

Universidad de las Ciencias Informáticas

Facultad 3



**Propuesta de expediente de proyecto para la gestión documental
en la Factoría de Software de la facultad 3.**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor(es): Yaniris Hinojosa Alón
Mayviris Martínez Valdivia

Tutor: Lic. Yaniel Díaz Rubio

Consultante: Lic. Liuris Rodríguez Castilla

Ciudad de la Habana

Junio 2007.

DECLARACIÓN DE AUTORÍA

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del 2007.

Mayviris Martínez Valdivia
Autor

Yaniris Hinojosa Alón
Autor

Lic. Yaniel Díaz Rubio
Tutor

“No hay más que asomarse a las puertas de la tecnología y la ciencia contemporánea para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimientos...”

Fidel Castro

AGRADECIMIENTOS COMPARTIDOS

A la Revolución Cubana, y en especial a Fidel por permitirnos ser parte de una generación de triunfadores y darnos la posibilidad de formarnos como profesionales comprometidos y conscientes del momento que estamos viviendo.

A la UCI, por ser nuestra segunda casa durante estos cinco años.

A nuestro tutor Yaniel, que pese a su juventud asumió esa difícil tarea. Gracias por su amabilidad, su apoyo y sus consejos.

A nuestros profesores de toda la carrera, gracias por tratar de regalarnos sus conocimientos y convertirnos hoy en lo que somos.

A nuestros compañeros de estudio y de cuarto por habernos hecho la vida tan alegre en todos estos años de estudio.

Al Lic. Lázaro Castillo por facilitarnos información para el desarrollo de este trabajo.

A Liuris por su ayuda incondicional.

A Abel por brindarnos su amistad y colaborar en la realización de este trabajo.

A todas aquellas personas que de un modo u otro ayudaron a la realización de esta tesis. Gracias por habernos brindado todo el apoyo, colaboración, ánimo, y sobre todo cariño y amistad.

A todos. Gracias

De Mayviris

A ti Dios que me diste la oportunidad de vivir y de regalarme una familia maravillosa.

A mi mamá Mary y a mi papá Vicente por su amor y paciencia, pero sobre todo por confiar en mí, apoyarme en la lucha por mis propios sueños y guiarme por un buen camino.

A mi hermano Marlon por hacerme feliz y tener que soportarme.

A mi abuelita Hivalia y mi abuelo Francisco, que aunque físicamente ya no están siempre me ayudaron, y me quisieron. Que Dios me los cuide por siempre.

Agradecer hoy y siempre a mi familia por haber estado a mi lado en todo momento respetando mi decisión. Gracias a todos por ayudarme.

A mi compañera de tesis Yaniris por su apoyo y tenerme la paciencia necesaria.

A mis compañeros de estudio y de cuarto; por todos los momentos compartidos, los quiero mucho y nunca los olvidaré.

Gracias a Adalberto por sus explicaciones a mis dudas siempre que las necesitaba.

A todas mis amistades, en especial a: Yani, Katy y Hany por compartir estos 5 años de universidad, por aconsejarme, regañarme, compartir risas y llantos en todo este tiempo.

De Yaniris

A mamá y papá, los mejores padres que puede tener cualquier hijo, los que me han dado todo su apoyo y se han sacrificado sobremanera, y que me alientan a dar siempre lo mejor de mí.

A mis hermanas Elita y Yanelis por darme tanta alegría y soportarme.

A mi abuelito, que aunque ya no esté físicamente siempre me apoyó y me quiso mucho. Abue tu sueño se hizo realidad.

A mi familia por su apoyo incondicional y por confiar en mí. Gracias por ser como son.

A Glenia por ser mi amiga de toda la vida y malcriarme tanto.

A Mayviris por ser la mejor compañera de trabajo que he tenido.

A ese piquete grande y feliz que hemos compartido tantas cosas desde primer año y seguiremos juntas a pesar de todo. Gracias May, Hany y Katy, las quiero mucho.

DEDICATORIA

A mis padres...

A mis hermanas...

A mi abuelito...

A toda mi familia...

Yaniris

A mis padres...

A mi hermano...

A toda mi familia...

Mayviris

RESUMEN

La Gestión Documental ha alcanzado en los últimos años un creciente auge como disciplina de apoyo a la producción de software. Es de gran importancia aplicar adecuadamente las técnicas de salvataje de la información con el objetivo de conservar la documentación que se genera del desarrollo de un software, siendo este un proceso complejo dentro de la Ingeniería de Software porque intervienen varios factores en él, tanto humanos como tecnológicos.

La introducción de las tecnologías informáticas en la gestión documental ha permitido el desarrollo de herramientas encargadas de automatizar la gestión de los documentos. El uso de estas herramientas soluciona la problemática de la gestión documental que utiliza como soporte el papel, y brinda innumerables beneficios.

El presente trabajo muestra la propuesta de un expediente de proyecto para el proceso de Gestión Documental en la Factoría de Software de la Facultad 3, apoyado en la selección de una herramienta para automatizar dicho expediente, y basado en la metodología RUP como proceso de desarrollo.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.....	6
1.2 Antecedentes de la Gestión Documental.	6
1.3 Gestión documental de proyectos de producción de software.	9
1.4 Metodología de desarrollo.	11
1.4.1 Proceso Unificado de Rational (RUP).....	12
1.5 Expediente de proyecto.....	21
1.5.1 Expediente de DeSoft S.A.....	22
1.5.2 Expediente de la Infraestructura Productiva (IP) de la UCI.	23
1.5.3 Expediente de la Facultad 3.....	25
1.6 Herramientas para la Gestión Documental de Proyectos.	26
1.7 Selección de la herramienta.....	35
1.8 Aspectos a tener en cuenta para trabajar con SVN.....	35
1.9 Conclusiones.....	40
CAPÍTULO 2: MONTAJE DE LA PROPUESTA DE EXPEDIENTE.....	41
2.1 Introducción.....	41
2.2 Propuesta de expediente de proyecto.....	41
2.3 Instalación de SVN.....	77
2.4 Configuración de la herramienta para montar el expediente de proyecto.....	80
2.5 Asociación de permisos con roles de RUP.	84
2.6 Proceso de las copias de trabajo para seguir la pista del repositorio.	84
2.7 StatSVN: métricas sobre Subversion.....	85
CAPÍTULO # 3: ANÁLISIS DE LOS RESULTADOS.....	89
3.1 Introducción.....	89
3.2 Criterios de especialistas.	89

3.3 Valoración de las entrevistas realizadas.	93
3.4 Conclusiones.	94
CONCLUSIONES	95
RECOMENDACIONES	96
REFERENCIAS BIBLIOGRÁFICAS.....	97
BIBLIOGRAFÍA CONSULTADA	100
GLOSARIO DE TÉRMINOS Y SIGLAS.....	103

INTRODUCCIÓN

El conjunto de principios y técnicas para la organización de los archivos en papel se ha ido desarrollando desde hace más de 1000 años. Éstas técnicas constituyen la disciplina de la documentación, que ha ido evolucionando en el último siglo debido a un crecimiento exponencial de la información. Con la aparición de los ordenadores y los soportes electrónicos, se habló inmediatamente de la oficina sin papeles. Al respecto se señala que durante la década del 50 comenzaron las primeras acciones de carácter práctico y luego se desarrolló en la literatura norteamericana el concepto de Gestor de Documentos [Guerrero, 2006].

