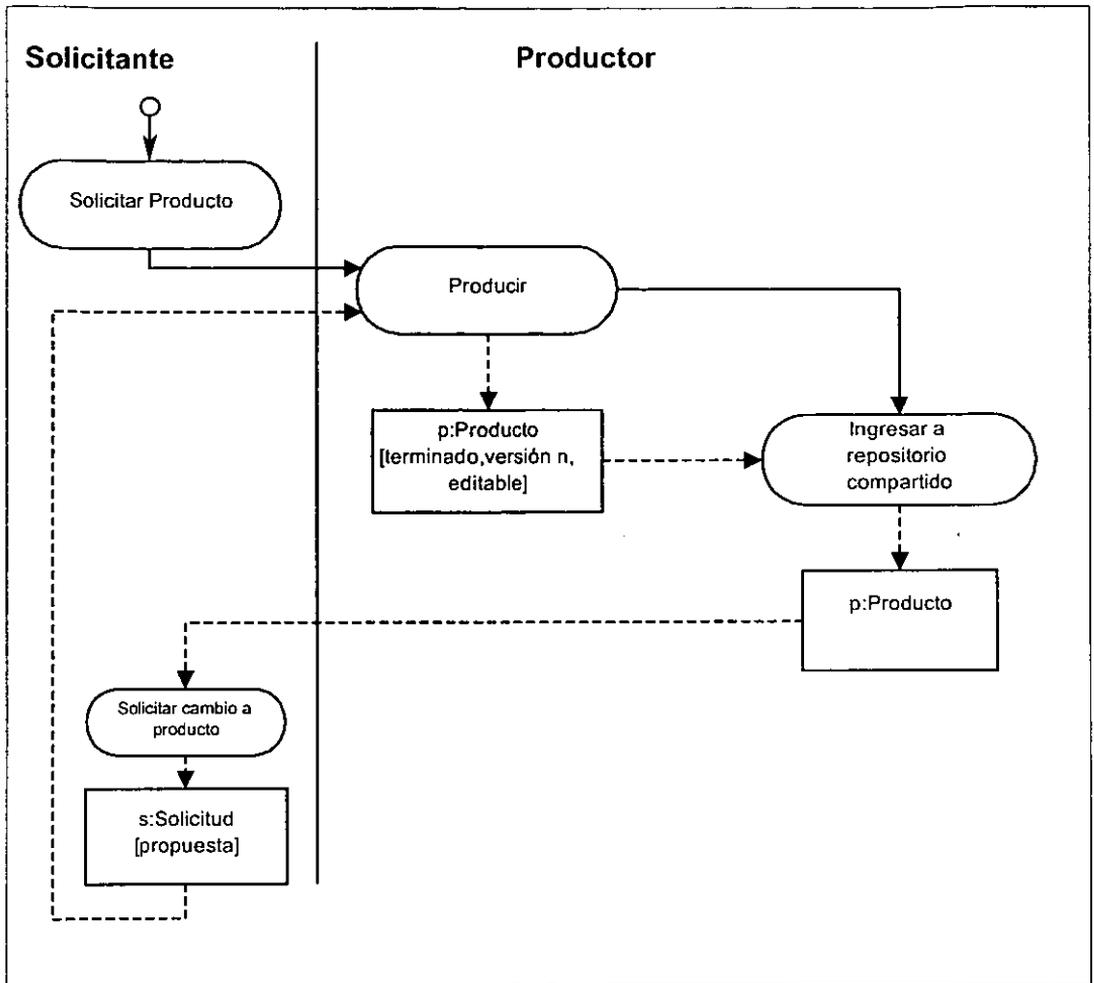


Figura 2.9 Control personal de producción y calidad (nivel 1) y control documentado de cambios a productos de software (nivel 2).

5. Control de producción con revisión de calidad (nivel 2) y control documentado de cambios a productos de software (nivel 2).

Característica General:

La calidad del producto depende de una revisión independiente al productor, el cual utiliza una especificación documentada de los requisitos. La solicitud del cambio es documentada.

Roles involucrados:

- El *Solicitante* quien solicita el producto por medio de la especificación de requisitos documentados y expresa sus necesidades de cambio al producto de una forma documentada.
- El *Productor* es el responsable de producir un producto de software de acuerdo con los requisitos especificados y documentados. Corregirá los defectos encontrados por el Revisor(es).
- *Revisor(es)* quien revisa utilizando técnicas de validación/verificación. Documenta los defectos y acepta el producto correcto.

La calidad depende de:

- La especificación documentada de los requisitos y su comprensión por parte del Productor y del Revisor(es).
- Las técnicas de revisión (verificación/validación).
- La claridad de la documentación de los defectos para el productor.

El control del cambio depende de:

- La buena comprensión de la solicitud documentada.

Este nivel de control es recomendado para toda clase de productos, especialmente para los entregados al cliente.

En cuanto a los cambios, es recomendado cuando no consumen tiempo y no impactan a otros productos.

6. Control de producción con doble control de calidad (nivel 3) y control documentado de cambios a productos de software (nivel 2).

Este esquema es riesgoso, porque un solicitante realiza la solicitud de cambio directamente al productor, ignorando el rol del aprobador.

7. Control personal de producción y calidad (nivel 1) con control de cambios evaluados a productos de software (nivel 3).

Este esquema no tiene sentido porque mientras que en el nivel 1 de control de producción y calidad se tiene un repositorio compartido, el nivel 3 de control de cambios exige una biblioteca para la línea base.

8. Control de producción con revisión de calidad (nivel 2) y control de cambios evaluados a productos de software (nivel 3).

Este esquema no tiene sentido porque mientras que en el nivel 2 de control de producción y calidad se tiene un repositorio compartido, el nivel 3 de control de cambios exige una biblioteca para la línea base, aunque exista una solicitud documentada del cambio.

9. Control de producción con doble control de calidad (nivel 3) y control de cambios evaluados a productos de software (nivel 3).

La figura 2.10, que incluye un modelo ideal de control de producción y calidad de software con control de cambios.

Característica general:

La calidad del producto depende no solo del cumplimiento de la especificación de los requisitos sino también de la minimización del impacto en otros productos de la línea base. Son evaluados el impacto y el costo de la solicitud de cambio.

Roles Involucrados:

- El *Solicitante*, quien solicita el producto por medio de la especificación de requisitos documentados y expresa sus necesidades del cambio al producto en una forma documentada.
- El *Productor*, quien produce un producto de software de acuerdo con los requisitos especificados y documentados. Corrige los defectos encontrados por el Revisor(es) y el Aprobador(es).
- *Revisor(es)*, quien revisa el producto utilizando técnicas de validación/verificación. Documenta los defectos y acepta el producto corregido.
- *Aprobador(es)*, quien analiza el producto revisado para asegurarse que puede introducirse a la línea base sin impactar inesperadamente a otros productos. También, evalúa el impacto y el costo de la solicitud de cambio y decide si es o no aprobada.
- El *Bibliotecario*, quien administra la biblioteca de los productos en la línea base.

La calidad del producto depende de:

- Lo mismo que en el nivel 2 de control de producción con revisión de calidad.
- Listas de verificación de elementos para evaluar antes la inserción del producto a la biblioteca de la línea base. Estos elementos deberán expresar la dependencia de puntos de riesgo del producto con respecto a otros productos del proyecto que será revisado.

El control del cambio depende de:

- La buena comprensión de la solicitud documentada.
- La evaluación del impacto sobre el producto y otros productos y la estimación de los costos del cambio.

Este esquema es recomendado para productos de gran importancia, con tiempo de vida largo y necesidades de mantenimiento y en la que las solicitudes de cambio pueden consumir tiempo/costo y tener impacto sobre otras líneas base.

II. Proceso de producción de software con control de calidad y cambios

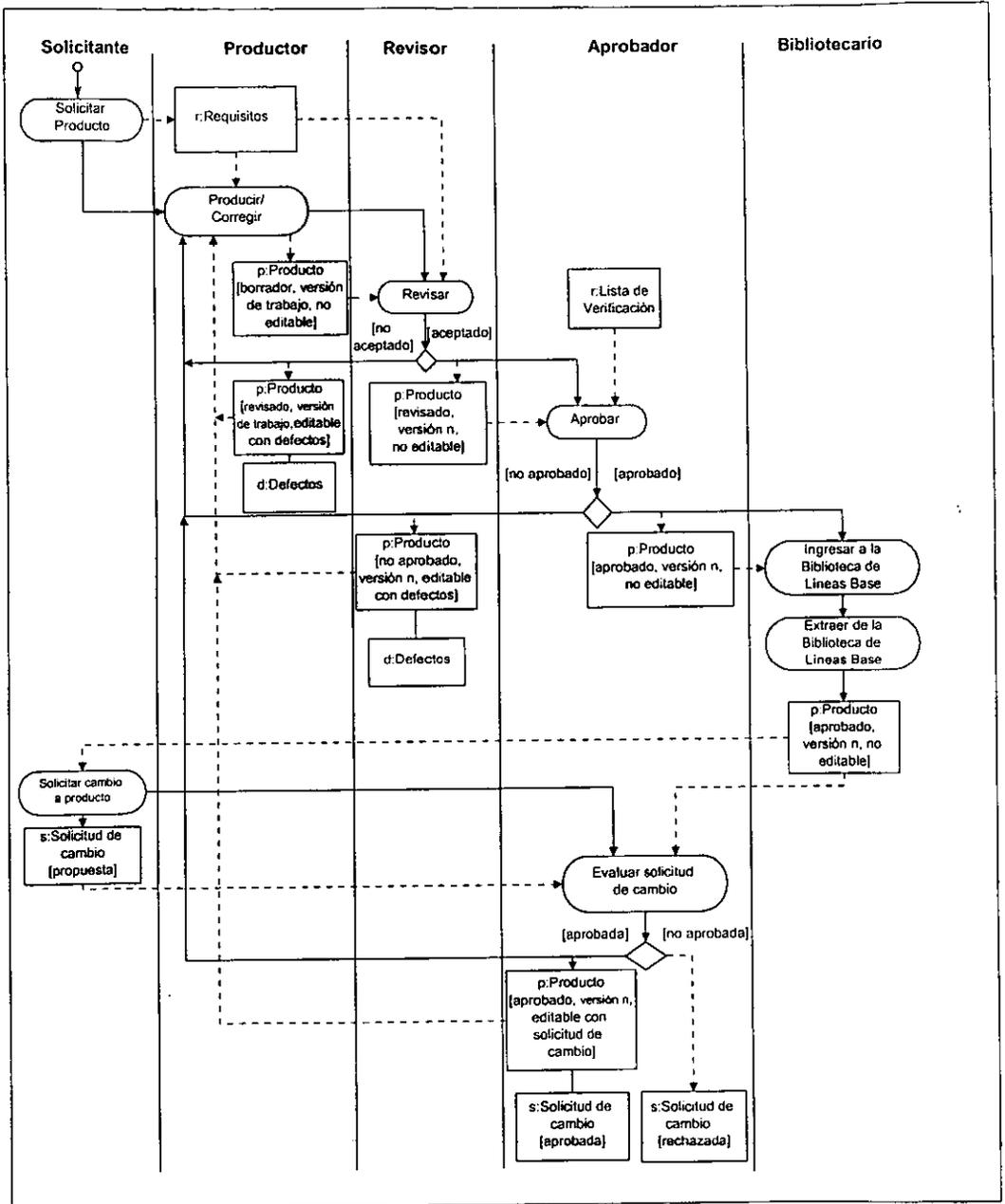


Figura 2.10 Control de producción con doble control de calidad (nivel 3) y control de cambios evaluados a productos de software (nivel 3).

2.3.4. Resumen de las posibles combinaciones de los diferentes niveles de control.

La tabla 2.7 resume las posibles combinaciones de los diferentes niveles de control de calidad y cambios a los productos de software:

- √ indica el esquema cuya combinación es factible de implementar, determinado por la cantidad de recursos disponibles o por las necesidades propias de los proyectos y/o de la empresa.
- X indica el esquema muy riesgoso para implementar.

Control de cambios	Control de producción y calidad		
	Nivel 1 Personal	Nivel 2 Con revisión de calidad	Nivel 3 Con doble control de calidad
Nivel 1, Verbal	√	√	X
Nivel 2, Documentado	√	√	X
Nivel 3, Evaluado	X	X	√

Tabla 2.7 Resumen de las posibles combinaciones de los diferentes niveles de control de calidad y cambios a los productos de software

2.3.5. Modelo Genérico de Control de Calidad y Cambios a los Productos de Software.

Del análisis realizado al integrar los modelos, podemos concluir que las opciones más factibles para ser utilizadas en la práctica son:

1. Control personal de producción y calidad (nivel 1) y control verbal de cambios a productos de software (nivel 1).
2. Control de producción con revisión de calidad (nivel 2) y control documentado de cambios a productos de software (nivel 2).
3. Control de producción con doble control de calidad (nivel 3) y control de cambios evaluados a productos de software (nivel 3).

Las características de estos niveles se detallaron en la sección anterior

Estas tres combinaciones lo llamaremos Modelo Genérico de tres niveles de Control de Calidad y Cambios en los Productos de Software.

- El primer nivel corresponde al caso clásico del *heroísmo*, donde todo depende de la visión limitada de una sola persona.
- El segundo nivel refleja el reconocimiento del *valor de colaborar en equipo y de la documentación escrita*.
- El tercer nivel puede considerarse para *el control ideal* de proyectos a largo plazo y de mayor complejidad.

El aspecto genérico del modelo está cubierto por múltiples significados de los elementos asociados al modelo. La tabla 2.8 ejemplifica las posibles substituciones por cada rol, actividad y objeto.

2.3.6. Aplicaciones del Modelo Genérico de Control de Calidad y Cambios a los Productos de Software.

El modelo genérico [OD2001] puede ser aplicado al menos para los tres siguientes propósitos:

1. Para definir el estándar de los procesos de control de calidad y cambios para una organización o proyecto.

La organización o proyecto puede clasificar los productos de trabajo de software en tres categorías: simple, término medio o a largo plazo, basado en las recomendaciones de cada nivel de control. Por cada categoría, una instancia del proceso de control de calidad y cambios substituye los elementos genéricos por uno en particular, utilizando la guía de la tabla 2.8. Cada producto de trabajo de software puede ser asignado al mismo nivel de control durante todo su ciclo de vida. Otra posibilidad es la de cambiar dinámicamente el nivel de control de acuerdo a la madurez del producto.

2. Para estimar el costo de producción y cambios , el cual incluye el costo del control de calidad.

Cada nivel de control del modelo, especifica las actividades necesarias que deben ser llevadas a cabo. Si podemos estimar para un producto su tamaño específico, el tiempo dedicado a cada actividad (producción, corrección, revisión, aprobación), y si podemos estimar el número de correcciones y cambios solicitados para este producto, entonces podemos estimar el costo total de la producción que incluye el control de la calidad y las modificaciones.

3. Para crear herramientas que administren diferentes niveles de control de calidad y cambios.

El modelo puede utilizarse para diseñar herramientas de producción de software que pueden administrar, en el mismo ambiente, diferentes niveles de producción, calidad y control de cambios. Las herramientas pueden permitir también cambios dinámicos en el nivel de control durante el ciclo de vida del producto.

La tabla 2.8 ejemplifica posibles substituciones de conceptos genéricos del modelo de control de calidad y cambios a los productos de software.

II. Proceso de producción de software con control de calidad y cambios

Conceptos genéricos		Posibles substituciones
Roles	Solicitante	Cliente, Usuario Final, Administrador, Ingeniero de Software
	Productor	Analista, Diseñador, Administrador, Ingeniero de Software
	Revisor	Cliente, Usuario Final, Par de Ingeniero de Software, Probador, Administrador
	Aprobador	Cliente, Administrador, Equipo de Aseguramiento de la Calidad, Ingeniero de Software, Experto, Mesa de Control de Configuración del Software
	Bibliotecario	Ingeniero de Software, Administración de la Configuración del Software
Actividades	Solicitar un producto	Mediante: Solicitud verbal, Solicitud documentada informalmente, Solicitud documentada formalmente
	Producir/Corregir	Para producir: Un plan, Especificación de Requisitos, Documentación en Análisis y Diseño, Planes de Prueba, Código, Manuales de Usuario
	Revisar	Revisión por el Administrador, Revisión con Cliente/Usuario Final, Revisión entre Pares, Inspecciones, Pruebas de Integración, Prueba del Sistema, Prueba de aceptación
	Aprobar	Revisión por el Administrador, Revisión por el Equipo de Aseguramiento de la Calidad, Revisión y Aprobación por parte de la Mesa de Control de Configuración del Software
	Insertar/Extraer de un repositorio común	Uso de directorio de archivos, Uso de herramientas simples de Administración de la Configuración, Uso de bibliotecas de líneas base.
Objetos	Producto	Plan, Especificación de Requisitos, Modelo de Análisis, Modelo de Diseño, Código, Casos de Prueba
	Requisitos	Requisitos Cliente/Usuario Final, Requisitos del Producto
	Defectos	Bitácora de defectos, bitácora de Pruebas
	Solicitud de Cambio	Solicitud de cambio verbal, Solicitud de cambio documentada informalmente, Solicitud de cambio documentada formalmente.
	Lista de Verificación	Lista implícita o explícita de elementos para revisar antes de aprobar.

Tabla 2.8 Ejemplos de posibles substituciones de conceptos genéricos del modelo de control de calidad y cambios a los productos de software.

El modelo genérico permite:

- La integración de la producción, la verificación corrección y el control de cambios de los productos de software en un solo proceso y que frecuentemente son referidos como prácticas o áreas separadas.
- El control de calidad y el control de los cambios en cada nivel puede verse como un patrón y especializarse de acuerdo a las necesidades o recursos particulares de la organización o proyecto
- Tres niveles de clasificación de las prácticas de calidad y control de cambios. El primer nivel es personal y aparentemente tiene un bajo costo en la producción pero probablemente un alto riesgo en la calidad. El segundo nivel tiene un mayor costo de producción pero la responsabilidad de la calidad no es más que de una persona. En el tercer nivel, los costos de producción parecen ser los más altos, pero el riesgo de la calidad puede ser la más baja. En la práctica, el tiempo que se pierde en las revisiones y aprobaciones puede ser recuperado para realizar las correcciones y especificaciones.

Además permite desarrollar un prototipo de una herramienta que soporte el modelo genérico y que también constituye el propósito de esta tesis.

Capítulo III

III. Herramientas de control de productos de software.

En este capítulo se revisan diversas herramientas que apoyan el control de los productos de software. Se inicia con la revisión de una tesis [PO1999] que muestra un prototipo para el control de documentos de administración de la calidad en empresas pequeñas, la cual fue un antecedente de este trabajo

Posteriormente se muestra una caracterización general de diversas herramientas disponibles en el mercado que controlan productos de software.

3.1. Prototipo para el control de los documentos de administración de la calidad en empresas pequeñas.

La tesis, donde se presenta este prototipo, realiza una revisión del estándar de calidad ISO 9000 y el modelo CMM para procesos de desarrollo de software, con el fin de conocer los requisitos y principios necesarios para el control de la documentación.

Estos conceptos forman parte del proceso de mejora de una organización y garantizan una producción de los documentos de calidad en forma disciplinada entre todos sus miembros.

El control se realiza principalmente en dos elementos, documentos y solicitudes de cambio, con las siguientes características:

1. Documentos.

- Identificación del documento
- Versión
- Título
- Estado (borrador, revisión, aprobado, implantado, obsoleto, rechazado).
- Identificación de quien lo aprueba
- Fecha de creación
- Fecha de aprobación
- Grupo al que pertenece el documento (Políticas, Procesos, Procedimientos, Formatos y Roles)
- Fecha de vigencia
- Breve descripción del documento

Los documentos se hacen disponibles en formato html¹ y se clasifican en grupos de políticas, procesos, procedimientos, formatos y roles.

Los documentos se pueden crear, editar, enviar a evaluación, leer su contenido, revisar, aprobar o rechazar a causa de una revisión o aprobación. Las actividades que se llevan a cabo durante su ciclo de evaluación son:

- Un documento al ser creado está en 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.
- 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 la que estaba vigente pasa al estado de 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.

2. Solicitudes de cambio a los documentos.

- Identificación de la solicitud de cambio
- Título
- Identificación del documento al que se solicita el cambio
- Fecha de la solicitud
- Estado (aprobada, rechazada, implantada)
- Identificación del que aprueba la solicitud
- 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 la evaluación e implantación

Las solicitudes de cambio pueden crearse, editarse o modificar su estado; e incluyen las siguientes actividades:

- La solicitud de cambio se crea con el estado de 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 cambia a implantada una vez que los cambios son incorporados en el documento.

¹HiperText Markup Language, el lenguaje utilizado para crear documentos en la World Wide Web.

Para llevar a cabo las actividades que controlan los documentos y las solicitudes de cambio, los usuarios del prototipo, tienen acceso a los documentos mediante una clave de usuario y una clave de acceso. Los usuarios, se agrupan en las siguientes categorías, dependiendo de los derechos que tengan:

- **Administrador.** Crea nuevos usuarios del sistema, establece las características de los grupos y define los estados de los documentos.
- **Propietario.** Crea y edita los documentos o solicitudes de cambio. Establece revisores y aprobadores para evaluar los documentos y los modifica cuando existen solicitudes de cambio.
- **Revisor.** Revisa el documento y puede cambiar su estado a revisado o rechazado, en cuyo caso elabora una solicitud de cambio justificando el rechazo.
- **Aprobador.** Aprueba 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.

Para el modelado y desarrollo del prototipo se utilizó principalmente el Modelo Orientado a Objetos, el Lenguaje de Modelado Unificado (UML) y como lenguaje de programación Java.

El esquema de este prototipo, constituye una excelente propuesta para una empresa, porque mejora la identificación y control de sus documentos de calidad y solicitudes de cambio. Sin embargo es un modelo de control único y hasta cierto punto puede clasificarse como rígido si las características del documento y/o recursos de las empresas no se ajustan a este esquema ya sea que el producto no lo requiera o no se tengan los suficientes recursos humanos.

Podríamos emplear este prototipo como base para obtener otro con mayor funcionalidad y flexibilidad al emplear la propuesta del Modelo Genérico de Control de Productos de Software del Capítulo II.

El modelo genérico incluye otros elementos de control como son los requisitos, defectos y los diferentes niveles de control de los productos de software acorde con algunas áreas clave del nivel 2 y 3 del modelo de capacidad del proceso de CMM como fueron definidas en dicho capítulo.

Con ello obtendríamos un mejor aseguramiento de la calidad del producto, al apoyar las revisiones mediante el control de sus requisitos y defectos y por consiguiente no solo se

lograría un mayor entendimiento con el solicitante y el grupo que lo desarrolla sino que también se reducirían los costos del mismo.

Por otra parte, si a la herramienta se le concibe para que funcione bajo un ambiente de desarrollo donde más de un usuario pueda acceder a los productos de software al mismo tiempo mejoraríamos la colaboración y comunicación entre los grupos de personas que intervienen en su ciclo de vida.

Hoy en día existen suficientes avances tecnológicos que hacen realidad este último punto y que serán aprovechados para el desarrollo del nuevo prototipo.

Las posibles mejoras al prototipo de control de documentos se resumen en las siguientes consideraciones:

1. Niveles configurables de control de productos de software que permitan controlar sus requisitos, defectos o solicitudes de cambio dependiendo de las características del producto o recursos del proyecto.
2. Hacer la herramienta disponible a más de una persona al mismo tiempo para mejorar la interacción con otros miembros del grupo de desarrollo del producto.
3. Implementar los mecanismos y características necesarias de los productos que permitan mantener su integridad cuando sean compartidos.
4. Permitir la participación de varios usuarios al mismo tiempo, en la evaluación de los productos y solicitudes de cambio.
5. La herramienta deberá funcionar bajo un ambiente de red cubriendo aspectos de seguridad y mejorar la comunicación entre los usuarios, con el uso del correo electrónico.
6. Simplificar la configuración del sistema bajo diferentes plataformas.
7. Contar con un registro histórico del ciclo de evaluación de los productos y de las personas que han participado en el mismo.
8. Facilitar la búsqueda de los productos

Los elementos básicos del prototipo de Control de Documentos de Calidad y las mejoras comprendidas en los puntos del 1 al 6, serán utilizados para crear el prototipo Control de Calidad y Cambios a los Productos de Software (C3PoS) que implementará el Modelo Genérico de Control de Calidad y Cambios expuesto en el capítulo II.

3.2 Características de las herramientas que apoyan el control de cambios a los productos de software.

Además de contar con un modelo para el control de la calidad y cambios a los productos de software, es de gran utilidad tener una herramienta como medio integrador de todas las actividades del grupo de desarrollo para la mejora de sus procesos.

En el mercado existen diversas herramientas que intentan implementar las funciones que se prescriben en las áreas clave de Nivel 2 de CMM, en particular el área clave de la Administración de la Configuración del Software (ACS). Las hay desde las que solo realizan el control de versiones hasta las orientadas a los procesos y que incluyen conceptos de aseguramiento de la calidad.

Las herramientas más comunes, que podrían clasificarse como básicas, solo realizan el control de versiones con un registro histórico de las modificaciones realizadas al contenido del producto, aunque sus proveedores por lo regular se refieran a esto como el control de cambios. Este control es muy simple ya que no incluye una solicitud de cambio a un producto sujeta a un proceso de evaluación por parte de un revisor y/o aprobador.

Por otra parte, existen herramientas de dominio público o libres, que no representa un costo económico adquirirlas, como Concurrent Versions System CVS o Revision Control System RCS (Cyclic Software) que por lo general solo ofrecen el control de versiones y el registro histórico de las modificaciones del contenido del producto.

Las herramientas que integran mayor funcionalidad de la Administración de la Configuración del Software y que incluyen elementos de aseguramiento de la calidad, por lo regular son de alto costo y es difícil disponer de versiones de prueba. Este es el ejemplo de eChange Man (SERENA Software, Inc.).

En el anexo A se presenta un resume de algunas herramientas disponibles en el mercado que apoyan el control de cambios a los productos de software.

Después de realizar un análisis de este tipo de herramientas, se obtuvo la siguiente caracterización:

1. **Control de versiones.** Incluye el control de las versiones del producto con sus respectivas revisiones.
2. **Registro histórico de las modificaciones al contenido de un producto.** Se realiza un registro histórico de las modificaciones que se hayan tenido que hacer al producto (agregar, eliminar o cambiar información).
3. **Control de cambios.** Utilizaremos esta característica para referirnos al control de las solicitudes de cambios a los productos sujetas a un proceso de evaluación.
4. **Soporte de la mesa de Control de la Configuración del Software.** Las personas que evalúan los productos y solicitudes de cambio.
5. **Reporte de los defectos encontrados al producto durante las verificaciones y validaciones.** La posibilidad de registrar y reportar los defectos encontrados a un producto como resultado de un proceso de evaluación.
6. **Administración de las líneas base.** La posibilidad de definir y administrar en determinada fase del ciclo de vida del proyecto a un conjunto de productos de software.
7. **Definición de los roles asociados a un producto.** Posibilidad de definir las responsabilidades y los derechos que tienen los usuarios en la producción y evaluación de los productos de software.
8. **Niveles configurables de control para un proyecto.** Posibilidad de definir un nivel único de control para la producción y evaluación de todos los productos de un proyecto.
9. **Niveles configurables de control para un producto de software.** Posibilidad de definir diferentes niveles de control para la producción y evaluación de cada uno de los productos de un proyecto.
10. **Control de seguridad, mediante el acceso autorizado a la herramienta y a los productos.** Por medio de claves de acceso que definan los derechos de los usuarios de la herramienta.
11. **Integración con otras herramientas.** La posibilidad de que la herramienta integre productos con formatos de diversos fabricantes como Microsoft Office, Lotus Notes, etc.

12. Trabajo en un ambiente en red. La posibilidad de compartir la producción, la consulta y evaluación de los productos entre varios miembros del grupo de desarrollo a la vez.

La característica 9 pretende indicar si la herramienta puede definir diferentes niveles de control de la calidad y cambios, que van desde la solicitud del producto y sus cambios en forma verbal hasta un control con mayor funcionalidad que incluye la definición de sus requisitos, reporte de defectos y evaluación de los cambios por un grupo de revisores o aprobadores. Tal y como se definió en el Modelo Genérico de Control de Calidad y Cambios en los Productos de Software en el capítulo II.

A continuación se proporciona un resumen de algunas herramientas seleccionadas que implementan las distintas funciones de la Administración de la Configuración del Software del Modelo CMM tomando como base la caracterización anterior:

A. ConDor. Del M. en C. Carlos Pérez Escobar.

Prototipo para el Control de los Documentos de Administración de la Calidad en Empresas Pequeñas [PO1999]. Prototipo que sirvió de base para el desarrollo de esta tesis.

B. eChange Man de SERENA Software, Inc.

Herramienta que incluye control sobre la administración del cambio a los productos de software [WEB5]. Centralmente administra el ciclo de vida de toda la aplicación en un ambiente multiplataforma y de red. Además proporciona seguridad en el control de versiones, análisis de impacto, auditoría, administración de la construcción y liberación del software y ofrece integración con otros productos

C. Aimware

Ofrece un conjunto de herramientas para la administración del software que permite a los administradores obtener el control de las solicitudes de los usuarios, información sobre guías de procesos, proyectos, desarrollo de productos de trabajo [WEB6]. La suite de imware que cubre Aimware son:

- Administración de requisitos
- Administración del producto
- Administración del proceso
- Administración de proyectos

D. PVCS Dimensions de MERANT.

Registra los cambios en el código, soporta múltiples espacios de trabajo y archiva revisiones de los archivos [WEB7]. Realiza un reporte de defectos, de requisitos, liberación al cliente y tareas de los proyectos.

E. CS-RCS de Component Software.

Esta herramienta básicamente realiza control de versiones y registro histórico de cambios al contenido del producto [WEB8].

F. Concurrent Versions System (CVS) de Cyclic Software.

Como la anterior esta herramienta básicamente realiza control de versiones y registro histórico de cambios al contenido del producto [WEB9].

En la figura 3.1 se muestra una tabla que resume las características de éstas herramientas:

No.	Característica	Producto					
		A	B	C	D	E	F
1	Control de versiones	√	√	√	√	√	√
2	Registro histórico de las modificaciones al contenido de un producto	X	√	√	√	√	√
3	Control de cambios	√	√	√	√	X	X
4	Soporte de la mesa de Control de la Configuración del Software	√	√	√	√	X	X
5	Reporte de los defectos encontrados al producto durante las verificaciones y validaciones	X	√	√	√	X	X
6	Administración de las líneas base	X	√	√	P	P	P
7	Definición de los roles asociados a un producto	√	√	√	X	X	X
8	Niveles configurables de control para un proyecto	X	√	√	X	X	X
9	Niveles configurables de control para un producto de software	X	X	X	X	X	X
10	Control de seguridad, mediante el acceso autorizado a la herramienta y a los productos	√	√	√	√	X	X
11	Integración con otras herramientas	X	√	√	√	√	√
12	Trabajo en un ambiente en red	X	√	√	√	√	√

Figura 3.1. Comparación de características de herramientas que apoyan el control de la calidad y cambios a los productos de software

- √ : Cumple con la característica
- X : No cumple con la característica
- P : Cumple parcialmente

La caracterización de herramientas que apoyan el control de la calidad y cambios a los productos de software puede servir de referencia para comprender y evaluar a otras herramientas de éste tipo.

Capítulo IV

IV. Marco contextual.

A continuación se resumen algunos elementos de tecnología que se tomarán en cuenta para el diseño e implementación del prototipo Control de Calidad y Cambios a los Productos de Software y que denominaremos C3PoS.

4.1. Redes de computadoras

En los últimos años, se han obtenido mejoras extraordinarias en la tecnología de las redes de computadoras tanto en los elementos de hardware y software que la constituyen, así como al uso de mejores medios de comunicación de alta velocidad y por nuevos protocolos de red.

El ejemplo más notable lo tenemos con el desarrollo explosivo de Internet, que logra un intercambio de información rápido y confiable entre los usuarios y sus procesos bajo diversos ambientes de trabajo de redes de computadoras, equipo y sistemas operativos de diferente proveedor.

El prototipo C3PoS trabajará bajo un ambiente de red con el fin de que los productos de software sean compartidos por diferentes miembros del grupo de desarrollo de un proyecto de software.

Internet/Intranet

Internet es una red global que conecta a millones de computadoras en más de 100 países. Hasta 1999 Internet tenía más de 200 millones de usuarios alrededor del mundo y su número aumenta día con día [WEB1].

El diseño descentralizado de Internet permite que cada computadora sea independiente, el operador puede escoger el servicio de Internet a utilizar y a su vez el operador puede determinar el tipo de servicio que desee hacer disponible al resto de la comunidad.

El origen de Internet se debió al esfuerzo de interconectar la red ARPAnet del Departamento de Defensa estadounidense con otras redes, todas enlazadas por medio de satélite y de radio. ARPAnet era una red experimental que apoyaba la investigación militar y su objetivo era lograr y mantener a toda costa la comunicación entre los centros de investigación y el Departamento de Defensa norteamericano [DU1995].

Si una organización desea que su información se comparta solo entre sus miembros y exportar a Internet una mínima parte de ésta, puede realizar una solución basada en Internet pero para fines internos, denominada Intranet.

Una Intranet dispone de herramientas y procedimientos estandarizados en Internet apoyándose en la red corporativa de la organización.

Para el prototipo C3PoS se emplearán los servicios que ofrece Internet mediante una conexión a una red de computadoras con acceso a Internet utilizando el protocolo de red TCP/IP y los servicios de correo electrónico. El objetivo es que los usuarios de C3PoS accedan y distribuyan los productos de software desde cualquier lugar.

4.2. World Wide Web (WWW)

World Wide Web también conocida como WWW o simplemente Web, es un sistema de diseminación y recuperación de información distribuida que se inició en Europa (CERN, Suiza) [Pau1999].

En la Web no existe un control o administración central, "cualquiera" con conexión a Internet puede recuperar y hacer disponible información. La Web Está formada por una gran colección de documentos que se localizan en computadoras de todo el mundo creados por organizaciones académicas, profesionales, gubernamentales, comerciales y por individuos.

La Web basa su diseño y funcionamiento en los siguientes elementos:

- **La idea de un mundo sin límites para la información.** Todos los elementos de información están identificados por una clave que permite recuperarlos, mediante un sistema de direcciones (URI¹/URL²) al margen del protocolo empleado.

¹ Universal Resource Identifiers

² Universal Resource Locators

- **Un protocolo de red (HTTP³).** Este protocolo se basa en el Protocolo de Internet IP (Internet Protocol).
- **Un lenguaje (HTML⁴).** Es un estándar para la descripción de documentos que permite integrar objetos de distinta naturaleza (texto, gráficos o sonidos) y el usuario de la Web lo usa para transferir información a través de la red.
- **Un servidor Web.** Administra las aplicaciones y los elementos de información que se hacen disponibles al usuario de la Web.

La comunicación en la Web sigue el modo de operación propio de esquemas cliente-servidor. En un extremo de la línea hay una computadora sobre la que corre la aplicación servidora HTTP, en el otro extremo existe una computadora con un programa cliente, llamado navegador, "browser" o visualizador como ejemplo tenemos a Internet Explorer de Microsoft y a Netscape de Netscape Communications Corporation, figura 4.1. [Cona1999].

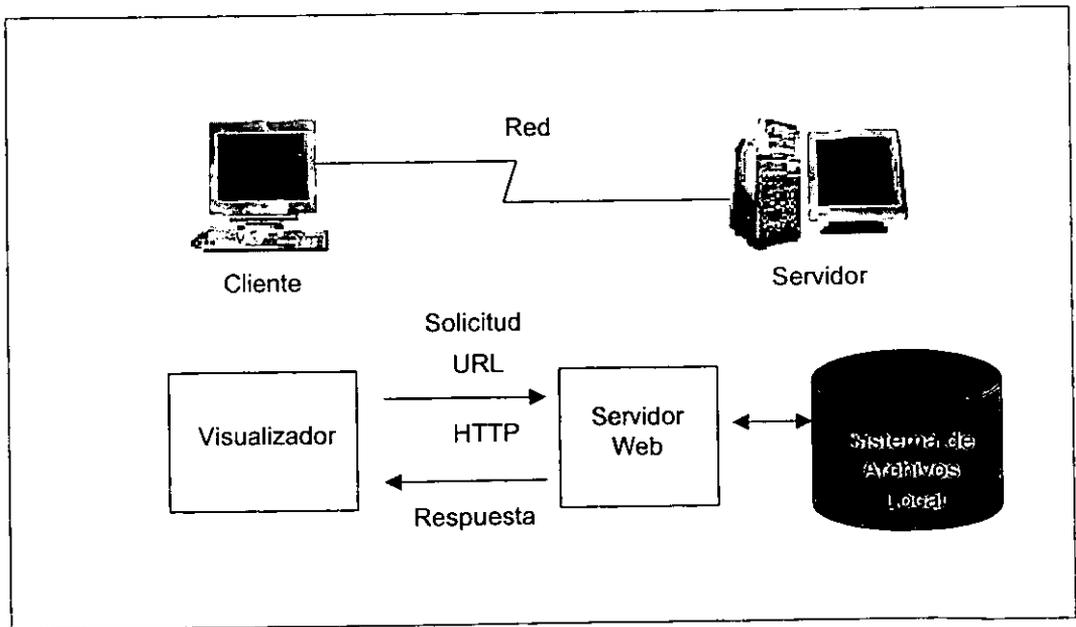


Figura 4.1 Arquitectura Web.

³ HyperText Transfer Protocol

⁴ HyperText Markup Language

El protocolo HTTP rige la forma en que el cliente y el servidor intercambian información.

Cuando se quiere obtener un documento HTML el cliente envía al servidor (cuya dirección conoce a través del URL del documento) una petición. El servidor Web recupera el documento y se lo envía al cliente. Si éste no existiera o el cliente no estuviera autorizado a leerlo, o se produjera cualquier problema, el servidor cancelaría la operación y devolvería un mensaje de error.

El documento también puede contener a su vez referencias a imágenes y a otros objetos.

Las ventajas de HTML son:

- Una sintaxis simple, con un conjunto fijo de etiquetas que le dan su funcionalidad.
- Permite crear fácilmente documentos multimedia al incorporar imágenes y sonido.
- Permite enlazar varios documentos entre sí.

Una de sus principales desventajas se debe a ese conjunto fijo de etiquetas. La única forma de mejorar la funcionalidad de HTML mediante etiquetas nuevas es elaborar una propuesta en la que venga detallada la funcionalidad deseada y remitirla para su discusión al W3C (World Wide Web Consortium) que es el consorcio involucrado en Internet y la Web. El proceso de debate puede ser largo y no todas las etiquetas son lo suficientemente generales para que puedan ser incluidas en la especificación HTML.

Por lo anterior existen otras opciones, como lo es XML (Extensible Markup Language) que constituye una especificación desarrollada por el W3C diseñada específicamente para documentos Web. Hace posible que los diseñadores creen sus propias etiquetas permitiendo la definición, transmisión, validación e interpretación de los datos entre las aplicaciones y entre las organizaciones que la empleen [MTU2000].

Sin embargo para el desarrollo del prototipo de C3PoS no se seleccionó como elemento tecnológico.

Se utilizará para el diseño e implementación del prototipo C3PoS un esquema cliente-servidor compuesto por un Servidor Web, para administrar la aplicación y los servicios que los usuarios de un proyecto de software soliciten para acceder a los productos a través de navegadores Web como Internet Explorer o Netscape.

4.3. Tecnologías orientadas a objetos.

Los principios de las tecnologías orientadas a los objetos aparecen en lenguajes de programación como Simula, a finales de los años 60, para aplicaciones industriales que fue creado para la simulación de procesos físicos complejos que podía simplificarse en gran medida por una división sistemática en módulos pequeños y que fueran directamente comprensibles [Cha1997].

La idea principal de Simula está basada en la división de estos módulos en objetos físicos que se quieren representar.

En la tecnología orientada a los objetos, tanto el modelado y la programación lo constituyen un conjunto de objetos que interactúan mediante el envío de mensajes y sus métodos que definen su comportamiento.

Un mensaje está constituido simplemente por el nombre del objeto destinatario, el nombre de un método de ese objeto y una lista eventualmente vacía de datos llamados *parámetros* o bien *argumentos* del mensaje.

Una aplicación exigirá muchos más objetos y categorías de objetos que un ejemplar único de cada categoría, es aquí donde interviene la noción de clase. Una clase es un modelo que define los atributos y los métodos comunes a una colección de objetos similares. Estos atributos y métodos compartidos se definen una vez y sólo una en la clase.

Los objetos que pertenecen a una clase son llamados las *instancias* de la clase y no difieren unos de otros más que por los valores de sus atributos.

Objetos, mensajes y clases son los elementos principales de las tecnologías orientadas a objetos. Estos conceptos aparecen actualmente en diversos lenguajes de programación, pero también en el interior de ciertas bases de datos, de interfaces hombre máquina, de modelos y metodologías de desarrollo de sistemas de información, de elementos de información en la Web y en general de Internet.

Para el desarrollo del prototipo C3PoS se utilizará la tecnología orientada a objetos por las siguientes ventajas:

- El análisis se facilita porque los objetos permiten un modelado directo de los objetos reales y sus comportamientos (usuarios, productos de software, solicitudes de cambio, etc.).
- El proceso de programación se simplifica porque estos objetos son reutilizables, fáciles de adaptar y de ensamblar, facilitando la rápida creación de prototipos.

Otras de las ventajas es la reutilización de los objetos conforme cambian las necesidades de los proyectos a diferencia de los procesos tradicionales de desarrollo de programas.

UML

El Lenguaje de Modelado Unificado UML (Unified Modeling Language) es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema.

UML proporciona una forma estándar de escribir los "planos" de un sistema, cubriendo tanto las cosas conceptuales tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes de software reutilizables [JBR999].

Debido a las facilidades que ofrece UML de visualizar, especificar, construir y documentar sistemas orientados a objetos se escogió como lenguaje de modelado para el prototipo C3PoS de esta tesis.