Al término de la Segunda Guerra Mundial, esta metodología fue empleada para organizar la información documental acumulada por el Estado; posteriormente dio el impulso para mejorar los aspectos administrativos en la generación de la documentación pública en los Estados Unidos. De ahí la aparición del término Gestión Documental (GD), que surge muy ligado a la disciplina de la documentación con el objetivo de controlar el almacenamiento y recuperación de la información generada en cualquier proceso. La GD comprende el ciclo de vida completo de la documentación; desde su producción hasta la eliminación final o su envío al archivo para su conservación permanente evitando lo innecesario [Guerrero, 2006].

La Gestión Documental consiste en el uso de tecnología y procedimientos que permiten la gestión y el acceso unificado a la información generada en la organización [Gestión Documental, 2005]. Pretende el tratamiento integral, consistente y fiable de los documentos, y la información que se genera en las transacciones y procesos de negocio. La GD proporciona una solución integral para la manipulación y protección de aquellos documentos que se desean preservar en los proyectos como soporte de su negocio; esta solución permite la gestión de todo tipo de documentos, bien sean documentos en soporte físico de papel, como documentos cuyo soporte sea digital. La aplicación de estos procesos no solo hará posible disciplinar a los involucrados, permitirá además almacenar

y disponer de los datos históricos necesarios para lograr un trabajo más predecible y eficiente.

La GD es una disciplina aplicable a muchas ramas de la ciencia y la técnica, pues siempre hay un producto o acción con un propósito definido, donde la documentación es un testimonio de lo que se realiza durante el proceso.

El desarrollo de proyectos de software es el tema de interés de este trabajo, en el cual se realizará un estudio profundo de cómo manejar la información generada en los mismos. Gestionar documentos para lograr una mayor eficiencia en los proyectos de producción de software, es algo necesario para alcanzar la calidad en los productos; es indispensable documentar el proceso de desarrollo y controlar cada fase, de esto depende el éxito de cada proyecto, y la satisfacción del equipo y del cliente al conocer en cualquier momento el grado de desarrollo, y gestionar los cambios de una manera ágil y dinámica.

Toda la documentación que se genera en la producción de un software queda recogida en un expediente de proyecto, el cual es único para cada proyecto y contiene todos los artefactos creados durante el proceso de desarrollo del software, asegurando así su posterior uso o consulta.

En la actualidad se hace muy difícil llevar toda la documentación de un proyecto de producción de software en formato duro, por lo que se aplican las tecnologías de la información a fin de lograr una solución factible. De aquí que se empleen herramientas de GD con el propósito de soportar este proceso y hacerlo más favorable.

La Universidad de las Ciencias Informáticas (UCI) es un proyecto que tiene como misión principal elaborar softwares que soporten la informatización del país, y promover la competitividad internacional de la industria cubana del software. Una de las facultades que integra dicha Universidad es la Facultad 3, la cual se encuentra inmersa en un proceso de transformación y cambios estructurales en cuanto a la forma en que se lleva a

cabo la producción, orientándose hacia la creación de una Factoría de Software, con el propósito de realizar una producción de software más organizada, estructurada y que ofrezca servicios diferenciados, orientados al incremento de la calidad del producto final.

La creación de esta Factoría genera necesidades, entre ellas la estandarización del proceso de GD de un proyecto de desarrollo de software, con el objetivo de lograr un formato común para la documentación que se crea durante el desarrollo de un producto.

Las metodologías de desarrollo de proyectos no son todas iguales, lo que implica enfocarse hacia una en específico. Como este trabajo pretende aplicarse en la facultad 3, donde la metodología mayormente utilizada en sus proyectos es RUP, se decidió usar dicha metodología.

Teniendo en cuenta la situación actual, deviene el siguiente **problema**: La inexistencia de un mecanismo de control influye negativamente en la gestión eficiente de la información generada en los proyectos productivos de la facultad 3.

Por tanto se tiene como **objeto de estudio**: la Gestión Documental de proyectos de software.

Estableciendo así el **campo de acción**, siendo este: el expediente de proyecto.

La investigación se sustenta en la siguiente **hipótesis**: Si se implementa un expediente de proyecto para la gestión documental en la futura Factoría de Software de la facultad 3, se logrará la gestión eficiente de la información manejada en los mismos.

El **objetivo general** es: realizar una propuesta de expediente de proyecto que garantice la GD eficiente de los proyectos que se desarrollen en la Factoría de Software de la facultad 3.

Para el cumplimiento del objetivo de esta investigación se proponen las siguientes **tareas de la investigación:**

1. Actualizar conocimientos sobre el proceso de gestión documental de proyectos de software.
2. Actualizar conocimientos sobre herramientas que dan soporte al proceso de gestión documental.
3. Evaluar herramientas de gestión documental y seleccionar una de ellas.
4. Diseñar una estructura de expediente de proyecto.
5. Proponer un sistema de métricas para evaluar la implementación de la propuesta.
6. Realizar entrevistas a especialistas en documentación.
7. Validar la propuesta realizada sobre la base de los criterios recogidos.

Métodos científicos de la investigación a utilizar.

En el desarrollo de este trabajo se emplea la estrategia explorativa para conocer lo referente a la gestión documental de proyectos. Este tipo de estrategia tiene como principal objetivo relacionar al investigador con el tema que se estudia y su estado actual.

Entre los métodos teóricos utilizados para la investigación se emplea la Modelación, mediante la cual se diseña un expediente de proyecto con el objetivo de implementarlo en los proyectos productivos de la facultad 3, y uniformar el proceso de gestión de la información que se genera en los mismos; y el método Histórico Lógico que permite constatar teóricamente la evolución de la gestión documental desde su origen hasta nuestros días. De los métodos empíricos se utiliza la entrevista para validar la propuesta realizada.

Resultados Esperados.

Contribuir con la propuesta de expediente de proyecto desarrollada a la Factoría de Software de la facultad 3, en función de mejorar la productividad y calidad de los proyectos productivos de esta facultad.

Aportes prácticos.

La investigación obtendrá un expediente de proyecto estándar para la facultad 3 de la UCI que permitirá uniformar la gestión documental de proyectos de software en esta facultad.

El presente trabajo se ha estructurado en tres capítulos:

Capítulo 1: Fundamentación Teórica: se hace un análisis de la Gestión Documental desde sus inicios hasta la actualidad, enfatizando en lo correspondiente a la documentación generada en los proyectos de desarrollo de software, donde se plantean aspectos teóricos y conceptos importantes sobre el tema. Se analizan las herramientas que se utilizan para el proceso de gestión documental y las diferentes propuestas de expediente de proyecto dadas por varias organizaciones.

Capítulo 2: Montaje de la Propuesta de Expediente: se define un expediente de proyecto, y se estudia la herramienta Subversion aplicada a la propuesta de expediente diseñada al inicio de este capítulo.

Capítulo 3: Análisis de los resultados: se valida la propuesta de expediente de proyecto a través de criterios de especialistas en el tema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción.

La gestión documental como tema de análisis y desarrollo data de hace siglos. Ya en el Egipto y la Grecia antigua existían personas que su única función era la de recolectar, preservar y poner al alcance de otros la información existente. La Biblioteca de Alejandría llegó a contener miles de papiros y tabletas de arcilla grabadas. Hoy día la gestión y el soporte documental, junto a las nuevas tecnologías informáticas, han hecho de este campo un camino lleno de logros y fracasos. En este capítulo se exponen conceptos importantes sobre la gestión documental, destacando su aplicación a la producción de software, y se caracterizan algunas de las herramientas de gestión documental existentes en el mercado actual. Además se analiza RUP como metodología de desarrollo para organizar los documentos que se generan del proceso de desarrollo del software, y se presentan diferentes prototipos de expediente de proyecto empleados en varias organizaciones.

1.2 Antecedentes de la Gestión Documental.

El factor inicial relacionado a como tratar la documentación es un problema que tiene como origen lograr un control claro y actualizado de los documentos que genera cada uno de los proyectos, procedimientos y procesos en una organización.

La Gestión Documental clásica ha consistido en la impresión en papel de la información que se desea comunicar, y ha sido válida hasta principios de siglo dado que la cantidad de documentos ha sido razonable.

En el mundo del papel, la función de la gestión documental englobaba dos grandes campos: la gestión de los documentos activos, y los tradicionales archivos encargados de la organización/ preservación de los documentos pasivos.

Los tradicionales archivos en papel se han mantenido en paralelo a la introducción de los sistemas informáticos hasta la actualidad más reciente, siendo realmente escasas las empresas que han conseguido una verdadera oficina sin papeles. Sin embargo el documento electrónico es una realidad que crece vertiginosamente desde el momento en que se produce la aparición de los ordenadores y la utilización masiva de las herramientas ofimáticas. La tangibilidad que proporcionaba el papel como soporte documental ha cambiado drásticamente con los documentos electrónicos, sin que los responsables de la documentación hayan sido, en muchos casos, capaces de adaptarse al rápido cambio del entorno.

El término Gestión Documental puede tener distintos significados en el contexto empresarial. Generalmente se entiende como la gestión de los papeles, que en casi todos los contextos, desbordan la capacidad de las personas para guardarlos y encontrarlos cuando los buscan. En otros, con un nivel tecnológico más alto, Gestión Documental se asocia a la digitalización de imágenes y a escáneres que milagrosamente hacen desaparecer los papeles.

La Gestión Documental actual nace porque las organizaciones necesitan soluciones que permitan gestionar la información disponible para hacerla accesible, dentro de su ciclo de vida y bajo estrictos controles de seguridad [e-doctimo, 2007]. Es un elemento clave en la actividad de una empresa y está dejando de ser un simple método de archivo masivo para convertirse en una herramienta de análisis de información y gestión del conocimiento, con una creciente demanda no sólo en grandes corporaciones sino también en las Pequeñas y Medianas Empresas (PYMEs). La GD contempla distribuir, compartir, proteger, cambiar, buscar y versionar la información base para la actividad de la empresa [Sagi, 2007].

La Norma ISO 15489: Gestión de Documentos, publicada en el 2001 adquirió gran importancia en el contexto específico a la gestión de los documentos, debido a los escándalos financieros en las empresas ENRON y WORLCOM, y los atentados terroristas del 11 de Septiembre. Desde entonces, existe un reconocimiento internacional creciente de esta norma como un referente primordial en la gestión de los documentos en las organizaciones. Esta tiene como objetivo ser una guía para la gestión de documentos de archivo de una organización, sea cual sea su soporte.

La norma ISO 15489 plantea que los gestores de documentos e información tienen la función de valorar los recursos informativos disponibles, para documentar los procesos que se están llevando a cabo en la organización en un momento dado [Werner, 2005].

Abordar un proyecto de gestión documental en cualquier tipo de empresa requiere de dos condiciones básicas [Ruesta, 2000]:

- 1) La necesidad de pensar y planificar antes de actuar: es muy frecuente encontrar organizaciones que se “embarcan” en la implantación de proyectos de gestión documental sin establecer objetivos, sin valorar o considerar las dimensiones, los tiempos y la cantidad de elementos simultáneos que hay que poner en juego para obtener el éxito.
- 2) El diseño de un buen sistema requiere un buen proceso de análisis: los sistemas de gestión documental son complejos, pues en ellos intervienen múltiples factores, tanto organizativos como tecnológicos, y numerosas interrelaciones entre las personas. Sin llegar hasta el fondo en la detección de las necesidades y los problemas derivados de los resultados de la gestión documental, y sin una comprensión real de la actividad de la organización y de las personas es muy difícil llegar a buenos planteamientos.

La GD a pesar de ser una de las áreas del software donde se ha investigado desde hace algún tiempo, todavía no ha sido bien comprendida, ni ampliamente aplicada en esta industria; su implementación mejora la forma de controlar la documentación de un proyecto. El principal objetivo de las organizaciones es resolver los problemas derivados del exceso de información a tratar, bien por la gran cantidad de papel que debe ser manejado, así como por el almacenamiento de otros objetos de información de uso corporativo como pueden ser ficheros ofimáticos, reportes de las propias aplicaciones de gestión empresarial, imágenes, sonidos, video, e-mail, faxes, etc.