Al utilizar el diseño orientado a objetos se distribuyen las funciones y las responsabilidades entre varios objetos del prototipo. Los objetos colaboran o interactúan para cumplir con sus responsabilidades y mediante UML los representamos con:

- **Casos de Uso.** Describen el dominio de los procesos.
- **Diagramas de Clase.** Describen la asignación de las responsabilidades y las interacciones de objetos que muestran la definición de clases y el flujo de mensajes entre los objetos de software.
- **Diagramas de Secuencia.** Muestra la interacción entre los objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos.
- **Diagramas de Actividades.** Para modelar la vida de los objetos destacando el flujo de control entre actividades.
- **Diagramas de Estados.** Para modelar la vida de los objetos mostrando el flujo de control entre estados.

Se organizarán las clases en grupos denominados *paquetes*.

Java.

Se utilizará al lenguaje de programación Java para el desarrollo del prototipo por su naturaleza orientada a objetos y su independencia de la plataforma que permite ser ejecutado en plataformas y sistemas operativos distintos [Jaw1999].

El lenguaje de programación Java nació apoyando la resolución de aplicaciones en la Web.

El código de Java puede proporcionar *contenido ejecutable* en documentos Web. Esto se logra incluyendo *applets*, que son una forma especial de su código, en documentos Web. Un navegador (Internet Explorer y Netscape) permite que una aplicación recupere y ejecute *applets* en la computadora del cliente, lo que hace que una página Web resulte mucho más dinámica, figura 4.2, [Pau1999].

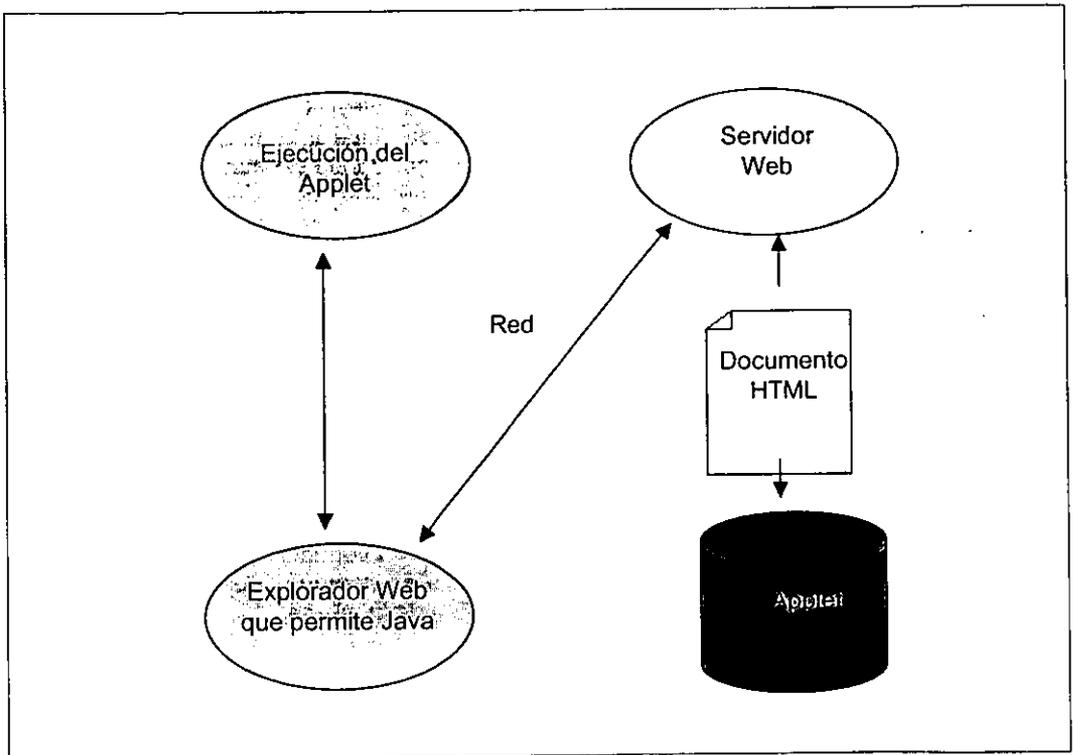


Figura 4.2 Applet en página Web.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Los applets se pueden usar para crear animación, figuras, juegos, formularios que respondan inmediatamente a la entrada del lector, u otros efectos interactivos entre textos y gráficos en las mismas páginas Web.

Los applets se descargan de la Web en la computadora del cliente igual que las páginas HTML, gráficos o cualquier otro elemento de un sitio Web. El applet iniciará su ejecución en cuanto esté descargado por completo.

Java tiene la característica de ser compilado e interpretado de forma simultánea. Cuando un programador realiza una aplicación en Java y lo compila, en realidad, el compilador Java únicamente genera el denominado ByteCode.

El ByteCode es un código intermedio entre el lenguaje de máquina del procesador y Java. Para ejecutar una aplicación Java es necesario disponer de un mecanismo que permita ejecutar el ByteCode que se denomina Máquina Virtual Java.

En cada plataforma (Solaris, Linux, Windows 95/98/NT, HP-UX, MacOS, etc.) existe una máquina virtual específica. Así que cuando el ByteCode llega a la máquina virtual, ésta lo interpreta convirtiéndolo a código máquina del procesador donde se esté trabajando para su ejecución.

Cuando el mismo ByteCode llega a diferentes plataformas, éste se ejecutará de forma correcta, pues en cada una de esas plataformas existirá la máquina virtual adecuada.

Por lo anterior se consigue la famosa multiplataforma de Java, que con sólo codificar una vez, podemos ejecutarlo en varias plataformas.

4.4. Integración de Bases de Datos en la Web

Como la Web es un medio para localizar, enviar, recibir información de diversos tipos, la integración de las bases de datos en la Web ayudarían a reducir costos y a almacenar información además de aumentar la rapidez de difusión de la misma.

Para la integración de bases de datos con la Web es necesario contar con una interfaz que realice las conexiones, extraiga la información de la base de datos, le dé un formato adecuado de tal manera que puede ser visualizada desde un navegador de la Web (Internet Explorer y Netscape) y permita lograr sesiones interactivas entre el usuario y la base de datos.

Un tipo de interfaz con cierto nivel de programación que permite comunicarse con las bases de datos lo constituye el elemento tecnológico Java y su Conectividad de Bases de Datos (JDBC) como se muestra en la figura 4.3 [Pau1999].

JDBC es una especificación de un conjunto de clases y métodos de operación que permiten a cualquier programa Java acceder a sistemas de bases de datos de forma homogénea. Lógicamente, al igual que ODBC, la aplicación de Java debe tener acceso a un driver JDBC adecuado. Este driver es el que implementa la funcionalidad de todas las clases de acceso a datos y proporciona la comunicación entre el API JDBC y la base de datos real [WEB2].

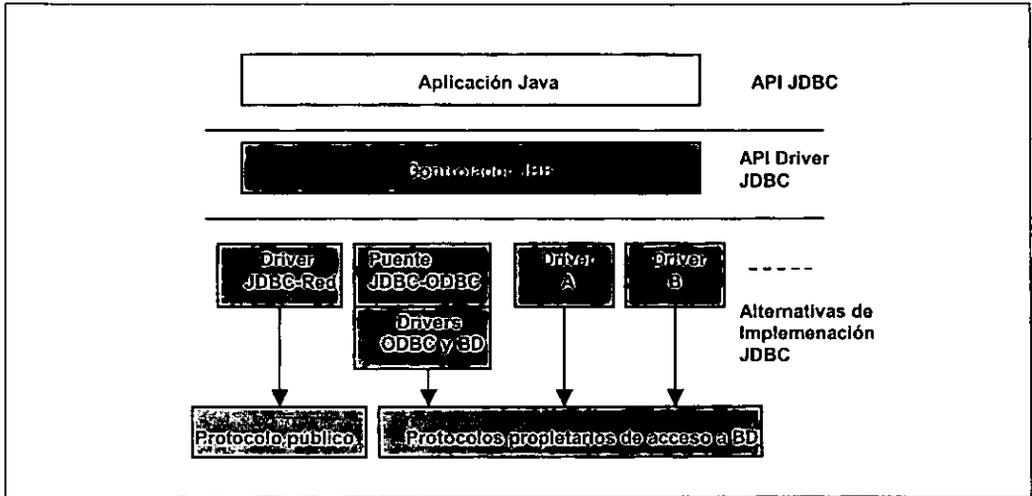


Figura 4.3 Conectividad JDBC.

El JDBC soporta dos modelos diferentes de acceso a Bases de Datos, los modelos de dos y tres capas. En el Modelo de dos capas la conexión entre la aplicación Java o el applet que se ejecuta en el navegador, se conectan directamente a la base de datos como se muestra en la figura 4.4 [WEB2].

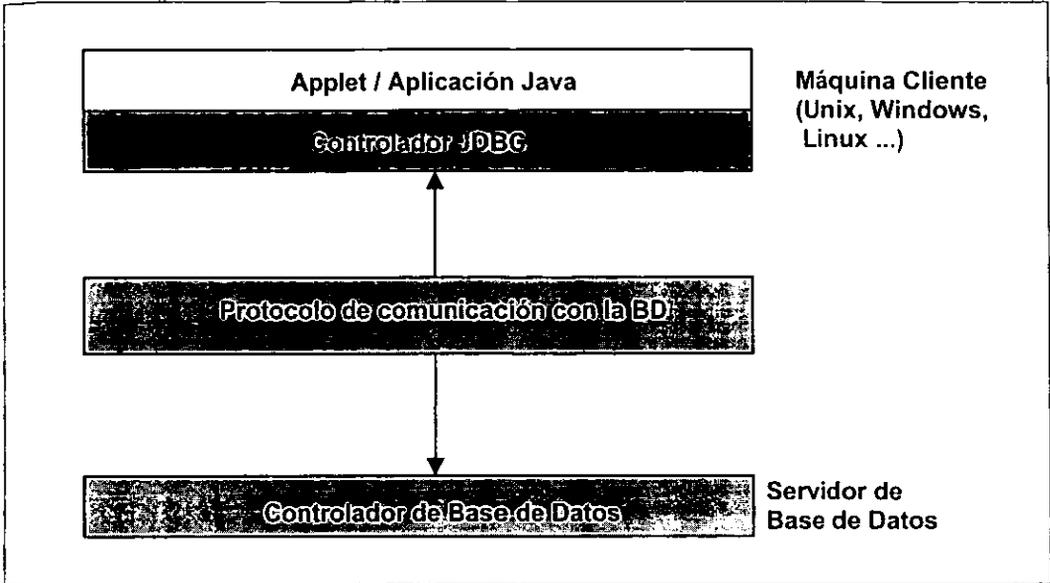


Figura 4.4 Modelo de dos capas.

Para el desarrollo del prototipo C3PoS se utilizará un Administrador de Base de Datos Relacional denominado PostgreSQL y el componente Java (JDBC) ofrecerá Conectividad de Bases de Datos que administrara los servicios que proporcionará el prototipo de Control de Calidad y Cambios a los Productos de Software (productos, solicitudes de cambio, defectos, requisitos).

4.5. Resumen de los elementos tecnológicos que se utilizarán para el desarrollo de un prototipo de Control de Calidad y Cambios a los Productos de Software.

A continuación se resumen los elementos de tecnológicos que se utilizarán para el desarrollo del prototipo Control de Calidad y Cambios a los Productos de Software (C3PoS):

Tecnología	Elemento	Nombre
Redes	Ambiente	Internet
World Wide Web	Identificador de recursos	URL
	Protocolo de red	TCP/IP, http
	Servidor de Web	Apache 1.3.14
	Cliente	Netscape Navigator 6 Internet Explorer 5.5
	Lenguaje de la página Web	HTML
Orientada a Objetos	Lenguaje de Modelado Unificado (UML)	UML para realizar diagramas de: <ul style="list-style-type: none"> • Casos de Uso • Clase • Secuencia • Objetos • Paquetes • Sistema
	Lenguaje de programación	Java, componentes <ul style="list-style-type: none"> - JDK 1.2 (herramienta de desarrollo) - Applets (tipo de programa) - Swing de JFC (desarrollo de la interfaz) [Sun1999] - JDBC (Conectividad a la base de datos)
Base de Datos	Manejador	PostgreSQL 7.1

Capítulo V

5. Análisis y Diseño de un Prototipo de Control de Calidad y Cambios en los Productos de Software.

En este capítulo se presentará el análisis y diseño orientado a objetos de un Prototipo de Control de Calidad y Cambios en los Productos de Software denominado C3PoS.

5.1. Consideraciones generales del prototipo

Con base en lo expuesto en el capítulo II, donde se expone un proceso de producción de software con control de calidad y cambios se propone un prototipo con distintos niveles de control utilizando para el diseño los elementos tecnológicos que se expusieron en el capítulo IV.

El prototipo permitirá configurar el nivel de control de calidad y cambios de productos de software que denominaremos:

Nivel de Control	Descripción
Nivel Simple (NS)	Control personal de producción y calidad con control verbal de cambios a productos de software
Nivel Intermedio (NI)	Control de producción con revisión de calidad y control documentado de cambios a productos de software
Nivel Completo (NC)	Control de producción con doble control de calidad y control de cambios evaluados a productos de software

El Nivel Simple corresponde al análisis que se realizó capítulo II, cuando se propuso la integración del control de producción de calidad (nivel 1) con control verbal de cambios a productos de software (nivel1) en la página 49 y como se muestra en la figura 2.13.

El Nivel Intermedio corresponde a la integración del control de producción con revisión de calidad (nivel 2) con control documentado de cambios a productos de software (nivel 2) que se propuso en el capítulo II (página 54).

El Nivel Completo corresponde al análisis que se realizó capítulo II, cuando se propuso la integración del control de producción con doble control de calidad (nivel 3) con control de cambios evaluados a productos de software (nivel 2) de la página 55 y figura 2.16.

A continuación se presentan las características generales de cada uno de estos niveles:

Nivel Simple. Control personal de producción y calidad y control verbal de cambios a productos de software.

- La calidad del producto depende totalmente del juicio del productor.
- La solicitud del cambio se realiza mediante una comunicación verbal.

En este nivel de control, la calidad depende completamente del Productor y no existe formalmente un control de cambios sobre el producto de software como se muestra en la figura 5.1. Los posibles defectos que el Productor pudiera encontrar no son documentados, ni los cambios que le son solicitados.

Este esquema, aunque simple e informal, resulta recomendable cuando el productor puede controlar todo, donde las solicitudes y productos son simples y de bajo riesgo. Sin embargo depende de la memoria y del buen juicio del programador.

Este nivel de control es recomendado para productos simples y de bajo riesgo y los cambios solicitados son pequeños, ni tan importantes para el contenido general del producto y no impactan a otros.

Nivel Intermedio. Control de producción con revisión de calidad y control documentado de cambios a productos de software

- La calidad del producto depende de una revisión independiente al productor, quien utiliza una especificación documentada de los requisitos.
- La solicitud del cambio es documentada.

En este esquema (figura 5.2), adicionando el rol del Revisor y el control de los requisitos del producto se obtiene un control formal de calidad.

El Revisor podrá reportar al Productor los defectos encontrados al producto el cual corregirá o en su caso los rechazará informando al Revisor.

Este nivel de control es recomendado para toda clase de productos, especialmente para los entregados al cliente. En cuanto a los cambios es recomendado cuando no consumen tiempo y no impactan a otros productos.

Nivel Completo. Control de producción con doble control de calidad y control de cambios evaluados a productos de software

- La calidad del producto depende no solo del cumplimiento de la especificación de los requisitos sino también de la minimización del impacto en otros productos de la línea base.
- Se evalúa el impacto y el costo de la solicitud de cambio.

Este esquema que incluye además el rol del aprobador y el Bibliotecario constituye el esquema ideal del control de los productos de software acorde con al área clave de la Administración de la Configuración del Software del modelo CMM, como se muestra en la figura 5.3.

El Aprobador con base a las listas de verificación y un análisis del impacto sobre otros productos determinará si el producto o la solicitud de cambio es aprobada o rechazada.

Este esquema es recomendado para proyectos que consumen gran tiempo y/o mantenimiento y las solicitudes de cambio pueden consumir tiempo/costo y tener impacto sobre otras líneas base.

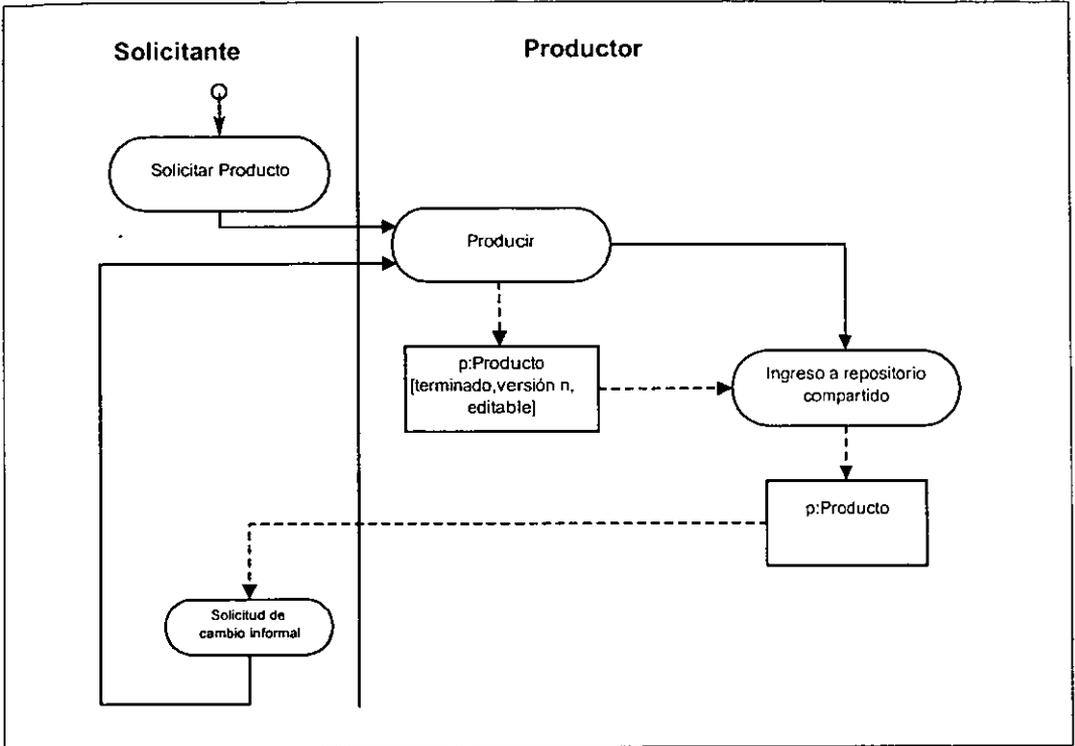


Figura 5.1. Nivel Simple (NS). Control personal de producción y calidad con control verbal de cambios a productos de software.

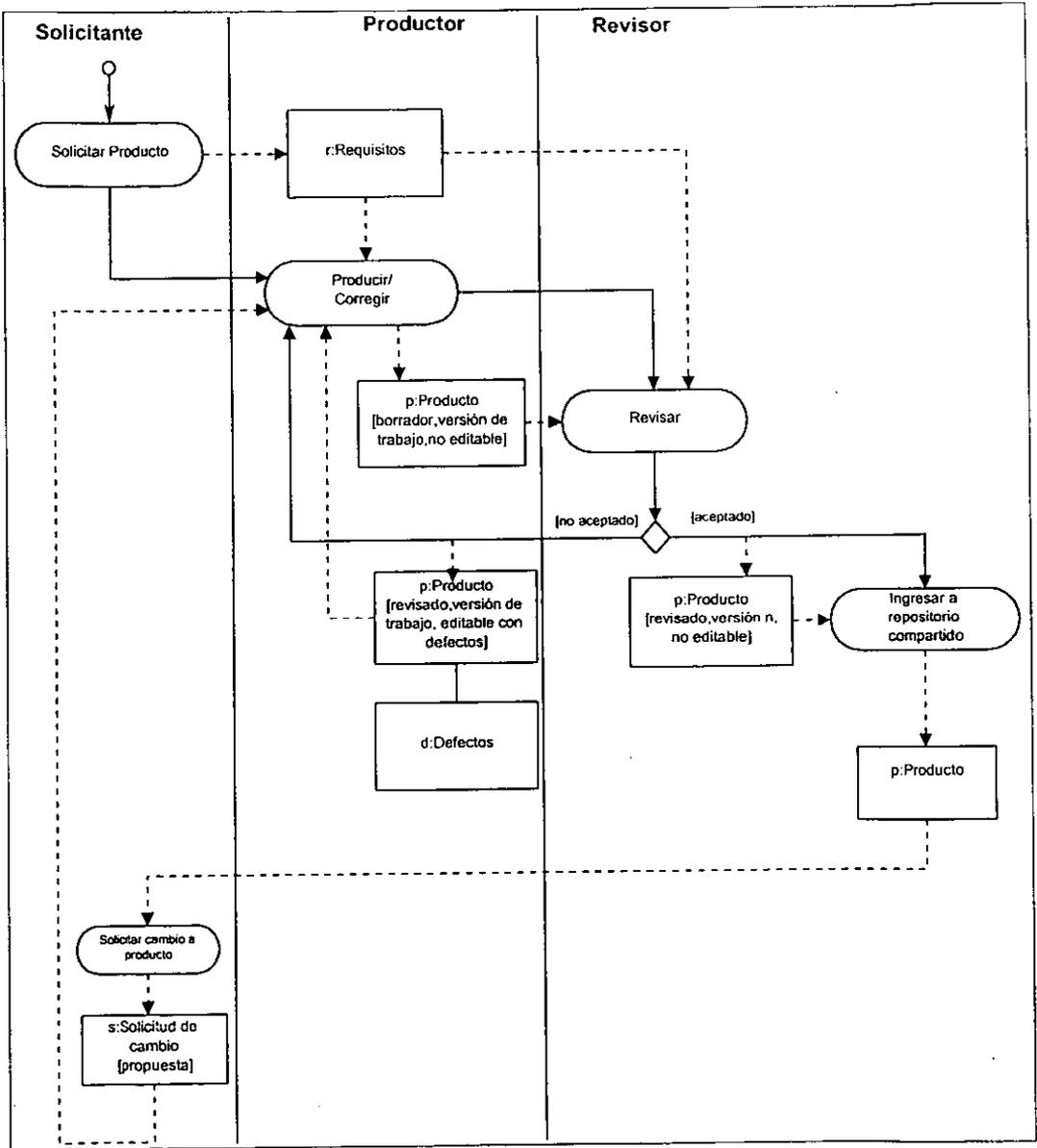


Figura 5.2. Nivel Intermedio (NI). Control de producción con revisión de calidad y control documentado de cambios a productos de software.

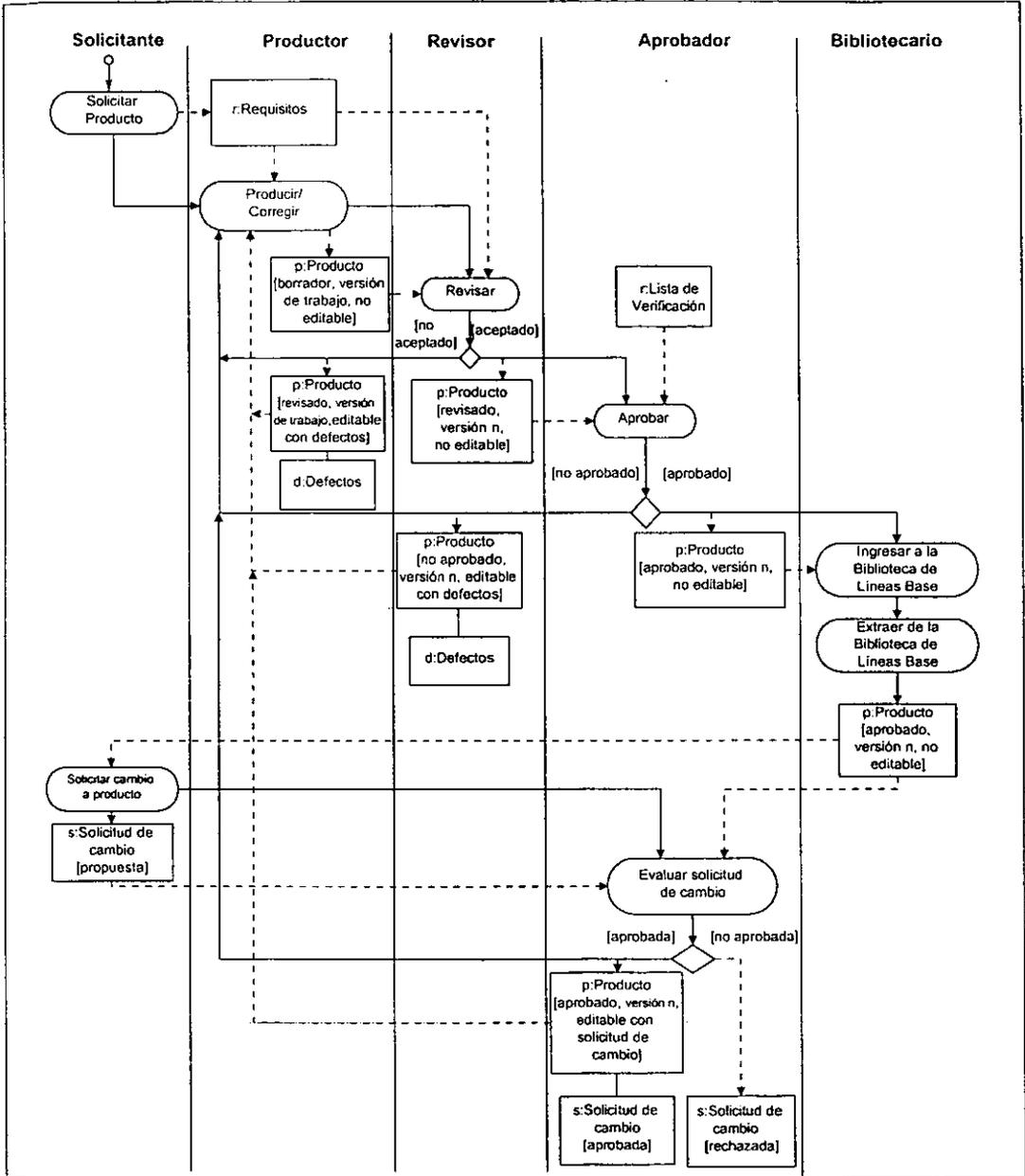


Figura 5.3. Nivel Completo (NC). Control de producción con doble control de calidad y control de cambios evaluados a productos de software.

Se utilizará la siguiente nomenclatura para referirnos a cada uno de los niveles de control:

- NS.** Nivel de Control Simple
- NI.** Nivel de Control Intermedio
- NC.** Nivel de Control Completo

El prototipo controlará los productos de software, las solicitudes de cambio, los requisitos y los defectos encontrados al producto. Las características con las que cumplen estos elementos son:

1. Productos de software.

Num.	Elemento a controlar	Nivel de control		
		NS	NI	NC
1.	Identificación del producto	√	√	√
2.	Título	√	√	√
3.	Versión	√	√	√
4.	Nivel de control	√	√	√
5.	Estado del producto (borrador, en revisión o aprobado)	√	√	√
6.	Identificación del responsable de su producción	√	√	√
7.	Fecha de creación de producto	√	√	√
8.	Fecha planeada de terminación del producto	√	√	√
9.	Fecha real de terminación del producto	√	√	√
10.	Descripción general del contenido del producto	√	√	√
11.	Identificación de los requisitos con los que cumple el producto		√	√
12.	Identificación del que realiza la revisión		√	√
13.	Fecha planeada de revisión		√	√
14.	Fecha real de revisión		√	√
15.	Identificación del responsable de aprobar el producto			√
16.	Fecha planeada de aprobación			√
17.	Fecha real de aprobación			√

√: Indica si el nivel controla este elemento

2. Requisito de un producto.

Los requisitos serán controlados en el Nivel de Control Intermedio (NI) y Completo (NC):

- Identificación del Requisito
- Título del requisito
- Identificación del producto que debe cumplir el requisito
- Estado (solicitado, cumplido).
- Identificación del responsable de la definición del requisito
- Fecha de creación de producto
- Descripción general del requisito

3. Defectos de un producto.

Los defectos, también serán controlados en el Nivel de Control Intermedio (NI) y en el Nivel de Control Completo (NC):

Se documentan los defectos encontrados a un producto de software, cuando es evaluado durante una revisión o aprobación donde se determina si el producto cumple con los requisitos solicitados.

Sus características son:

- Identificación de los defectos
- Título de los defectos
- Identificación del producto al que se le encontraron los defectos
- Estado (reportado, rechazado y corregido).
- Identificación del responsable que encontró los defectos
- Fecha de los defectos encontrados
- Descripción de los defectos encontrados

4. Solicitudes de cambios a los productos de software

Las solicitudes de cambio serán controladas en el Nivel de Control Intermedio (NI) y Completo (NC):

Num.	Elemento a controlar	Nivel de control	
		NI	NC
1.	Identificación de la solicitud de cambio	√	√
2.	Título de la solicitud de cambio	√	√
3.	Identificación del producto al que se le solicita el cambio	√	√
4.	Estado de la solicitud de cambio (aprobada, rechazada, implantada)	√	√
5.	Identificación del responsable de la solicitud de Cambio	√	√
6.	Fecha de la solicitud de cambio	√	√
7.	Descripción de la solicitud de cambio y del resultado de la evaluación e implantación	√	√
8.	Identificación del que aprueba la solicitud de cambio		√
9.	Fecha de aprobación de la solicitud de cambio		√
10.	Identificación del que implanta el cambio		√
11.	Fecha de implantación del cambio		√

√: Indica si el nivel controla este elemento

5.2. Requisitos del prototipo

El prototipo cumple con los siguientes requisitos no funcionales y funcionales:

Requisitos no funcionales:

1. El prototipo permitirá implantarse fácilmente, sin depender de la plataforma en la que fue desarrollado.
2. Permitir el acceso a los productos de software en forma distribuida, de manera que se compartan con diferentes miembros del proyecto en un ambiente de red.
3. Desvincular el formato de los productos de un procesador de textos en específico.
4. Minimizar el uso de pantallas y utilizar términos conocidos en español para facilitar el aprendizaje y permitir una rápida aplicación del prototipo.
5. Contar con un esquema que garantice al personal autorizado tener acceso controlado a los productos de software.

Con el fin de cumplir con estos requisitos, una buena opción es el lenguaje de programación Java, emplear un Servidor Web, así como el formato HTML para los productos de software y el uso de claves de acceso para los usuarios.

Requisitos funcionales

Las funciones que realiza el prototipo son las siguientes:

1. Niveles Configurables de control de productos y solicitudes de cambio, dependiendo de las necesidades del proyecto.
2. Dependiendo del nivel de control se podrá crear, editar y consultar el contenido de los productos, solicitudes de cambio, requisitos y reporte de defectos del producto
3. Notificar a los respectivos responsables mediante el uso del correo electrónico de la producción, solicitud de revisión o aprobación de los productos de software, solicitudes de cambio y defectos encontrados.
4. Cambiar el estado de los productos, solicitudes de cambio, requisitos y defectos encontrados.
5. Permitir que los revisores o aprobadores realicen la evaluación de los productos de software y las solicitudes de cambio. Y determinar si los requisitos fueron cumplidos o los defectos corregidos.
6. Generar nuevas versiones de los productos aprobados.
7. Hacer disponible los productos de software cuando se solicita un cambio.
8. Garantizar el acceso a mas más de un usuario a la vez a la lista de productos, solicitudes de cambio, requisitos y defectos.

Utilizando la caracterización que se obtuvo sobre las herramientas que apoyan el control de cambios a los productos de software que se expuso en el capítulo 3, en la siguiente tabla se incluye el Prototipo C3PoS para obtener una comparación de sus características con esas mismas herramientas.

- A. ConDor. Del M. en C. Carlos Pérez Escobar.
- B. eChange Man de SERENA Software, Inc.
- C. Aimware
- D. PVCS Dimensions de MERANT.
- E. CS-RCS de Component Software.
- F. Concurrent Versions System (CVS) de Cyclic Software.
- G. Prototipo C3PoS

No.	Característica	Producto						
		A	B	C	D	E	F	G
1	Control de versiones	√	√	√	√	√	√	√
2	Registro histórico de las modificaciones al contenido de un producto	X	√	√	√	√	√	X
3	Control de cambios	√	√	√	√	X	X	√
4	Soporte de la mesa de Control de la Configuración del Software	√	√	√	√	X	X	√
5	Reporte de los defectos encontrados al producto durante las verificaciones y validaciones	X	√	√	√	X	X	√
6	Administración de las líneas base	X	√	√	P	P	P	X
7	Definición de los roles asociados a un producto	√	√	√	X	X	X	√
8	Niveles configurables de control para un proyecto	X	√	√	X	X	X	X
9	Niveles configurables de control para un producto de software	X	X	X	X	X	X	√
10	Control de seguridad, mediante el acceso autorizado a la herramienta y a los productos	√	√	√	√	X	X	√
11	Integración con otras herramientas	X	√	√	√	√	√	X
12	Trabajo en un ambiente en red	X	√	√	√	√	√	√

- √ : Cumple con la característica
- X : No cumple con la característica
- P : Cumple parcialmente

5.3. Actores.

Las funciones definidas por los requisitos de prototipo serán llevadas a cabo por el usuario o grupo de usuarios que interactúan con el mismo y que se denominan "actores". Un usuario puede pertenecer a varios grupos dependiendo de la función que realice.

A continuación se describen brevemente las responsabilidades de cada rol con respecto a un producto de trabajo de software y su nivel de control.

- **Administrador**, es el único que puede crear nuevos usuarios del prototipo, establece las características de los niveles de control y estados de los productos. Es el autorizado para eliminar productos. En principio es el responsable de todos los productos.
- **Solicitante**, realiza una solicitud de creación de un producto o la solicitud de cambio al mismo. Un solicitante puede ser el cliente o cualquier miembro del grupo del proyecto.
- **Productor**, dependiendo del nivel de control es el que crea y edita los productos o solicitudes de cambio, requisitos y defectos. Envía a evaluación los documentos estableciendo revisores y aprobadores para los productos y los modifica cuando existen solicitudes de cambio. Indica si los requisitos fueron cumplidos o si los defectos encontrados fueron corregidos o en su caso los rechaza por no encontrar motivo.
- **Revisor**, verifica y/o valida el producto para asegurar la calidad de su contenido y determinar si los requisitos fueron cumplidos. Cambia el estado del producto a revisado o lo rechaza, en cuyo caso regresa el producto al productor con los defectos documentados.

El revisor ingresa el producto a un repositorio compartido en el Nivel de Control Intermedio o lo envía a un Aprobador para su evaluación en el Nivel de Control Completo.

- **Aprobador**, es el que con la lista de verificación evalúa el producto y cambia el estado del producto a aprobado o lo rechaza, en cuyo caso regresa el producto al productor con los defectos documentados.

Si el aprobador determina que el producto es aprobado y cumple con los criterios de entrada a la Biblioteca de la línea base, lo envía al Bibliotecario.

- **Bibliotecario**, organiza las diferentes bibliotecas del proyecto para cada línea base, guarda los productos y los proporciona cuando se solicita un cambio y en su caso actualiza las bibliotecas cuando se aprueba y se realiza el cambio.

Los actores involucrados en el desarrollo de un producto dependerá de su nivel de control:

Actor	Nivel		
	Simple	Intermedio	Completo
Administrador	√	√	√
Solicitante	√	√	√
Productor	√	√	√
Revisor		√	√
Aprobador			√
Bibliotecario			√

5.4. Análisis.

Utilizando el Modelado Orientado a Objetos y dadas las características del modelo de tres niveles de control de calidad y cambios se planteó un análisis incremental.

Se comenzó con la definición de las clases para Nivel de Control Simple (NS) y conforme avanzamos de nivel se utilizó una jerarquía de clases que permitió reutilizar algunas clases del nivel anterior y se definieron otras con nueva funcionalidad hasta llegar al Nivel de Control Completo (NC).

Además se realizó una agrupación de clases en diferentes paquetes conforme se cambió de Nivel de Control.

Durante todo el diseño se agruparon las clases siguiendo el modelo de tres capas que consiste en:

1. **Dominio del problema (DP).** Esta formado por el grupo de clases que son significativas al prototipo.
2. **Interfaz Humana (IH).** Formado a partir de las clases del dominio del problema y son con las que el usuario interactúa con el prototipo por medio de Ventanas.
3. **Manejo de Datos (MD).** Se obtiene de cada una de las clases del dominio del problema y que se desea sean persistentes.

5.4.1. Nivel de Control Simple (NS)

5.4.1.1. Clases Principales del Nivel de Control Simple (NS)

Clases principales que cumplen con los requisitos para controlar un producto de software con Nivel de Control Simple:

Componente	Clases
Interfaz Humana C3PoS.NS-IH	LogonWindow C3PoSWindow EditProductWindow DefineProductWindow ProductWindow UserConfigWindow OptionConfigWindow ErrorWindow
Dominio del Problema C3PoS.NS-DP	Product User ControlLevel StateProduct ProductRequestSave ProductManager
Manejo de Datos C3PoS.NS-MD	ProductDB UserDB ControlLevelDB StateProductDB ProductServerSave

Los diagramas de clase están en el anexo B de las páginas B-11 a la B-13.

LogonWindow, es la ventana para escribir la clave de usuario y su clave secreta que permitirá acceder al prototipo.

C3PoSWindow, es la ventana principal mediante la cual se tendrá acceso a las principales funciones del prototipo. Se muestra si el usuario introdujo su clave de acceso y clave secreta correctos.

EditProductWindow, es la Ventana para editar el contenido de un producto. Se muestra cuando el usuario selecciona crear o editar un producto.

DefineProductWindow, es la Ventana para definir las características de control del Producto. Se muestra al llenar datos de control del producto cuando se crea o se edita.

ProductWindow, es la ventana para editar el contenido de un producto. Se muestra cuando el usuario selecciona mostrar un producto.

UserConfigWindow, es la ventana para editar las características de un usuario.

OptionConfigWindow, es la ventana para establecer niveles de control y estados del producto

ErrorWindow, para mostrar los mensajes del prototipo para que el usuario atienda un mensaje o confirme una acción.

Product, es el producto con todos los elementos que lo definen.

User, define la información de los usuarios que tienen acceso al sistema.

ControlLevel, contiene información de los niveles en los que se puede definir un producto.

StateProduct, los estados que puede contener un producto o una solicitud de cambio.

ProductRequestSave, la clase que controla el almacenamiento del producto.

ProductManager, administra los productos del sistema, los estados, el nivel de control, los usuarios con acceso al sistema.

ProductDB, Contiene información que se necesita para guardar cada producto.

UserDB, Contiene información de los Usuarios.

ControlLevelDB, Contiene información de los niveles en los que se puede definir un producto.

StateProductDB, Contiene información de los estados de los Productos.

ProductServerSave, la clase que controla el almacenamiento del producto en el repositorio compartido.

C3PoS.NS-IH, es el componente que agrupa las clases de la Interfaz Humana en este nivel

C3PoS.NS-DP, es el componente que agrupa las clases del Dominio del problema en este nivel.

C3PoS.NS-MD, es el componente que agrupa las clases del Manejo de Datos en este nivel.

5.4.1.2. Casos de Uso del Nivel de Control Simple (NS)

La parte dinámica del análisis es definida mediante los diferentes casos de uso que se dan en el Nivel de Control Simple y son:

Num.	Caso de Uso	Descripción
1.	Entrar al sistema	Validar la clave de acceso y la clave secreta del usuario
2.	Crear Producto	El usuario seleccionará el nivel de control e introducirá datos de control del producto así como el contenido del mismo.
3.	Llenar datos de control del producto	El Productor introducirá datos que definen al producto o los actualiza.
4.	Editar Producto	El Productor podrá editar el contenido de un Producto seleccionado, si es editable, y guardarlo.
5.	Mostrar Producto	El usuario podrá leer el contenido de un Producto seleccionado.
6.	Seleccionar Producto	El usuario podrá seleccionar el producto de una lista que se muestra de acuerdo al nivel de control.
7.	Modificar un Producto	El usuario selecciona un producto y hará una copia del producto con estado de borrador con nueva versión y será editable.
8.	Crear Usuario	El Administrador podrá dar de alta, baja o modificar los datos de un usuario.
9.	Crear Nivel de Control y Estados	El Administrador podrá dar de alta, baja o modificar información de los Niveles de Control y Estados para los productos.

Los diagramas de casos de uso se describen en el anexo B de las páginas B-2 a la B-10

5.4.1.3. Diagrama de Estado del Nivel de Control Simple (NS)

Un diagrama de estado que indica los diferentes estados que pasa un producto debido a su comportamiento dinámico provocado por un estímulo.

En el Nivel de Control Simple (NS) un producto al ser creado está en estado de borrador y editable en una versión inicial.

Se podrá editar un producto mientras el Productor no determine que el producto esta terminado.

Una vez que el Productor termine de editar el producto, lo ingresará en el repositorio compartido y cambiará su estado a terminado.

Si el productor atiende una solicitud de cambio verbal a un producto, creará una nueva versión para editarlo y cambia su estado a borrador.

El diagrama de estados correspondiente a este nivel de control se muestra en el Anexo B (página B-14).

5.4.1.4. Diccionario de Clases del Nivel de Control Simple (NS).

Se proporciona en el Anexo B (B-15 a B-33) el prototipo de interfaz y el diccionario de clases que lista los atributos y métodos que deben contar las clases para realizar la funcionalidad del prototipo en este nivel de control.

5.4.2 Nivel de Control Intermedio (NI)

5.4.2.1. Clases Principales del Nivel de Control Intermedio (NI).

Este nivel incluye clases del Nivel de Control Simple y se definen nuevas clases debido a que en este nivel de control se adicionan los casos de uso que implican la revisión del producto por parte de un Revisor.

Mediante una jerarquía de herencia de clases reutilizaremos clases del nivel de control simple por lo cual solo se agregarán los atributos y métodos necesarios o se modifica el comportamiento de algunos métodos.