1.3 Gestión documental de proyectos de producción de software.

El tratamiento de la información y el conocimiento, en particular, en el ámbito de la informática, es un tema complejo en la actualidad. La Industria del Software, que abarca toda la creación y desarrollo de los sistemas informáticos, es una de las más competitivas a nivel internacional, debido a que su mercado se encuentra cubierto por un sinnúmero de grandes, medianas y pequeñas empresas con productos adecuadamente posicionados y clientes fijos. Por esto, es una garantía presentarse al mundo de la producción de software con efectivos métodos y formas de trabajo que demuestren eficacia y alta productividad.

La humanidad se encuentra sometida a un cambio constante, debido a la rápida evolución tecnológica de estos tiempos. Nadie escapa de este fenómeno aunque no a todos les llegue por igual. La producción de software es una práctica en la que diariamente se debe estar actualizado, pues avanza al paso de la revolución de la tecnología. Actualmente se construyen numerosos sistemas, apuntando a un mayor tamaño y complejidad, para satisfacer a usuarios cada vez más exigentes.

En la actualidad se habla mucho de la Ingeniería de Software, disciplina encargada de planificar los procesos de desarrollo y mantenimiento de un software. La ingeniería de software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadoras, y la documentación asociada requerida para desarrollar, operar y mantener dichos programas. Se conoce también como desarrollo de software o producción de software [Pressman, 2002].

Para realizar de forma exitosa un proyecto de software es necesaria la correcta ejecución de cuatro tipos de funciones:

1. Gestión del Proyecto: incluye fundamentalmente la Estimación, Planificación y Seguimiento del proyecto, Organización, Dirección y Gestión de Recursos Humanos.
2. Desarrollo Técnico: actividades de Ingeniería del Software a lo largo de todo el ciclo de vida del producto: Análisis, Diseño, Codificación.
3. Sistema de Calidad: incluye las actividades de Validación: construcción del producto correcto, Verificación: comprobar si se está construyendo el producto correctamente y Pruebas: verificar si funciona el código, y las actividades de Garantía de Calidad: Medidas encaminadas a asegurar que el producto se construye con unos determinados niveles de calidad.
4. Sistema de Gestión de Configuración: incluye principalmente la identificación, organización y control de las modificaciones que sufre el software.

La realidad muestra que la mayor cantidad de información que se maneja en un proyecto debe quedar plasmada en documentos para su inmediata y posterior consulta. La masiva implantación de herramientas de GD da lugar a que la mayoría de los documentos se produzcan de manera electrónica. Todo esto provoca que la Gestión Documental de Proyectos pretenda lograr un adecuado tratamiento de los documentos para optimizar su aprovechamiento, permitiendo a los desarrolladores de un producto de software guardar

toda la información que se genera en este proceso y poder recuperarla de forma precisa e inmediata; esto reduce tiempo de desarrollo y evita la duplicación de datos, perfeccionando la comunicación entre usuarios, clientes y entidad. Además se puede ver como la solución para el control de los documentos con una perspectiva superior, debido a que está presente en toda la etapa de su creación; desde su origen hasta que finalmente quede elaborado y se almacene, estableciendo una organización y actualización constante de los mismos.

El objetivo de las herramientas usadas en los proyectos de producción de software es presentar la información de forma ordenada, estándar, rápida y oportuna para la toma de decisiones. Esto proporciona una mejor perspectiva para realizar los proyectos con mayor éxito, dependiendo de la habilidad con la que se usen se obtendrá una mayor o menor eficacia [Herramientas, 2006].

1.4 Metodología de desarrollo.

Para llevar a cabo un proyecto de desarrollo de software, es fundamental contar con una metodología de desarrollo [Pressman, 2002], la cual se apoya en un conjunto de actividades que generan documentación. Esta forma de trabajo, si bien no asegura el éxito del proyecto, permite administrar de forma eficiente los recursos asignados al mismo. La mayoría de las metodologías de desarrollo actuales se basan en ciclos de vida de desarrollo iterativos. Para poder aplicar una metodología de desarrollo se debe contar con una herramienta que permita proteger los documentos del proyecto, asociar cada uno de estos con la actividad que lo generó, y con capacidad de versionar dichos documentos.

Dentro de cada una de las fases de desarrollo de un proyecto es normal encontrar un conjunto de subfases que generan documentos opcionales, los cuales dependiendo de las características del proyecto que se está desarrollando podrán ser completados o no.

Por último, es importante destacar que en la actualidad, salvo raras excepciones, los proyectos de desarrollo de software que se desarrollan en las distintas empresas difícilmente se asignan a una única persona. Por lo cual, el proceso de gestión de documentación que se aplique al proyecto deberá permitir vincular a cada uno de los elementos del proyecto con su responsable [Fernández, 2005].

El objetivo de un proceso de desarrollo es elevar la calidad del software en todas las fases por las que pasa, a través de una mayor transparencia y control sobre el proceso. No importa si es algo informal o para un cliente, hay que producir lo esperado en el tiempo y con el coste estimado. Es labor del proceso de desarrollo hacer que esas medidas para aumentar la calidad sean reproducibles en cada desarrollo. Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología o proceso de desarrollo de por medio, se lograrán clientes insatisfechos con el resultado y desarrolladores más insatisfechos aún.

1.4.1 Proceso Unificado de Rational (RUP).

El Proceso Unificado de Rational (RUP) es una metodología pesada que surge como resultado de la convergencia de Rational Approach y Objectory, su desarrollo estuvo a cargo de Ivan Jacobson. En la misma se han unificado varias técnicas de desarrollo, centrándose en el Lenguaje de Modelado Unificado (UML) como herramienta principal de modelado. La versión que se ha estandarizado vio la luz en 1998 y se conoció en sus inicios como Proceso Unificado de Rational 5.0; de ahí las siglas con las que se identifica a este proceso de desarrollo. RUP es uno de los procesos más generales de los existentes actualmente, debido a que está pensado para adaptarse a cualquier proyecto.

En el caso particular de la metodología RUP, por la propiedad de adaptación que presenta en cuanto a las condiciones del proyecto mediante su configuración previa, puede considerarse ágil [Fowler, 2003].

La metodología RUP define cuatro elementos principales:

1. Actividades: son los procesos que se llegan a determinar en cada iteración.
2. Trabajadores: van a hacer las personas involucradas en cada proceso.
3. Artefactos: productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
4. Flujo de actividades: secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

RUP ha estructurado su funcionamiento en grupos lógicos definiéndose 9 flujos de trabajo principales, los seis primeros se denominan flujos de ingeniería y los tres restantes flujos de apoyo.

Flujos de Ingeniería:

1. Modelación del Negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
2. Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen; traslada las necesidades del negocio a un sistema automatizado.
3. Análisis y Diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y los requerimientos, por lo que indica con precisión lo que se debe programar.
4. Implementación: Define cómo se organizan las clases en componentes, la ubicación de éstos en los nodos, y la estructura de capas de la aplicación, creando un software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
5. Prueba: Busca los defectos a lo largo del ciclo de vida asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

6. Despliegue: Produce un lanzamiento del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

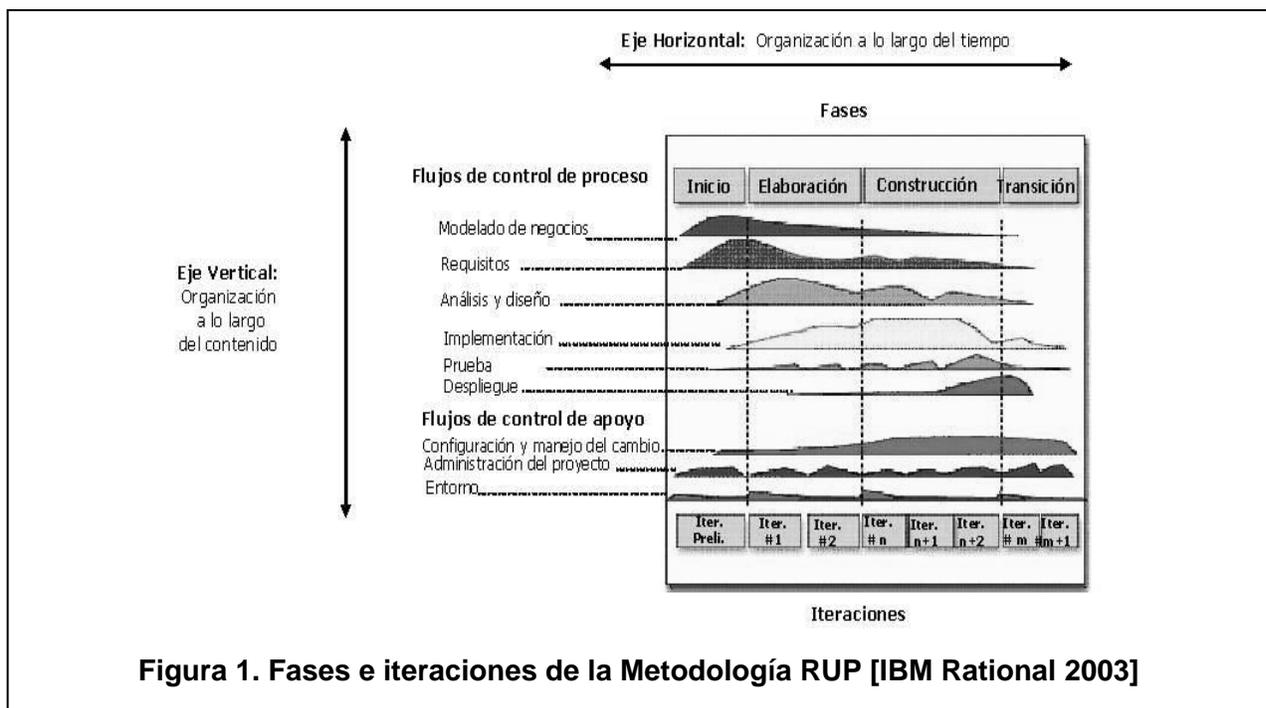
Flujos de Apoyo:

7. Gestión de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc, guardando todas las versiones del proyecto.
8. Gestión de proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
9. Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto, así como el procedimiento para implementar el proceso en una organización.

Un proyecto realizado siguiendo la Metodología RUP se divide en cuatro fases (Ver Figura 1):

1. Inicio: determina la visión del proyecto.
2. Elaboración: determina la arquitectura óptima.
3. Construcción: se obtiene la capacidad operacional inicial.
4. Transmisión: se obtiene una versión del producto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.



Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Una particularidad de la metodología RUP es que, en cada ciclo de iteración se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

Características de RUP [RUP, 2006]:

- ✓ Iterativo e incremental: Consta de una secuencia de iteraciones; cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina. Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los

detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

- ✓ Centrado en la arquitectura: La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.
- ✓ Guiado por casos de uso: Los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido, representan los requisitos funcionales del sistema.

En esta investigación se trabaja con la metodología RUP porque ofrece una solución a los problemas existentes en el proceso de producción de software de la facultad 3, por tal motivo los proyectos productivos de la misma utilizan esta metodología de desarrollo.

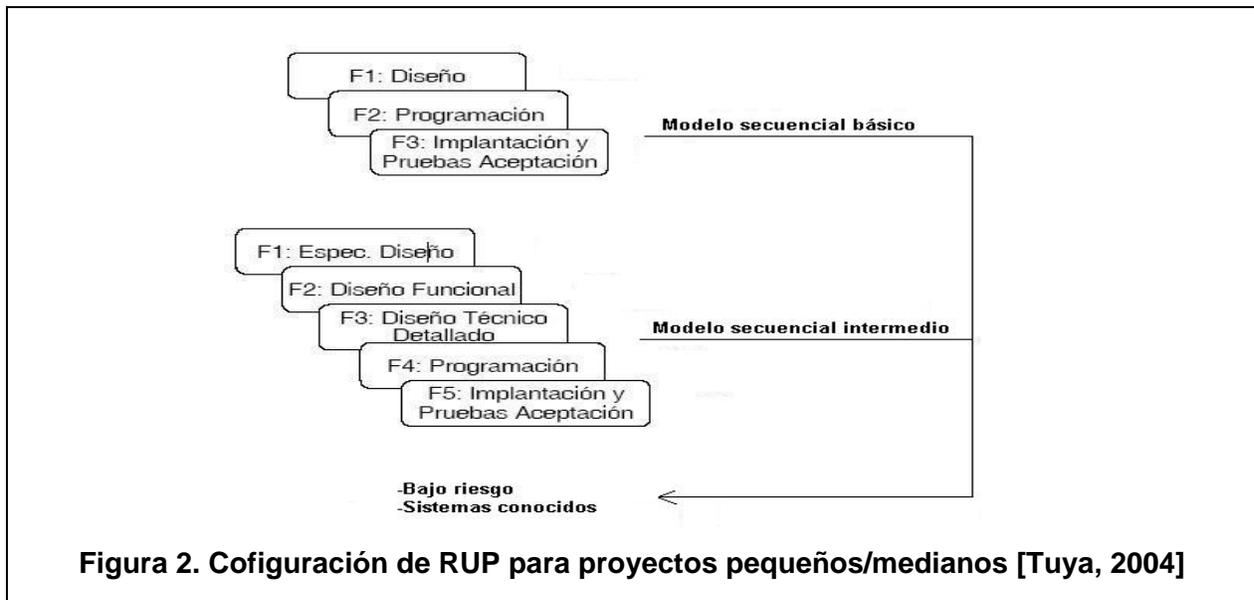
Abordar un proyecto de acuerdo a su magnitud basándose en RUP.