Además, se agregan otras clases debido a que en este nivel de control se pretende controlar los requisitos del producto, los defectos encontrados al mismo debido a una revisión y además los cambios solicitados a un producto.

En el anexo C (C-23 a C-25) se proporcionan los diagramas de clase.

Componente	Clase Principal	Observación
Interfaz Humana C3PoS.NI-IH	C3PoSWindowNI EditProductWindowNI DefineProductWindowNI DefineRequerimentWindow DefineDefectWindow DefineChangeWindow EvaluateWindow MailWindow	Hereda de C3PoSWindow Hereda de EditProductWindowNI Hereda de DefineProductWindow Nueva clase Nueva clase Nueva clase Nueva clase Nueva clase
	ProductWindow LogonWindow UserConfigWindow OptionConfigWindow ErrorWindow	(El grupo de clases definidas en el Nivel de Control Simple (NS) y que no se modifican)
Dominio del Problema C3PoS.NI-DP	ProductManagerNI ProductNI Requeriment Defect ChangeRequest MailRequest	Hereda de ProductManager Hereda de Producto Nueva clase Nueva clase Nueva clase Nueva clase
	User ControlLevel StateProduct ProductRequerimentSave	(El grupo de clases definidas en el Nivel de Control Simple y que no se modifican)
Manejo de Datos C3PoS.NI-MD	ProductNI DB RequerimentDB DefectDB ChangeRequestDB Mail	Hereda de ProductDB Nueva clase Nueva clase Nueva clase Nueva clase
	UserDB ControlLevelDB StateProductDB ProductServerSave	(El grupo de clases definidas en el Nivel de Control Simple y que no se modifican)

C3PoSWindowNI, es la ventana principal que hereda de C3PoSWindow mediante la cual se tendrá acceso a las funciones del producto, requisitos, defectos, solicitudes de cambio y configuración, de este nivel de control. Se muestra si el usuario introdujo su clave de acceso y clave secreta correctos.

EditProductWindowNI, es la ventana para editar el contenido de un producto. Se muestra cuando el usuario selecciona crear o editar un producto con nivel de control Intermedio.

DefineProductWindowNI, es la ventana para definir las características de control del Producto. Se muestra al llenar datos de control del producto cuando se crea o se edita, incluye información del revisor.

DefineRequerimentWindow, ventana para definir los requisitos de un producto.

DefineDefectWindow, ventana para definir los defectos encontrados a un producto.

DefineChangeWindow, ventana para definir las características de la solicitud de cambio.

EvaluateWindow, ventana para definir la aprobación o el rechazo de un producto después de un proceso de evaluación.

MailWindow, ventana para definir las características del mensaje por correo electrónico.

ProductManagerNI, Administra el producto del sistema que en este nivel incluye las características para el proceso de revisión.

ProductNI, es el producto que incluye información para el proceso de revisión.

Requeriment, requisitos que debe cumplir un producto de software y que será necesario conocer el revisor para la evaluación del producto.

Defect, defectos encontrados a un producto con todos los elementos que lo definen.

ChangeRequest, solicitud de cambio al producto con todas los elementos que la definen.

MailRequest, solicitud de envío de mensaje por correo electrónico.

ProductNIDB, Contiene el registro del producto que incluye información para el proceso de revisión.

RequerimentDB. Contiene el registro de los requisitos del producto.

DefectDB. Contiene el registro de los defectos encontrados a un producto.

ChangeRequestDB. Contiene el registro de las Solicitudes de Cambio.

Mail, envío de mensaje por correo electrónico.

C3PoS.NI-IH, es el componente que agrupa las clases de la Interfaz Humana en este nivel

C3PoS.NI-DP, es el componente que agrupa las clases del Dominio del problema en este nivel.

C3PoS.NI-MD, es el componente que agrupa las clases del Manejo de Datos en este nivel.

5.4.2.2. Casos de Uso del Nivel de Control Intermedio (NI).

Incluye los casos de uso del Nivel de Control Simple, excepto los que se modifican por el hecho de existir un Revisor y se necesita identificar datos de control del mismo.

Se adicionan otros casos de uso debido a que en este nivel se requiere controlar los requisitos y los posibles defectos encontrados al productos cuya función realiza el Revisor y además existe la solicitud de cambio.

Asumiremos que se utilizan los mismos casos del Nivel de Control Básico (NS) y solo mencionaremos los que se modifican y los que se adicionan.

En este nivel de control se incluye el caso de evaluar el producto y enviar un mensaje por correo electrónico para notificar a los demás usuarios involucrados

En el anexo C (C-2 a C-22) se proporciona la descripción de los diagramas de casos de uso.

V. Análisis y Diseño de un Prototipo de Control de Calidad y Cambios en los Productos de Software

Num.	Caso de Uso	Descripción	Observación
10.	Llenar datos de control del producto	Actualizar datos de control del producto	Se modifica
11.	Editar Producto	Modificar el contenido del producto	Se modifica
12.	Enviar a evaluación un producto	Seleccionar un producto, cambiar el estado del producto a no editable.	Se adiciona
13.	Revisar producto	Seleccionar un producto, cambia el estado del producto y enviarlo a un Aprobador para su evaluación, el usuario deberá ser el Revisor.	Se adiciona
14.	Rechazar producto	Seleccionar un producto y modificar datos de control así como reportar defectos encontrados si los hubiera.	Se adiciona
15.	Crear requisito	Introducir datos de control del requisito así como la descripción del cambio	Se adiciona
16.	Llenar datos de control del requisito	Introducir datos de control que definen al requisito	Se adiciona
17.	Editar requisito	Editar datos de control del requisito	Se adiciona
18.	Seleccionar requisito	Seleccionar el requisito asociado a un producto	Se adiciona
19.	Mostrar requisito	Mostrar las características del requisito	Se adiciona
20.	Crear reporte de defectos	Introducir datos de control y descripción de los defectos encontrados a un producto cuando ha sido evaluado	Se adiciona
21.	Llenar datos de control del defecto	Introducir datos de control del defecto	Se adiciona
22.	Editar defecto	Editar datos de control del reporte de defecto	Se adiciona
23.	Seleccionar reporte de defectos	Seleccionar un reporte de defectos asociados a un producto	Se adiciona
24.	Mostrar Reporte de Defectos	Mostrar las características de un reporte de defectos	Se adiciona
25.	Crear solicitud de cambio	Introducir datos de control y descripción de la solicitud de cambio a un producto	Se adiciona
26.	Llenar datos de control de la solicitud de cambio	Introducir datos de control de la solicitud de cambio	Se adiciona
27.	Editar solicitud de cambio	Editar datos de control de la solicitud de cambio	Se adiciona
28.	Seleccionar solicitud de cambio	Seleccionar una solicitud de cambio	Se adiciona
29.	Mostrar solicitud de cambio	Mostrar las características de una solicitud de cambio	Se adiciona
30.	Enviar mensaje	Enviar un mensaje a personas involucradas	Se adiciona

5.4.2.3. Diagramas de Estado del Nivel de Control Intermedio (NI).

En este nivel de control los objetos con un comportamiento dinámico se obtiene de las clases de ProductNI, Requirement, defect y ChangeRequest que corresponden al producto, al requisito, el defecto y solicitud de cambio.

Los diagramas de estado correspondiente a este nivel de control se muestra en el Anexo C (C-26 a C29).

Diagrama de estado del producto.

En el Nivel de Control Intermedio (NI) un producto al ser creado está en estado de borrador y editable en una versión inicial.

Se podrá editar un producto mientras el Productor no lo envíe a revisión.

Si el productor envía a un Revisor el producto para su evaluación y si es aceptado se genera una nueva versión aprobada y el Revisor lo ingresa al repositorio compartido.

Si el producto que se está revisando es rechazado se envía al Productor y pasa nuevamente al estado de borrador con una nueva versión.

Diagrama de estado del requisito.

Un requisito es creado por un Productor con el estado de Solicitado y editable

Una vez que el Productor termine de editar el producto y si cumple con el requisito cambiará el estado del requisito a Cumplido.

Diagrama de estado del reporte de defecto.

Un Revisor crea un reporte de defecto en estado de Reportado y editable y lo envía a un Productor.

El productor podrá aceptar el defecto encontrado al producto y cambiará su estado a aceptado y no editable, una vez corregido el defecto cambiará su estado a Corregido.

El productor podrá rechazar el defecto encontrado al producto y cambiará su estado a Rechazado y no editable.

Diagrama de estado de la solicitud de cambio.

Un solicitante puede crear una solicitud de cambio en estado de borrador y editable y enviársela al Productor.

El Productor podrá rechazar la solicitud de cambio y en este caso cambia su estado a Rechazado.

El Productor podrá aprobar la solicitud de cambio y en este caso cambia su estado a Aprobado y una vez atendida cambia su estado a Procesado.

5.4.2.4. Diccionario de Clases del Nivel de Control Intermedio (NI).

Se proporciona en el Anexo C (C30 a C49) el prototipo interfaz y el diccionario de clases que lista los atributos y métodos que deben contar las clases para realizar la funcionalidad del prototipo en este nivel de control.

5.4.3 Nivel de Control Completo (NC)

5.4.3.1. Clases Principales del Nivel de Control Completo (NC).

Incluye las clases del Nivel de Control Intermedio, excepto las que se modifican porque en este nivel existe un Aprobador y un Bibliotecario y se necesitan datos de control del Aprobador.

También se modifican algunas clases que se relacionan con la solicitud de cambio debido a que un Aprobador la evalúa y reporta defectos encontrados al producto de software.

En la siguiente tabla se muestran las clases que intervendrían en este nivel de control.

En el apéndice D (D9 a D11) se describen los diagramas de clase.

Componente	Clase Principal	Observación
Interfaz Humana C3PoS.NC-IH	DefineProductWindowNC DefineChangueWindowNC	Hereda de DefineProductWindowNI Hereda de DefineChangueWindow
	C3PoSWindowNI EditProductWindowNI DefineRequerimentWindow DefineDefectWindow MailWindow ProductWindow LogonWindow UserConfigWindow OptionConfigWindow ErrorWindow	(El grupo de clases definidas en el Nivel de control Intermedio (NI) y en el Nivel de Control Simple y que no se modifican)
Dominio del Problema C3PoS.NC-DP	ProductManagerNC ProductNC ChangeRequestNC	Hereda de ProductManagerNI Hereda de ProductNI Hereda de ChangeRequest
	Requeriment Defect MailRequest User ControlLevel StateProduct ProductRequerimentSave	(El grupo de clases definidas en el Nivel de control Intermedio (NI) y en el Nivel de Control Simple y que no se modifican)
Manejo de Datos C3PoS.NC-MD	ProductNCDB ChangueRequestNCDB	Hereda de ProductNIDB Hereda de ChangueRequestDB
	RequerimentDB DefectDB Mail UserDB ControlLevelDB StateProductDB ProductServerSave	(El grupo de clases definidas en el Nivel de control Intermedio (NI) y en el Nivel de Control Simple y que no se modifican)

DefineProductWindowNC, es la ventana para definir las características de control del Producto. Se muestra al llenar datos de control del producto cuando se crea o se edita, incluye información del aprobador y de la aprobación del producto.

DefineChangueWindowNC, ventana para definir las características de la solicitud de cambio, incluye información del que aprobará la solicitud de cambio.

ProductManagerNC, administra el producto del sistema que en este nivel incluye las características para el proceso de aprobación.

ProductNC, es el producto que incluye información para el proceso de revisión.

ChangeRequestNC, solicitud de cambio que incluye información del que la aprobará.

ProductNCDB, contiene el registro del producto que incluye información para el proceso de aprobación.

ChangeRequestNCDB, contiene el registro de la solicitud de cambio que incluye información para su aprobación.

C3PoS.NC-IH, es el componente que agrupa las clases de la Interfaz Humana en este nivel

C3PoS.NC-DP, es el componente que agrupa las clases del Dominio del problema en este nivel.

C3PoS.NC-MD, es el componente que agrupa las clases del Manejo de Datos en este nivel.

5.4.3.2. Casos de Uso del Nivel de Control Completo (NC).

Incluye los casos de uso del Nivel de Control Intermedio, excepto los que se modifican por el hecho de existir un Aprobador y porque existe la necesidad de identificar datos de control del mismo.

Se modifican algunos casos de uso que se relacionan con la solicitud de cambio debido a que en este nivel el Aprobador es el que la evalúa.

Asumiremos que se utilizan los mismos casos de uso del Nivel de Control Intermedio (NI) y solo mencionaremos los que se modifican y los que se agregan:

En el anexo D (D2 a D8) se describen los diagramas de casos de uso.

Num.	Caso de Uso	Descripción	Observación
31.	Llenar datos de control del producto	Actualizar datos de control del producto	Se modifica
32.	Revisar un Producto	El Revisor selecciona un producto, cambia el estado del producto y lo envía a un Aprobador para su evaluación	Se modifica
34.	Aprobar un Producto	El Aprobador selecciona un producto, cambia el estado del producto y notifica su evaluación ya sea de aprobación o de los defectos encontrados al producto.	Se agrega
35.	Crear Solicitud de cambio	Introducir datos de control y descripción de la solicitud de cambio a un producto	Se modifica
36.	Llenar datos de control de la solicitud de cambio	Introducir datos de control de la solicitud de cambio	Se modifica
37.	Editar solicitud de cambio	Editar datos de control de la solicitud de cambio	Se modifica

5.4.3.3. Diagrama de Estado del Nivel de Control Completo (NC).

Los diagramas de estado para el producto y la solicitud de cambio se modifican porque tienen que ser evaluados por un Aprobador.

El diagrama de estados correspondiente a este nivel de control se muestra en el Anexo D (D12 a D13).

Diagrama de estado del producto.

En el Nivel de Control Completo (NC) un producto al ser creado está en estado de borrador y editable en una versión inicial.

Se podrá editar un producto mientras el Productor no lo envíe a revisión.

Si el Productor envía a un Revisor el producto para su evaluación y si es aceptado se genera una nueva versión aprobada, el Revisor cambia el estado del producto a revisado y lo envía a un Aprobador para su evaluación.

Si el Revisor rechaza el producto lo envía al Productor y se mantiene en estado de borrador con una nueva versión.

Si el Aprobador al evaluar el producto lo aprueba lo enviará a un Bibliotecario para su ingreso a la Biblioteca y cambia su estado a aprobado.

Si el Aprobador rechaza el producto lo envía al Productor y pasa nuevamente al estado de borrador con una nueva versión.

Diagrama de estado de la solicitud de cambio.

Un solicitante puede crear una solicitud de cambio en estado de borrador y editable y enviársela al Aprobador.

El Aprobador podrá rechazar la solicitud de cambio y en este caso cambia su estado a Rechazado.

Si el Productor aprueba la solicitud de cambio y en este caso cambia su estado a Aprobado y una vez atendida cambia su estado a Procesado.

5.4.3.4. Diccionario de Clases del Nivel de Control Completo (NC).

Se proporciona en el Anexo D (D14 a D20) el prototipo interfaz y el diccionario de clases que lista los atributos y métodos que deben contar las clases para realizar la funcionalidad del prototipo en este nivel de control.

5.5. Diseño

Durante esta fase se refinan los modelos que se obtuvieron durante la fase de análisis y se adaptan los modelos obtenidos de los niveles de control al ambiente de implementación.

Características del ambiente de implementación:

- Utilizar el lenguaje de programación Java, las clases que forman los diferentes paquetes de JDK 1.2 o superior y las clases de Swing que son una extensión del AWT para crear interfaces.
- El prototipo funcionaría como un applet insertado en una página html que corre desde un servidor Web, en Internet, utilizando el protocolo http
- El prototipo funciona recibiendo órdenes del usuario a través de menús y el uso de paneles y ventanas para visualizar las respuestas recibidas del sistema
- Se utilizarían las clases de JDBC y el manejador de base de datos PostgreSQL v7.1.3 para el manejo de los datos persistentes.

5.5.1. Clases resultantes del diseño.

Al realizar el refinamiento de las clases obtenidas en la fase del análisis y mediante el uso de la herencia se crearon nuevas clases:

- **DefineAbstractWindow**, clase que se crea en el grupo de la Interfaz Humana de la cual heredarán:
 - DefineRequeriment
 - DefineDefect
 - DefineChangeRequest

Con el fin de generalizar el funcionamiento de estas clases.

- **DataBase**, clase que se crea en el grupo del Manejo de Datos que se encarga de la conexión y de los detalles del manejo de la base de datos en general.
- **Grupo de clases Ejecutantes (Command)**, se relacionan con cada uno de los subsistemas de la arquitectura física y heredan de la clase AbstractCommand. Este nuevo grupo se crea para separar la funcionalidad entre los grupos de manera que sólo las clases ejecutantes conozcan la forma en que interactúan las clases entre sí cuando se realice una acción con el fin de lograr una mayor independencia, facilitar el mantenimiento y minimizar las interfaces

Estas clases ejecutantes tiene todo el control del sistema y las otras clases son pasivas por la razón de que sólo pueden hacer lo que les corresponde y si se requiere interactuar con otras clases necesitan invocar el método correspondiente en la clase ejecutante.

El grupo de clases Ejecutantes (Command) se agruparán en el componente **C3PoS.Command**:

Componente	Clase Principal
Ejecutantes C3PoS.Command	AbstractCommand C3PoSCommand EditCommand

Una descripción detallada de estas clases se encuentran en el Anexo E (E9 a E11).

5.5.2. Resumen de componentes del prototipo

En resumen, de acuerdo a la agrupación de las clases en componentes para cada uno de los niveles de control definidos en la fase de análisis tenemos los siguientes tipos:

Tipo de Componente	Nombre del Componente	Paquetes de Java que se requieren
Interfaz Humana	C3PoS.NS-IH C3PoS.NI- IH C3PoS.NC- IH	java.awt.* java.util.* java.awt.event.* javax.swing.*
Dominio del Problema	C3PoS.NS-DP C3PoS.NI-DP C3PoS.NC-DP	java.util.*
Manejo de datos	C3PoS.NS-MD C3PoS.NI-MD C3PoS.NC-MD	java.sql.* java.net.* java.util.*
Ejecutantes	C3PoS.Command	java.util.*

En el anexo E (E2 a E8) se describen estas clases.

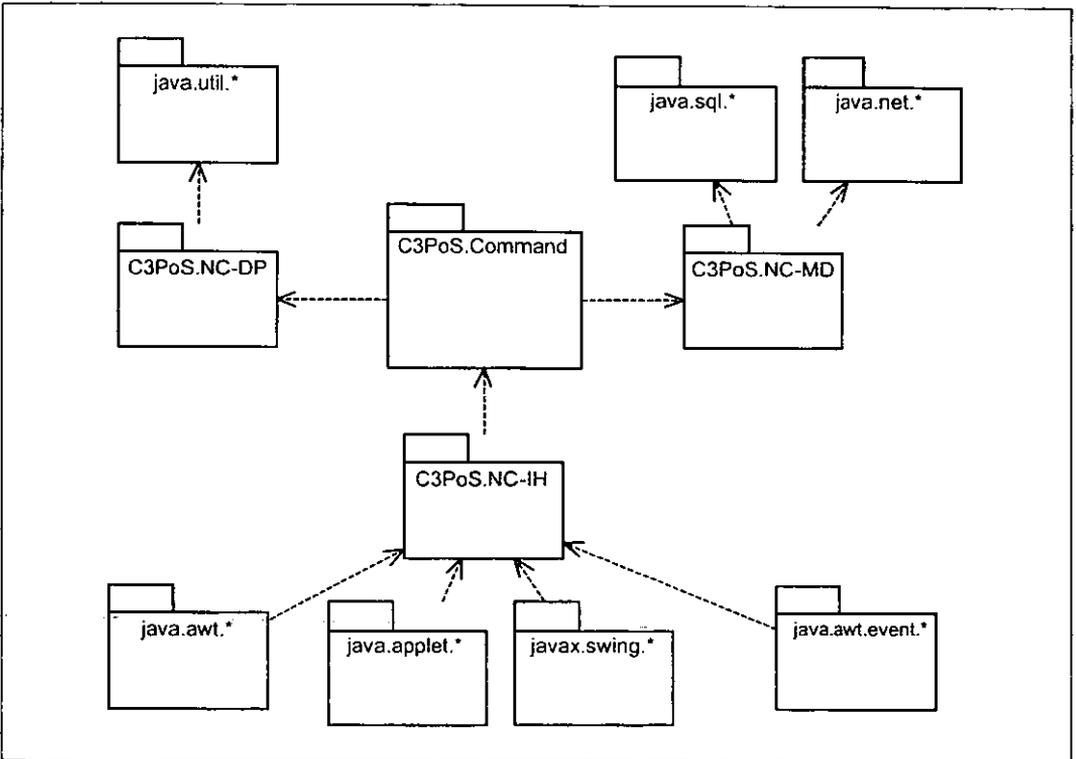
5.5.3. Arquitectura del Prototipo.

La arquitectura del sistema se describe mediante:

- **Arquitectura lógica.** Muestra la agrupación de las clases en categorías
- **Arquitectura física.** Muestra el agrupamiento de los módulos en subsistemas

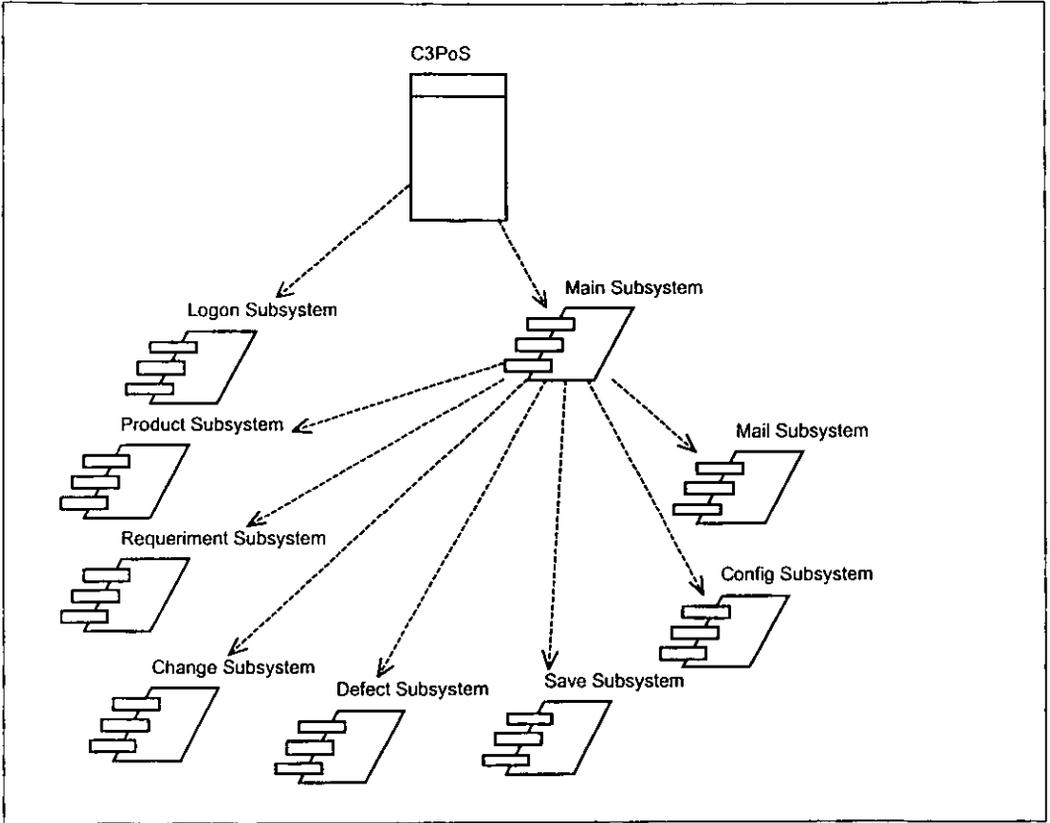
5.5.3.1. Arquitectura lógica.

La arquitectura lógica del sistema estaría formada por los componentes que integran el Dominio del Problema, Interfaz Humana y Manejo de Datos, y el grupo de clases que integran el componente Ejecutante (Command) así como los paquetes Java que se requieren.



5.5.2.2. Arquitectura física.

La arquitectura física comprende el agrupamiento de los módulos en subsistemas y esta compuesto por:



- **Logon**, verifica el derecho de acceso de los usuarios que desean ingresar al sistema.
- **Main**, realiza todas las funciones correspondientes al control de los productos. Este subsistema a su vez esta dividido en otros subsistemas:

Subsistema	Descripción
Product	Control del producto, donde se crea y editan, se muestra su contenido y se cambia su estado
Requeriment	Control de requisitos, donde se crea y editan los requisitos con los que debe cumplir un producto, se muestra su contenido y se cambia su estado.
Defect	Control de defectos, donde se crea y editan los defectos encontrados a un producto después de un proceso de evaluación se muestra su contenido y se cambia su estado.
Change	Control de cambios, donde se crea y editan las solicitudes de cambio a los productos, se muestra su contenido y se cambia su estado
Config	Configuración del sistema, donde se establecen las opciones del sistema y los usuarios con acceso al sistema
Save	Guardar el producto el contenido del producto o consultarlo
Mail	Para el envío de mensajes por correo electrónico

5.5.2.3. Diseño de la Base de Datos.

Para la creación de la base de datos se consideran las clases que son persistentes, éstas pertenecen al grupo de Manejo de Datos.

Se utilizó un esquema de base de datos relacional. Para cada una de las clases del Manejo de Datos se creó una relación, obteniéndose las relaciones:

Nivel de Control	Clase del Manejo de Datos	Relación	Descripción
Simple	ProductDB	PRODUCT	Productos
	UserDB	USER	Usuarios
	ControlLevelDB	LEVELS	Niveles de Control
	StatesDB	STATES	Estados
Intermedio	ProductNIDB	PRODUCTNI	Productos
	RequerimentDB	REQUERIMENT	Requisitos
	DefectDB	DEFECT	Defectos
	ChangesDB	CHANGES	Solicitudes de Cambio
Completo	ProductNCDB	PRODUCTNC	Productos
	ChangesNCDB	CHANGUESNC	Solicitudes de cambio

La tabla muestra el total de relaciones obtenidas y que conformarán la Base de Datos.

Con el fin de dar mayor claridad en el Diagrama Entidad-Relación de la Base de Datos obtenida (C3PoS) se dividió en los niveles de control como se muestra a continuación

Para un mayor detalle sobre las relaciones que conforman la Base de Datos, se proporciona más información en el Anexo E (E5 a E8).

Diagrama Entidad-Relación para el Nivel de Control Simple

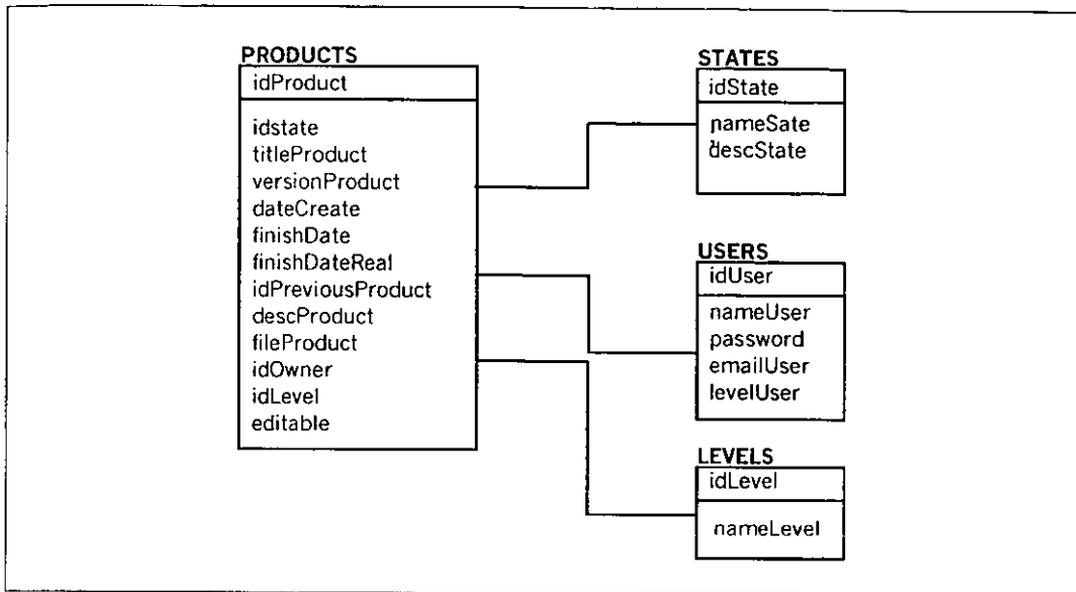


Diagrama Entidad-Relación para el Nivel de Control Intermedio

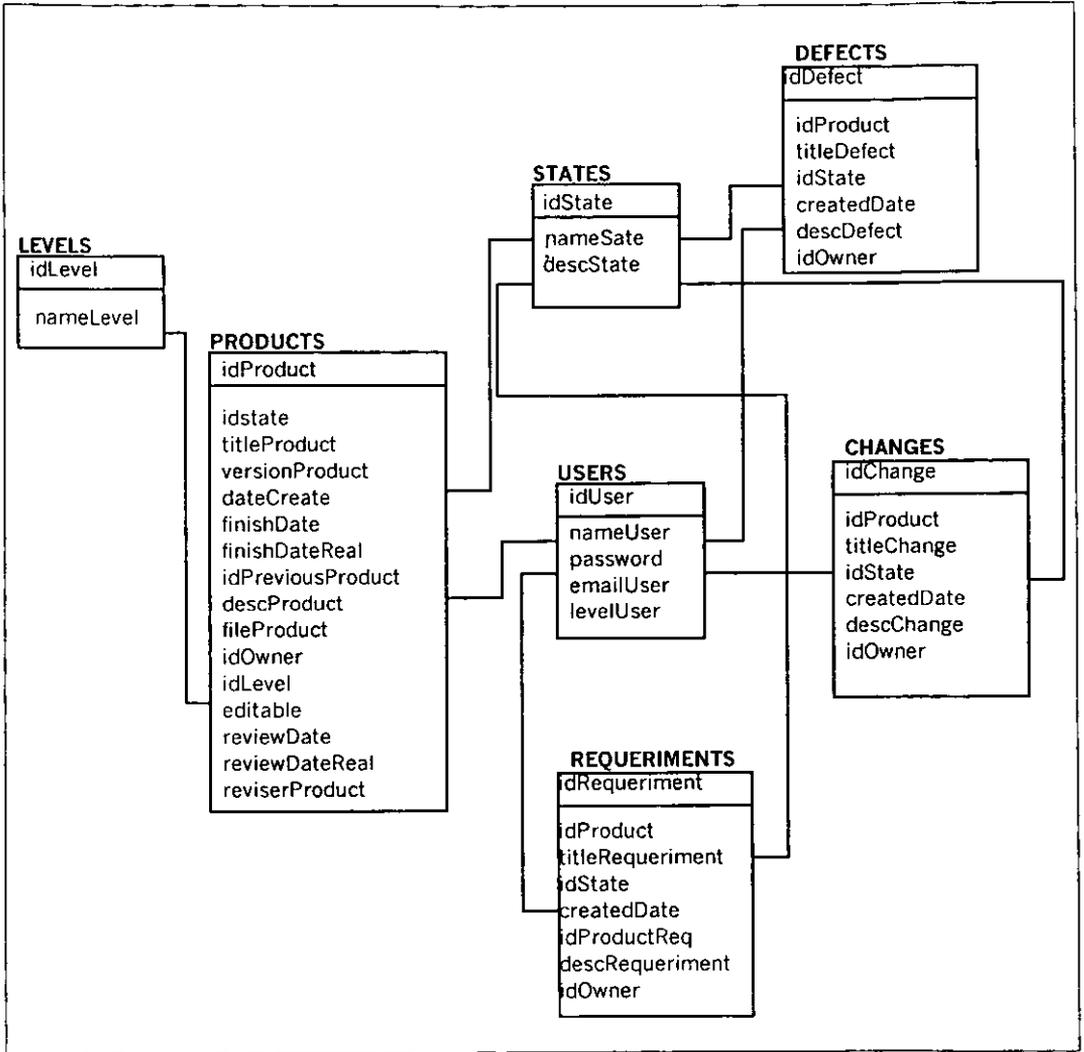
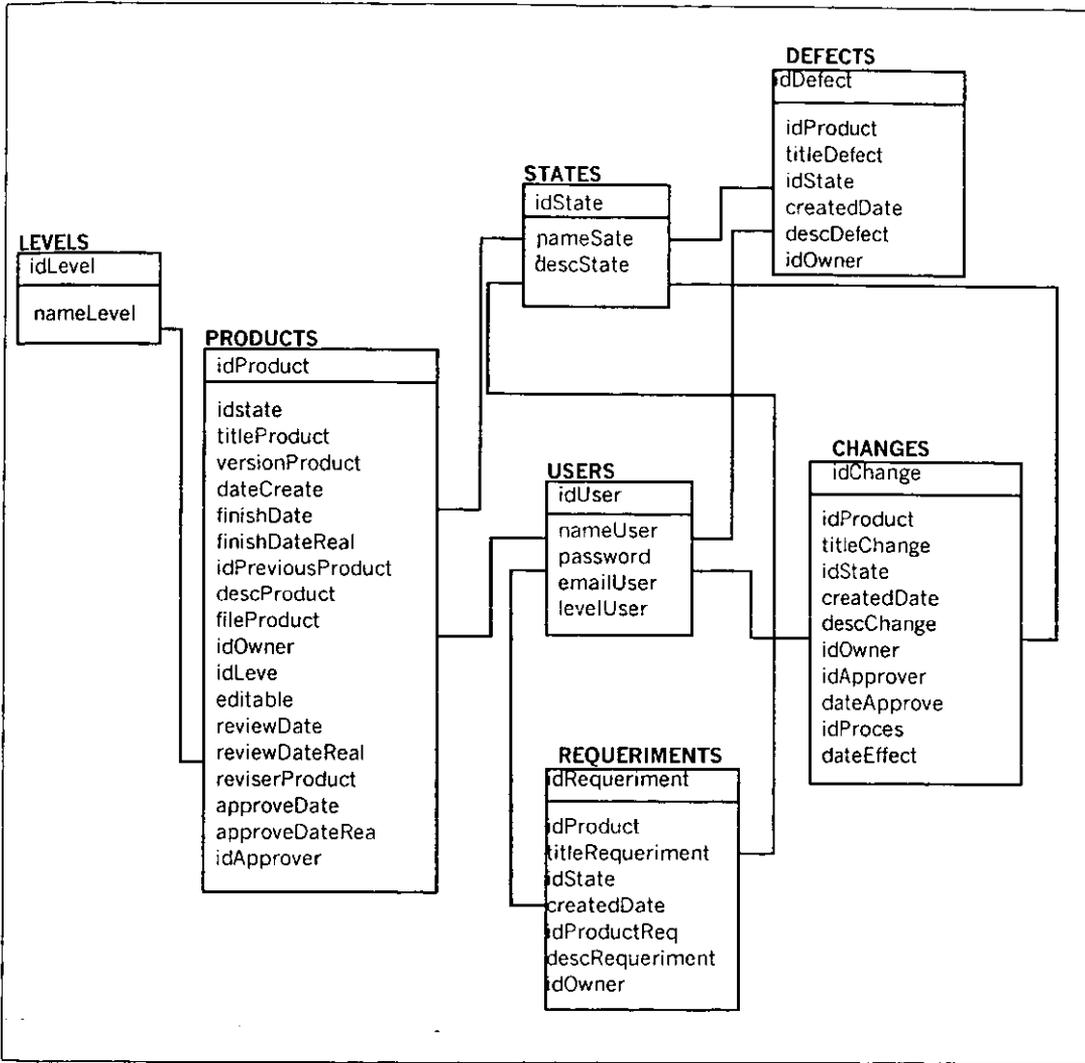


Diagrama Entidad-Relación para el Nivel de Control Completo



5.6. Aportaciones y limitaciones

Aportaciones

El prototipo de Control de Calidad y Cambios a los productos de software realizó mejoras con respecto al prototipo de Control de los Documentos de Administración de la Calidad en empresas pequeñas, que sirvió de base para el desarrollo de ésta tesis, en cuanto a:

1. Definición de niveles configurables de control de calidad y cambios a los productos de software que permiten controlar sus requisitos, defectos o solicitudes de cambio dependiendo de las características del producto o recursos del proyecto.
2. Disponibilidad bajo un ambiente de red, para más de una persona, mejorando la interacción con otros miembros del grupo de desarrollo del producto con la participación de varios usuarios al mismo tiempo en la evaluación de los productos o las solicitudes de cambio a los mismos e incluyendo el uso del correo electrónico.
3. Se aprovechó la ventaja de uso de la tecnología orientada a objetos y el Lenguaje de Modelado Unificado (UML) durante el análisis y diseño, para ampliar las abstracciones correspondientes en cuanto a casos de uso, diagramas de estado, objetos, componentes y arquitectura del prototipo (tanto lógica como física). El trabajo se facilitó además por el empleo de un diseño y desarrollo iterativo incremental al utilizar una jerarquía de clases. Esto permitió no solo la reutilización de diagramas de análisis y diseño sino incluso de código de programación en lenguaje Java con el que fue implementado y documentado el prototipo.
4. Rediseño de la base de datos, incrementando el número de relaciones necesarias para soportar lo objetos a controlar y sus atributos para cada uno de los niveles de control.

Limitaciones

Entre las limitaciones que se consideran en el prototipo C3PoS se encuentran:

1. Es difícil de tratar a los gráficos con el editor con que cuenta el prototipo, aunque el editor permite la visualización de los gráficos, presenta mayores facilidades para la edición de texto con formato html.
2. Durante la edición de un producto es necesario mantener la conexión via Internet, con el repositorio de los productos, lo que puede representar una desventaja en caso de fallas de la red ya que el prototipo no considera la tolerancia a fallas.

3. La herramienta carece de un servicio de transferencia de archivos entre una computadora local y la computadora en donde se ubicaría el repositorio para los productos, por lo que se hace necesario utilizar por separado un servicio de Internet como FTP.
4. La herramienta no permite registrar, si fuera necesario, a más de un revisor o aprobador.
5. No se realiza un registro histórico de las modificaciones al contenido de un producto.
6. No existe una administración de las líneas base.
7. El prototipo no se integra con otras herramientas que existen en el mercado y que pudieran permitir la edición de productos con formato diferente al de html.

Para probarlo en proyectos reales habrá que eliminar estas limitaciones.

Capítulo VI

VI. Propuesta de integración de la herramienta en un proyecto

En este capítulo se realiza una propuesta para la integración de la herramienta de control de calidad y cambios en los productos de software (C3PoS) a un proyecto que se realizó durante el estudio de la maestría en el curso de Ingeniería de Software Orientada a Objetos (ISOO), impartido por la Dra. Hanna Oktaba y la M.C. Ma. Guadalupe Ibargüengoitia [SIPOCIC1999].

6.1. El proyecto SIPOCIC

Durante el curso de ISOO impartido en el semestre escolar 200-1 en el Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), se realizó un proyecto cuya finalidad fue realizar un sistema de inscripción a los cursos impartidos en la Maestría en Ciencias e Ingeniería de Computación, y cuyo nombre recibió "Sistema de Inscripciones al Posgrado en Ciencia e Ingeniería de la Computación de la UNAM (SIPOCIC)".

Para el desarrollo del proyecto se utilizó una metodología que combinó las prácticas sugeridas por el Modelo de Madurez de la Capacidad del Proceso de Software (CMM) nivel 2 y el Proceso Unificado.

6.1.1. El ciclo de vida del proyecto.

El ciclo de vida que se utilizó para desarrollar los productos de software fue el del Proceso Unificado por basarse en Casos de Uso, por ser incremental e iterativo y enfocado a la arquitectura. Su estructura básica se conformó por las siguientes:

Fases:

- Inicio.
- Elaboración.
- Construcción.
- Transición.

Actividades para cada fase:

- Requerimientos.
- Análisis.
- Diseño.
- Implementación.
- Pruebas

Entregables Generales:

- Modelo de Requerimientos.
- Modelo de Análisis.
- Modelo de Diseño.
- Modelo de Implantación.
- Documento de Implementación.
- Documento de Transición.
- Documentación para el usuario y operativa.

6.1.2. Organización.

El equipo estuvo integrado por nueve personas que desempeñaron diferentes roles, como se muestra en la tabla 6.1 y en la tabla 6.2 se mencionan a otras personas involucradas:

Nombre	Rol	Identificador
Blanca Gil Castellanos	Administrador de Proyecto	AP
Pablo Castillejo García	Administrador de Calidad	AQ
Martín De Jesús	Administrador de Configuración	AC
Sandra Gómez	Lider de Proyecto y Administrador de Requerimientos	LP
Judith Jaramillo	Ingeniero de Software	IS-1
Henry Pérez Luna	Ingeniero de Software	IS-2
Alicia Rojas Estevez	Ingeniero de Software	IS-3
Jesús Sandoval Lugo	Ingeniero de Software	IS-4
Moisés Bautista Osorno	Ingeniero de Software	IS-5
Blanca Gil Castellanos	Ingeniero de Software	IS-6
Pablo Castillejo García	Ingeniero de Software	IS-7
Pablo Castillejo García	Coordinador de la Construcción	CC
Henry Pérez Luna	Coordinador de Pruebas	CP
Pablo Castillejo García	Probador	PR-1
Martín De Jesús	Probador	PR-2
Henry Pérez Luna	Probador	PR-3

Tabla 6.1. Equipo de desarrollo

Nombre	Rol	Identificador
Coordinadora de la Maestría Dra. Hanna Oktaba	Cliente	C
M.C. Ma. Guadalupe Inbargüengoitia	Asesora externa	AE-1
	Asesora externa	AE-2

Tabla 6.2. Otras personas involucradas.

6.1.3. Áreas estratégicas.

Se contó con las siguientes áreas estratégicas como estándar de trabajo en general:

- Administración de la Configuración
- Administración de la Calidad.
- Generación de Reportes Mensuales del Avance y del Éxito del proyecto.