RUP permite mayor productividad en equipo y la realización de mejores prácticas de software a través de plantillas y herramientas que lo guían en todas las actividades de desarrollo crítico del software, además es un proceso de desarrollo adaptable a las necesidades particulares de cualquier proyecto.

RUP está pensado para proyectos y equipos grandes, en cuanto a tamaño y duración, debido a que es una buena posibilidad de reducir el trabajo a realizar. Para el desarrollo de software por medio de proyectos pequeños, hasta unas diez personas, es definitivamente muy grande y prácticamente inalcanzable. Se deben repartir 31 roles y generar más de 100 artefactos distintos. Esto supone que antes de implantar RUP se debe adaptar, lo que también requiere su tiempo y tiene su coste [Molpeceres, 2003].

Configuración de RUP para proyectos pequeños o medianos (Ver figura 2).

Muchas veces no se toma en cuenta utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyecto es separar rápidamente la aplicación en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo. Al iniciar un proyecto pequeño o mediano se emplean un número mínimo de personas o desarrolladores, lo cual sugiere que no existirá un mayor control, y entonces sucede la catástrofe, o se documenta en lo mínimo o lo que es peor, no se documenta. Por ejemplo, los requerimientos no son detallados más que en la mente del desarrollador o en algún diseño del jefe de proyecto o solo en el documento inicial del proyecto, provocando en el futuro un descontrol en el manejo de estos si ocurren cambios o aparecen nuevos requerimientos derivados de los inicialmente planteados.



En esta posible configuración de RUP para un proyecto pequeño/mediano se incluyen muy pocos artefactos y roles de la metodología, manteniendo los más esenciales. Dicha configuración está basada en la siguiente selección de artefactos que se generarán y usarán durante el proyecto [Letelier, 2007]:

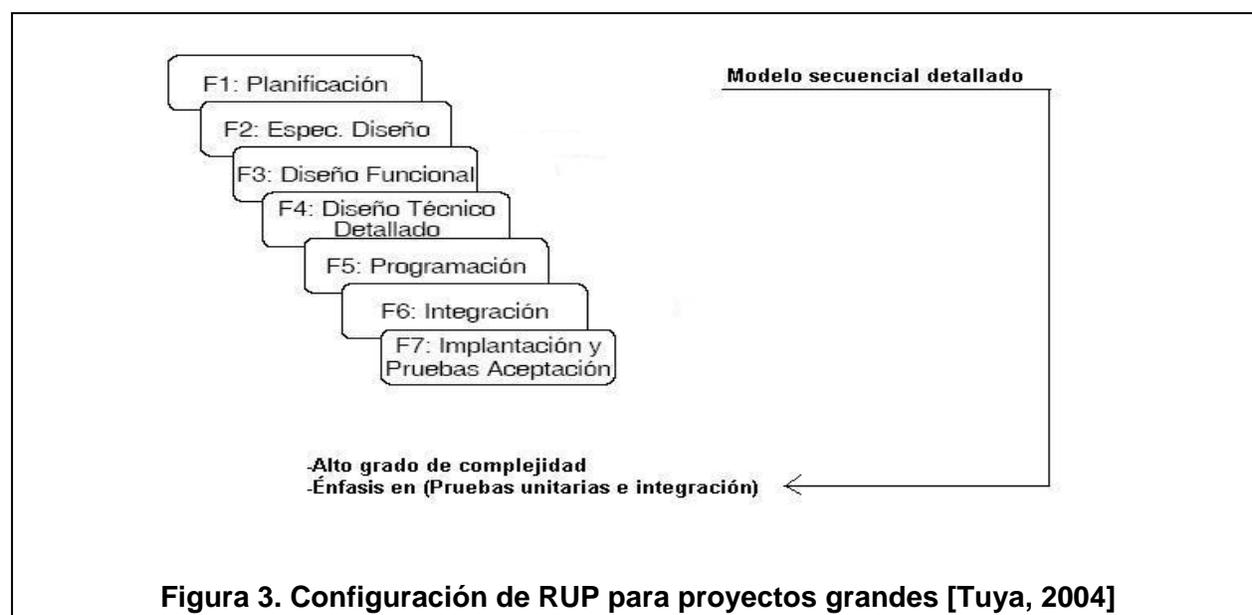
1. Flujos de Trabajo: se utilizarán Diagramas de Actividad para modelar los Flujos de Trabajo del área con problema, tanto los actuales; previos a la implantación del nuevo sistema, como los propuestos; que serán soportados por el sistema desarrollado.
2. Características del Producto Software: es una lista de las características principales del producto, deseables desde una perspectiva de las necesidades del cliente.
3. Glosario de Términos: es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología común.
4. Modelo de Casos de Uso: este modelo presenta la funcionalidad del sistema y los actores que hacen uso de ella. Se representa mediante Diagramas de Casos de Uso.

5. Especificaciones de Casos de Uso: para los casos de uso que lo requieran, cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa, se realiza una descripción detallada utilizando una plantilla, donde se incluyen: precondiciones, postcondiciones, flujo de eventos, requisitos no funcionales asociados.
6. Modelo de Análisis y Diseño: este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación), hacia una de diseño (incluyendo una orientación hacia el entorno de implementación). Está constituido esencialmente por un Diagrama de Clases y algunos Diagramas de Estados para las clases que lo requieran.
7. Modelo Lógico Relacional: previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Tablas donde se muestran las tablas, claves, etc.
8. Modelo de Implementación: este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.
9. Modelo de Pruebas: para cada Caso de Uso se establecen pruebas de Aceptación que validarán la correcta implementación del Caso de Uso. Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados.
10. Manual de Instalación: este documento incluye las instrucciones para realizar la instalación del producto.
11. Material de Usuario: corresponde a un conjunto de documentos y facilidades de uso del sistema para el usuario.
12. Producto: todos los ficheros fuente y ejecutables del producto.