6.1.3.1. Administración de la Configuración

Esta área era la responsable de la coordinación de todos los procesos referentes al control de versiones, seguridad de la información, respaldo y control de los documentos.

Se definió la nomenclatura para identificar a los productos de Software:

AANNN-VV.ext

- AA Abreviatura del área

Abreviatura	Nombre del Área
AP	Administración del Proyecto
AC	Administración de la Configuración
AQ	Administración de la Calidad
LP	Liderazgo de equipo de trabajo
RR	Modelo de Requerimientos
AA	Modelo de Análisis
DD	Modelo de Diseño
CC	Modelo de Construcción
PP	Pruebas
CA	Solicitudes de cambios
Ix	Registro de tiempo de Ingenieros de Software, donde x es 1, 2, 3, 4 y 5 asignado a cada Ingeniero de Software

Tabla 6.3. Áreas controladas

- NN Número consecutivo
- V Versión (primer dígito)
- V Versión (segundo dígito)
- .ext Extensión del tipo de documento

Ejemplo: AP001-01.doc

Documento creado por el Administrador del Proyecto (AP) correspondiente al plan (001) con versión 0.1, elaborado con el editor Word.

- **Características de los formatos utilizados.**

En la figura 6.1. se muestra el formato utilizado por los diferentes documentos donde se muestra la información requerida, en el encabezado y pie de página del documento. En la figura el producto corresponde a un Diagrama general de paquetes donde se proporcionaba su identificador y versión así como las fechas, firmas y nombres de los responsables de su producción, revisión o aprobación. El formato indica que el documento está en borrador.

Se observa que no es claro quien lo revisaba o lo aprobaba. De acuerdo a la experiencia en el proyecto, en este caso el Líder del Proyecto lo revisaba y el Administrador del Proyecto lo aprobaba.

Entregable: Diagrama general de paquetes	Ref. DD042-10.DOC	Versión 1.0	Pág. 126	/1
---	-------------------	-------------	----------	----

(Información del diagrama general de paquetes)

Fase: Elaboración	Sistema: Sistema de Inscripción al Posgrado en Ciencia e Ingeniería de la Computación de la UNAM	B	R	A
		X		
Etapa: Diseño	Realizó: Judith Jaramillo López	Fecha: 11/01/2000		
Líder: Sandra Gómez	Calidad: Pablo Castillejos	Administrador: Blanca Gil		
Firma:	Fecha:	Firma:	Fecha:	Firma:

Figura 6.1. Formato de los documentos utilizados en el proyecto SIPOCIC.

Herramienta para la administración de documentos.

Durante el desarrollo del proyecto SIPOCIC se utilizó una herramienta de software como apoyo para la administración de la configuración llamado BSCW 3.2, que permitía compartir con un grupo de usuarios documentos en red bajo el ambiente de Internet [WEB10].

El acceso a dicha información se realizó a través de una clave y un password, todos los miembros del equipo podían consultar los productos de trabajo pero sólo el Administrador de la Configuración tenía derecho de colocar dichos documentos en el repositorio.

El Administrador de la Configuración contaba con una guía para el uso y configuración de la herramienta que era de su total responsabilidad.

6.1.3.2. Administración de la Calidad.

Esta área se encargaba del aseguramiento de la calidad del trabajo entregado, de la verificación y propuesta de estándares de procedimientos y procesos para cada rol y de auditorías para corroborar el cumplimiento de lo anterior.

6.1.3.2. Generación de Reportes Mensuales del Avance y del Éxito del proyecto.

Se elaboraron reportes de las métricas principales del proyecto, como tamaños de elementos estratégicos (original, actual, planeado y realizado), seguimiento del presupuesto, graficación de las métricas principales. Así como el seguimiento a los riesgos.

Cada área estratégica marcó sus métodos y procedimientos para ejercer sus objetivos principales. El método para el seguimiento del proyecto cubrió lo siguiente:

- Contó con un plan de desarrollo de software.
- El Administrador del Proyecto debía estar informado del estado real del proyecto.
- Se contaron con Planes de Contingencia para tomar acciones correctivas en los casos en los que no se estaba cumpliendo el plan.
- Los cambios del plan fueron realizados por los afectados.
- Debía existir una supervisión a los cambios comprometidos.

6.2. Productos de Software del proyecto SIPOCIC y su clasificación con niveles de control.

A continuación se proporciona una propuesta de asignación del tipo de control de los productos de software por área, para el proyecto SIPOCIC, utilizando los niveles de control definidos en la herramienta de Control de Calidad y Cambios a los Productos de Software C3PoS, de acuerdo a las características de control del producto.

Analizando la experiencia obtenida durante el desarrollo del proyecto SIPOCIC, en las siguientes tablas se presenta la asignación de niveles de control, así como los roles que intervinieron en su producción, revisión o aprobación

6.2.1. Asignación de niveles de control y roles.

La asignación de niveles de control y roles se realizó mediante los siguientes criterios:

- **Nivel de Control Simple.**

Se asignó en casos de que el documento era definido por un solo propietario sin requerir de aprobación.

En estos casos la calidad dependía del productor y no existía un control de solicitudes de cambio o registro de defectos. Su producción no impactaba a otros productos de trabajo.

- **Nivel de Control Intermedio.**

Para los productos que requerían un control de producción con revisión independiente al productor, quien utilizaba una especificación documentada de los requisitos.

Un revisor reportaba al Productor los defectos encontrados con una solicitud documentada del cambio al producto, el Productor realizaba las correcciones.

- **Nivel de Control Completo.**

Se asignó en casos en que al producto era necesario realizarle un control de producción con doble control de calidad y control de cambios.

Un revisor verificaba el cumplimiento de los requisitos y un Aprobador realizaba un análisis del impacto y costo sobre otros productos para determinar si el producto o la solicitud de cambio era aprobada o rechazada.

VI. Propuesta de integración de la herramienta en un proyecto

En las siguientes tablas utilizaremos la nomenclatura:

Para indicar los niveles de control:

- NS** Nivel de Control Simple
- NI** Nivel de Control Intermedio
- NC** Nivel de Control Completo

Para indicar los roles que intervinieron de acuerdo a la herramienta

- P** Productor
- R** Revisor
- A** Aprobador

Y utilizaremos los identificadores de la Tabla 6.1. para indicar las personas que participaron en los roles de producción, revisión o aprobación del producto.

Identificador	Nombre	Nivel de Control	P	R	A
AP001-12.MPP	Plan de Trabajo General	NC	AP	LP	AE-1
AP008-10.DOC	Estructura	NC	AP	LP	AE-1
AP009-10.DOC	Planeación	NC	AP	LP	AE-1
AP010-10.DOC	Análisis de Factibilidad	NC	AP	LP	AE-1
AP011-10.DOC	Cotización	NC	AP	LP	C
AP012-10.DOC	Contrato	NC	AP	LP	C
AP013-10.DOC	Reporte Informativo General	NC	AP	LP	AE-2
LP007-10.DOC	Plan de Contención	NC	LP	AP	AE-2

Tabla 6.4. Documentos de la Administración del Proyecto

Estos productos debían ser acordados por todos los miembros involucrados para su revisión o aprobación, debido a que su producción o modificación impactaba a los planes y actividades del desarrollo de todos los productos.

El Administrador del Proyecto sometía a revisión el producto al Líder del Proyecto y una vez aceptado dependía de la aprobación de los Asesores externos o del Cliente.

VI. Propuesta de integración de la herramienta en un proyecto

Identificador	Nombre	Nivel de Control	P	R	A
AC001-20.DOC	Plan de Administración de la Configuración	NC	AC	AP	AQ
AC002-20.DOC	Formato de registro de tiempo de actividades	NC	AC	AP	AQ
AC003-20.DOC	Formato de carátula	NC	AC	AP	AQ
AC004-20.DOC	Formato de hojas de presentación. Para diagramas, hoja horizontal y área de texto la hoja vertical	NC	AC	AP	AQ
AC005-20.DOC	Formato de hoja de requerimientos	NC	AC	LP	AQ
AC006-20.DOC	Formato de casos de uso	NC	AC	LP	AQ
AC007-20.DOC	Formato de glosario de términos y sinónimos	NC	AC	LP	AQ
AC008-20.DOC	Formato Minuta de sesión	NC	AC	AP	AQ
AC009-20.DOC	Formato Bitácora	NC	AC	AP	AQ
AC010-20.DOC	Formato Registro de control de configuración	NC	AC	AP	AQ
AC011-10.DOC	Relación de información controlada bajo el ambiente de BSCW	NC	AC	AP	AQ
AC012-20.PDF	Guía para el uso y configuración de la herramienta de administración de documentos BSCW	NS	AC		
AC013-10.DOC	Guía para el registro de documentos	NC	AC	AP	AQ
AC016-10.DOC	Guía para la elaboración de minutas	NC	AC	AP	AQ
AC017-10.DOC	Guía para la solicitud de cambios	NC	AC	AP	AQ

Tabla 6.5. Documentos de la Administración de la Configuración

Los productos elaborados por el Administrador de la Configuración eran revisados por el área que los empleaba, por lo general era el Administrador de Proyectos o el Líder del Proyecto y éste último a su vez los distribuía, en caso necesario, a los Ingenieros de Software.

Sin embargo el Administrador de Calidad (AQ) evaluaba éstos documentos porque era el responsable de aprobar formatos y estándares.

En la tabla 6.5 se muestra que estos documentos fueron clasificados como Nivel de Control Completo (NC) a excepción de la Guía para la Administración de la Configuración de la Herramienta utilizada (BSCW) que se clasifica en el Nivel de Control Simple, por corresponder a la total responsabilidad técnica del Administrador de la Configuración.

VI. Propuesta de integración de la herramienta en un proyecto

Identificador	Nombre	Nivel de Control	P	R	A
AQ002-02.DOC	Aseguramiento de la Calidad del Software según el nivel 2 del CMM	NI	AQ	AP	
AQ004-01.DOC	Procedimiento de Interpretación del Rol	NI	AQ	AP	
AQ006-01.DOC	Procedimiento de Revisión de Producto de trabajo	NC	AQ	LP	AP
AQ007-01.DOC	Procedimiento de Petición de Consulta o Revisión	NC	AQ	LP	AP
AQ008-01.DOC	Revisión de Interpretación de Rol de Administrador del Proyecto	NI	AQ	AP	
AQ009-01.DOC	Revisión de Interpretación de Rol de Líder del Proyecto	NC	AQ	LP	AP
AQ010-01.DOC	Informe de auditoria de producto de trabajo	NC	AQ	LP	AP
AQ011-01.DOC	Estándares de Programación	NC	AQ	LP	AP
AQ012-01.DOC	Resumen sumario de revisión del proyecto	NC	AQ	LP	AP

Tabla 6.6. Documentos de la Administración de la Calidad

Los estándares propuestos por el Administrador de Calidad para los productos e interpretación de los roles, debían ser revisados por las áreas de interés para su aprobación con el fin de incluir la descripción completa de las actividades así como de los roles involucrados.

Identificador	Nombre	Nivel de Control	P	R	A
RR000-10.DOC	Modelo de Requerimientos	NC	IS-1 al IS-5	LP	AP
RR001-10.DOC	Documento de Presentación	NC	IS-1 al IS-5	LP	AP
RR002-10.DOC	Documento de Requerimientos	NC	IS-1 al IS-5	LP	AP
RR003-10.DOC	Diagrama de Casos de Uso del Negocio	NC	IS-1 al IS-5	LP	AP
RR006-10.DOC	Especificación de Infraestructura de Desarrollo y Aplicación	NC	IS-1 al IS-5	LP	AP
RR008-10.DOC	Glosario de Términos	NC	IS-5	LP	AP

Tabla 6.7. Documentos del Modelo de Requerimientos

VI. Propuesta de integración de la herramienta en un proyecto

En los productos para el Modelo de Requerimientos participó más de una persona y fueron revisados por el Líder del Proyecto y aprobados por el Administrador del Proyecto. Lo mismo sucedió para el modelo de Análisis y Diseño.

Identificador	Nombre	Nivel de Control	P	R	A
AA014-10.DOC	Diagrama de Casos de Uso para el Control de Catálogos.	NC	IS-1	LP	AP
AA015-10.DOC	Diagrama de Casos de Uso para el Control de Horario Personal	NC	IS-4	LP	AP
AA016-10.DOC	Diagrama de Casos de Uso para el Control de Reportes	NC	IS-3	LP	AP
AA020-10.DOC	Diagrama de Estados para el Control de Catálogos	NC	IS-1	LP	AP
AA021-10.DOC	Diagrama de Estados para el Control de Horario Personal	NC	IS-4	LP	AP
AA022-10.DOC	Diagrama de Estados para el Control de Reportes	NC	IS-3	LP	AP
AA029-10.DOC	Diagrama de Interacción para el Control de Catálogos	NC	IS-1	LP	AP
AA030-10.DOC	Diagrama de Interacción para el Control de Horario Personal	NC	IS-4	LP	AP
AA031-10.DOC	Diagrama de Interacción para el Control de Reportes	NC	IS-3	LP	AP
AA035-10.DOC	Diagrama de Actividades para el Control de Catálogos	NC	IS-1	LP	AP
AA038-10.DOC	Manual de Operación	NC	IS-5	LP	C
AA039-10.DOC	Manual del usuario	NC	IS-5	LP	C
AA041-10.DOC	Diagrama de Paquetes General	NC	IS-1 al IS-5	LP	AP

Tabla 6.8. Documentos del Modelo de Análisis

VI. Propuesta de integración de la herramienta en un proyecto

Identificador	Nombre	Nivel de Control	P	R	A
DD043-10.DOC	Diagrama de Casos de Uso Reales para el Control de Catálogos.	NC	IS-1	LP	AP
DD044-10.DOC	Diagrama de Casos de Uso para el Control de Horario Personal	NC	IS-4	LP	AP
DD045-10.DOC	Diagrama de Casos de Uso Reales para el Control de Reportes	NC	IS-3	LP	AP
DD046-10.DOC	Diagrama general de clases	NC	IS-1 IS-3, IS-4	LP	AP
DD050-10.DOC	Diagrama General de Paquetes	NC	IS-1 IS-3, IS-4	LP	AP
DD051-10.DOC	Arquitectura de la aplicación	NC	IS-2	LP	AP
DD052-10.DOC	Prototipo	NC	IS-4	LP	AP
DD053-10.DOC	Lista de Errores	NC	IS-1 IS-3, IS-4	LP	AP

Tabla 6.9. Documentos del Modelo de Diseño.

Identificador	Nombre	Nivel de Control	P	R	A
CC001-10.MPP	Plan de Construcción	NI	LP	AP	
CC010-30.DOC	Descripción de la clase CargaCatalogos	NI	IS-6	LP	
CC011-30.DOC	Descripción de la clase DiaMensajes	NI	IS-6	LP	
CC021-10.DOC	Descripción de la clase LogPass	NI	IS-5	LP	
CC023-10.DOC	Descripción de la clase CierreInsc	NI	IS-2	LP	
CC025-10.DOC	Descripción de la clase Solicitud	NI	IS-7	LP	
CC035-10.DOC	Descripción de la clase AlumnosPorMateria	NI	IS-2	LP	
CC036-10.DOC	Descripción de la clase MateriaPorSemestre	NI	IS-2	LP	
CC037-10.DOC	Descripción de la clase PaginaSemestral	NI	IS-2	LP	
CC038-10.DOC	Descripción de la clase Formatosol	NI	IS-2	LP	
CC040-30.DOC	Clase CargaCatalogos	NI	IS-6	LP	
CC041-30.DOC	Clase DiaMensajes	NI	IS-6	LP	
CC051-10.DOC	Clase LogPass	NI	IS-5	LP	
CC053-10.DOC	Clase CierreInsc	NI	IS-5	LP	
CC055-10.DOC	Clase Solicitud	NI	IS-7	LP	
CC065-10.DOC	Clase AlumnosPorMateria	NI	IS-2	LP	
CC066-10.DOC	Clase MateriaPorSemestre	NI	IS-2	LP	
CC067-10.DOC	Clase PaginaSemestral	NI	IS-2	LP	
CC068-10.DOC	Clase Formatosol	NI	IS-2	LP	

Tabla 6.10. Documentos del Modelo de Construcción.

VI. Propuesta de integración de la herramienta en un proyecto

El Plan de Construcción del Sistema debía ser revisado por el Administrador del Proyecto.

En cuanto a los productos de los Ingenieros de Software, debían ser revisados por el Líder del Proyecto durante la fase de construcción.

Identificador	Nombre	Nivel de Control	P	R	A
PP124-10.DOC	Plan de Pruebas	NC	CP	LP	AP
PP010-10.DOC	Prueba de la clase DiaMensajes	NI	PR-1	LP	
PP013-10.DOC	Prueba de la clase Mensaje	NI	PR-3	LP	
PP014-10.DOC	Prueba de la clase CierraVentana	NI	PR-3	LP	
PP025-10.DOC	Prueba de la clase SolicitudInscripcion	NI	PR-3	LP	
PP035-10.DOC	Prueba de la clase Formato	NI	PR-1	LP	
PP036-10.DOC	Prueba de la clase HttpServlet	NI	PR-1	LP	
PP100-10.DOC	Datos de prueba	NI	PR-2	LP	
PP110-10.DOC	Pruebas de generar información bajo la responsabilidad del usuario administrativo: Materias, profesores, alumnos, lugares y áreas	NI	PR-2	LP	
PP119-10.DOC	Pruebas de realizar las altas, cambios, consultas a los horarios semestrales	NI	PR-2	LP	
PP121-10.DOC	Pruebas de validar Acceso al Sistema	NI	PR-2	LP	
PP125-10.DOC	Pruebas de generación de la solicitud de inscripción de los alumnos	NI	PR-2	LP	
PP128-10.DOC	Pruebas de integración del Sistema de Inscripción al Posgrado en Ciencia e Ingeniería de la Computación de la UNAM	NI	PR-2	LP	
PP135-10.DOC	Pruebas de generación de las listas de alumnos inscritos por materia	NI	PR-2	LP	
PP135-10.DOC	Pruebas de Creación de Horarios Semestrales	NI	PR-2	LP	

Tabla 6.11. Documentos de las pruebas

Los resultados de las pruebas debían ser revisadas por el Líder del Proyecto para su revisión y si fuera el caso para la incorporación de los cambios necesarios.

VI. Propuesta de integración de la herramienta en un proyecto

Identificador	Nombre	Nivel de Control	P	R	A
CA001-03.DOC	Solicitud de cambio a DD046-01.DOC	NC	I6	IS-1 IS-3, IS-4	LP
CA002-03.DOC	Solicitud de cambio a DD043-01.DOC	NC	I6	IS-1 IS-3, IS-4	LP

Tabla 6.12. Documentos de las Solicitudes de Cambio.

Identificador	Nombre	Nivel de Control	P	R	A
MM001-01.DOC	Minutas de sesión	NC	IS	LP	AP
LP002-09.DOC	Registro de Tiempos del Administrador del Proyecto	NI	AP	LP	
AQ002-09.DOC	Registro de Tiempos del Líder del Proyecto	NI	AP	LP	
AC002-09.DOC	Registro de Tiempos del Administrador de la Calidad	NI	AP	LP	
LP002-09.DOC	Registro de Tiempos del Líder del Proyecto	NI	AP	LP	
I1001-04.DOC	Registro de tiempo de actividades del Ingeniero de Software 1	NI	IS-1	LP	
I2001-04.DOC	Registro de tiempo de actividades del Ingeniero de Software 2	NI	IS-2	LP	
I3001-04.DOC	Registro de tiempo de actividades del Ingeniero de Software 3	NI	IS-3	LP	
I4001-04.DOC	Registro de tiempo de actividades del Ingeniero de Software 4	NI	IS-4	LP	
I5001-04.DOC	Registro de tiempo de actividades del Ingeniero de Software 5	NI	IS-5	LP	
AP006-04.DOC	Bitácora del Administrador del Proyecto	NS	AP		
LP006-13.DOC	Bitácora del Líder del Proyecto	NS	LP		
AC006-13.DOC	Bitácora del Administrador de la Configuración	NS	AC		
AC006-13.DOC	Bitácora del Administrador de la Calidad	NS	AQ		
I1002-04.DOC	Bitácora del Ingeniero de Software 1	NS	IS1		
I2002-04.DOC	Bitácora del Ingeniero de Software 2	NS	IS2		
I3002-04.DOC	Bitácora del Ingeniero de Software 3	NS	IS3		
I4002-04.DOC	Bitácora del Ingeniero de Software 4	NS	IS4		
I5002-04.DOC	Bitácora del Ingeniero de Software 5	NS	IS5		

Tabla 6.13. Documentos de reportes.

En resumen, en la Tabla 6.13 se muestra el número de documentos por su tipo y nivel de control.

Area	No. NS	No. NI	No. NC
AP - Administración del Proyecto			8
AC - Administración de la Configuración	1		14
AQ - Administración de la Calidad		3	6
RR - Modelo de Requerimientos			6
AA - Modelo de Análisis			13
DD - Modelo de Diseño			8
CC - Modelo de Construcción		18	
PP - Pruebas		14	1
CA - Solicitudes de cambios			2
Ix - Registro de tiempo de Ingenieros de Software		5	
A - Registro de tiempo de Administradores y Líder de Proyecto		4	
Ix - Bitácoras de Ingenieros de Software	5		
A - Bitácoras de Administradores y Líder de Proyecto	4		
TOTAL	10	44	58

Tabla 6.14. Resumen del número de documentos por tipo y nivel de control.

Debido a que se utilizó una metodología que combinó las prácticas sugeridas por el Modelo de Madurez de la Capacidad del Proceso de Software (CMM) nivel 2, la mayoría de los documentos fueron clasificados en los Niveles de Control Intermedio o Completo.

Los documentos que fueron clasificados en el Nivel Simple realmente no tenían la misma relevancia que los demás y su producción no impactaban el desarrollo de éstos últimos.

6.2.2. Controles para el proyecto.

Se determinaron procedimientos para:

- Controlar los productos que están activos o están siendo utilizados
- Controlar el acceso y distribución de los mismos
- Controlar los cambios y modificaciones a los productos

- **Elementos de control de los productos**

Para identificar un producto se definieron las características que debían contar los mismos y consistieron en:

- Identificador o referencia del producto.
- Título
- Versión
- Número de páginas
- Estado del documento (borrador, revisión o aprobación)
- Identificación del responsable de su elaboración
- Fecha de elaboración
- Firmas de los responsables (líder y/o Administradores)
- Identificación de los que realizan la revisión
- Descripción general de su contenido

Estas características debían estar contenidas en el formato que se muestra en la figura 6.1.

Con la integración de la herramienta C3PoS a este proyecto los productos contarían con un mayor número de características que proporcionarían mayor información para el apoyo a los reportes y determinar avances del proyecto. Las características que se adicionarían serían las siguientes:

- Fecha planeada de terminación
- Fecha real de terminación
- Integración del registro y control de los requisitos con los que debe cumplir
- Integración del registro y control de los defectos encontrados al producto
- Integración del registro de las solicitudes de cambio.
- Fecha planeada de revisión
- Fecha real de revisión
- Fecha planeada de aprobación
- Fecha real de aprobación

El uso de la herramienta C3PoS simplificaría el control de todos estos elementos debido a que el formato que se utilizó en el proyecto SIPOCIC (figura 6.1) resultaría insuficiente para el registro de los mismos. También se facilitaría el registro y control de los defectos asociados al producto, los cuales serían reportados vía correo electrónico y cuya opción está integrada en la misma herramienta.

La herramienta C3PoS permite definir fácilmente y con claridad el responsable de la producción, revisión o aprobación aún cuando un miembro del equipo de trabajo desempeñe diferentes roles.

- **Elementos de control de las solicitudes de cambio**

Para el control de solicitudes de cambio éstas debían tener las siguientes características:

- 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 responsable de su elaboración
- 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

Durante el proyecto ya de por sí era difícil identificar oportunamente el documento con la última versión y estado, también debían ser controladas las diferentes solicitudes de cambio que podría tener un mismo producto. Había que darles seguimiento a las solicitudes de cambio hasta su implementación o rechazo.

Los productos y las solicitudes de cambio estaban dispuestos en directorios diferentes por lo no existía una integración que permitiera identificar fácilmente las solicitudes de cambio asociadas a un producto. En estas mismas solicitudes había que describir los defectos encontrados al producto.

Además como sucedía con los productos, también había que reemplazar las solicitudes de cambio cada vez que se modificaba su estado, por estar definido en el formato.

Con el uso de la herramienta C3PoS se integran los requisitos, defectos encontrados así como las solicitudes de cambio propuestas, disminuyendo el tiempo de búsqueda y de reporte de los mismos, mediante su aviso por correo electrónico y su consulta en red. De tal forma que solo basta con consultar el producto y pedir que se muestren sus requisitos, defectos encontrados o solicitudes de cambio asociados al producto, como resultado de un proceso de evaluación

- **Control de acceso y distribución de los productos**

Con el uso de la herramienta BSCW durante el desarrollo del proyecto, los miembros del equipo tenían a su disposición, a través de Internet, los productos para su consulta y si lo deseaban obtenían una copia de los mismos. Sin embargo sólo el Administrador de la Configuración tenía derecho de colocar dichos documentos en el repositorio.

La labor de actualización del repositorio era difícil debido a que la recepción de dichos documentos por parte del Administrador de la Configuración era a través del correo electrónico, y debía clasificar los documentos para colocarlos en el directorio correspondiente para su consulta, con base a la información enviada por el productor.

Además el formato era insuficiente y la herramienta BSCW no estaba diseñada para registrar otros elementos de control que sirvieran a los miembros del equipo para su rápida consulta, como por ejemplo, el estado del documento. También, estrictamente había que reemplazar el documento cuando se modificaba el estado del producto, ya que así lo requería el formato.

Otra de las dificultades se encontraba en manejo de diferentes versiones que a su vez podían tener diferentes estados (borrador, en revisión o aprobación), esto duplicaba la labor de actualización del documento y la consulta por parte de otros miembros del equipo porque un documento podría estar disponible en Internet en estado de revisión pero el Administrador de la Configuración recientemente le había llegado por correo la versión de aprobación y le llevaba tiempo reemplazarlo.

Con el uso de la herramienta C3PoS el mismo propietario del documento puede crearlo y editarlo lo que facilita la puesta a disposición de los demás miembros del equipo, avisándoles por correo electrónico, cuya opción trae consigo la misma herramienta. Además no sería necesario reemplazar el documento por otro cuando cambie su estado durante su ciclo de evaluación y en la información general que muestra la herramienta C3PoS en la lista de los productos se indica el estado y versión que guarda el producto.

6.3. Limitaciones de la herramienta C3PoS.

Los productos fueron realizados en diversos formatos (word, pdf, ppt) y la herramienta en su versión actual no los soporta. Para darlos de alta en la herramienta sería necesario convertirlos en formato html. Los gráficos serían difícil de manipular, ya que el editor con que cuenta la herramienta esta enfocado al trabajo con texto.

Durante la edición de un producto es necesario mantener la conexión vía Internet lo que puede representar una desventaja en caso de fallas de la red.

La herramienta carece de un servicio de transferencia de archivos entre una computadora local y la computadora en donde se concentrarían los productos por lo que sería necesario utilizar por separado un servicio de Internet como FTP.

La herramienta no permite registrar por producto a más de un propietario, revisor o aprobador como lo requieren algunos documentos de requisitos, análisis o diseño del proyecto SIPOCIC.

Para probarlo en proyectos reales habrá que eliminar estas limitaciones.

Conclusiones

En este trabajo se logró el objetivo de construir un modelo genérico de control de calidad y cambios a los productos de software representado con el Lenguaje de Modelado Unificado (UML). Este modelo tiene las siguientes características:

1. El modelo genérico integra en un solo proceso la producción, corrección, revisión, aprobación y el control de cambios de los productos de software que frecuentemente son referidos como prácticas o áreas separadas en otros modelos o herramientas.
2. Cada uno de los tres niveles de control de calidad y cambios de los productos de software del modelo genérico, describe las actividades que deben ser llevadas a cabo por los diferentes roles del grupo de ingeniería de software. El nivel de control simple podría clasificarse como al caso clásico del heroísmo donde todo depende de la visión limitada de una persona. El nivel intermedio refleja el reconocimiento del valor de colaborar en equipo y de la documentación escrita. El nivel completo puede considerarse para el control ideal de proyectos a largo plazo y de mayor complejidad.
3. El control de calidad y el control de los cambios en cada nivel puede verse como un patrón genérico y especializarse de acuerdo a las necesidades o recursos particulares de la organización o proyecto. Se utilizarían los correspondientes elementos genéricos para el desarrollo del producto de software dependiendo del nivel de control en el que haya sido clasificado.
4. El modelo genérico puede ser utilizado para definir estándares de los procesos de control de calidad y cambios para una organización o proyecto debido a que incluye conceptos que definen estándares de calidad para la mejora de los procesos obtenidos del Modelo de Madurez de la Capacidad del Proceso de Software (CMM). Además como apoyo para estimar los costos en cuanto a la producción, los cambios y el control de calidad de los productos de software.

El modelo genérico de control de calidad y cambios a los productos de software fue presentado en el 3er. Encuentro Internacional de Ciencias de la Computación ENC'01, celebrado en la Ciudad de Aguascalientes, México, el 18 de septiembre del 2001 [OD2001].

Mediante el desarrollo del prototipo se mostró que el modelo genérico puede ser soportado por una herramienta que cuente con niveles configurables de control de calidad y cambios a los productos de software. Además se logran las siguientes aportaciones.

1. Obtener una mayor integración y coordinación de las actividades de los miembros que intervienen en el desarrollo y evaluación de los productos de software, mejorando la comunicación y sobre todo la colaboración, que representa la piedra angular de un grupo de trabajo.
2. Para la realización del prototipo, fue de gran ayuda la utilización de la tecnología orientada a objetos y en particular el Lenguaje de Modelado Unificado (UML). Mediante diferentes niveles de abstracción que ofrece UML no solo fue posible realizar con facilidad una descripción de los roles, actividades y objetos descritos en el modelo genérico sino que también el modelado del prototipo desde su etapa de análisis hasta su diseño con el suficiente detalle para facilitar su implementación.
3. Con respecto al uso del modelado con UML, con facilidad se comprendió el análisis y diseño plasmado en el prototipo de la tesis anterior, que sirvió como base para el desarrollo de ésta tesis. Por lo que fue más fácil introducir modificaciones a un sistema de software que cuenta con un modelado en UML que si no lo tuviera.

El uso de la tecnología de la Web y las facilidades ofrecidas por el lenguaje de programación Java, en el desarrollo de aplicaciones que trabajan bajo un ambiente de red, representó una buena opción para hacer que el prototipo fuera un medio integrador de todas las actividades del grupo de desarrollo de los productos de software.

El modelo genérico así como la caracterización que se obtuvo de las herramientas relacionadas con el control de los productos y cambios puede servir de referencia para comprender y evaluar otras herramientas.

Con el caso de estudio de la propuesta de integración de la herramienta a un proyecto, se demuestra su viabilidad al clasificar los productos del proyecto en diferentes niveles de control de calidad y cambios, que controla la herramienta, con lo que se obtendrían los siguientes beneficios:

1. Se mejorarían y simplificarían las actividades de control de calidad y solicitudes de cambio para los productos de software, al ampliar y definir claramente los elementos a controlar y especificar los miembros que participarían en su producción, revisión o aprobación.
2. Con la integración al producto de sus requisitos, de los defectos encontrados, así como de las solicitudes de cambio, se disminuiría el tiempo de búsqueda y de reporte de los mismos, obteniéndose mayor claridad para los procesos de evaluación del producto.
3. Agilizaría la colaboración entre todos los miembros del equipo de desarrollo al mantener la disponibilidad en red de los productos a otros grupos involucrados en la evaluación de los mismos. Se facilitaría la comunicación al utilizar para las notificaciones la opción de correo electrónico que tiene integrada la herramienta.

También, con este caso de estudio se reconocen las limitaciones de la herramienta que habría que eliminar para su uso en proyectos reales.

Trabajos a futuro

Este prototipo constituye la base para la implementación de una herramienta con mayor funcionalidad y con elementos que no fueron incluidos en este prototipo para lo cual se hacen las siguientes recomendaciones:

- Cambiar dinámicamente el nivel de control del producto de acuerdo a la madurez del mismo.
- Incluir en la evaluación del producto y de sus solicitudes de cambio a más de un revisor o aprobador.
- Permitir facilidades de búsqueda y de impresión.
- Integrar el registro histórico de las modificaciones al contenido de un producto.
- Administración de las líneas base.
- La posibilidad de incluir un control a nivel de proyecto.

- Incluir las métricas con el fin de determinar tendencias de algún producto o proyecto.
- Mejorar la edición y visualización de los productos por medio de la integración de este prototipo con otras herramientas que existen en el mercado para permitir crear además productos con formatos diferentes al de html.
- Permitir la transferencia de productos entre repositorios ubicados localmente y repositorios compartidos en forma central.
- Soportar la tolerancia a fallas debido a su trabajo bajo un ambiente de red.

Bibliografía

- [Ber 1992] Berlack, H. Ronald, *Software Configuration Management*; Addison-Wesley; EUA 1992.
- [BMT1999] Brown, William J.; McCormick III, Hays W.; Thomas, Scott W.; *Anti-Patterns and Patterns in Software Configuration Management*; EUA 1999.
- [Cad1996] Cadena, Eduardo; *ISO 9000, una visión general*”, *Soluciones Avanzadas*; México, 1996.
- [Cha1997] Chauvet, Jean Marie; *Corba, ActiveX y Java Beans*, Ediciones Gestión 2000, París 1997.
- [CMM®] Paul, Mark C., et. Al.; *The Capability Maturity Model, Guidelines for Improving the Software Process*; SEI Series in Software Engineering, Addison-Wesley, EUA1995.
- [Con1999] Conallen Jim; *Building Web Applications with UML*; Addison-Wesley; EUA 1999.
- [Hum1997] Humphrey, Watts S.; *Introduction to the Personal Software Process*; SEI Series; Addison Wesley, EUA 1997.
- [Hum1999] Humphrey, Watts S.; *Introduction to the Team Software Process*; SEI Series; Addison Wesley; EUA 1999.
- [Ibar1999] Ibarguengoitia, Guadalupe; *Un procedimiento para administrar la configuración del software acorde con CMM nivel 2*; México, 1999.
- [Jal2000] Jalote, Pankaj; *CMM in Practice: Processes for Executing Software Projects at Infosys*; (The SEI Series in Software Engineering); EUA 2000.
- [Jaw1999] Jaime Jaworski, *Java 1.2 Al descubierto*; EUA 1999.
- [JBR1999] Jacobson, Ivar; Booch, Grady; Rumbaugh, James; *The Unified Modeling Language User Guide*; Addison-Wesley; EUA 1999.
- [JBRu1999] Jacobson, Ivar, Booch Grady; Rumbaugh, James; *The Unified Software Development Process*; Addison-Wesley; EUA 1999.

- [KP1999] Koomen, Tim; Pol, Martin; *Test Process Improvement*; Addison-Wesley; Inglaterra 1999.
- [MTU2000] Maruyama, Hiroshi; Tamura, Kent; Uramoto, Naohiko; *Creación de sitios Web con XML y Java*; España 2000.
- [OD2001] Oktaba, Hanna; De Jesús, Martín; *Multilevel Generic Model of Quality and Change Control for Software Products*, México 2001.
- [Pau1999] Paul S. Wang; *Java With Object-Oriented and World Wide Web Applications*; EUA 1999.
- [PauM1993] Paul, Mark C., et. Al.; *The Capability Maturity Model for Software. Version 1.1*; SEI CMU/SEI-93-TR-24; EUA 1993.
- [PO1999] Pérez, Escobar Carlos J.; Oktaba, Hanna; *Un prototipo para el Control de los Documentos de Administración de la Calidad en Empresas Pequeñas*; Tesis de Maestría, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas; UNAM, México 1999.
- [Sch1995] Schmauch, Charles H.: *ISO 9000 for Software Developers*, Revised Edition, ASQC Quality Press, EU., 1995.
- [Sun1999] Sun Microsystem, Inc; *Graphic Java Mastering the JFC Volume II Swing*; EUA 1999.
- [DU1995] De Urquijo Francisco, Carlos Vizcaíno; *Internet para Todos, Soluciones Avanzadas No. 27*, México 1995.
- [SIPOCIC1999] Gil, Blanca; Castillejo, Pablo; De Jesús, Martín; Gómez, Sandra; Jaramillo, Judith; Pérez, Henry; Rojas ,Alicia; Sandoval, Jesús; Bautista, Moisés; Oktaba, Hanna; Inbargüengoitia, Guadalupe; "Sistema de Inscripciones al Posgrado en Ciencia e Ingeniería de la Computación de la UNAM; Documentación del proyecto; UNAM, México 2000.

Sitios en Internet

- [WEB1] <http://webopedia.internet.com/TERM//Internet.html>; *Dictionary and search engine for computer and Internet technology*, 2001.
- [WEB2] <http://www.dis.um.es/~bmoros/Tutorial/parte21/cap21-3.html>; *Tutorial de Java - Conectividad JDBC*, 2001.
- [WEB3] <http://www.sei.cmu.edu/>; Software Engineering Institute (SEI) Home Page, 2001.
- [WEB4] <http://java.sun.com/>; Sun Microsystems' Java Technology Home Page, 2001
- [WEB5] <http://www.serena.com/>; Serena Home Page, 2001
- [WEB6] <http://www.aimware.com/>; Aimware Home Page, 2001
- [WEB7] <http://www.merant.com/>; Merant PVCS Enterprise Change Management, 2001
- [WEB8] <http://www.componentsoftware.com/>; ComponentSoftware RCS (CS-RCS) Home Page, 2001
- [WEB9] <http://www.cvshome.org/>, Concurrent Versions System Home Page, 2001
- [WEB10]. <http://bscw.gmd.de>; BSCW (Basic Support for Cooperative Work) Home Page, 2001

ANEXO A

Herramientas que apoyan el control de cambios a los productos de software

A continuación se proporciona un resumen de la información sobre herramientas existentes en el mercado y que apoyan el control de cambios a los productos de software. Con dicha información se obtuvo la caracterización de este tipo de herramientas y que fue mostrada en el capítulo 3 de ésta tesis.

eChange Man (SERENA Software, Inc.) [WEB5]

Proporciona control sobre la administración del cambio a los productos de software. Centralmente administra el ciclo de vida de toda la aplicación en un ambiente multiplataforma (Web, Windows, UNIX, LINUX, OS/400 y MPE/iX).

Como trabaja bajo una arquitectura de red, se administra el desarrollo del software distribuido geográficamente.

Proporciona seguridad en el control de versiones, análisis de impacto, auditoría, administración de la construcción y liberación del software y ofrece integración con otros productos como Visual Studio .NET, VisualAge Java y WebSphere Studio.

Es una herramienta que no ofrece versiones de prueba y su costo es mayor.

Algunos de los productos que integran ésta herramientas son:

- **Serena™ ChangeMan™ DS (Formalmente eChange Man™)**
Software Change Manager for Distributed Systems
Administra los cambios involucrados en la administración, desarrollo y mantenimiento de las aplicaciones de software un ambiente multiplataforma
- **Serena™ ChangeMan™ WCM (Formalmente ChangeContent™)**
Web Code & Content Manager
Serena ChangeMan WCM proporciona la infraestructura para administrar todo el ciclo de vida para desarrollar contenido Web.

- **Serena™ ChangeMan™ ALM (Formalmente eRequestMan™)**
Application Lifecycle Manager

Da un seguimiento a reporte sobre las personas, los procesos y tareas para la resolución de solicitudes de cambio. Mejorando la productividad del desarrollador con la organización de las actividades y la información.

Aimware

El objetivo de aimware [WEB6] es proporcionar una solución completa para administrar el ciclo de vida del desarrollo de software, los productos de aimware están diseñados para tener el control completo del desarrollo de software y el proceso de pruebas bajo un ambiente integrado y consisten de:

Administración de requisitos. "El objetivo es administrar los requisitos, los problemas y cambios que se susciten sobre los mismos ayudando a la organización a comprender las solicitudes que son requeridas y permite al ubicarse en que momento esta en el proceso mediante una adecuada comprensión de como está asignado el trabajo entre los miembros del equipo y que requisitos están cumpliendo con sus usuarios.

Administración del producto. Permite crear, almacenar y administrar diseños del proyecto, requisitos del software, especificaciones de diseño, características técnicas del plan, descripciones del programa, liberaciones, sugerencias de mejora, planes de prueba y evaluación, objetivos, grupos de prueba, procedimientos, pruebas de ejecución y cualquiera tipo de comentarios a la evaluación, análisis o reportes.

Administración del proceso. Con este producto se puede crear y mantener procedimientos de operación ampliamente estandarizados. Permite al usuario un alto grado de control sobre la creación y publicación de guías del proceso organizacional que incluye Políticas, descripciones del Proceso, Procedimientos, Listas de verificación, Ciclos de Vida, y Plantillas. Se puede publicar y mejorar la información crítica que refleja las mejores prácticas de la organización, proporcionando un alto grado de control sobre la ejecución de las mismas y el flujo de trabajo. Todo esto representa una base para todos los procesos de mejora de la organización.

Administración del proyecto. Para registrar, analizar y dar seguimiento al proyecto. Proporciona la habilidad para que electrónicamente compartan el estatus y avance de la información proporcionando acceso al administrador a datos importantes y reportes que se necesitan para hacer decisiones críticas de negocios basadas en información de los proyectos que son obtenidos en tiempo real.

PVCS Dimensions (MERANT)

MERANT PVCS [WEB7] es una empresa dedicada a la creación de herramientas para la administración del cambio a productos de software.

PVCS Profesional 3.6 propone una solución para la Administración de la Configuración del software y lo realiza mediante los productos:

- **PVCS para la administración de las versiones.**

Registra los cambios en el código, soporta múltiples espacios de trabajo y archiva revisiones de los archivos..

Mantiene auditoria de los cambios, crea reportes basados en HTML y controla los cambios basándose en el ciclo de vida.

Soporta plataforma heterogéneas

- **PVCS para el seguimiento.**

Reporte de defectos, de requisitos, liberación al cliente, tareas de los proyectos. Enfocado principalmente para los desarrolladores, para el personal de Aseguramiento de la Calidad, los Usuarios.

Establece prioridades y responsables. Notificación inmediata por E-mail y ayuda a medir el progreso del proyecto.

- **PVCS para la construcción de la configuración.**

Explora e identifica dependencias. Corrige las versiones construidas. Rastreo de las versiones construidas. Construcción en varias plataformas. Seguridad en capas para acceso seguro vía Internet.

CS-RCS (Component Software)

ComponentSoftware RCS (CS-RCS) CS-RCS es una herramienta de administración del cambio y/o control de versiones [WEB8] y algunas de sus características son:

- Administra la revisión de documentos. Utilizada para monitorear los cambios realizados en archivos que son accedados por estaciones de trabajo que se desempeñan en una sola aplicación o que se encuentran en un ambiente de red.
- Esta herramienta está basada en otra herramienta, la GNU RCS y está completamente integrada con Windows 95 y Windows NT.
- Soporta multiplataforma para trabajo en grupo y es una opción para los sitios que comparten archivos en plataformas UNIX y Windows.
- Maneja varios tipos de documentos que incluyen archivos de programa, documentos HTML documentos MS-Word, imágenes y dibujos.
- Puede utilizar servidores de archivos o manejadores locales que almacenen los archivos. Las conexiones a la red al repositorio de archivos puede ser una red de área local o de gran área, realizada a través de conexión telefónica e Internet.
- Existe versión para su evaluación.

Concurrent Versions System (CVS) (Cyclic Software)

Las características [WEB9] de ésta herramienta son:

- CVS es el Sistema de Versiones Concurrente
- Herramienta de dominio público para el control de versiones.
- Realiza un registro histórico de los cambios que fueron hechos a los archivos fuente.
- Está disponible tanto para desarrolladores individuales como para grandes equipos de trabajo.
- Controla las actividades de los miembros del grupo de desarrollo.
- Utilizan el método de acceso cliente-servidor permite que los desarrolladores tengan acceso a su código desde cualquier lugar contando con una conexión a Internet
- El programa cliente está disponible en la mayoría de las plataformas, se encuentra disponible en plataformas Linux, Machintosh, OS2, UNIX, VMS, Windows 95/NT.

ANEXO B

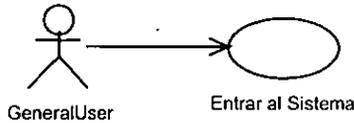
Documento de análisis para productos con nivel de control simple

Este documento contiene información obtenida durante la fase de análisis del prototipo para el control de productos y lo componen:

1. Casos de uso y escenarios
2. Diagramas de clase
3. Diagrama de estado
4. Diccionario de clases

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

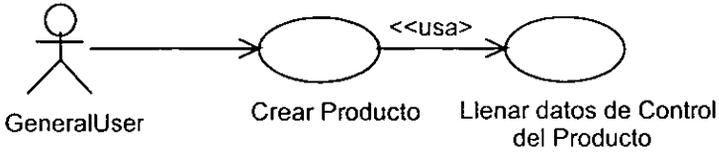
Documento de Análisis
Casos de uso y escenarios, Nivel Simple (Versión 0.8)
ENTRAR AL SISTEMA



- Caso de Uso:** Entrar al sistema
- Actores:** Usuario
- Propósito:** Validar la clave de acceso y la clave secreta del usuario
- Descripción:** El usuario introducirá su clave de acceso y clave secreta para que el sistema consulte el catálogo de claves de acceso y verifique si son correctas.
- Escenario normal:**
- 1: El usuario inicia el sistema.
 - 2: El sistema pide clave de acceso y clave secreta
 - 3: El usuario proporciona clave de acceso y clave secreta
 - 4: El sistema valida clave de acceso y clave secreta
 - 5: El usuario entra al sistema
- Flujos excepcionales:**
- 1: Clave de acceso y/o clave secreta no valida
 - 2: El usuario no tiene acceso al sistema si su clave no es válida y el sistema mostrará la ventana de acceso mientras no exceda el número máximo de intentos permitidos.
 3. El usuario cancela entrar al sistema
- Precondición:** Existe catálogo de usuarios con claves
- Postcondición:** Claves de acceso válidas

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Simple (Versión 0.8)
CREAR PRODUCTO



Caso de Uso: Crear Producto

Actores: Usuario (Productor)

Propósito: Si se selecciona crear producto, llenar datos de control para producto nuevo

Descripción: El usuario introducirá datos de control del producto así como el contenido del mismo.

Escenario normal:

- 1: El usuario entra al Sistema.
- 2: El sistema presenta menú de operaciones
- 3: El usuario selecciona Crear Producto y determina el nivel de control (*Nivel Simple*) del producto
- 4: El usuario introduce la información necesaria definir los datos de control del producto.
- 5: El usuario pide al sistema guardar datos de control
- 6: El usuario introduce el contenido del producto
- 7: El usuario pide al sistema guardar el producto

Flujos excepcionales:

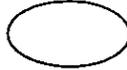
1. Los datos no cumplen reglas de tipo
- 2: El usuario cancela la operación

Precondición: No debe existir un producto con el mismo nombre.

Postcondición: Catálogo con producto guardado

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

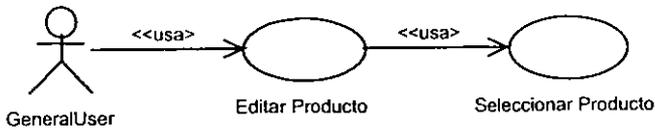
Documento de Análisis
Casos de uso y escenarios, Nivel Simple (Versión 0.8)
LLENAR DATOS DE CONTROL DEL PRODUCTO



Llenar datos de Control del Producto

Caso de Uso:	Llenar datos de control del producto
Actores:	Usuario (Productor)
Propósito:	Llenar datos de control que definen al producto
Descripción:	El usuario introducirá datos que definen al producto o los actualiza.
Escenario normal:	<p>Si el producto es nuevo:</p> <ol style="list-style-type: none">1: El sistema establece la versión 0.1, el estado de borrador y la fecha de creación.2: El sistema asigna el responsable de su producción.3: El usuario introduce la información necesaria para llenar datos de control del producto:<ul style="list-style-type: none">• Identificación del producto• Título• Fecha planeada de terminación del producto• Descripción general del contenido del producto4: El usuario revisa la información que capturó.5: El usuario pide al sistema guardar datos de control del producto.
Curso alterno:	<p>Si el producto ya existe:</p> <ol style="list-style-type: none">1: El sistema no establece la versión, ni el estado, ni la fecha de creación.2: El usuario actualiza datos de control, como:<ul style="list-style-type: none">• Estado• Fecha real de terminación de producto3: El usuario revisa la información que capturó.
Flujos excepcionales:	<ol style="list-style-type: none">1. Los datos no cumplen reglas de tipo2: El usuario cancela la operación
Precondición:	1:Exista catálogo de productos
Postcondición:	Datos de control del producto actualizados

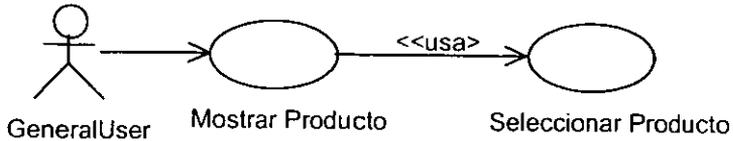
C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Casos de uso y escenarios, Nivel Simple (Versión 0.8) EDITAR PRODUCTO
---	--



- Caso de Uso:** Editar Producto
- Actores:** Usuario (Productor)
- Propósito:** Modificar el contenido del producto.
- Descripción:** El Usuario podrá editar el contenido de un Producto seleccionado, si es editable, y guardarlo.
- Escenario normal:**
- 1: El Usuario Selecciona el producto a editar.
 - 2: El sistema mostrará el contenido del producto y permitirá su edición.
 - 3: El Usuario editará el producto
 - 4: El Productor guarda el producto
- Flujos excepcionales:**
- 1: El Usuario cancela la edición del producto
 - 2: El producto no esta en estado editable.
 - 3: El Usuario no es el dueño del producto.
- Precondición:** Exista catálogo de productos y el producto esta en estado editable
- Postcondición:** Producto editado y guardado.

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Simple (Versión 0.8)
LEER PRODUCTO



Caso de Uso: Mostrar Producto

Actores: Usuario (Productor, Solicitante)

Propósito: Consultar el contenido del producto.

Descripción: El usuario podrá leer el contenido de un Producto seleccionado.

Escenario normal:

- 1: El Usuario Selecciona el producto a leer.
- 2: El sistema mostrará el contenido del producto.
- 3: El Usuario lee el contenido del producto.

Flujos excepcionales:

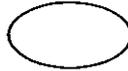
- 1: El usuario cancela la lectura del producto.
- 2: Si producto esta en estado editable no podrá mostrarse.

Precondición: Exista catálogo de productos y el producto está en estado no editable.

Postcondición: Lectura del contenido del Producto consultado.

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Simple (Versión 0.8)
SELECCIONAR PRODUCTO

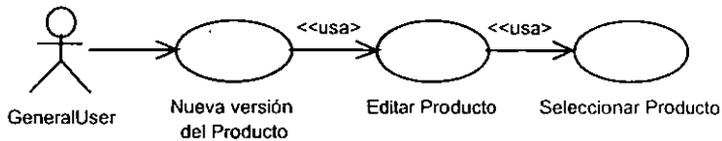


Seleccionar Producto

Caso de Uso:	Seleccionar Producto
Actores:	Usuario
Propósito:	Mostrar el conjunto de productos elaborados.
Descripción:	El usuario podrá seleccionar el Producto de una lista de productos que se mostraron de acuerdo a un Nivel de Control.
Escenario normal:	<ol style="list-style-type: none">1: El usuario selecciona el grupo de productos que pertenecen al nivel de Control deseado para obtener una lista de los mismos.2: El sistema muestra la lista de productos del grupo seleccionado.3: El Usuario selecciona un producto.
Flujos excepcionales:	<ol style="list-style-type: none">1: El sistema no muestra productos del grupo del nivel de control seleccionado, por no existir en el catálogo.
Precondición:	Exista catálogo de productos
Postcondición:	Mostrar lista de productos.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

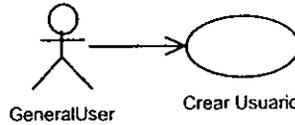
Documento de Análisis
 Casos de uso y escenarios, Nivel Simple (Versión 0.8)
MODIFICAR UN PRODUCTO



- Caso de Uso:** Nueva versión del Producto
- Actores:** Usuario (Productor)
- Propósito:** El usuario hará una nueva copia del producto.
- Descripción:** El usuario selecciona un producto y hará una copia del producto con estado de borrador con nueva versión y será editable.
- Escenario normal:**
- 1: El Usuario selecciona un producto.
 - 2: El usuario selecciona editar el producto
 - 3: El usuario selecciona nueva versión
 - 3: El sistema crea nueva versión del producto en estado editable incrementando el número de versión.
 - 4: El Usuario edita el producto.
- Flujos excepcionales:** Si el producto está en estado editable no podrá generarse una nueva versión
- Precondición:** Exista catálogo de productos.
- Postcondición:** Se crea un nuevo producto con una nueva versión y sus datos de control del producto son actualizados

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Simple (Versión 0.8)
CREAR USUARIO



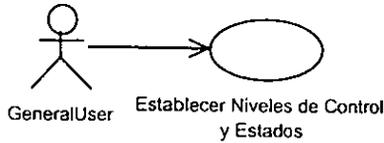
- Caso de Uso:** Crear Usuario
- Actores:** Usuario (Administrador)
- Propósito:** Dar de alta o de baja a un usuario o modificar sus datos.
- Descripción:** El Administrador podrá dar de alta, baja o modificar los datos de un usuario.
- Escenario normal:**
- 1: El Administrador entra al Sistema.
 - 2: El sistema presenta menú de operaciones.
 - 3: El Administrador selecciona la opción de Configuración
 - 4: El Administrador selecciona la Opción de Usuarios.
 - 5: Si se desea dar de alta un usuario, el Administrador introduce la información necesaria para llenar datos de control del usuario.
 - 6: El Administrador revisa la información que capturó
 - 7: El Administrador pide al sistema guardar la actualización.
- Curso alterno:**
- 4: Si se desea dar de baja o modificación de datos de un usuario, el Administrador introduce la información que lo identifica y selecciona eliminar o modificar.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo.
 - 2: El Administrador cancela la operación.
- Precondición:** Exista catálogo de usuarios
- Postcondición:** Se modifica el catálogo de usuarios.

C3PoS Versión 0.1

Prototipo para el Control de Calidad y Cambios en los Productos de Software

Documento de Análisis

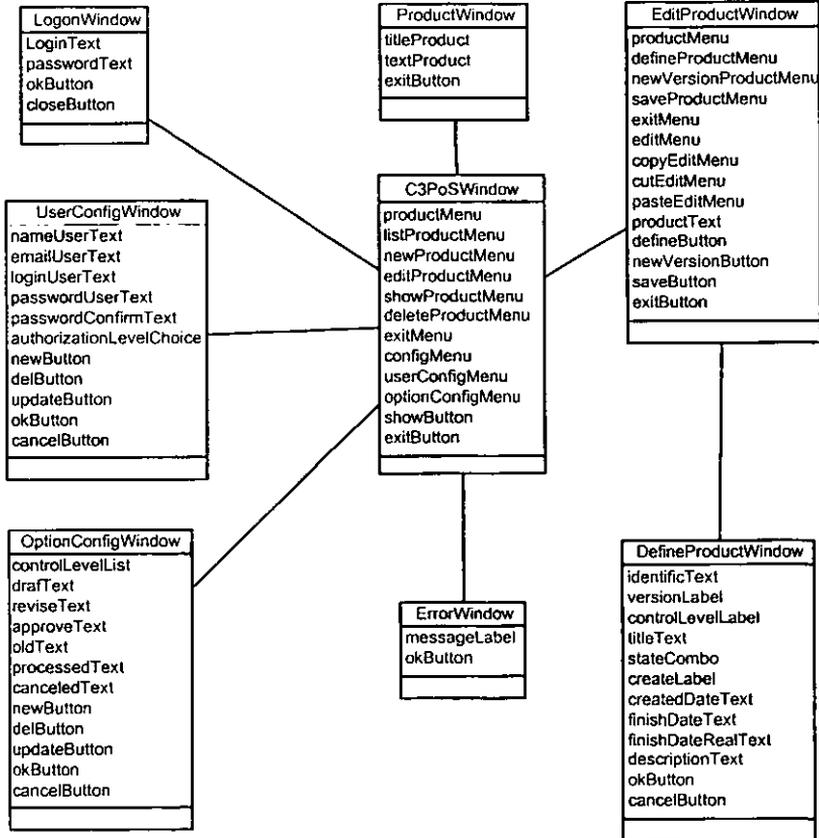
Casos de uso y escenarios, Nivel Simple (Versión 0.8)
ESTABLECER NIVELES DE CONTROL Y ESTADOS



- Caso de Uso:** Establecer Niveles de Control y Estados.
- Actores:** Usuario (Administrador)
- Propósito:** Dar de alta, baja o modificar información de los Niveles de Control y los nombres de los Estados para los productos.
- Descripción:** El Administrador podrá dar de alta o baja los niveles de control o los estados que se encontrará el producto con lo cual quedarán definidos en el sistema.
- Escenario normal:**
- 1: El Administrador entra al Sistema.
 - 2: El sistema presenta menú de operaciones
 - 3: El Administrador selecciona la opción de Configuración
 - 3: El Administrador selecciona la opción de Opciones.
 - 4: El Administrador podrá dar de alta los nombres y descripción de los diferentes niveles de control o en su caso podrá modificarlos o eliminarlos.
Podrá modificar los nombres asignados para los diferentes niveles de control.
 - 5: El Administrador revisa la información que capturó
 - 6: El Administrador pide al sistema guardar o cancelar las modificaciones.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo.
 - 2: El Administrador cancela la operación.
- Precondición:** Exista catálogo de opciones
- Postcondición:** Se modifica el catálogo de opciones.

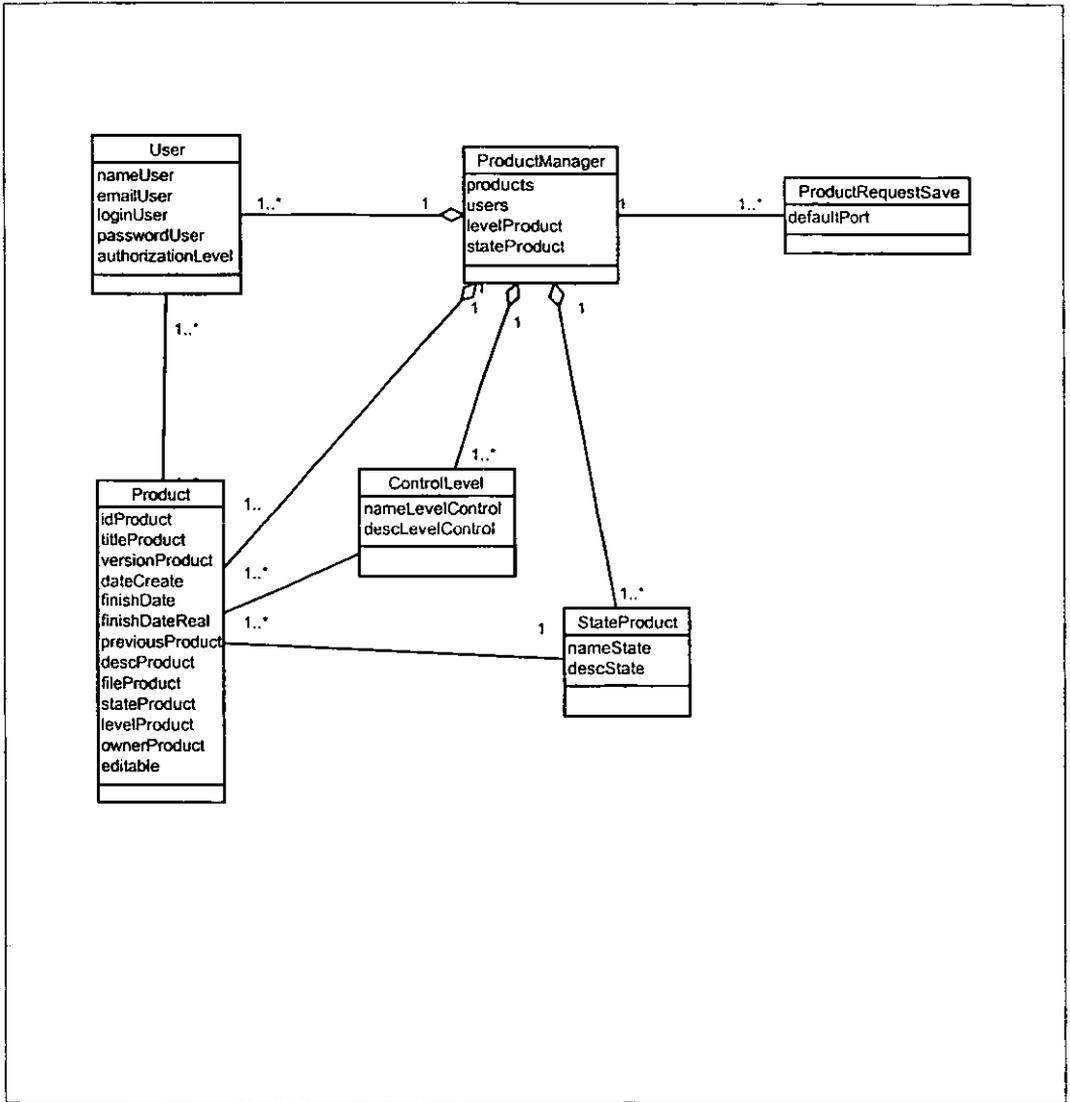
C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software
 Martín De Jesús Jiménez

Documento de Análisis
 Diagramas de Clases, Nivel Simple (Versión 0.8)
INTERFAZ HUMANA



C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software
 Martín De Jesús Jiménez

Documento de Análisis
 Diagramas de Clases, Nivel Simple (Versión 0.8)
DOMINIO DEL PROBLEMA



C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software
Martín De Jesús Jiménez

Documento de Análisis
Diagramas de Clases, Nivel Simple (Versión 0.8)
MANEJO DE DATOS

UserDB
nameUser
emailUser
loginUser
passwordUser
authorizationLevel

ControlLevelDB
idControlLevel
nameControlLevel

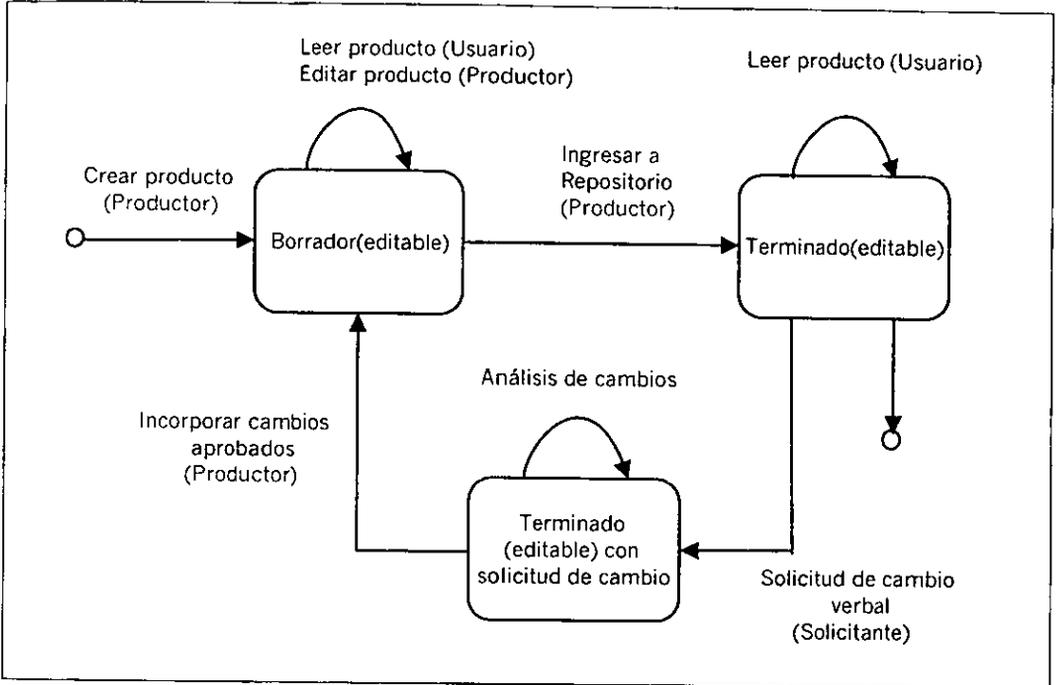
StateProductDB
nameState
descState

ProductDB
idProduct
titleProduct
versionProduct
dateCreate
finishDate
finishDateReal
idPreviousProduct
descProduct
fileProduct
idOwner
idLevel
idState
editable

ProductServerSave
defaultPort
port
listen_socket

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software
Martín De Jesús Jiménez

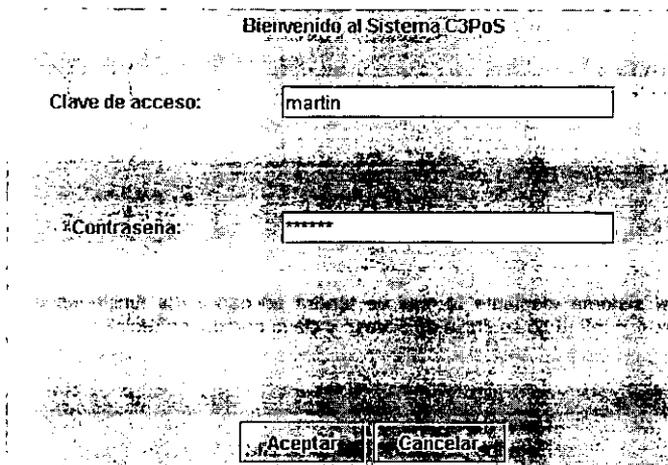
Documento de Análisis
Diagramas de estados, Nivel Simple (Versión 0.8)
PRODUCTO



C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Simple (Versión 0.8)
INTERFAZ HUMANA

LogonWindow



Es la ventana para escribir la clave de acceso y la clave secreta del usuario, con el fin de comprobar si tiene derecho de acceso al sistema

Nombre del atributo	Propiedad
loginText	Área de texto para introducir la clave de acceso del usuario
passwordText	Área de texto para introducir la clave secreta del usuario
okButton	Botón para verificar el acceso al sistema
cancelButton	Botón para abandonar la aplicación sin entrar al sistema

Nombre del método	Comportamiento
LogonWindow()	Crear la ventana
checkUser()	Comprobar los datos del usuario para verificar si tiene derecho de acceso al sistema
exitWindow()	Abandonar el sistema sin revisar los datos del usuario
login()	Inicializa la entrada del usuario

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) INTERFAZ HUMANA
---	--

C3PoSWindow

Producto Configuración		Lista de Productos			
Estado	Título	Identificador	Versión	Fecha	Nivel de Control
Borrador	Actividades del Nivel de Control Simple	C3PoS-NS	0.1	01/11/2001	Simple

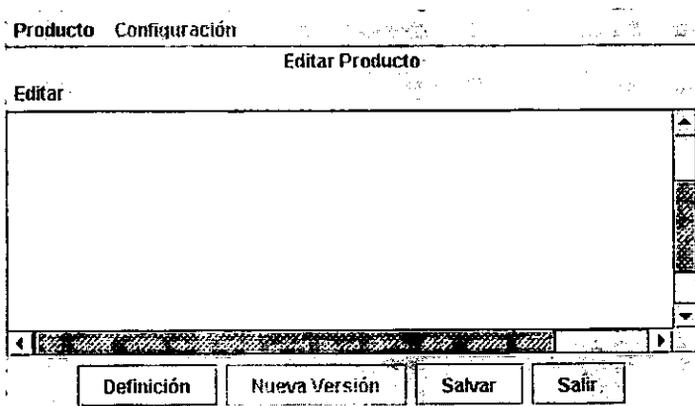
Ventana principal del sistema mediante la cual se tendrá acceso a las funciones del sistema. Se muestra después de que el usuario introdujo su clave de acceso y clave secreta correctos.

Nombre del atributo	Propiedad
productMenu	Menú del producto
listProductMenu	Lista los productos dados de alta
newProductMenu	Crea nuevo producto
editProductMenu	Edita Nuevo producto
showProductMenu	Muestra el contenido del producto
deleteProductMenu	Borra un producto seleccionado
exitMenu	Salir del sistema
viewMenu	Menú para ver la lista de documentos
configMenu	Menú de configuración del Sistema
userConfigMenu	Mantiene el catalogo de usuarios
optionConfigMenu	Mantiene las opciones de trabajo del sistema
showButton	Botón para mostrar un producto seleccionado
exitButton	Botón para salir del Sistema

Nombre del método	Comportamiento
C3PoSWindow()	Crear la ventana y mostrarla
ShowListProduct()	Muestra la lista de productos
newProduct()	Crear un nuevo Producto
editProduct()	Editar un producto que fue previamente seleccionado
showProduct()	Mostrar el contenido del Producto
deleteProduct()	Borrar un producto que fue previamente seleccionado
levelView()	Mostrar los productos del nivel seleccionado
userConfig()	Editar los datos de un usuario
optionConfig()	Editar la configuración del sistema
exitC3PoS()	Cerrar la ventana y salir del sistema
showProduct()	Mostrar el producto seleccionado

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) INTERFAZ HUMANA
---	--

EditProductWindow



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

Nombre del atributo	Propiedad
productMenu	Menu del Producto
defineProductMenu	Para definir las características del producto
newVersionProductMenu	Para crear una nueva versión del producto
saveProductMenu	Para guardar el producto después de haberlo editado
exitMenu	Para salirse de la Edición del producto
editMenu	Menú de edición del producto
copyEditMenu	Para copiar información del contenido del producto
cutEditMenu	Para borrar parte del contenido del producto
pasteEditMenu	Para pegar información al contenido del producto
productText	Para capturar el contenido de la información del producto
defineButton	Para definir las características del producto
newVersionButton	Para crear una nueva versión del producto
saveButton	Para guardar el producto después de haberlo editado
exitButton	Para salirse de la Edición del producto

Nombre del método	Comportamiento
EditDocumentWindow()	Crear la ventana y mostrarla si recibe un producto como parámetro, el contenido del documento para editarlo
defineDocument	Definir las características de un producto
saveProduct()	Salvar la información asociada al producto
newVersionProduct()	Crea una versión en borrador del producto
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

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) INTERFAZ HUMANA
---	--

DefineProductWindow

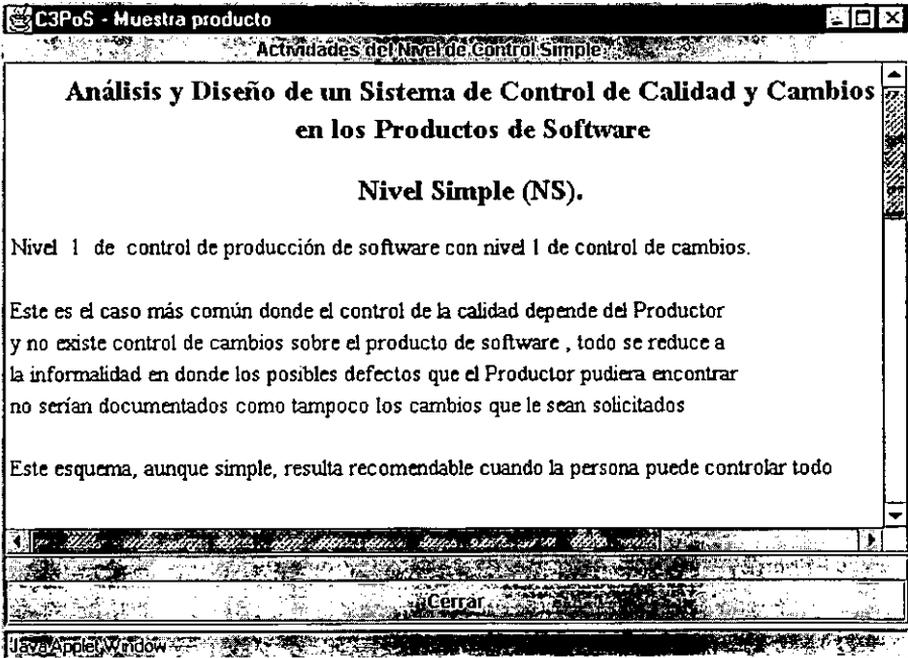
Ventana para definir las características de control del producto. Se muestra cuando se crea o se edita el producto.

Nombre del atributo	Propiedad
identificText	Identificador del producto de software
versionLabel	Versión del producto
titleText	Titulo del producto
controlLevelLabel	Nivel de control del producto
createLabel	Nombre del creador del producto
createDateText	Fecha de creación del producto
stateCombo	Estado del producto
finishDateText	Fecha planeada de terminación del producto
finishDateRealText	Fecha real de terminación del producto
descriptionText	Descripción general del contenido del producto
okButton	Guardar la información de definición del producto
cancelButton	Permite cerrar la ventana sin actualizar la definición del producto

Nombre del método	Comportamiento
DefineProductWindow()	Crear la ventana y mostrar, si recibe un producto como parámetro, las características del producto para editarlas.
defineProduct()	Definir las características de un producto
close()	Cerrar la ventana y regresar a la ventana de edición, sin definir el producto
setValue()	Inicializa los valores de los componentes
setActiveOwner	Establece las opciones de edición si el usuario es el propietario
setFirstTime	Establece las opciones de edición cuando el producto es creado

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) INTERFAZ HUMANA
---	--

ProductWindow



Ventana para mostrar el contenido del producto.

Nombre del atributo	Propiedad
titleProduct	Título del Producto que se Muestra
textProduct	Area de texto donde se muestra el contenido del Producto
exitButton	Para cerrar la ventana que muestra el producto

Nombre del método	Comportamiento
ProducWindow()	Crear la ventana y mostrarla
close()	Cerrar la ventana

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Simple (Versión 0.8)
INTERFAZ HUMANA

UserConfigWindow

Ventana para editar las características de un usuario.

Nombre del atributo	Propiedad
nameUserText	Nombre del usuario
emailUserText	Dirección de correo electrónico
loginUserText	Clave de acceso del usuario
passwordUserText	Clave secreta de acceso del usuario
passwordConfirmText	Confirmación de clave secreta de acceso del usuario
authorizationLevelChoice	Nivel de autorización
newButton	Para crear un nuevo usuario
delButton	Para borrar un usuario que existe en el sistema
updateButton	Actualizar los datos de un usuario
okButton	Guardar la información del usuario y cerrar la ventana
cancelButton	Cerrar la ventana sin actualizar la información del usuario

Nombre del método	Comportamiento
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

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) INTERFAZ HUMANA
--	--

OptionConfigWindow

The screenshot shows a window titled "Configuración de Opciones". On the left, there is a list box labeled "Niveles de Control" containing the items "Simple", "Intermedio", and "Completo". Below this list are three buttons: "Nuevo", "Borrar", and "Modificar". On the right side of the window, there is a vertical list of state options: "Borrador", "Revisado", "Aprobado", "Terminado", "Solicitado", "Cumplido", "Reportado", "Aceptado", "Rechazado", "Corregido", "Procesado", and "Obsoleto". At the bottom center of the window are two buttons: "Aceptar" and "Cancelar".

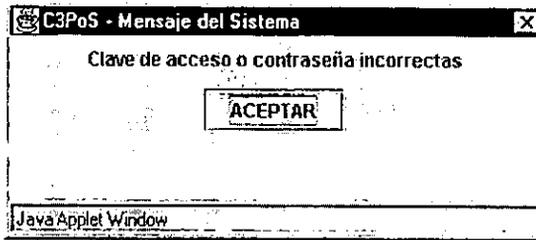
Ventana para establecer los niveles de control y estados del sistema.

Nombre del atributo	Propiedad
ControlLevellist	Lista con los diferentes niveles de control del producto de software
draftText	Definición del estado de borrador
reviseText	Definición del estado de revisión
approveText	Definición del estado de aprobación
oldText	Definición del estado de obsoleto
processedText	Definición del estado de implantado
canceledText	Definición del estado de rechazado
newButton	Para crear un nuevo nivel de control
delButton	Para borrar un nuevo nivel de control
updateButton	Para modificar un nivel seleccionado
okButton	Para guardar la información de configuración y cerrar la ventana
cancelButton	Para cerrar la ventana sin actualizar la información

Nombre del método	Comportamiento
OptionConfigWindow()	Crear la ventana y mostrar las opciones
newLevel()	Crea un nuevo nivel de productos
delLevel()	Borrar el nivel de productos seleccionados
updateLevel()	Actualizar el nivel de productos seleccionado
setOption()	Establecer las nuevas opciones y cerrar la ventana
close()	Cerrar la ventana sin establecer las opciones y regresar a la ventana principal

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) INTERFAZ HUMANA
---	--

ErrorWindow



Ventana que muestra los mensajes del sistema para que el usuario atienda una falla del sistema o confirme una acción (este mensaje corresponde al caso en que un usuario introdujo sus claves de acceso en forma incorrecta)

Nombre del atributo	Propiedad
messageLabel	Muestra el contenido del mensaje al usuario
okButton	Botón para aceptar el mensaje

Nombre del método	Comportamiento
ErrorWindow()	Crea la ventana

Documento de análisis para productos con nivel de control simple

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

Product. Es el producto con todas los elementos que lo definen.

Nombre del atributo	Propiedad
idProduct	Texto con el identificador único del documento, establecido por el productor
titleProduct	Texto con el título del documento, establecido por el productor
versionProduct	Número de versión del producto, establecida por el sistema
dateCreate	Fecha en que se crea el producto, establecido por el productor
finishDate	Fecha planeada de terminación del producto
finishDateReal	Fecha real de terminación del producto
previousProduct	Objeto producto qprobado con anterioridad, establecido 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.
descProduct	Texto con la descripción del contenido del producto con carácter de información general establecida por el usuario.
fileProduct	Texto con la referencia al archivo que contiene el texto del producto, establecida por el sistema
stateProduct	Texto con la descripción del estado del producto, establecido por el Productor
levelProduct	Texto con el identificador del nivel de control del producto, establecido por el productor
ownerProduct	Objeto usuario que crea el producto, establecido por el sistema
editable	Información que indica si el documento es modificable (0) o si solo se permite la lectura (1)

Nombre del método	Comportamiento
Product()	Crear el objeto producto
setText()	Escribir la referencia al archivo con el contenido del producto
setId()	Escribir el identificador del producto
setTitle()	Escribir el título del producto
setDateCreate()	Escribir la fecha de creación
setLevel()	Escribir el nivel de control al que pertenece el producto
setDesc()	Escribir la descripción del producto
setEditable()	Escribir si el documento es modificable
setState()	Escribir el estado del producto
setFinishDate()	Escribir la fecha en que se planeo terminar el producto
setFinishDateReal()	Escribir la fecha real de terminación del producto
setVersion()	Escribir la versión del producto
replicate()	Replicar el producto para crear un borrador con una nueva versión
getLevel	Leer el nivel del producto
getTitle	Leer el título del producto
getId	Leer el identificador del producto
getVersion	Leer la versión del producto
getDate	Leer las fecha del producto
getState	Leer el estado del producto
getDesc	Leer la descripción del producto
getEditable()	Leer si el producto es modificable
getPrevious()	Leer el producto aprobado anteriormente
getOwner()	Leer el usuario que creó el producto

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

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

Nombre del atributo	Propiedad
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	Nivel de autorización, 0 si el usuario es el administrador y 1 si es otro usuario

Nombre del método	Comportamiento
User()	Crear el objeto usuario
setLogin()	Escribir la clave de acceso 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

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

ControlLevel. Contiene información acerca del nivel de control del producto. Un producto solo puede pertenecer a un nivel de control y una vez definido, no será posible cambiarlo.

Nombre del atributo	Propiedad
nameLevel	Texto con el nombre del nivel, establecido por el administrador

Nombre del método	Comportamiento
LvelProduct()	Crear el objeto nivel de control par el producto
setName()	Establece el nombre del estado
setDesc()	Establece el nombre descriptivo del estado
getName()	Obtiene el nombre del estado
getDesc()	Obtiene el nombre descriptivo del estado

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

StateProduct. Contiene los estados que puede tener un producto o una solicitud de cambio.

Nombre del atributo	Propiedad
nameState	Texto con el nombre del estado, establecido por el sistema
descState	Texto con el nombre del estado para el usuario, establecido por el administrador

Nombre del método	Comportamiento
StateProduct()	Crear el objeto estado de los productos
setName()	Establece el nombre del nivel de control de los productos
setId()	Establece el identificador del nivel de control de los productos
getName()	Obtiene el nombre del nivel de control de los productos
getId()	Obtiene el identificador del nivel de control de los productos

Documento de análisis para productos con nivel de control simple

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

ProductRequestSave. Para grabar el producto en el repositorio compartido

Nombre del atributo	Propiedad
defaultPort	Puerto de comunicación
file	Nombre del archivo
operation	Tipo de operación
textdocument	Contenido del documento

Nombre del método	Comportamiento
ProductRequestSave()	Crea el objeto

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

ProductManager. Administra los productos del sistema, los niveles establecidos, los estados posibles y los usuarios.

Nombre del atributo	Propiedad
products	Objetos producto creados en el sistema
users	Objetos usuario con acceso al sistema
levelProduct	Objetos nivel establecidos en el sistema
stateProduct	Objetos estado establecidos en el sistema

Nombre del método	Comportamiento
ProductManager()	Crear el objeto
checkUser()	Revisar si existe el usuario
saveProduct()	Guardar la información del producto
getProductData()	Leer la información de los productos del sistema
getProduct()	Leer el producto indicado
setUser()	Escribir la información del usuario
SetControlLevel	Escribir la información del nivel de control de los productos
setSateProduct()	Escribir la información del estado de los productos

Documento de análisis para productos con nivel de control simple

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) MANEJO DE DATOS
---	--

ProductDB. Contiene la información que se necesita para guardar cada producto.

Nombre del atributo	Propiedad
idProduct	Identificador del producto de software
titleProduct	Título del producto
versionProduct	Versión del producto
dateCreate	Fecha planeada de terminación del producto
finishDate	Fecha real de terminación del producto
finishDateReal	Fecha de creación del producto
idPreviousProduct	Identificador del producto aprobado anteriormente
descProduct	Descripción general del contenido del producto
fileProduct	Texto con la referencia al archivo que contiene el texto del producto, establecido por el sistema
idOwner	Identificador del usuario que es el creador del producto
idLevel	Identificador del nivel de control al que pertenece el producto
idstate	Identificador del estado en el que se encuentra el producto
editable	Información que indica si el documento es modificable (0) o si sólo se permite su lectura (1).

Nombre del método	Comportamiento
ProductDB	Crea el objeto
saveProduct()	Guarda información del producto en la Base de Datos
loadProduct()	Trae información del producto en la Base de Datos
deleteProduct()	Borra información del producto en la Base de Datos
updateProduct()	Actualiza información del producto en la Base de Datos

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) MANEJO DE DATOS
---	--

UserDB. Contiene la información de los Usuarios

Nombre del atributo	Propiedad
nameUser	Nombre del usuario
emailUser	Dirección de correo electrónico
loginUser	Clave de acceso del usuario
passwordUser	Clave secreta de acceso del usuario
authorizationLevel	Nivel de autorización

Nombre del método	Comportamiento
UserDB()	Crea el objeto usuario
saveUser()	Crear un nuevo usuario del sistema
loadUser()	Carga información del usuario del sistema
deleteUser()	Borra información del usuario
updateUser()	Actualiza información del usuario

Documento de análisis para productos con nivel de control simple

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Simple (Versión 0.8)
MANEJO DE DATOS

ControlLevelDB. Contiene información de los niveles de control del producto. Un producto sólo puede tener un nivel de control.