Configuración de RUP para proyectos de gran envergadura (Ver figura 3).

Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí si toma sentido el basarse en una metodología de desarrollo, y se empieza a buscar cual será la más apropiada para el caso. Lo cierto es que muchas veces no se encuentra la más adecuada y se termina adaptando una ya existente a las necesidades del proyecto que se desea desarrollar.

Mientras más grande es el proyecto, más formal tiene que ser la especificación y por lo general, se utiliza RUP como metodología de trabajo, la cual obliga a utilizar UML como notación específica para el producto de software. Si el proyecto es suficientemente grande como para compensar la adaptación a RUP, es una buena base para el proceso, consiguiendo una mejor estructura y disciplina del proceso de desarrollo. Una buena posibilidad de reducir el trabajo a realizar es la reutilización de modelos, procesos, etc. ya definidos en utilizaciones previas de RUP en distintos ámbito.



En esta posible forma de RUP para abordar proyectos grandes se tienen en cuenta todos los artefactos y roles que propone RUP para el desarrollo de un software, en función de lograr un producto de alta calidad. Puede obviarse o no un artefacto o rol específico, eso es decisión de los involucrados en el proyecto, pero de forma general se tiene en cuenta lo definido por esta metodología de desarrollo, pues RUP fue creada para proyectos grandes a pesar de su posibilidad de adaptación a las exigencias de cualquier proyecto.

Para aplicar RUP en pequeños equipos y proyectos se deberá mapear los diferentes roles entre los distintos miembros del equipo, pero la diferencia clave con un proyecto de mayor envergadura, es el grado de formalidad a la hora de usar los distintos artefactos, planes del proyecto, requisitos, clases [RUP, 2006].

1.5 Expediente de proyecto.

Cuando se comienza la realización de cualquier actividad o proceso se piensa en una forma de que este quede explicado y guardado en papel para su posterior consulta y uso, por tal motivo se crea un expediente, donde se incluye todo lo referente al proceso que se pretende desarrollar. Almacenar esta información en formato duro es algo arcaico, pues con el avance de las Tecnologías de la Informática y las Comunicaciones (TICs) existen diferentes tipos de formatos digitales que se pueden utilizar para salvar la información generada. De ahí la necesidad de gestionar un expediente de forma digital, lo cual permite el uso compartido de la información dentro de una organización, y garantiza un almacenamiento de datos más cómodo y seguro.

Un expediente de proyecto se abre cuando el cliente solicita un servicio para realizar un proyecto. Normalmente el expediente se compone de un único proyecto, pero también puede ocurrir en otros casos que sea preciso por la envergadura del mismo realizar varios expedientes. En este caso se crearían otros expedientes para los nuevos

proyectos, de manera tal que quede proporcionado la cantidad de proyectos con la cantidad de expedientes.

Expediente del proyecto: contiene la información técnica generada durante el desarrollo del proyecto y respalda la correcta aplicación de los recursos del proyecto. Su integración, actualización y resguardo son obligatorios para los responsables técnico y administrativo del proyecto [Sistema, 2006].

1.5.1 Expediente de DeSoft S.A.

La Empresa DeSoft S.A. surgió a partir de la integración y asociación de capital de entidades que funcionaban de manera aislada, pertenecientes al Grupo de Tecnologías de la Información (GTI) que pertenecía al Ministerio de la Informática y las Comunicaciones. Su objetivo radica en la comercialización en el mercado nacional e internacional de productos y servicios, para lo cual cuenta con 15 divisiones territoriales, repartidas en todas las provincias, y una casa matriz en Ciudad de La Habana.

DeSoft S.A. facilita una amplia gama de productos y servicios como la implantación de sistemas informáticos para la gestión empresarial, desarrollo de aplicaciones para Internet y comercio electrónico, consultorías y asesorías especializadas en tecnologías de la información, publicaciones de soporte magnético y multimedia, cursos y adiestramientos, así como planificación, desarrollo y evaluación de proyectos integrales especializados [Santos, 2004].

A continuación se muestra la estructura que presenta el expediente aplicado en el desarrollo de proyectos de producción de software en esta institución, el cual se basa en RUP ajustándose a las necesidades de la entidad:

- Ingeniería de software
 - Artefactos resultantes del proceso de Ingeniería de software organizados por flujos de trabajo (negocio, requisitos, análisis, diseño, etc).
- Planificación
 - Cronograma
 - Riesgos
 - Información del proyecto
 - Recursos
- Gestión de configuración
 - Control de versiones
 - Herramientas para el control de versiones
 - Bibliotecas de trabajo
 - Administración de ECS (elementos de configuración del software)
- Aseguramiento de la calidad
 - Administración de la calidad del software
 - Plantillas
 - Evaluación de la calidad del software

1.5.2 Expediente de la Infraestructura Productiva (IP) de la UCI.

En la Infraestructura Productiva (IP), área principal de producción de software en la UCI, se está desarrollando una propuesta de expediente de proyecto, que aún no se ha puesto en práctica porque no se ha seleccionado la herramienta que se va a utilizar para soportar el proceso de GD. Hasta el momento lo que se tiene pensado está establecido de la siguiente forma:

- Ingeniería de software
 - Artefactos resultantes del proceso de Ingeniería de software, organizados por flujos de trabajo (negocio, requisitos, análisis, diseño, etc).

- Gestión de proyectos.
 - Contratación
 - Planificación
 - Riesgos
 - Cronograma
 - Recursos
 - Información del Proyecto
 - Información del cliente
 - Distribución de laboratorios
 - Horarios
- Soporte
 - Gestión de configuración
 - Solicitudes de cambio
 - Elementos de configuración
 - Repositorios
 - Aseguramiento de la calidad
 - Estándares de información
 - Estándares de codificación
 - Metodologías
 - Patrones
 - Plantillas.

(Aquí estarían disponibles todas las plantillas que se vayan a utilizar por el proyecto pero no se ha previsto como se organizarían)

- Gestión de no conformidades

(No está diseñado aún)

1.5.3 Expediente de la Facultad 3.

En la facultad 3 de la UCI, se ha realizado un prototipo de expediente de proyecto para ser implantado en todos los proyectos de la facultad con el objetivo de estandarizar este proceso, pero éste no se utiliza, y cada quien organiza la información que se genera del desarrollo de un software a su estilo. Además el expediente que posee dicha facultad presenta algunas partes que no están bien definidas y existen otras de más, debido a que fue elaborado por un proyecto de esta facultad sin generalizar las necesidades de todos los proyectos de la misma.