Nombre del atributo	Propiedad
idControlLevel	Identificador del nivel de control
nameControlLevel	Nombre del nivel de control

Nombre del método	Comportamiento
ControlLevelDB()	Crea el Objeto
saveControlLevel()	Guarda información de los niveles de control en la Base de Datos
loadControlLevel()	Trae información de los niveles de control en la Base de Datos
deleteControlLevel()	Borra información de los niveles de control en la Base de Datos
updateControlLevel()	Actualiza información de los niveles de control en la Base de Datos

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Simple (Versión 0.8) MANEJO DE DATOS
--	--

StateProductDB. Contiene la información de los estados de los productos.

Nombre del atributo	Propiedad
nameState descState	Texto con el nombre del estado, establecido por el sistema Texto con el nombre del estado para el usuario, establecido por el administrador

Nombre del método	Comportamiento
StateDocDB() saveState() loadState() deleteState() UpdateState()	Crear el objeto Guarda información de los estados en la Base de Datos Trae información de los estados en la Base de Datos Borra información de los estados en la Base de Datos Actualiza información de los estados en la Base de Datos

Documento de análisis para productos con nivel de control simple

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Simple (Versión 0.8)
MANEJO DE DATOS

ProductServerSave. Controla el almacenamiento del producto en el repositorio compartido.

Nombre del atributo	Propiedad
defaultPort	El puerto de comunicación que se utilizará para enviar el mensaje
dataInputStream	Información recibida que se guardará
socket	El tipo de conexión realizada
operationType	Tipo de operación que se realiza

Nombre del método	Comportamiento
ProductServerSave()	Crea el objeto
Read	Lee información del contenido del producto
Write()	Escribe información del contenido del producto

ANEXO C

Documento de análisis para productos con nivel de control intermedio

Este documento contiene información obtenida durante la fase de análisis del prototipo para el control de productos y lo componen:

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

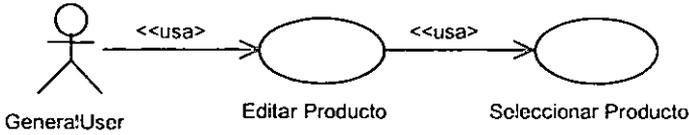
C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
LLENAR DATOS DE CONTROL DEL PRODUCTO

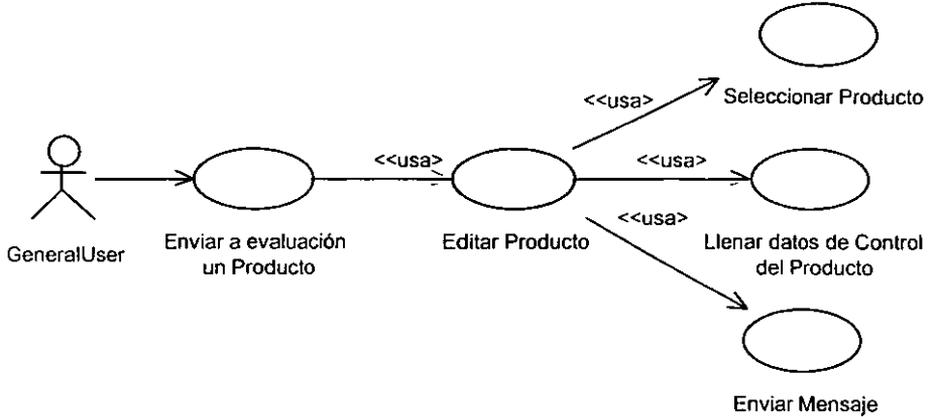


Llenar datos de control del
producto

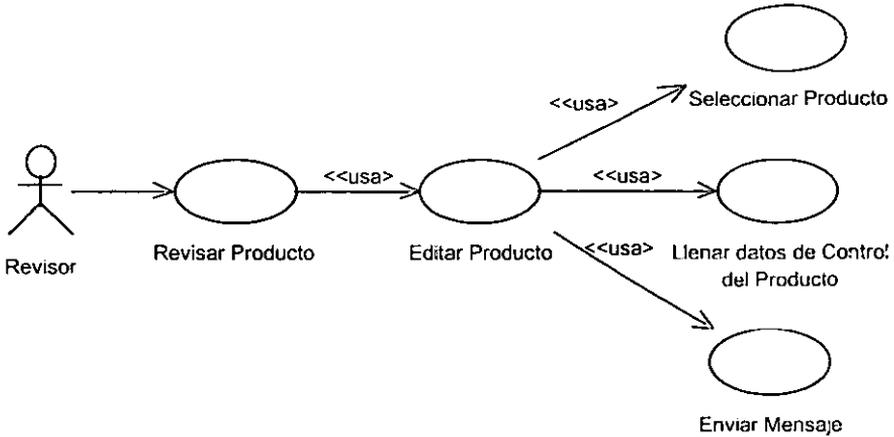
- Caso de Uso:** Llenar datos de control del producto
- Actores:** Usuario (Productor, Revisor)
- Propósito:** Llenar datos de control que definen al producto
- Descripción:** El Usuario introducirá datos que definen al producto o los actualiza.
- Escenario normal:** Si el producto es nuevo:
- 1: El sistema establece la versión 0.1, el estado de borrador y la fecha de creación.
 - 2: El sistema asigna el responsable de su producción.
 - 3: El usuario Productor introduce la información necesaria para llenar datos de control del producto, incluye los mismo datos solicitados en el caso de uso del nivel de control Simple además de:
 - Nombre del Revisor
 - Fecha Planeada de Revisión
 - 4: El usuario Productor revisa la información que capturó.
 - 5: El usuario pide al sistema guardar datos de control del producto.
- Curso alterno:** Si el producto ya existe:
- 1: El sistema no establece la versión, ni el estado, ni la fecha de creación.
 - 2: El usuario actualiza datos de control, como:
 - Fecha real de revisión del producto
 - 3: El usuario Productor revisa la información que capturó.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo
 - 2: El usuario cancela la operación
- Precondición:** Exista catálogo de productos
- Postcondición:** Datos de control del producto actualizados



- Caso de Uso:** Editar Producto
- Actores:** Usuario (Productor)
- Propósito:** Modificar el contenido del producto
- Descripción:** El Usuario podrá editar el contenido de un Producto seleccionado, si es editable, y guardarlo.
- Escenario normal:**
- 1: El Usuario Selecciona el producto a editar.
 - 2: El sistema mostrará el contenido del producto y permitirá su edición.
 - 3: El Usuario editará el producto
 - 4: El Usuario guarda el producto
 - 5: El Usuario podrá enviar un mensaje a las demás personas involucradas en el desarrollo del producto
- Flujos excepcionales:**
- 1: El Usuario cancela la edición del producto
 - 2: El producto no está en estado editable.
 - 3: El Usuario no es el dueño del producto.
- Precondición:** Exista catálogo de productos y el producto está en estado editable
- Postcondición:** Producto editado y guardado.



- Caso de Uso:** Enviar a evaluación un producto
- Atores:** Usuario (Productor)
- Propósito:** El Productor envía al Revisor el producto para su evaluación.
- Descripción:** El Productor seleccionará un producto, cambia el estado del producto a no editable y lo envía a un Revisor para su evaluación.
- Escenario normal:**
- 1: El Usuario selecciona un producto en borrador y editable.
 - 2: El Usuario entra a la opción de edición del producto
 - 3: Usuario selecciona definir producto
 - 4 El sistema mostrará datos de control del producto.
 - 5 El Usuario cambiará el producto a no editable.
 - 6 El Usuario guarda datos de control del producto.
 - 7 El Usuario envía el producto al Revisor avisándole con un mensaje
- Flujos excepcionales:**
- 1: El producto esta en estado no editable.
 - 2: El Usuario cancela el envío del producto.
 - 3: El Usuario no es el dueño del producto.
- Precondición:** Exista catálogo de productos y el producto esta en estado editable.
- Postcondición:** El producto cambia a estado no editable.



Caso de Uso: Revisar un Producto

Actores: Usuario (Revisor)

Propósito: El Revisor selecciona un producto, así como los requisitos que deberá cumplir el producto para su evaluación.

Descripción: El Revisor seleccionará un producto, cambiará el estado del producto a revisado.

Escenario normal:

- 1: El Revisor Selecciona un producto en estado de borrador y no editable.
- 2: El sistema mostrará datos de control del producto
- 3: El Revisor consultará los requisitos del producto
- 4: El Revisor con ayuda de los requisitos revisará el producto
- 5: El Revisor actualiza datos de control del producto
- 6: El Revisor evalúa el producto y cambia el estado del producto a revisado si cumple con los requisitos, ingresa el producto a un repositorio común y lo notifica al productor con un mensaje

Flujos excepcionales:

- 1: Si el producto es editable no podrá realizar esta operación.
- 2: El usuario cancela la revisión del producto

Precondición:

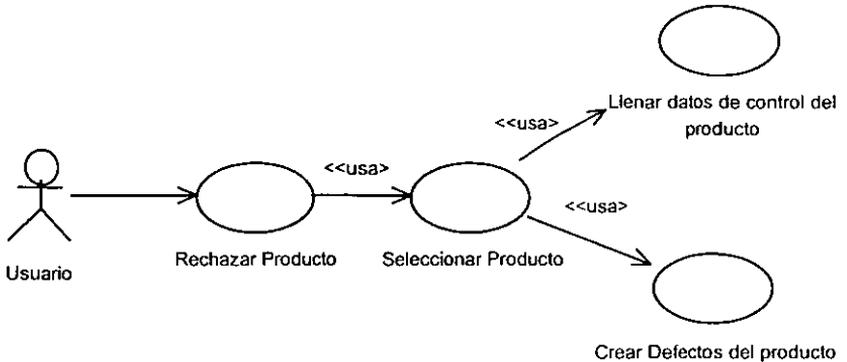
Exista catálogo de productos y el producto está en estado de borrador y es no editable.

Postcondición:

Los datos de control son actualizados y guardados.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

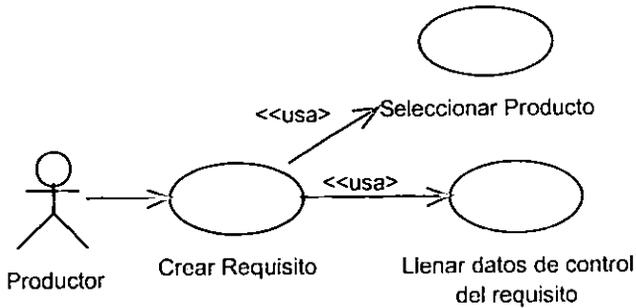
Documento de Análisis
 Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
RECHAZAR PRODUCTO



- Caso de Uso:** Rechazar un Producto
- Actores:** Usuario (Revisor).
- Propósito:** El usuario selecciona un producto y modificará datos de control y notificará al productor.
- Descripción:** El Usuario selecciona un producto, cambia el estado del producto y notificará al productor de los defectos encontrados.
- Escenario normal:**
- 1: El Revisor Selecciona un Producto en estado de borrador y no editable.
 - 2: El sistema mostrará datos de control del producto.
 - 3: El Revisor evalúa el producto como no aceptado si no cumple con los requisitos.
 - 4: El Revisor guarda datos de control del producto.
 - 5: El Revisor avisa al Productor de su rechazo y de los defectos encontrados con un mensaje.
 - 6: El Revisor crea un reporte de los defectos encontrados.
- Flujos excepcionales:**
- 1: Si el producto es editable no podrá realizar esta operación.
- Precondición:** Exista catálogo de productos y el producto está en estado de borrador y es no editable.
- Postcondición:** Los datos de control del producto son actualizados y guardados.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
CREAR REQUISITO



- Caso de Uso:** Crear Requisito
- Actores:** Usuario (Productor)
- Propósito:** Crear un requisito con el que debe cumplir un producto
- Descripción:** El Productor introducirá datos de control que corresponden al requisito que debe cumplir el producto cuando éste ha sido creado
- Escenario normal:**
- 1: El Productor selecciona un producto
 - 2: El Productor selecciona crear requisito.
 - 3: El Revisor introduce la información que corresponden a datos de control del Requisito:
 - 4: El Productor introduce la descripción del requisito solicitado
 - 5: El Productor pide al sistema guardar el requisito
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo.
 - 2: El Usuario cancela la operación.
 - 3: Al producto no se le puede asociar un requisito por no pertenecer a este nivel de control.
- Precondición:** Exista catálogo de productos que permitan la definición del requisito.
- Postcondición:** El requisito es guardado en el Catálogo de requisitos

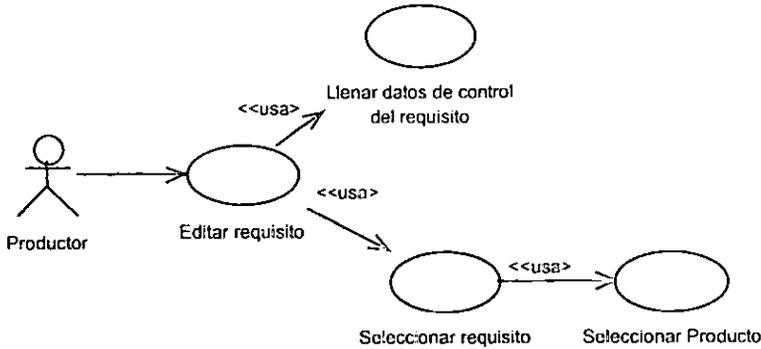
C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
**LLENAR DATOS DE CONTROL DEL REPORTE DE
DEFECTOS**



Llenar datos de control
del requisito

- Caso de Uso:** Llenar datos de control del requisito
- Actores:** Usuario (Productor)
- Propósito:** Llenar datos de control que definen al requisito
- Descripción:** El Productor introducirá datos que definen al requisito o los actualiza.
- Escenario normal:** Si el requisito es nuevo:
- 1: El sistema establece su identificador, nombre del producto a quien se le asociará el requisito , quien lo crea y la fecha de creación.
 - 2: El Revisor introduce la información necesaria para llenar datos de control del requisito:
 - Título
 - Nombre del producto como requisito con el que debe cumplir el producto
 - Descripción general del requisito
 - 3: El Productor revisa la información que capturó.
 - 4: El Productor pide al sistema guardar datos de control del requisito.
- Curso alterno:** El Productor podrá modificar el estado del defecto cuando haya sido terminada y modificar la descripción del mismo.
- Flujos excepcionales:**
- 1: El usuario cancela la operación
- Precondición:** 1:Exista catálogo de productos
- Postcondición:** Datos de control actualizados del requisito.



Caso de Uso:	Editar requisito
Actores:	Usuario (Productor)
Propósito:	Editar datos de control del requisito
Descripción:	El Productor modificará datos que definen al requisito
Escenario normal:	<ol style="list-style-type: none"> 1:El usuario selecciona un producto 2:El usuario selecciona la opción de edición del defecto 3:El usuario podrá modificar datos de control como: <ul style="list-style-type: none"> • Estado • Descripción del requisito 4: El Usuario revisa la información que capturó. 5: El Usuario pide al sistema guardar datos de control del requisito.
Flujos excepcionales:	<ol style="list-style-type: none"> 1. Los datos no cumplen reglas de tipo 2: El usuario cancela la operación
Precondición:	Exista lista de requisitos para ese producto
Postcondición:	Datos de control del requisito actualizados

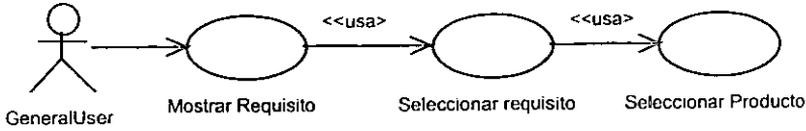
C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
SELECCIONAR REQUISITO



Seleccionar requisito

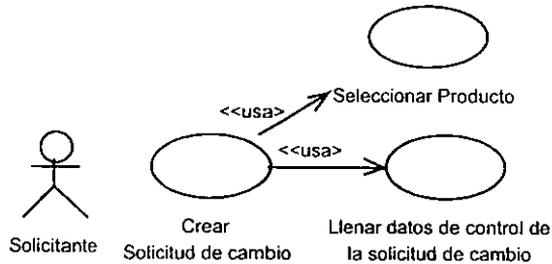
- Caso de Uso:** Seleccionar requisito
- Actores:** Usuario (Revisor, Productor)
- Propósito:** Seleccionar lista de reportes de defectos asociados a un producto
- Descripción:** El usuario podrá seleccionar un requisito asociado a un producto
- Escenario normal:**
- 1:El Usuario selecciona un producto
 - 2:El Usuario solicita al sistema le sean mostradas los requisitos asociados a ese producto.
 - 3: El sistema muestra la lista de requisitos asociados a un producto.
 - 4: El Usuario selecciona un requisito.
- Flujos excepcionales:**
- 1:En caso de no existir defectos asociados al producto, el sistema envía un mensaje notificando
 - 2:El Usuario cancela la lista de requisitos mostrados.
- Precondición:** Exista lista de requisito asociado al producto.
- Postcondición:** Requisito seleccionado



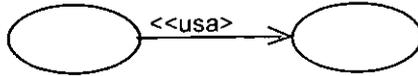
Caso de Uso	Mostrar requisito
Actores:	Usuario
Propósito:	Mostrar las características e información del requisito.
Descripción:	El usuario podrá leer el contenido de un requisito asociado a un producto
Escenario normal:	<ol style="list-style-type: none">1:El usuario selecciona un Requisito2: El sistema mostrará el contenido del Requisito3: El usuario lee el contenido del Requisito
Flujos excepcionales:	<ol style="list-style-type: none">1:El usuario cancela la lectura del requisito.
Precondición:	Exista catálogo de productos y requisitos.
Postcondición	Requisito consultado.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
CREAR SOLICITUD DE CAMBIO



- Caso de Uso:** Crear solicitud de cambio
- Actores:** Usuario (Solicitante)
- Propósito:** Crear solicitud de cambio a un producto.
- Descripción:** El Solicitante introducirá datos de control que corresponden a la solicitud de cambio a un producto cuando desea que sea modificado.
- Escenario normal:**
- 1: El Solicitante selecciona un producto
 - 2: El Solicitante selecciona Crear Cambio.
 - 3: El Solicitante introduce la información que corresponden a datos de control del Solicitud de cambios.
 - 4: El Solicitante introduce la descripción del cambio solicitado.
 - 5: El Solicitante pide al sistema guardar el solicitud de cambios.
 - 6: El Solicitante notifica de la solicitud al productor mediante un mensaje por correo electrónico.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo.
 2. El Usuario cancela la operación.
 - 3: Al producto no pueden generarse solicitud de cambios por no pertenecer a este nivel de control.
- Precondición:** Exista catálogo de productos que permitan solicitud de cambios.
- Postcondición:** El solicitud de cambios es guardado en el Catálogo de solicitudes de cambios y el productor recibe notificación de la misma.

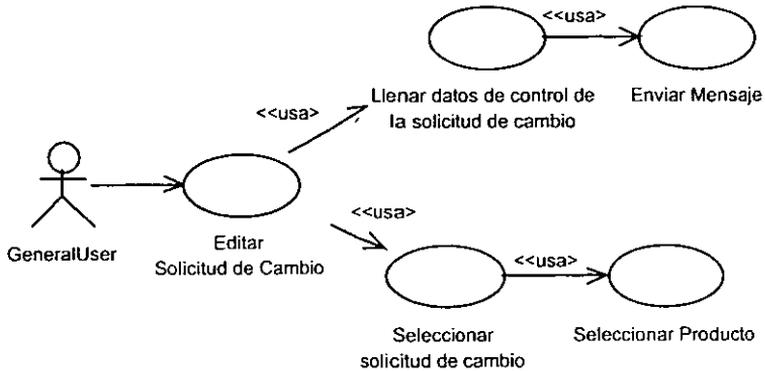


Llenar datos de control de la solicitud de cambio Enviar Mensaje

- Caso de Uso:** Llenar datos de control de la solicitud de cambio
- Actores:** Usuario (Solicitante, Productor)
- Propósito:** Llenar datos de control que definen la solicitud de cambios
- Descripción:** El usuario introducirá datos que definen la solicitud de cambios o los actualiza
- Escenario normal:** Si el solicitud de cambios es nuevo.
- 1: El sistema establece su identificador, nombre del producto a quien se le reportará el cambio, quien lo crea y la fecha de creación.
 - 2: El Solicitante introduce la información necesaria para llenar datos de control del solicitud de cambios:
 - Título
 - Descripción de la solicitud de cambios
 - 3: El Usuario revisa la información que capturó.
 - 4: El Usuario pide al sistema guardar datos de control del solicitud de cambios y envía un mensaje al Productor.
- Curso alterno:** Si el usuario es el Productor podrá modificar el estado del cambio y modificar la descripción del mismo.
- Flujos excepcionales:**
- 1: El usuario cancela la operación
- Precondición:** Exista catálogo de productos
- Postcondición:** Datos de control de la solicitud de cambios actualizados

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
EDITAR SOLICITUD DE CAMBIO



- Caso de Uso:** Editar solicitud de cambio
- Actores:** Usuario (Productor)
- Propósito:** Editar datos de control de la solicitud de cambios
- Descripción:** El Usuario modificará datos que definen la solicitud de cambios.
- Escenario normal:**
- 1:El Usuario selecciona un producto
 - 2:El Usuario selecciona la opción de edición de solicitud de cambios
 - 3:El usuario podrá modificar datos de control como:
 - Estado
 - Descripción de la solicitud
 - 4: El Usuario revisa la información que capturó.
 - 5: El Usuario pide al sistema guardar datos de control del solicitud de cambios y si es el Solicitante envía un mensaje al Productor o viceversa.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo
 - 2: El usuario cancela la operación
- Precondición:** Exista lista de solicitud de cambios para ese producto.
- Postcondición:** Datos de control de la solicitud de cambios actualizados

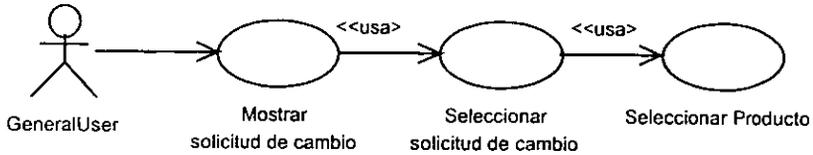


Seleccionar
solicitud de cambio

- Caso de Uso:** Seleccionar solicitud de cambio
- Actores:** Usuario
- Propósito:** Seleccionar lista de solicitud de cambio asociados a un producto
- Descripción:** El usuario podrá seleccionar una solicitud de cambio a un producto
- Escenario normal:**
- 1: El Usuario selecciona un producto
 - 2: El Usuario solicita al sistema le sean mostradas las solicitudes de cambio asociadas a ese producto.
 - 3: El sistema muestra las solicitudes de cambio asociadas a un producto
 - 4: El Usuario selecciona una solicitud de cambio.
- Flujos excepcionales:**
- 1: En caso de no existir defectos asociados al producto, el sistema envía un mensaje notificando
 - 2: El Usuario cancela la lista defectos mostrados.
- Precondición:** Exista lista de solicitudes de cambio asociadas al producto.
- Postcondición:** Solicitudes de cambio seleccionada.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
MOSTRAR SOLICITUD DE CAMBIO

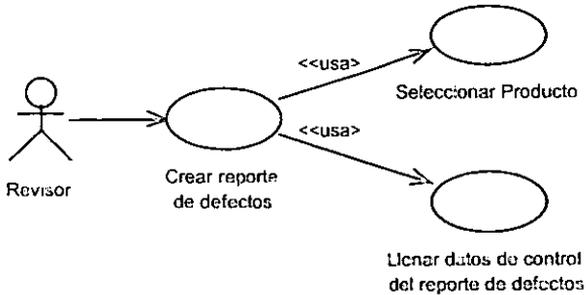


- Caso de Uso:** Mostrar solicitud de cambio
- Actores:** Usuario
- Propósito:** Mostrar las características y contenido de una solicitud de cambio.
- Descripción:** El usuario podrá leer el contenido de una solicitud de cambio asociada a un producto
- Escenario normal:**
- 1: El usuario selecciona una solicitud de cambio
 - 2: El sistema mostrará el contenido del solicitud de cambio
 - 3: El usuario lee el contenido de la solicitud de cambio
- Flujos excepcionales:**
- 1: El usuario cancela la lectura de l solicitud de cambio.
- Precondición:** Exista catálogo de productos y solicitudes de cambio.
- Postcondición:** Solicitud de cambio consultada.

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

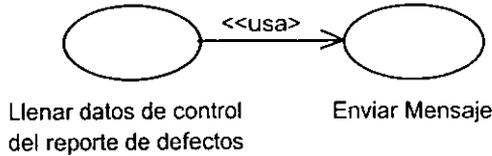
Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
CREAR SOLICITUD DE CAMBIO



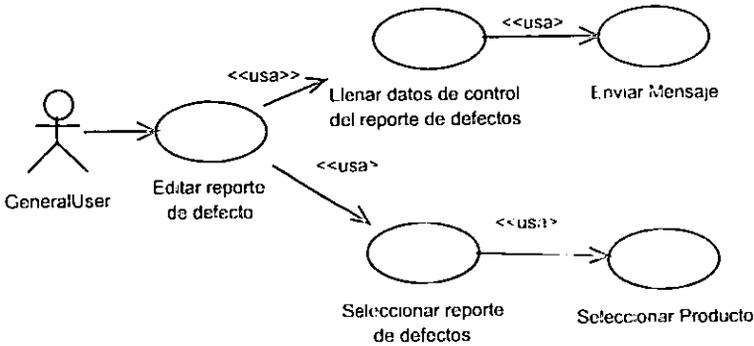
Caso de Uso	Crear Reporte de Defectos
Actores:	Usuario (Revisor)
Propósito:	Crear un reporte de los defectos encontrados a un producto después de ser revisado.
Descripción:	El Revisor introducirá datos de control que corresponden a los defectos encontrados a un producto cuando ha sido evaluado.
Escenario normal:	<ol style="list-style-type: none">1: El Revisor selecciona un producto2: El Revisor selecciona Crear Defecto.3: El Revisor introduce la información que corresponden a datos de control del Reporte de Defectos.4. El Revisor introduce la descripción de los defectos encontrados.5: El Revisor pide al sistema guardar el reporte de defectos.6: El Revisor notifica los defectos encontrados al productor mediante un mensaje por correo electrónico.
Flujos excepcionales:	<ol style="list-style-type: none">1. Los datos no cumplen reglas de tipo.2: El Usuario cancela la operación3 Al producto no pueden generarse reporte de defectos por no pertenecer a este nivel de control.
Precondición:	Exista catálogo de productos que permitan reporte de defectos.
Postcondición:	El reporte de defectos es guardado en el Catálogo de Defectos y el productor recibe notificación de los mismos.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
LLENAR DATOS DE CONTROL DEL REPORTE DE DEFECTOS



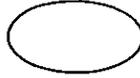
- Caso de Uso:** Llenar datos de control del reporte de defectos
- Actores:** Usuario (Revisor, Productor)
- Propósito:** Llenar datos de control que definen al reporte de defectos
- Descripción:** El Revisor introducirá datos que definen al reporte de defectos o los actualiza.
- Escenario normal:** Si el reporte de defectos es nuevo:
- 1: El sistema establece su identificador, nombre del producto a quien se le reportará el defecto, quien lo crea y la fecha de creación.
 - 2: El Revisor introduce la información necesaria para llenar datos de control del reporte de defectos:
 - Título
 - Descripción del reporte de defectos
 - 3: El Revisor revisa la información que capturó.
 - 4: El Revisor pide al sistema guardar datos de control del reporte de defectos y envía un mensaje al Productor.
- Curso alterno:** Si el usuario es el Productor podrá modificar el estado del defecto y modificar la descripción del mismo.
- Flujos excepcionales:**
- 1: El usuario cancela la operación
- Precondición:** Exista catálogo de productos
- Postcondición:** Datos de control del reporte de defectos actualizados



- Caso de Uso:** Editar reporte de defectos
- Actores:** Usuario (Revisor, Productor)
- Propósito:** Editar datos de control del reporte de defectos
- Descripción:** El Revisor modificará datos que definen al reporte de defectos
- Escenario normal:**
- 1: El usuario selecciona un producto
 - 2: El usuario selecciona la opción de edición del defecto
 - 3: El usuario podrá modificar datos de control como
 - Estado
 - Descripción del defecto
 - 4: El Usuario revisa la información que capturó
 - 5: El Usuario pide al sistema guardar datos de control del reporte de defectos y si es el Revisor envía un mensaje al Productor o viceversa
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo
 2. El usuario cancela la operación
- Precondición:** Exista lista de reporte de defectos para ese producto
- Postcondición:** Datos de control del reporte de defectos actualizados

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
SELECCIONAR REPORTE DE DEFECTOS



Seleccionar reporte
de defectos

- Caso de Uso:** Seleccionar reporte de defectos
- Actores:** Usuario
- Propósito:** Seleccionar lista de reportes de defectos asociados a un producto
- Descripción:** El usuario podrá seleccionar un reporte de un defecto asociado a un producto
- Escenario normal:**
- 1:El Usuario selecciona un producto
 - 2:El Usuario solicita al sistema le sean mostradas los Reportes de Defectos asociados a ese producto.
 - 3: El sistema muestra los reportes de defectos asociadas a un producto.
 - 4: El Usuario selecciona un reporte de defectos.
- Flujos excepcionales:**
- 1:En caso de no existir defectos asociados al producto, el sistema envía un mensaje notificando
 - 2:El Usuario cancela la lista defectos mostrados.
- Precondición:** Exista lista de reporte de defectos asociado al producto.
- Postcondición:** Reportes de defectos seleccionado.

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
MOSTRAR REPORTE DE DEFECTOS



Caso de Uso	Mostrar reportes de defectos
Actores:	Usuario
Propósito:	Mostrar las características de un reporte de defectos.
Descripción:	El usuario podrá leer el contenido de un reporte de defectos asociado a un producto
Escenario normal:	<ol style="list-style-type: none">1:El usuario selecciona un Reportes de Defectos2: El sistema mostrará el contenido del Reporte de Defectos3: El usuario lee el contenido del Reporte de Defectos
Flujos excepcionales:	<ol style="list-style-type: none">1:El usuario cancela la lectura del Reportes de Defectos.
Precondición:	Exista catálogo de productos y Reportes de Defectos.
Postcondición	Reportes de Defectos consultado

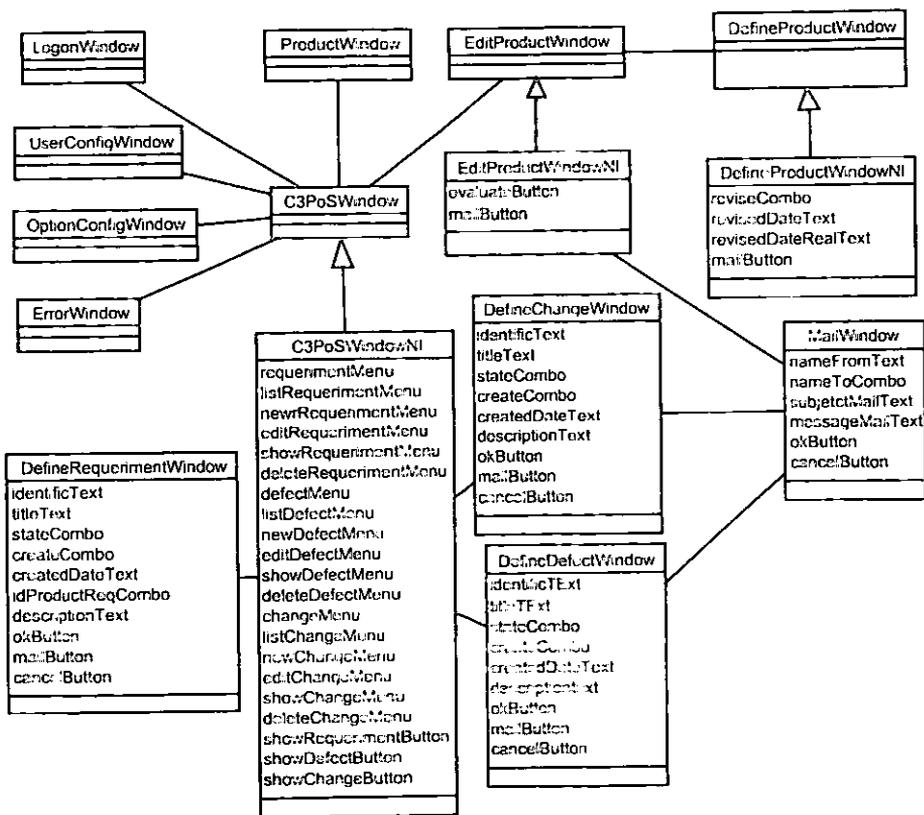
C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Intermedio (Versión 0.8)
ENVIAR MENSAJE



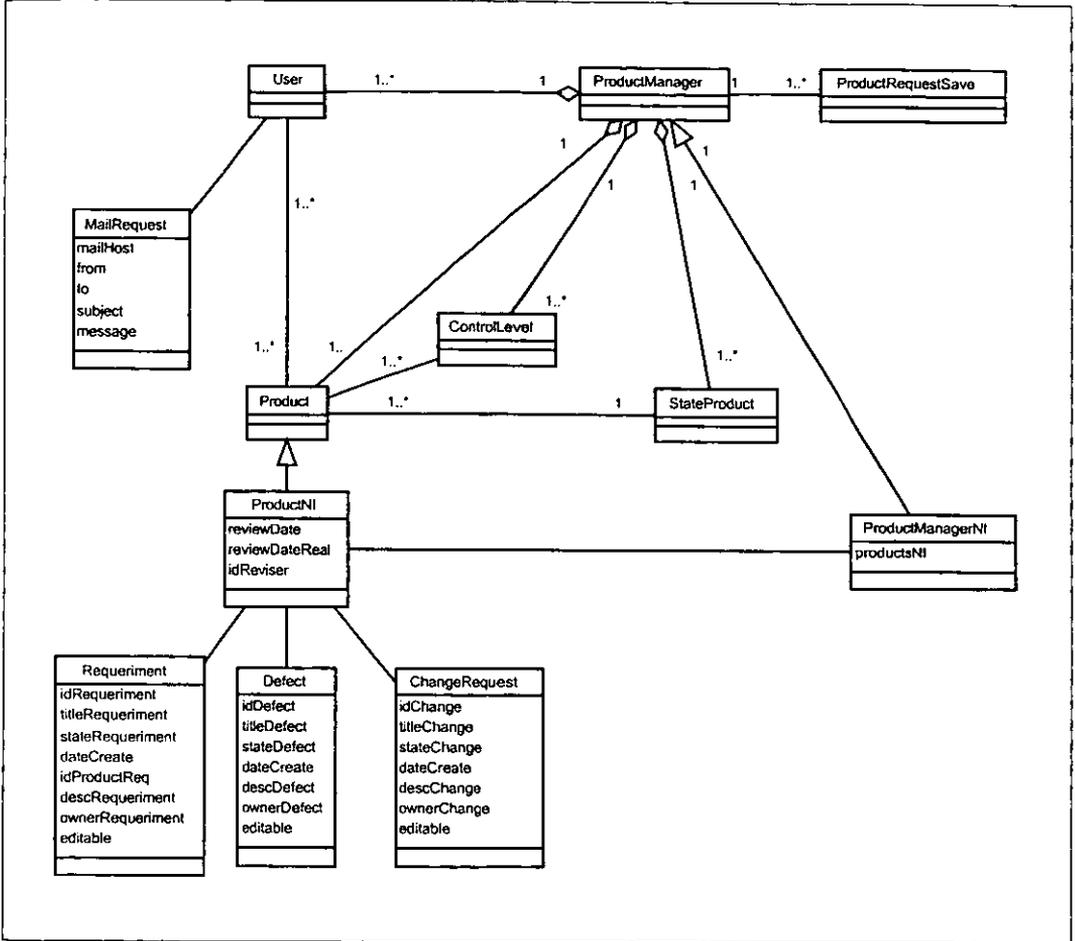
Enviar Mensaje

- Caso de Uso:** Enviar mensaje
- Actores:** Usuario (Solicitante, Productor, Revisor)
- Propósito:** Enviar un mensaje a otros actores del sistema
- Descripción:** El usuario podrá enviar un mensaje por correo electrónico al terminar de editar el producto, el defecto o la solicitud de cambio a otro actores involucrados en el desarrollo y evaluación del producto.
- Escenario normal:**
- 1: El usuario introduce información necesaria para el mensaje:
 - Las persona a quien se le envía el mensaje
 - El título de mensaje
 - Descripción del mensaje
- Flujos excepcionales:**
- 1: El usuario cancela el envío del mensaje
- Precondición:** 1:Exista catálogo de usuarios con dirección de correo electrónico.
- Postcondición:** 2:Mensaje recibido por las personas destinatarias del mensaje.



C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software
 Martín De Jesús Jiménez

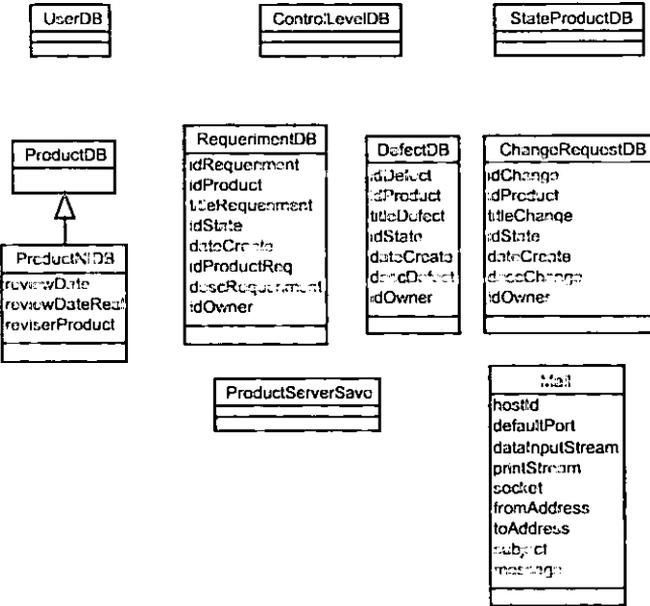
Documento de Análisis
 Diagramas de Clases, Nivel Intermedio (Versión 0.8)
DOMINIO DEL PROBLEMA



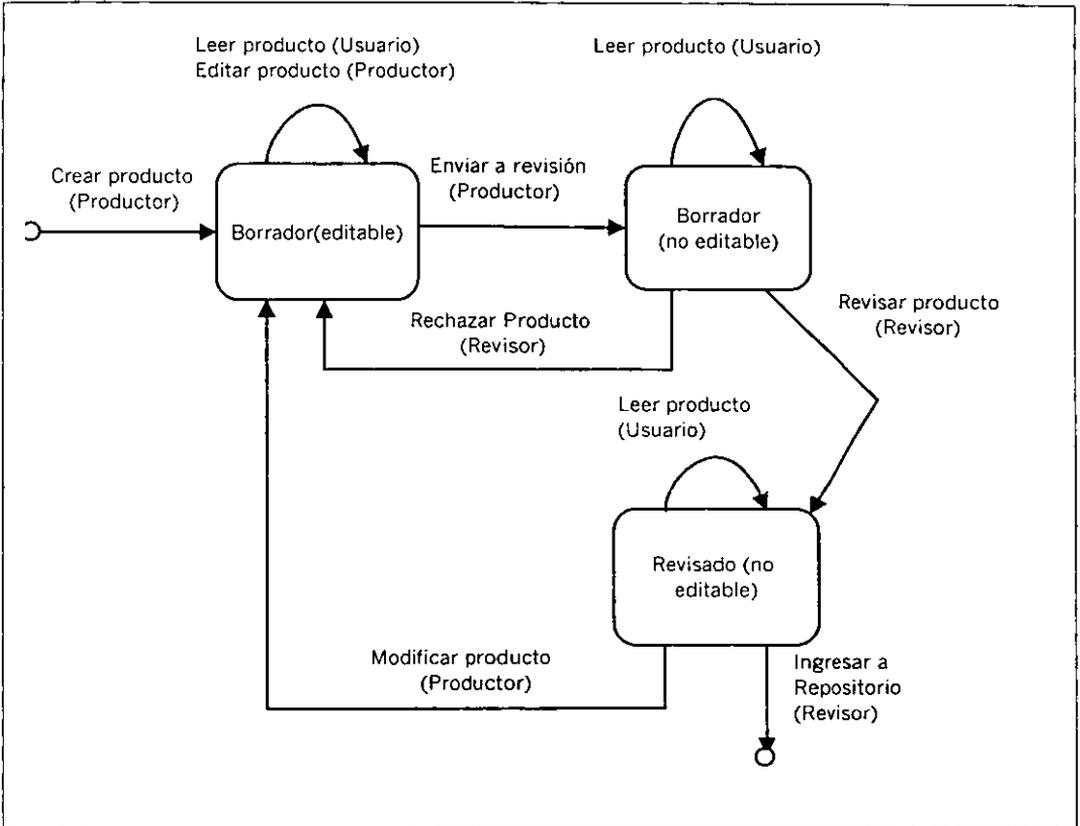
Documento de análisis para productos con nivel de control intermedio

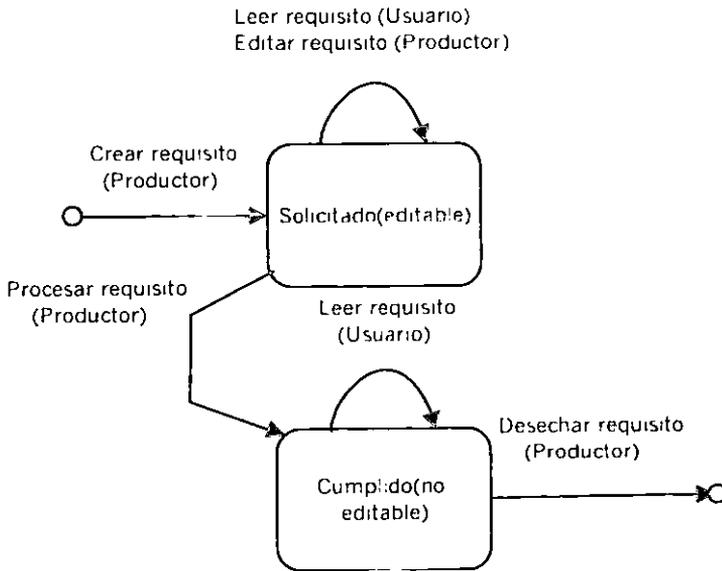
C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software
 Martín De Jesús Jiménez

Documento de Análisis
 Diagramas de Clases, Nivel Intermedio (Versión 0.8)
MANEJO DE DATOS



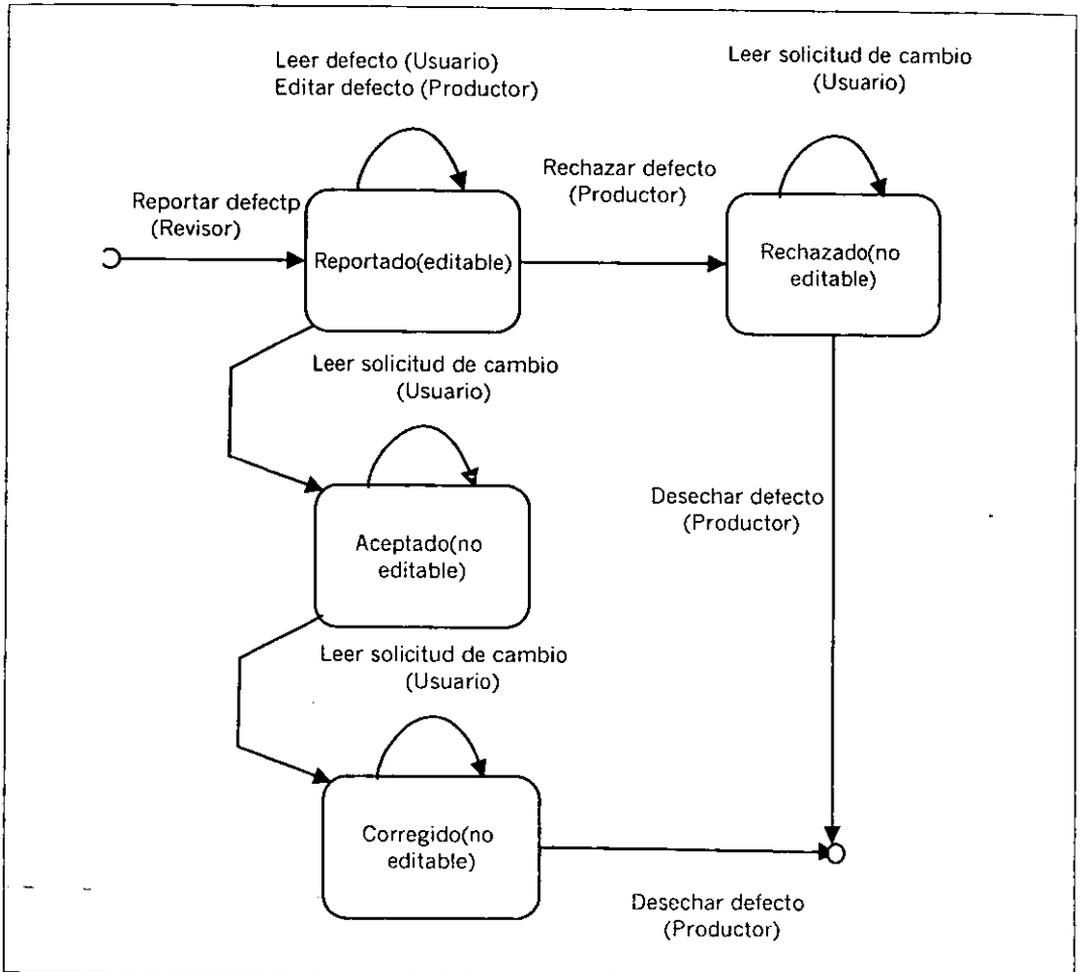
C3PoS Versión 0.1 Prototipo para elControl de Calidad y Cambios en los Productos de Software Martín De Jesús Jiménez	Documento de Análisis Diagramas de estados, Nivel Intermedio (Versión 0.8) PRODUCTO
---	--

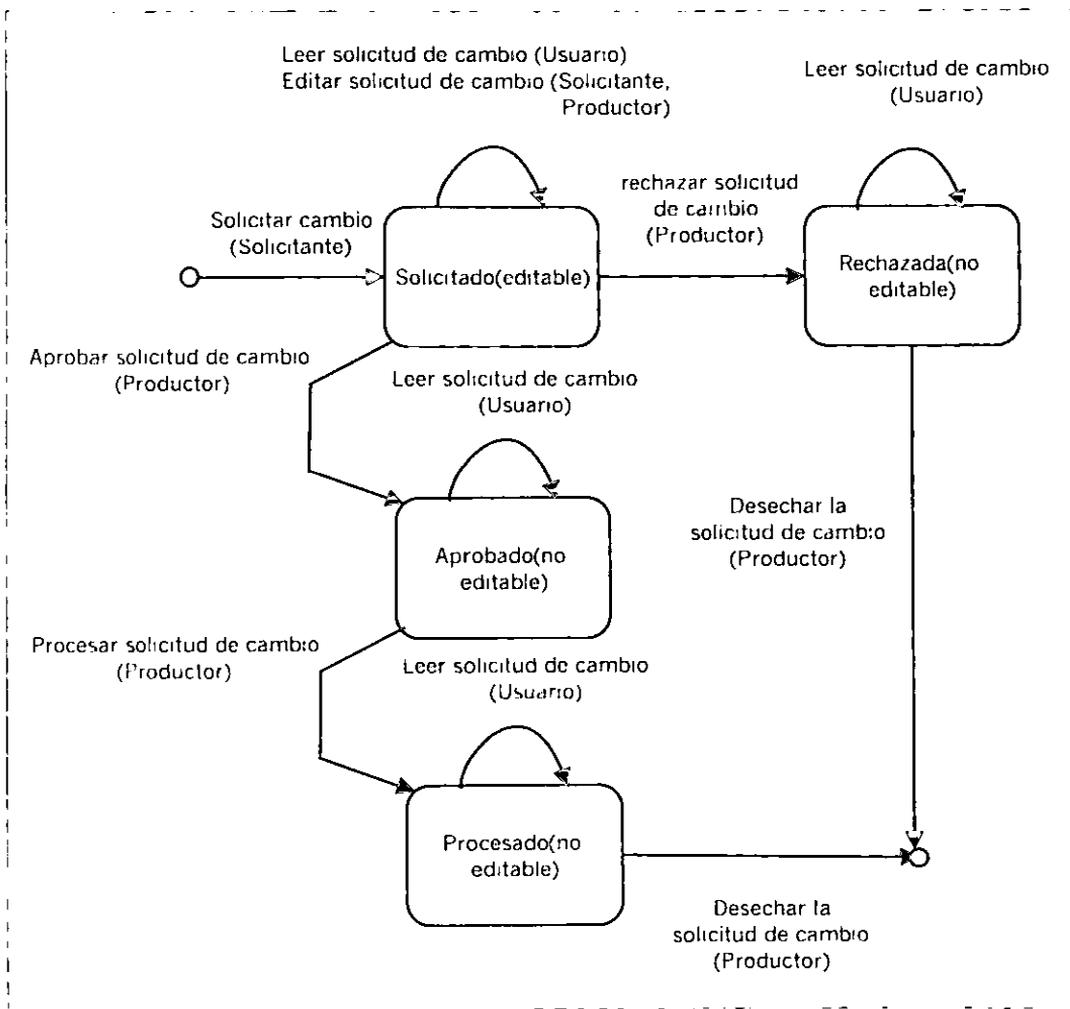




C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software
Martín De Jesús Jiménez

Documento de Análisis
Diagramas de estados, Nivel Intermedio (Versión 0.8)
DEFECTO





C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Intermedio (Versión 0.8) INTERFAZ HUMANA
---	--

C3PoSWindowNI

Producto	Requisitos	Defectos	Cambio	Configuración	
Lista de Productos					
Estado	Título	Identificador	Versión	Fecha	Nivel de Control
Borrador	Actividades del Nivel Simple	C3PoS-NS	0.1	01/11/2001	Simple
Revisado	Actividades del Nivel Intermedio	C3PoS-NI	0.1	01/11/2001	Intermedio

Ventana principal del sistema mediante la cual se tendrá acceso a las funciones del sistema. Se muestra después de que el usuario introdujo su clave de acceso y clave secreta correctos.

Nombre del atributo	Propiedad
RequerimentMenu	Menú del requisito
listRequerimentMenu	Lista los requisitos dados de alta
newRequerimentMenu	Crea nuevo requisito
editRequerimentMenu	Edita requisito
showRequerimentMenu	Muestra las características del requisito
deleteRequerimentMenu	Borra un requisito seleccionado
defectMenu	Menú del defecto
listDefectMenu	Lista los defectos dados de alta
newDefectMenu	Crea nuevo defecto
editDefectMenu	Edita defecto
showDefectMenu	Muestra las características del defecto
deleteDefectMenu	Borra un defecto seleccionado
changeMenu	Menú de la solicitud de cambio
listChangeMenu	Lista las solicitudes de cambios dados de alta
newChangeMenu	Crea nueva solicitud de cambio
editChangeMenu	Edita solicitud de cambio
showChangeMenu	Muestra las características de la solicitud de cambio
deleteChangeMenu	Borra una solicitud de cambio seleccionada

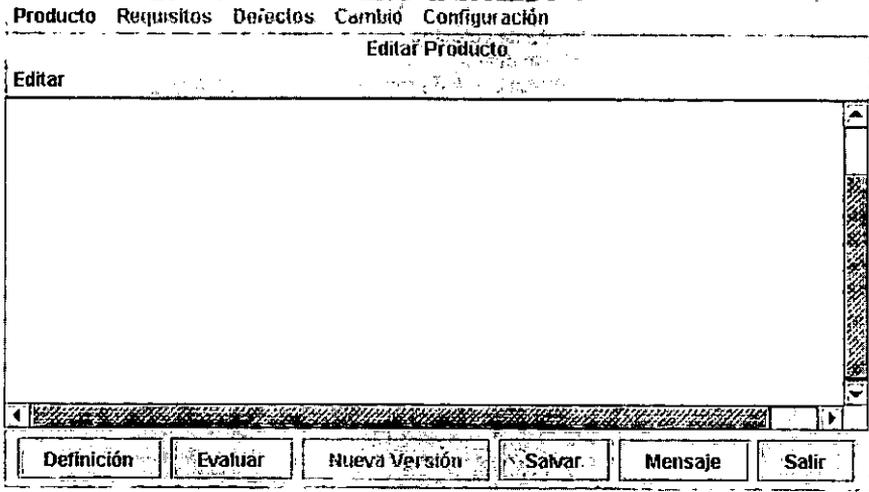
Documento de análisis para productos con nivel de control intermedio

Nombre del método	Comportamiento
C3PoSW:ndowNI()	Crear la ventana y mostrarla
showListRequeriment()	Muestra la lista de requisitos
newRequeriment()	Crear un nuevo Requisito
editRequeriment()	Editar un requisito que fue previamente seleccionado
showRequeriment()	Mostrar el contenido del requisito seleccionado
deleteRequeriment()	Borrar un requisito que fue previamente seleccionado
showListDefect()	Muestra la lista de defectos
newDefect()	Crear un nuevo defecto
editDefect()	Editar un defecto que fue previamente seleccionado
showDefect()	Mostrar el contenido del defecto seleccionado
deleteDefect()	Borrar un defecto que fue previamente seleccionado
showListChange()	Muestra la lista de solicitudes de cambio
newChange()	Crear una nueva solicitud de cambio
editChange()	Editar una solicitud de cambio que fue previamente seleccionada
showChange()	Mostrar el contenido de la solicitud de cambio seleccionada
deleteChange()	Borrar una solicitud de cambio que fue previamente seleccionada

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
INTERFAZ HUMANA

EditProductWindowNI



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

Nombre del atributo	Propiedad
EvaluateButton	Para evaluar el producto
mailButton	Para enviar un mensaje

Nombre del método	Comportamiento
EditDocumentWindowNI()	Crear la ventana y mostrarla si recibe un producto como parámetro, el contenido del documento para editarlo
evaluate()	Para evaluar el producto
mailRequest()	Para realizar una solicitud de mensaje por correo electrónico

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
INTERFAZ HUMANA

DefineProductWindowNI

C3PoS - Definir Producto con Nivel de Control Intermedio

C3PoS-NI		Nombre del producto		Fecha de creación	
Actividades del Nivel Intermedio		Fecha de revisión		31/11/2001	
<input type="text" value=""/>		Fecha de aprobación		10/11/2001	
<input type="text" value="Sandra Gómez D"/>		Fecha de validación		11/11/2001	
				13/11/2001	

Este producto describe las actividades del nivel de control simple para el control de calidad y cambios

Aceptar Salir

Ventana para definir las características de control del producto. Se muestra cuando se crea o se edita el producto.

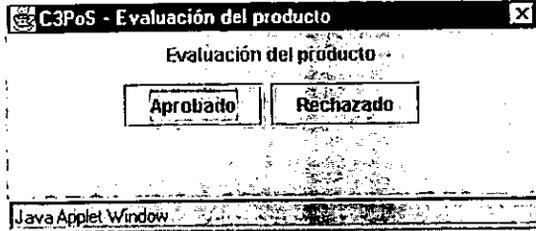
Nombre del atributo	Propiedad
reviseCombo	Nombre del Revisor del producto
revisedDateText	Fecha planeada de revisión del producto
revisedDateRealText	Fecha real de revisión del producto
mailButton	Para enviar un mensaje

Nombre del método	Comportamiento
DefineProductWindow()	Crear la ventana y mostrar, si recibe un producto como parámetro, las características del producto para editarlas
defineProduct()	Definir las características de un producto
setValue()	Inicializa los valores de los componentes
setActiveOwner	Establece las opciones de edición si el usuario es el propietario
setFirstTime	Establece las opciones de edición cuando el producto es creado
mailRequest()	Para realizar una solicitud de mensaje por correo electrónico

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
INTERFAZ HUMANA

EvaluateWindow



Ventana para definir la aprobación o el rechazo de un producto después de un proceso de evaluación.

Nombre del atributo	Propiedad
okButton	Botón para aprobar el mensaje
cancelButton	Botón para rechazar el producto

Nombre del método	Comportamiento
EvaluateWindow ()	Crea la ventana

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1

Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis

Diccionario de Clases, Nivel Intermedio (Versión 0.8)

INTERFAZ HUMANA

DefineRequerimientoWindow

Producto: Requisitos

Definición del Requisito

Identificador	5	El campo de texto debe aceptar el intermedio
estado	Documento de Análisis	estado Solicitado
fechaCreacion	11/11/2001	11/11/2001
idProductoReq	C3PoS-NS	

Se debe tomar como requisito el documento que corresponde al análisis del nivel de control simple

Aceptar Salir

Ventana para definir las características de un requisito con el cual debe cumplir un producto.

Nombre del atributo	Propiedades
idRequerimiento	Identificador del requisito
titulo	Título del requisito
estado	Estado del requisito
createLabel	Nombre del producto que definió el requisito
createdDateText	Fecha de creación del requisito
idProductoReqCombo	Identificador del producto como requisito
descriptionText	Descripción general del requisito
okButton	Guardar la información de la definición del requisito
cancelButton	Permitir cerrar la ventana sin actualizar la información de la definición del requisito.

Nombre del método	Comportamiento
defineRequerimientoWindow()	Crear la ventana y mostrar
defineRequerimiento()	Definir las características del requisito
close()	Cerrar la ventana sin definir el requisito

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Intermedio (Versión 0.8) INTERFAZ HUMANA
--	--

DefineDefectWindow

Ventana para definir las características del reporte de defectos encontrados a un producto como resultado de la revisión.

Nombre del atributo	Propiedad
IdentificText	Identificador del requisito
titleText	Título del requisito
stateCombo	Estado del requisito
createLabel	Nombre del producto que definió el requisito
createdDateText	Fecha de creación del requisito
descriptionText	Descripción general del requisito
okButton	Botón para guardar la información de la definición del requisito
mailButton	Botón para enviar un mensaje
cancelButton	Permitir cerrar la ventana sin actualizar la información de la definición del requisito.

Nombre del método	Comportamiento
DefineDefectWindow()	Crear la ventana y mostrar
defineDefect()	Definir las características de los defectos encontrados
close()	Cerrar la ventana sin definir los defecto
mailRequest()	Para realizar una solicitud de mensaje por correo electrónico

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
 Protocolo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
INTERFAZ HUMANA

DefineChangeWindow

Producto

Cambio

Ventana para definir el cambio

Identificador	Producto	Fecha de creación
Titulo	Estado	Fecha
Cambio en el Análisis	Solicitado	15/11/2001
Descripcion del cambio		

Se solicita realizar el cambio correspondiente a la definición del problema.

Aceptar Mensaje Salir

Ventana para definir las características de una solicitud de cambio a un producto.

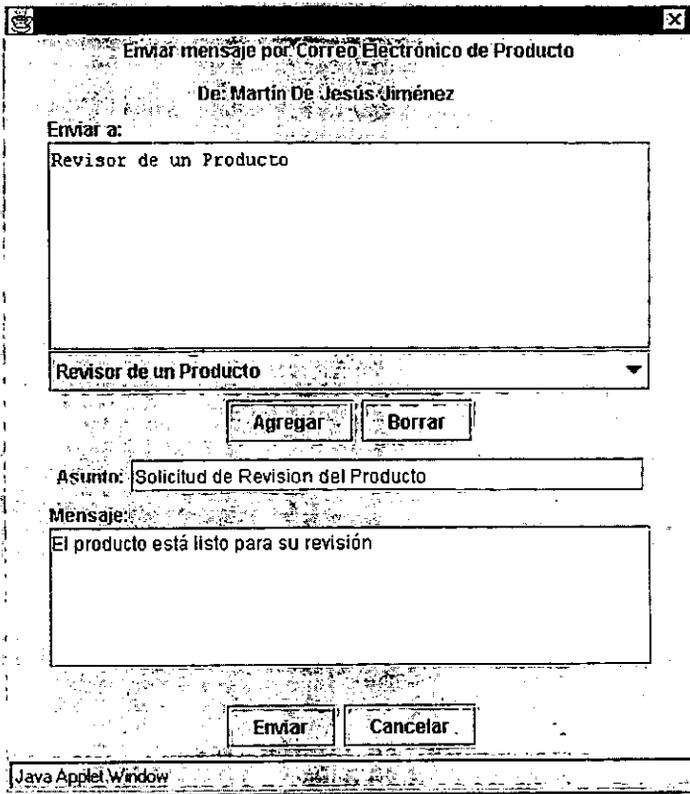
Nombre del atributo	Propiedad
IdentificText	Identificador del cambio
titleText	Título del cambio
stateCombo	Estado del cambio
createLabel	Nombre del Producto que definió el cambio
createdDateText	Fecha de creación del cambio
descriptionText	Descripción general del cambio
okButton	Botón para guardar la información de la definición del cambio
mailButton	Botón para enviar un mensaje
cancelButton	Permitir cerrar la ventana sin actualizar la información de la definición del cambio.

Nombre del método	Comportamiento
DefineDefectWindow()	Crear la ventana y mostrar
defineChange()	Definir las características de la solicitud de cambio
close()	Cerrar la ventana sin definir la solicitud de cambio
mailRequest()	Para realizar una solicitud de mensaje por correo electrónico

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
INTERFAZ HUMANA

MailWindow



Nombre del atributo	Propiedad
nameFromText	Nombre del que envía el mensaje
nameToCombo	Nombre del que se le enviará el mensaje
subjectMailText	Título del mensaje
messageMailText	Descripción del mensaje
okButton	Botón para aceptar el envío del mensaje
cancelButton	Botón para cancelar el mensaje

Nombre del método	Comportamiento
MailWindow	Crea el objeto MailWindow
defineMail	definir el mensaje
close()	cerrar la ventana sin definir el mensaje

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1

Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis

Diccionario de Clases, Nivel Intermedio (Versión 0.8)

DOMINIO DEL PROBLEMA

ProductManagerNI Administra los productos del sistema, los niveles establecidos, los estados posibles y los usuarios

Nombre del atributo	Propiedad
productsNI	Objetos producto con nivel de control intermedio

Nombre del método	Comportamiento
ProductManagerNI()	Crear el objeto
saveProduct()	Guardar la información del producto
getProductData()	Leer la información de los productos del sistema
getProduct()	Leer el producto indicado

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Intermedio (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

ProductNI. Es el producto con todas los elementos que lo definen.

Nombre del atributo	Propiedad
ReviewDate	Fecha planeada de revisión del producto, establecida por el Productor
reviewDateReal	Fecha real de revisión del producto
reviserProduct	Objeto usuario que revisa el producto, establecido por el Productor

Nombre del método	Comportamiento
ProductNI()	Crear el objeto producto
setReviewDate()	Escribir la fecha en que se planea revisar el producto
setReviewDateReal()	Escribir la fecha real de revisión del producto
setReviser()	Escribir el usuario que revisa el producto
getDate	Leer las fecha del producto
getReviser()	Leer el usuario que revisa el producto

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
DOMINIO DEL PROBLEMA

Requeriment. Es el requisito con todas los elementos que lo definen.

Nombre del atributo	Propiedad
IdRequeriment	Identificador del requisito
titleRequeriment	Título del requisito
stateRequeriment	Estado del requisito
dateCreate	Fecha de creación del requisito
idProductReq	Identificador del producto que es requisito de este producto
descRequeriment	Descripción general del requisito
ownerRequeriment	Nombre del productor que definió el requisito
editable	Información que indica si el requisito es modificable (0) o si solo se permite la lectura (1)

Nombre del método	Comportamiento
Requeriment()	Crea el objeto Requisito
setId()	Escribir el identificador del requisito
setTitle()	Escribir el título del requisito
setState()	Escribir el estado del requisito
setDateCreate()	Escribir la fecha en que se crea el requisito
setProduct	Escribir el identificador del producto que es requisito
setDesc()	Escribir la descripción general del requisito
getTitle()	Leer el título del requisito
getState()	Leer el estado del requisito
getDate()	Leer la fecha en que se crea el requisito
getProduct	Leer el nombre del identificador que es requisito
getDesc()	Leer la descripción del requisito
getOwner()	Leer el usuario que creó el requisito

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Intermedio (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

Defect. Es el defecto encontrado al producto, con todas los elementos que lo definen.

Nombre del atributo	Propiedad
IdDefect	Identificador del defecto
titleDefect	Titulo del defecto
stateDefect	Estado del defecto
dateCreate	Fecha de creación del defecto
descDefect	Descripción general del defecto
ownerDefect	Nombre del productor que definió el defecto
editable	Información que indica si el defecto es modificable (0) o si solo se permite la lectura (1)

Nombre del método	Comportamiento
Defect()	Crea el objeto Defecto
setId()	Escribir el identificador del defecto
setTitle()	Escribir el título del defecto
setState()	Escribir el estado del defecto
setDateCreate()	Escribir la fecha en que se crea el defecto
setDesc()	Escribir la descripción general del defecto
getTitle()	Leer el título del defecto
getState()	Leer el estado del defecto
getDate()	Leer la fecha en que se crea el defecto
getDesc()	Leer la descripción del defecto
getOwner()	Leer el usuario que creó el defecto

ChangeRequest. Es la solicitud de cambio con todos los elementos que lo definen.

Nombre del atributo	Propiedad
idChangeRequest	Identificador de la solicitud de cambio
titleChangeRequest	Título de la solicitud de cambio
stateChangeRequest	Estado de la solicitud de cambio
dateCreate	Fecha de creación de la solicitud de cambio
descChangeRequest	Descripción general de la solicitud de cambio
ownerChangeRequest	Nombre del productor que definió la solicitud de cambio
editable	Información que indica si la solicitud de cambio es modificable (0) o si solo se permite la lectura (1)

Nombre del método	Comportamiento
ChangeRequest()	Crear el objeto Defecto
setId()	Escribir el identificador de la solicitud de cambio
setTitle()	Escribir el título de la solicitud de cambio
setState()	Escribir el estado de la solicitud de cambio
setDateCreate()	Escribir la fecha en que se crea la solicitud de cambio
setDesc()	Escribir la descripción general de 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 en que se crea la solicitud de cambio
getDesc()	Leer la descripción de la solicitud de cambio
getOwner()	Leer el usuario que creó la solicitud de cambio

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Intermedio (Versión 0.8) DOMINIO DEL PROBLEMA
--	---

MailRequest, crea el objeto mensaje para enviarse por correo electrónico a un usuario

Nombre del atributo	Propiedad
nameFromText	Nombre del que envía el mensaje
nameToCombo	Nombre del que se le enviará el mensaje
subjectMailText	Título del mensaje
fromAddress	Dirección de correo electrónico de la persona que envía el mensaje
toAddress	Dirección de correo electrónico de la persona que envía el mensaje
message	Descripción del mensaje

Nombre del método	Comportamiento
MailRequest	Crea el objeto MailWindow
defineMailRequest	definir el mensaje
sendMail()	Solicitar enviar un mensaje

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Intermedio (Versión 0.8)
MANEJO DE DATOS

ProductNIDB. Contiene la información que se necesita para guardar cada producto.

Nombre del atributo	Propiedad
reviewDate	Fecha planeada de revisión del producto
reviewDateReal	Fecha real de revisión del producto
reviserProduct	Identificador del usuario que es el Revisor del producto

Nombre del método	Comportamiento
ProductNIDB	Crea el objeto
saveProduct()	Guarda información del producto en la Base de Datos
loadProduct()	Trae información del producto en la Base de Datos
deleteProduct()	Borra información del producto en la Base de Datos
updateProduct()	Actualiza información del producto en la Base de Datos

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Intermedio (Versión 0.8) MANEJO DE DATOS
---	--

RequerimentDB

Contiene información que se necesita guardar del requisito que debe cumplir el producto

Nombre del atributo	Propiedad
idRequeriment	Identificador del requisito
idProduct	Identificador del producto que debe cumplir con el requisito
titleRequeriment	Titulo del requisito
idState	Estado del requisito
createdDate	Fecha de creación del requisito
idProductReq	Identificador del producto que es requisito
descRequeriment	Descripción general del requisito
idOwner	Nombre del productor que definió el requisito

Nombre del método	Comportamiento
RequerimentWindow()	Crea el objeto
saveRequeriment()	Guarda información del requisito en la Base de Datos
loadRequeriment()	Trae información del requisito en la Base de Datos
deleteRequeriment()	Borra información del requisito en la Base de Datos
updateRequeriment()	Actualiza información del requisito en la Base de Datos

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis
Diccionario de Clases, Nivel Intermedio (Versión 0.8)
MANEJO DE DATOS

DefectDB

Contiene información de algún defecto encontrado al producto después de un proceso de revisión.

Nombre del atributo	Propiedad
idDefect	Identificador del defecto
idProduct	Identificador del producto que debe cumplir con el defecto
titleDefect	Título del defecto
idState	Estado del defecto
createdDate	Fecha de creación del defecto
descDefect	Descripción general del defecto
idOwner	Nombre del productor que definió el defecto

Nombre del método	Comportamiento
DefectWindow()	Crea el objeto
saveDefect()	Guarda información del defecto en la Base de Datos
loadDefect()	Trae información del defecto en la Base de Datos
deleteDefect()	Borra información del defecto en la Base de Datos
updateDefect()	Actualiza información del defecto en la Base de Datos

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Intermedio (Versión 0.8) MANEJO DE DATOS Diccionario de Clases, Nivel Intermedio
--	---

ChangeRequestDB

Contiene información de la solicitud de cambio a un producto

Nombre del atributo	Propiedad
idChange	Identificador del cambio
idProduct	Identificador del producto al que se le hace una solicitud de cambio
titleChange	Titulo del cambio
idState	Estado del cambio
createdDate	Fecha de creación del cambio
descChange	Descripción general del cambio
idOwner	Nombre del productor que definió el cambio

Nombre del método	Comportamiento
ChangeRequestDB()	Crea el objeto
saveChange()	Guarda información del cambio en la Base de Datos
loadChange()	Trae información del cambio en la Base de Datos
deleteChange()	Borra información del cambio en la Base de Datos
updateChange()	Actualiza información del cambio en la Base de Datos

Documento de análisis para productos con nivel de control intermedio

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Análisis
Diccionario de Clases, Nivel Intermedio (Versión 0.8)
MANEJO DE DATOS

Mail

Contiene información del mensaje enviado.

Nombre del atributo	Propiedad
hostid	Identificador del host de donde se enviarán los mensajes
defaultPort	El puerto de comunicación que se utilizará para enviar el mensaje
dataInputStream	Información recibida que se enviará
socket	El tipo de conexión realizada
fromAddress	Dirección de correo electrónico de la persona que envía el mensaje
toAddress	Dirección de correo electrónico de la persona que envía el mensaje
subject	Título del mensaje que se envía
message	Mensaje que se desea enviar

Nombre del método	Comportamiento
Mail	Crea el objeto
setValue()	Establece los valores que serán enviados
send()	Envía el mensaje
errorSend()	Identifica algún error durante el envío del mensaje

ANEXO D

Documento de análisis para productos con nivel de control completo

Este documento contiene información obtenida durante la fase de análisis del prototipo para el control de productos y lo componen:

1. Casos de uso y escenarios
2. Diagramas de clase
3. Diagrama de estado
4. Diccionario de clases

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Casos de uso y escenarios, Nivel Completo (Versión 0.8)
LLENAR DATOS DE CONTROL DEL PRODUCTO

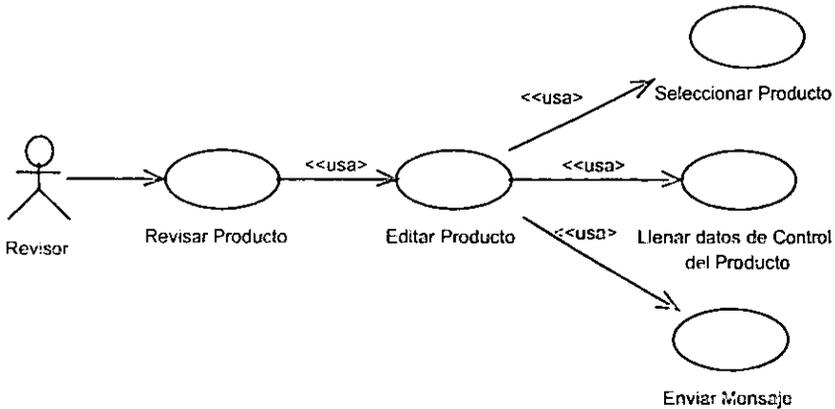


Llenar datos de control del producto

- Caso de Uso:** Llenar datos de control del producto
- Actores:** Usuario (Productor, Revisor)
- Propósito:** Llenar datos de control que definen al producto
- Descripción:** El Usuario introducirá datos que definen al producto o los actualiza.
- Escenario normal:** Si el producto es nuevo:
- 1: El sistema establece la versión 0.1, el estado de borrador y la fecha de creación.
 - 2: El sistema asigna el responsable de su producción.
 - 3: El usuario Productor introduce la información necesaria para llenar datos de control del producto, incluye los mismo datos solicitados en el caso de uso del nivel de control Simple además de:
 - Nombre del Revisor
 - Fecha Planeada de Revisión
 - Nombre del Aprobador
 - Fecha Planeada de Aprobación
 - 4: El usuario Productor revisa la información que capturó.
 - 5: El usuario pide al sistema guardar datos de control del producto.
- Curso alternativo:** Si el producto ya existe:
- 1: El sistema no establece la versión, ni el estado, ni la fecha de creación.
 - 2: El usuario actualiza datos de control, como:
 - Fecha real de revisión del producto
 - 3: El usuario Productor revisa la información que capturó.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo
 - 2: El usuario cancela la operación
- Precondición:** Exista catálogo de productos
- Postcondición:** Datos de control del producto actualizados

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

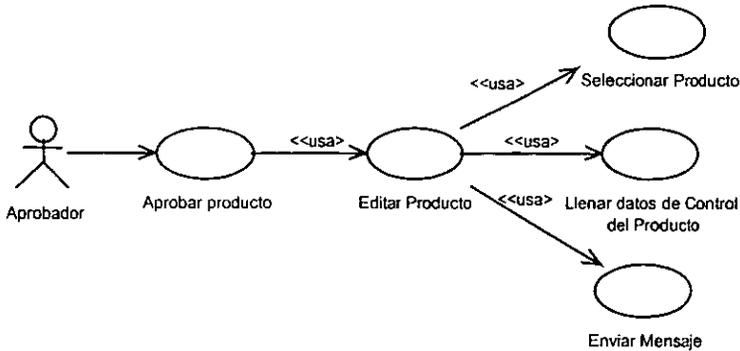
Documento de Análisis
Casos de uso y escenarios, Nivel Completo (Versión 0.8)
REVISAR PRODUCTO



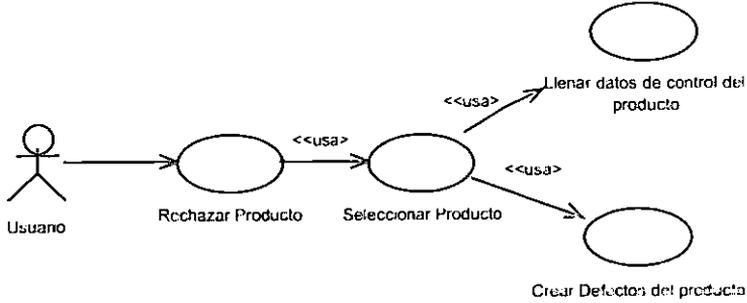
- Caso de Uso:** Revisar un Producto
- Actores:** Usuario (Revisor)
- Propósito:** El Revisor selecciona un producto, así como los requisitos que deberá cumplir el producto para su evaluación.
- Descripción:** El Revisor seleccionará un producto, cambiará el estado del producto a revisado.
- Escenario normal:**
- 1: El Revisor Selecciona un producto en estado de borrador y no editable.
 - 2: El sistema mostrará datos de control del producto.
 - 3: El Revisor consultará los requisitos del producto
 - 4: El Revisor con ayuda de los requisitos revisará el producto.
 - 5: El Revisor actualiza datos de control del producto
 - 6: El Revisor evalúa el producto y cambia el estado del producto a revisado si cumple con los requisitos, notifica al Productor y al Aprobador con un mensaje.
- Flujos excepcionales:**
- 1: Si el el producto es editable no podrá realizar esta operación.
 - 2: El Revisor cancela la revisión del producto
- Precondición:** Exista catálogo de productos y el producto está en estado de borrador y es no editable.
- Postcondición:** Los datos de control son actualizados y guardados.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Completo (Versión 0.8)
REVISAR PRODUCTO



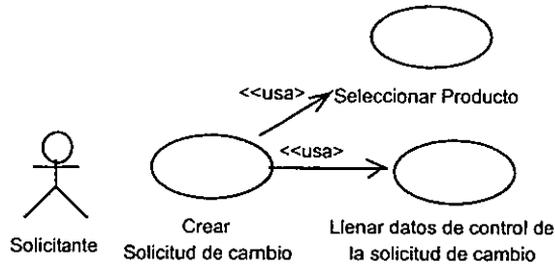
- Caso de Uso:** Aprobar un Producto
- Actores:** Usuario (Aprobador)
- Propósito:** El Aprobador selecciona un producto, así como los requisitos que deberá cumplir el producto para su evaluación.
- Descripción:** El Aprobador seleccionará un producto, cambiará el estado del producto a aprobado.
- Escenario normal:**
- 1: El Aprobador Selecciona un producto en estado de revisado y no editable.
 - 2: El sistema mostrará datos de control del producto.
 - 3: El Aprobador leerá la lista de verificación
 - 4: El Aprobador con ayuda de la lista de verificación aprobará el producto.
 - 5: El Aprobador actualiza datos de control del producto
 - 6: El Aprobador evalúa el producto y cambia el estado del producto a aprobado si cumple con la lista de verificación y lo notificará al productor y al revisor con un mensaje
- Flujos excepcionales:**
- 1: Si el producto es editable no podrá realizar esta operación.
 - 2: El Aprobador cancela la revisión del producto
- Precondición:** Exista catálogo de productos y el producto está en estado de revisado y es no editable.
- Postcondición:** Los datos de control son actualizados y guardados.



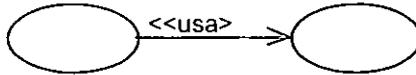
- Caso de Uso:** Rechazar un Producto
- Actores:** Usuario (Revisor, Aprobador).
- Propósito:** El usuario selecciona un producto y modificará datos de control y notificará al productor.
- Descripción:** El Usuario selecciona un producto, cambia el estado del producto y notificará al productor de los defectos encontrados.
- Escenario normal:**
- 1: El Usuario Selecciona un Producto en estado de borrador y no editable
 - 2: El sistema mostrará datos de control del producto.
 - 3: El Usuario evalúa el producto como no aceptado.
 - 4: El Usuario guarda datos de control del producto
 - 5: El Usuario avisa de su rechazo y de los defectos encontrados con un mensaje. Si es el Revisor se lo notifica al Productor y si es el Aprobador lo notifica al Revisor.
 - 6: El Usuario crea un reporte de los defectos encontrados.
- Flujos excepcionales:**
- 1: Si el producto es editable no podrá realizar esta operación.
- Precondición:** Exista catálogo de productos y el producto está en estado de borrador y es no editable.
- Postcondición:** Los datos de control del producto son actualizados y guardados.

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Casos de uso y escenarios, Nivel Completo (Versión 0.8)
CREAR SOLICITUD DE CAMBIO



- Caso de Uso:** Crear solicitud de cambio
- Actores:** Usuario (Solicitante)
- Propósito:** Crear solicitud de cambio a un producto.
- Descripción:** El Solicitante introducirá datos de control que corresponden a la solicitud de cambio a un producto cuando desea que sea modificado.
- Escenario normal:**
- 1: El Solicitante selecciona un producto
 - 2: El Solicitante selecciona Crear Cambio.
 - 3: El Solicitante introduce la información que corresponden a datos de control del Solicitud de cambios.
 - 4: El Solicitante introduce la descripción del cambio solicitado.
 - 5: El Solicitante pide al sistema guardar el solicitud de cambios.
 - 6: El Solicitante notifica de la solicitud al productor mediante un mensaje por correo electrónico.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo.
 - 2: El Usuario cancela la operación.
 - 3: Al producto no pueden generarse solicitud de cambios por no pertenecer a este nivel de control.
- Precondición:** Exista catálogo de productos que permitan solicitud de cambios.
- Postcondición:** El solicitud de cambios es guardado en el Catálogo de solicitudes de cambios y el productor recibe notificación de la misma.

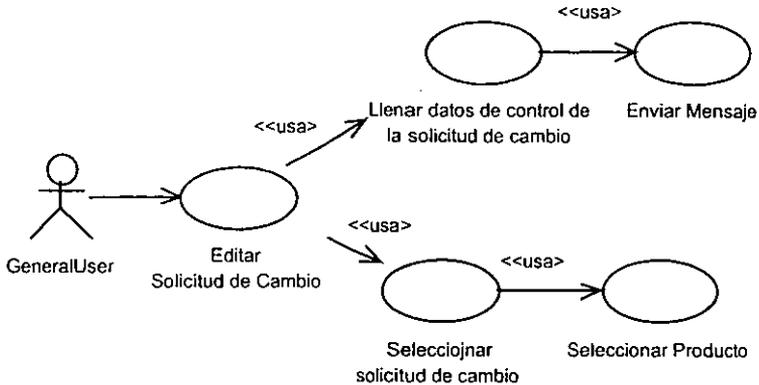


Llenar datos de control de la solicitud de cambio Enviar Mensaje

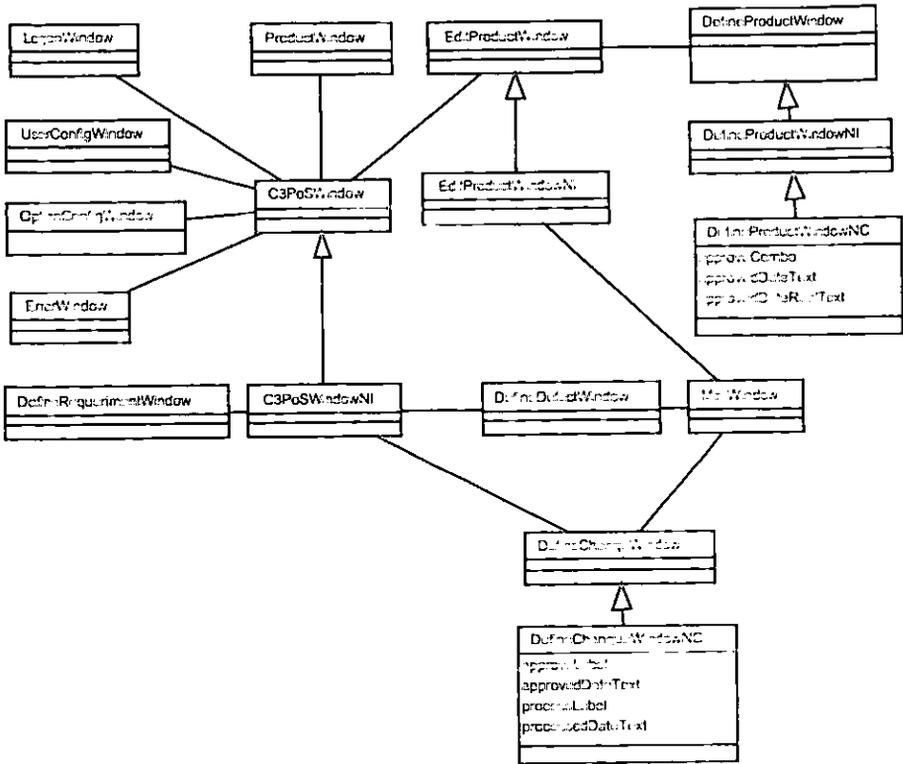
- Caso de Uso:** Llenar datos de control de la solicitud de cambio
- Actores:** Usuario (Solicitante, Aprobador)
- Propósito:** Llenar datos de control que definen la solicitud de cambios
- Descripción:** El usuario introducirá datos que definen al solicitud de cambios o los actualiza.
- Escenario normal:** Si el solicitud de cambios es nuevo:
- 1: El sistema establece su identificador, nombre del producto a quien se le reportará el cambio, quien lo crea y la fecha de creación.
 - 2: El Solicitante introduce la información necesaria para llenar datos de control del solicitud de cambios:
 - Título
 - Descripción de la solicitud de cambios
 - Nombre del aprobador de la solicitud del cambio
 - 3: El Usuario revisa la información que capturó.
 - 4: El Usuario pide al sistema guardar datos de control del solicitud de cambios y envía un mensaje al Aprobador.
- Curso alterno:** 1: Si el usuario es el Aprobador podrá introducir los datos
- Nombre de quien implanta el cambio
 - Modificar el estado del cambio y modificar la descripción del mismo
- Flujos excepcionales:**
1. El usuario cancela la operación
- Precondición:** Exista catálogo de productos
- Postcondición:** Datos de control de la solicitud de cambios actualizados

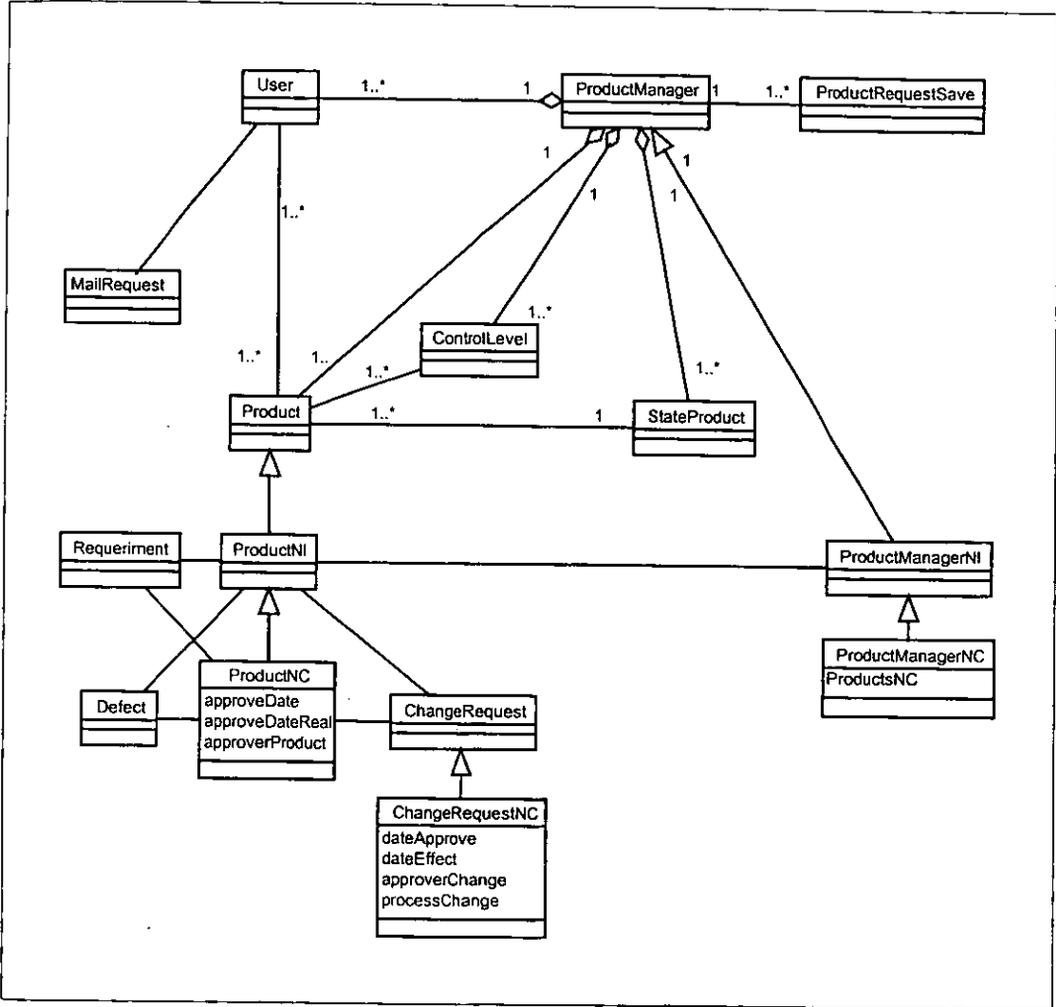
C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

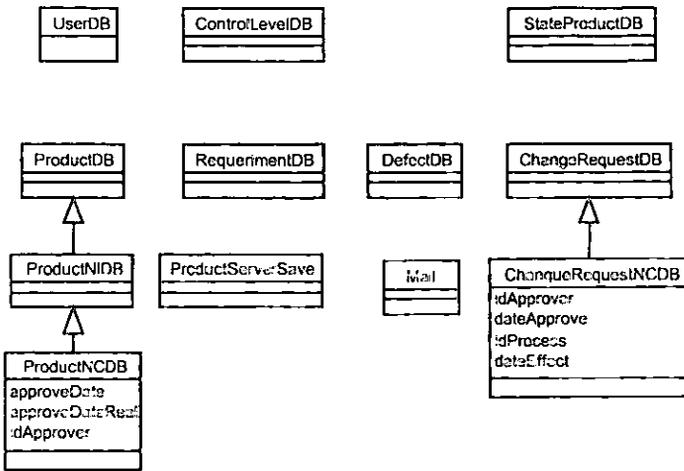
Documento de Análisis
 Casos de uso y escenarios, Nivel Completo (Versión 0.8)
EDITAR SOLICITUD DE CAMBIO

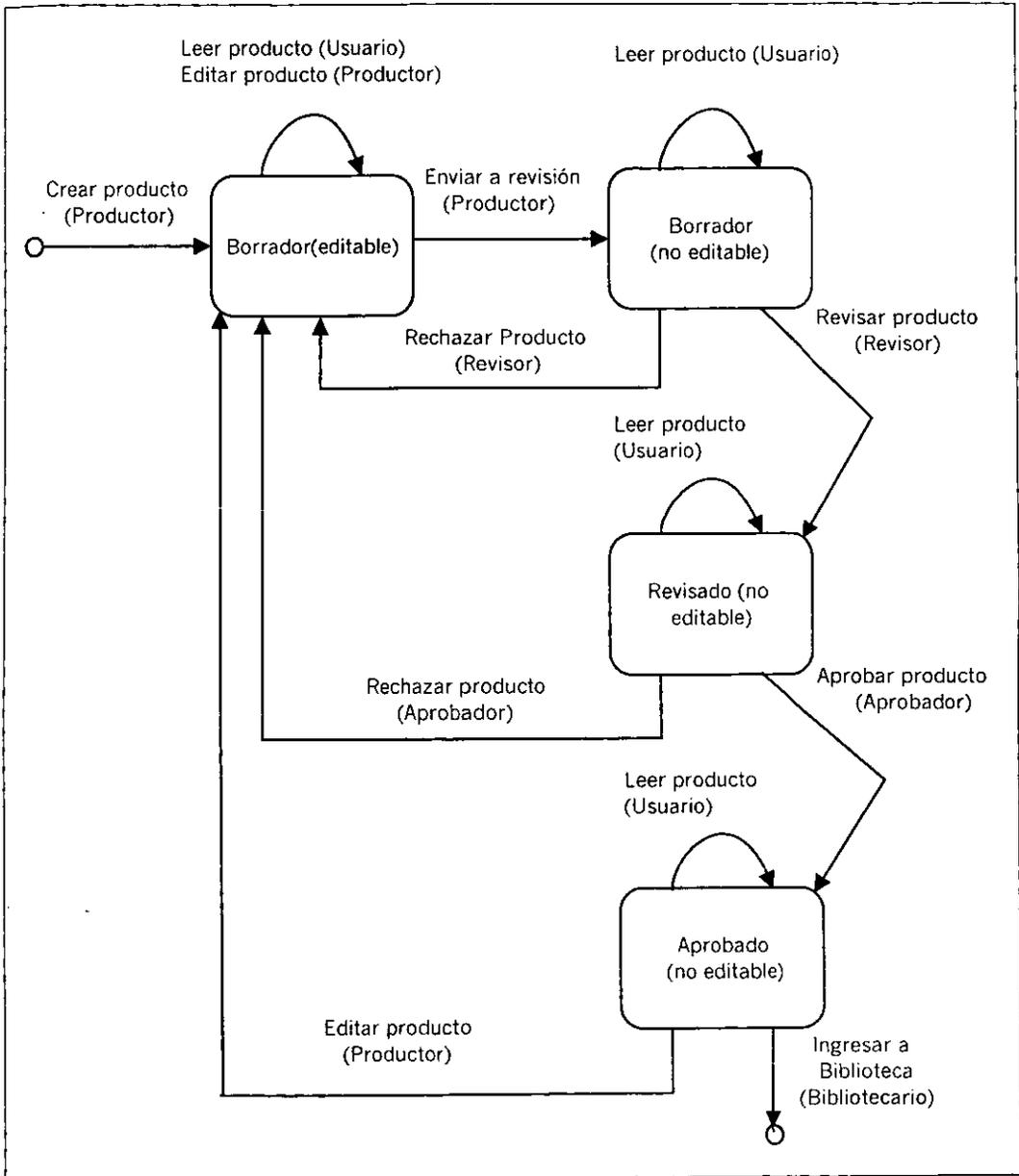


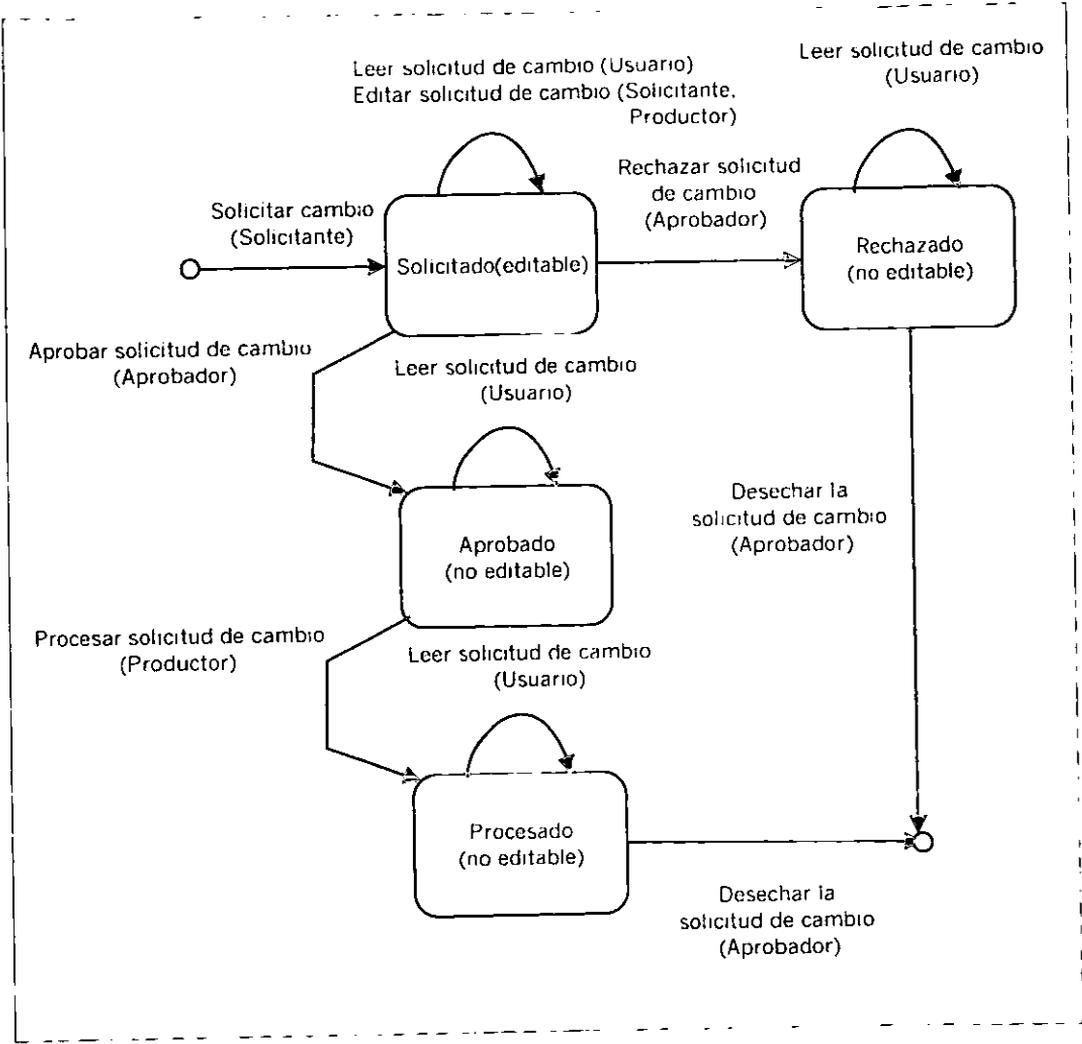
- Caso de Uso:** Editar solicitud de cambio
- Actores:** Usuario (Solicitante, Aprobador)
- Propósito:** Editar datos de control de la solicitud de cambios
- Descripción:** El Usuario modificará datos que definen la solicitud de cambios.
- Escenario normal:**
- 1:El Usuario selecciona un producto
 - 2:El Usuario selecciona la opción de edición de solicitud de cambios
 - 3:El usuario podrá modificar datos de control de la solicitud de cambio:
 - 4: El Usuario revisa la información que capturó.
 - 5: El Usuario pide al sistema guardar datos de control del solicitud de cambios y si es el Solicitante envía un mensaje al Aprobador. Si el usuario es el Aprobador enviará un mensaje al Productor y al Solicitante para informar de la solicitud del cambio.
- Flujos excepcionales:**
1. Los datos no cumplen reglas de tipo
 - 2: El usuario cancela la operación
- Precondición:** Exista lista de solicitud de cambios para ese producto.
- Postcondición:** Datos de control de la solicitud de cambios actualizados











C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Completo (Versión 0.8) INTERFAZ HUMANA
--	--

DefineProductWindowNC

C3PoS - Definir Producto con Nivel de Control Completo

Identificador	C3PoS-NC	Versión	0.1
Título	Documento de Análisis NS	Nivel de Control	Completo
Creado por	Henry Pérez Luna	con fecha	09/10/2001
Estado	Borrador	Fecha de terminación	20/10/2001
Revisado por	Sandra Gómez D.	Fecha real de terminación	21/10/2001
Aprobado por	Blanca Gil Castellanos	Fecha de revisión	25/10/2001
		Fecha real de revisión	27/10/2001
		Fecha de aprobación	30/10/2001
		Fecha real de aprobación	30/10/2001

Descripción

Se describe el análisis para el Nivel de Control Completo ...

Acceptar Salir

Java Applet Window

Ventana para definir las características de control del producto. Se muestra cuando se crea o se edita el producto.

Nombre del atributo	Propiedad
approveCombo	Nombre del Revisor del producto
approvedDateText	Fecha planeada de aprobación del producto
approvedDateRealText	Fecha real de aprobación del producto

Nombre del método	Comportamiento
DefineProductWindowNC()	Crear la ventana y mostrar, si recibe un producto como parámetro, las características del producto para editarlas.
defineProduct()	Definir las características de un producto
setValue()	Inicializa los valores de los componentes
setActiveOwner	Establece las opciones de edición si el usuario es el propietario
setFirstTime	Establece las opciones de edición cuando el producto es creado

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Completo (Versión 0.8)
INTERFAZ HUMANA

DefineChangeWindowNC

Producto

Cambio

Definición del Cambio

Identificador	Descripción	Estado	Fecha de inicio	Fecha de fin
1	Cambio en el Diseño	Aprobado	20/10/2001	
2			25/10/2001	
3			30/10/2001	

El cambio solicitado en el diseño fue aprobado y será implantado en la fecha de fin.

Aceptar Mensaje Salir

Ventana para definir las características de una solicitud de cambio a un producto

Nombre del atributo	Propiedad
approveLabel	Nombre del Aprobador del cambio
approvedDateText	Fecha de aprobación del cambio
processLabel	Nombre del que implanta el cambio
processedDateText	Fecha de implantación del cambio

Nombre del método	Comportamiento
DefineDefectWindowNC()	Crear la ventana y mostrar
defineChange()	Definir las características de la solicitud de cambio
evaluateChange()	Mandar a evaluar la solicitud de cambio

Documento de análisis para productos con nivel de control completo

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Completo (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

ProductManagerNC. Administra los productos del sistema, los niveles establecidos, los estados posibles y los usuarios.

Nombre del atributo	Propiedad
ProductsNC	Objetos producto con nivel de control completo

Nombre del método	Comportamiento
ProductManagerNC() saveProduct() getProductData() getProduct()	Crear el objeto Guardar la información del producto Leer la información de los productos del sistema Leer el producto indicado

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios
 en los Productos de Software

Documento de Análisis
 Diccionario de Clases, Nivel Completo (Versión 0.8)
DOMINIO DEL PROBLEMA

ProductNC. Producto con todas los elementos que lo definen.

Nombre del atributo	Propiedad
approveDate	Fecha planeada de aprobación del producto, establecida por el Productor
approveDateReal	Fecha real de aprobación del producto
approverProduct	Objeto usuario que aprueba el producto, establecido por el Productor

Nombre del método	Comportamiento
ProductNC()	Crear el objeto producto
setApprovedDate()	Escribir la fecha en que se planea aprobar el producto
setApprovedDateReal()	Escribir la fecha real de aprobación del producto
setApprover()	Escribir el usuario que aprueba el producto
getDate()	Leer la fecha del producto
getApprover()	Leer el usuario que aprueba el producto

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Completo (Versión 0.8) DOMINIO DEL PROBLEMA
---	---

ChangeRequestNC. Es la solicitud de cambio con todos los elementos que lo definen.

Nombre del atributo	Propiedad
dateApprove	Fecha en que se aprueba la solicitud de cambio, establecida por el Aprobador
dateChange	Fecha en que se incorpora la solicitud de cambio, establecida por el Productor
approverChange	Objeto usuario que aprueba la solicitud de cambio, establecido por el Productor
processChange	Objeto usuario que implanta la solicitud de cambio, establecido por el Productor

Nombre del método	Comportamiento
ChangeRequestNC	Crea el objeto
SetApprover()	Escribir el usuario que aprueba el cambio
SetDate()	Escribir fechas
setApprover()	Escribir el aprobador del cambio
setProcess()	Escribir el que implanta el cambio
getDate()	Leer fechas
getApprover()	Leer Aprobador
getProcess()	Leer el que implanta el cambio

Documento de análisis para productos con nivel de control completo

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios
en los Productos de Software

Documento de Análisis
Diccionario de Clases, Nivel Completo (Versión 0.8)
MANEJO DE DATOS

ProductNCDB Contiene la información que se necesita para guardar cada producto.

Nombre del atributo	Propiedad
approveDate	Fecha planeada de aprobación del producto
approveDateReal	Fecha real de aprobación del producto
idApprover	Identificador del usuario que aprueba el producto

Nombre del método	Comportamiento
ProductNCDB	Crece el objeto
saveProduct()	Guarda información del producto en la Base de Datos
loadProduct()	Trae información del producto en la Base de Datos
deleteProduct()	Borra información del producto en la Base de Datos
updateProduct()	Actualiza información del producto en la Base de Datos

C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Análisis Diccionario de Clases, Nivel Completo (Versión 0.8) MANEJO DE DATOS
---	--

ChangeRequestNCDB. Contiene información de la solicitud de cambio a un producto

Nombre del atributo	Propiedad
idApprover	Identificador del Aprobador del cambio
dateApprove	Fecha de aprobación del cambio
idProcess	Identificador del que implanta el cambio
dateEffect	Fecha de implantación del cambio

Nombre del método	Comportamiento
ChangeRequestNCDB()	Crea el objeto
saveChange()	Guarda información del cambio en la Base de Datos
loadChange()	Trae información del cambio en la Base de Datos
deleteChange()	Borra información del cambio en la Base de Datos
updateChange()	Actualiza información del cambio en la Base de Datos

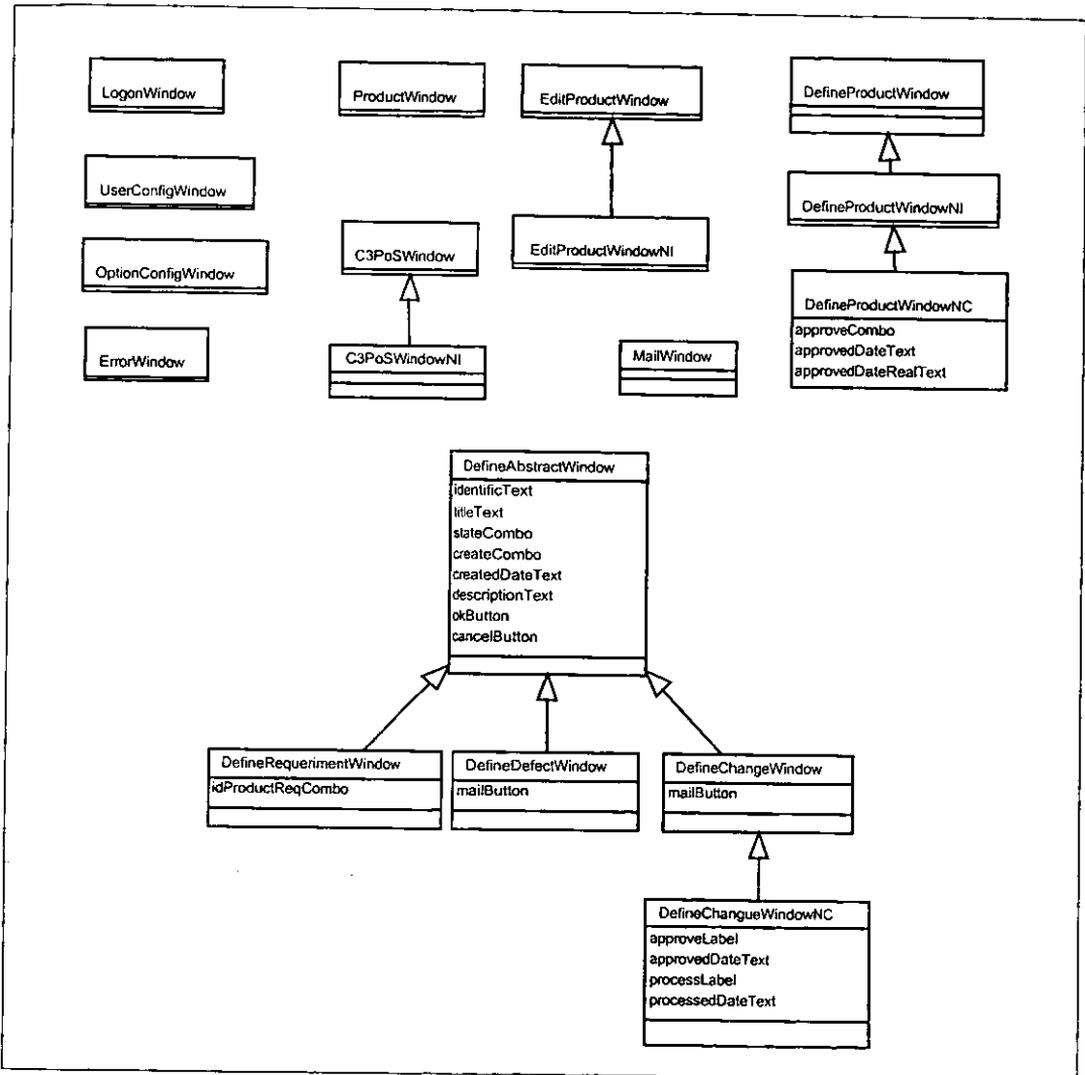
ANEXO E

Documento de diseño.

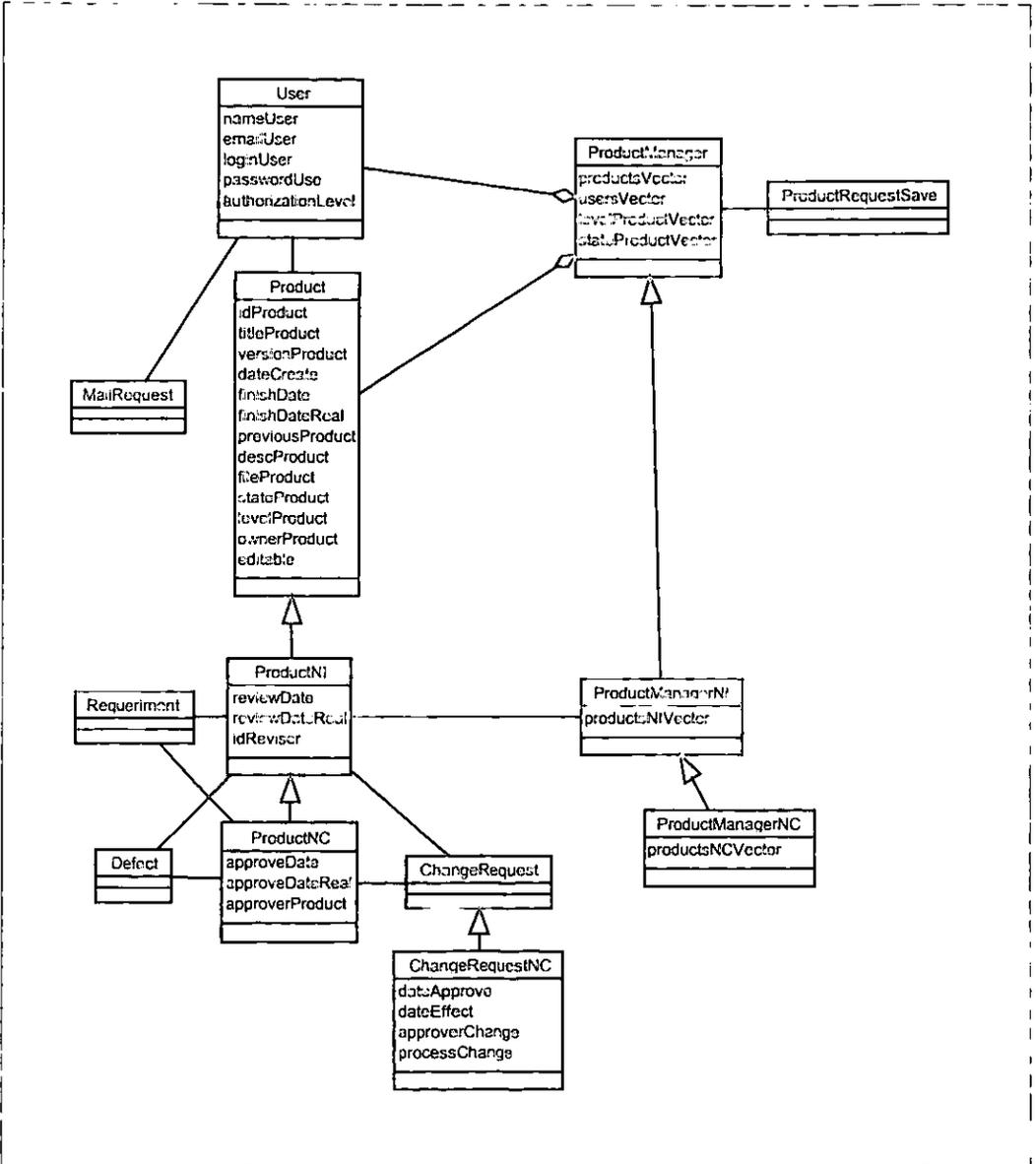
Este documento contiene información obtenida durante la fase de diseño del prototipo para el control de productos y lo componen:

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

En la interfaz humana se agrega la clase DefineAbstractWindow de la que heredan las definiciones de requisitos, defectos, solicitudes de cambio tanto para el nivel de control intermedio como el del completo. Se muestra para el nivel de control completo.

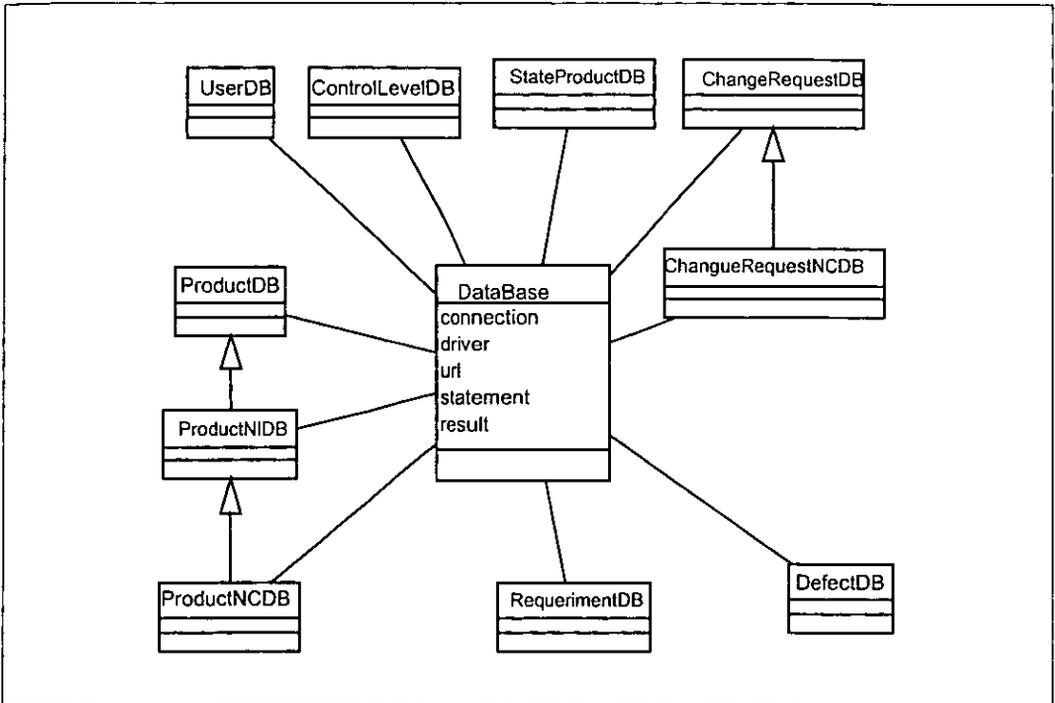


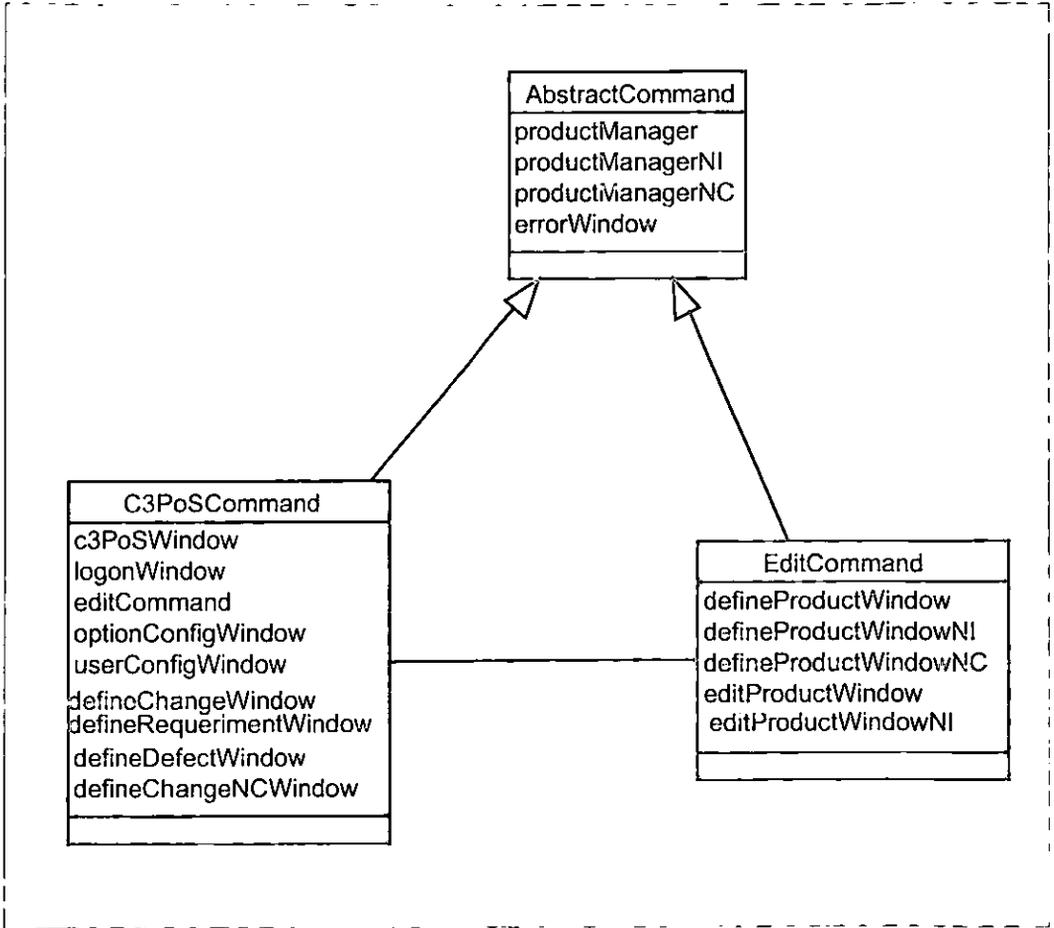
Se muestra el diagrama de clases para el dominio del problema del nivel de control completo:



A continuación se proporciona el diagrama de clases del Manejo de Datos del Nivel de Control Completo.

DataBase, es la clase que se encarga de la conexión y de los detalles del manejo de la base de datos en general. Aquí se muestra para el nivel de control completo:





C3PoS Versión 0.1 Prototipo para el Control de Calidad y Cambios en los Productos de Software	Documento de Diseño Diseño e la Base de Datos (Versión 0.8)
---	--

Para cada una de las clases del manejo de Datos se creó una Relación, que a continuación se detalla:

Clases pertenecientes al Manejo de Datos	Relación	Nulo	Dominio
ProductDB	PRODUCTS		
idProduct	{ idProduct }	No	Númérico
idstate	{ idState }	No	Texto
titleProduct	titleProduct	No	Texto
versionProduct	versionProduct	No	Texto
dateCreate	dateCreate	No	Texto
finishDate	dateFinish	No	Texto
finishDateReal	dateFinishReal	No	Texto
idPreviousProduct	idPreviousProduct	Si	Númérico
descProduct	descProduct	No	Númérico
fileProduct	fileProduct	No	Texto
idOwner	idOwner	No	Texto
idLevel	idLevel	No	Númérico
editable	editable	No	S/N
UserDB	USERS		
nameUser	{ idUser }	No	Texto
emailUser	nameUser	No	Texto
loginUser	passwordUser	No	Texto
passwordUser	emailUser	Si	Texto
authorizationLevel	levelUser	No	S/N
ControlLevelDB	LEVELS		
idControlLevel	{ idLevel }	No	Númérico
nameControlLevel	nameLevel	No	Texto
StateProductDB	STATES		
nameSate	{ idState }	No	Texto
descState	nameState	No	Texto

Clases pertenecientes al Manejo de Datos	Relación	Nulo	Dominio
ProductNIDB	PRODUCTSNI		
idProduct	{ idProduct }	No	Númeroico
idstate	idState	No	Texto
titleProduct	titleProduct	No	Texto
versionProduct	versionProduct	No	Texto
dateCreate	dateCreate	No	Texto
finishDate	dateFinish	No	Texto
finishDateReal	dateFinishReal	No	Texto
idPreviousProduct	idPreviousProduct	Si	Númeroico
descProduct	descProduct	No	Númeroico
fileProduct	fileProduct	No	Texto
idOwner	idOwner	No	Texto
idLeve:	idLevel	No	Númeroico
editable	editable	No	S/N
reviewDate	dateReview	No	Texto
reviewDateReal	dateReviewReal	No	Texto
reviserProduct	idReviser	No	Texto
RequerimentDB	REQUERIMENTS		
idRequeriment	{ idRequeriment }	No	Númeroico
idProduct	idProduct	No	Númeroico
titleRequeriment	titleRequeriment	No	Texto
idState	idState	No	Texto
createdDate	createdDate	No	Texto
idProductReq	idProductReq	No	Númeroico
descRequeriment	descRequeriment	Si	Texto
idOwner	idOwner	No	Texto
DefectDB	DEFECTS		
idDefect	{ idDefect }	No	Númeroico
idProduct	idProduct	No	Númeroico
titleDefect	titleDefect	No	Texto
idState	idState	No	Texto
createdDate	createdDate	No	Texto
descDefect	descDefect	Si	Texto
idOwner	idOwner	No	Texto
ChangeRequestDB	CHANGES		
idChange	idChange	No	Númeroico
idProduct	idProduct	No	Númeroico
titleChange	titleChange	No	Texto
idState	idState	No	Texto
createdDate	createdDate	No	Texto
descChange	descChange	Si	Texto
idOwner	idOwner	No	Texto

Clases pertenecientes al Manejo de Datos	Relación	Nulo	Dominio
ProductNCDB	PRODUCTSNC		
idProduct	{ idProduct }	No	Numérico
idstate	idState	No	Texto
titleProduct	titleProduct	No	Texto
versionProduct	versionProduct	No	Texto
dateCreate	dateCreate	No	Texto
finishDate	dateFinish	No	Texto
finishDateReal	dateFinishReal	No	Texto
idPreviousProduct	idPreviousProduct	Si	Numérico
descProduct	descProduct	No	Numérico
fileProduct	fileProduct	No	Texto
idOwner	idOwner	No	Texto
idLevel	idLevel	No	Numérico
editable	editable	No	S/N
reviewDate	dateReview	No	Texto
reviewDateReal	dateReviewReal	No	Texto
reviserProduct	idReviser	No	Texto
approveDate	approveDate	No	Texto
approveDateReal	approveDateReal	No	Texto
idApprover	idApprover	No	Texto
ChangeRequestNCDB	CHANGESNC		
idChange	idChange	No	Numérico
idProduct	idProduct	No	Numérico
titleChange	titleChange	No	Texto
idState	idState	No	Texto
createdDate	createdDate	Si	Texto
descChange	descChange	No	Texto
idOwner	idOwner	No	Texto
idApprover	idApprover	No	Texto
dateApprove	dateApprove	No	Texto
idProcess	idProcess	No	Texto
dateEffect	dateEffect	No	Texto

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Diseño
 Diccionario de Clases del Diseño (Versión 0.8)
 EJECUTANTES

C3PoSCommand. Es la clase principal de la aplicación que controla la mayor parte de las acciones del prototipo. Define los valores constantes que se utilizan, tales como estados de los productos y los parámetros de conexión a la base de datos. Las funciones principales son crear y editar un producto, requisito, defecto o solicitud de cambio, mostrar los requisitos, defecto o solicitudes de cambio de un producto seleccionado, mostrar los productos del nivel de control seleccionado, modificar las opciones y usuarios del prototipo y salir.

Nombre del método	Comando utilizado
main ()	Verificar la conexión con la base de datos Crear y actualizar ProductManager Crear LogonWindow y mostrarla Crear C3PosWindow
newProduct()	Crear EditCommand y ejecutar
editProduct()	Leer el producto seleccionado Crear EditCommand on el producto seleccionado y ejecutar
newRequeriment()	Crear DefineRequerimentWindow y mostrarla
editRequeriment	Leer el requisito seleccionado Crear DefineRequerimentWindow con el requisito seleccionado y mostrarlo
newDefect()	Crear DefineDefectWindow y mostrarla
editDefect()	Leer el defecto seleccionado Crear DefineDefectWindow con el defecto seleccionado y mostrarlo
newChange()	Crear DofineChangeNCWindow y mostrarla
editChange()	Leer la solicitud de cambio seleccionada Crear DefineChangeNCWindow con la solicitud de cambio seleccionada y mostrarla
userConfig()	Crear UserConfigWindow y mostrarla
optionConfig()	Crear OptionConfigWindow y mostrarla
viewLevelControl()	Mostrar productos del nivel de control seleccionado en C3PoSWindow
exitC3PosWindow	Cerrar la conexión con la base de datos Salirse del prototipo

C3PoS Versión 0.1
Prototipo para el Control de Calidad y Cambios en los
Productos de Software

Documento de Diseño
Diccionario de Clases del Diseño (Versión 0.8)
EJECUTANTES

EditCommand. Esta clase controla las acciones de edición de un producto. Las funciones principales son mostrar la ventana de edición y definición de las características de los productos, indicar que el producto está en evaluación y generar una nueva versión del producto.

Nombre del método	Comportamiento
editProduct()	Crear EditProductWindow y mostrarla
defineProduct()	Crear DefineProductoWindow y mostrarla
evaluateProduct()	Cambiar el producto a no editable
newVersionProduct()	Copiar el producto con versión 0.1 y estado borrador
saveProduct()	Salvar la información del producto

C3PoS Versión 0.1
 Prototipo para el Control de Calidad y Cambios en los
 Productos de Software

Documento de Diseño
 Diccionario de Clases del Diseño (Versión 0.8)
 MANEJO DE DATOS

DataBase. Es la clase que se utiliza para la administración del acceso a la base de datos mediante la clase JDBC.

Esta clase inicia y cierra la conexión con la base de datos.
 Permite a las clases en el manejo de datos realizar consultas y actualizaciones de la información.

Las clases del manejo de datos deben crear la expresión en SQL necesaria y la clase DataBase la procesa y envía el resultado.

Nombre del método	Comportamiento
openConnection()	Inicia la conexión de acuerdo con el driver y url establecido y verifica que no existan errores al iniciar la conexión
closeConnection()	Terminar la conexión y verificar que no existan errores al terminar la conexión
updateConnection()	Ejecutar la operación de actualización establecida por la expresión SQL y verificar que no existan errores en la actualización
openCursorConnection()	Ejecutar una operación de consulta establecida por la expresión SQL y verificar que no existan errores en la consulta.