Estructura que presenta este expediente:

- Desarrollo técnico: todos los elementos relacionados con la solución del proyecto.
 - Artefactos de ingeniería de software
 - Fuentes
 - Herramientas utilizadas para el desarrollo del producto
 - Los ejecutables y la ayuda
 - Uso de herramientas para el control de versiones y el trabajo en equipo: SourceSafe y Subversion.
- Planificación
- Gestión de configuración: todos los elementos relacionados con garantizar la propia calidad y evolución del proyecto a partir de la integración de las tres funciones anteriores.
 - Establecimiento de procedimientos generales que regulan las funciones anteriores
 - El uso de los servidores de la universidad para la organización y las copias de los proyectos
- Seguridad Informática
- Gestión de proyectos: elementos relacionados con la organización del proyecto.
 - Información de los recursos humanos

- Correos
 - Decisiones
 - Informes de balance
 - Mecanismos de chequeo de la evolución
- Sistema de calidad: los elementos relacionados con garantizar la calidad del proyecto.
- Listas de comprobación y listas de chequeo
 - Planificación de las pruebas y los casos de uso de pruebas

1.6 Herramientas para la Gestión Documental de Proyectos.

No cabe duda que el archivo físico tradicional no es el más adecuado para gestionar con agilidad, y total conocimiento las necesidades requeridas en la explotación de información. Gestionar y mantener organizado un archivo físico de expedientes acarrea grandes problemas como [Integrado, 2007]:

- Dificultad para disponer de información de forma ágil y sencilla
- Pérdida de originales durante la manipulación de estos
- Imposibilidad de su utilización de forma simultánea

La no materialización de la información en su soporte tradicional surge del uso extendido de la informática en todos los ámbitos. Si bien en la actualidad hay herramientas para realizar esta gestión, abordar proyectos encaminados a la gestión documental involucra aspectos adicionales a las herramientas informáticas, la redefinición de procesos, la seguridad en términos de confidencialidad, integridad y disponibilidad de la información, y en general un cambio total de la cultura del papel. En un marco ambicioso, se espera que dichos proyectos abarquen el ciclo vital de los documentos desde su etapa de creación y/o recepción hasta su disposición final. Una de las maneras de lograr este nivel de control y seguridad en el medio electrónico, es utilizando herramientas de gestión documental [Bárcenas, 2005].

La Gestión Documental de Proyectos adquiere cada vez mayor relevancia, debido a la necesidad de mejorar los procesos de los proyectos de forma continua. Para ello es necesario contar con una herramienta que permita gestionar toda la información que se maneja en el proyecto de una manera eficiente, segura y desde diferentes puntos distantes entre si. La introducción de las TIC ha permitido el desarrollo de herramientas cada vez más sofisticadas para manejar esos volúmenes de información.

En toda organización donde se requiera la implantación de una herramienta de gestión de documentos es importante una fase de estudio y análisis que precise los requisitos necesarios para el diseño e implantación de un sistema de este tipo dentro de la institución, para que la herramienta elegida resulte la más práctica según las necesidades y recursos de la organización. Éstas brindan varias ventajas tales como: ahorro en espacio físico y equipamiento, reducción de costos salariales, administrativos y de los derivados del uso del papel, acceso centralizado y consulta distribuida, rápida localización de los documentos por múltiples vías de acceso y rápido intercambio de información.

Las herramientas para la gestión documental se apoyan a su vez en técnicas de gestión del flujo de trabajo, para controlar la forma en que las personas crean y modifican los datos proporcionando los medios para que se ejecuten las acciones correspondientes. "Comprender la información es poder" y para esto son necesarias nuevas herramientas que lo posibiliten [Lloret, 2004].

Características generales de una herramienta utilizada para la GD.

1. Compatible con la plataforma corporativa y capaz de funcionar en entornos cliente/servidor de arquitectura web.
2. Seguridad.
3. Establecer categorías y un registro de usuarios.

4. Escalabilidad.
5. Controlar el documento desde la propia génesis, el almacenamiento y las futuras consultas.
6. Registro de acceso para obtener estadísticas acerca del uso de los documentos.
7. Migración de documentos.
8. Recuperar en múltiples formatos.
9. Normalización de la documentación para su almacenamiento.
10. Acceso y búsqueda de los documentos.

Esta investigación se centra en analizar varias de las herramientas de GD que existen en la actualidad, con el objetivo de seleccionar la más apropiada para dar soporte a un software, de acuerdo a las características definidas anteriormente.

Las herramientas que se describen a continuación, fueron seleccionadas de una amplia gama de productos del mercado. Estas incorporan en mayor o menor medida los diferentes campos que abarca la GD: control de versiones, gestión electrónica de documentos, gestión de imágenes, etc.; utilizando los últimos avances de la tecnología. Haciendo un análisis de las mismas se arrojarán a resultados que contribuirán a la toma de decisiones durante el desarrollo de la propuesta.

➤ **Sistema de Gestión de Contenidos (CMS).**

La diversidad de CMS de código abierto que existe es extraordinariamente amplia, cada uno de ellos poseen las funcionalidades generales de cualquier manejador de contenidos, además de características particulares que los identifican. Habitualmente todo el software de código abierto es de acceso libre, sin ningún coste en licencias, lo cual es muy ventajoso en este caso. Muchos de estos CMS son utilizados como herramientas de gestión documental, en dependencia de las necesidades de la organización que lo vaya a utilizar. Permiten la recuperación y reutilización de

documentos, y en general de cualquier objeto publicado o almacenado. Presentan control de acceso gestionando permisos aplicados a grupos o individuos. Dentro de los CMS más conocidos se encuentran:

- Plone.

Plone permite que se creen y adicionen diferentes tipos de contenido; todos los contenidos son adicionados y editados de forma similar. Como miembro del sitio se cuenta con una carpeta donde se puede guardar el contenido creado. Permite el trabajo con documentos, imágenes, ficheros, vínculos, tópicos, carpetas y artículos noticiosos. Puede utilizarse como servidor intranet o extranet, un Sistema de Publicación de documentos y una herramienta de trabajo en grupo para colaborar entre entidades distantes. Realiza el proceso de gestión de objetos en el sitio mediante un sistema de flujo de trabajo por defecto basado en los estados de los objetos y los roles de usuario [Project, 2006].

- Mambo.

Mambo es una aplicación escrita en lenguaje PHP. Permite la creación y mantenimiento de sitios web y portales de manera fácil y dinámica, permitiendo al dueño o administrador de una página web la simplicidad para actualizarla y hacerla accesible a todo tipo de usuarios a través de una variedad de instrumentos. La simplicidad de Mambo radica en que no son necesarios conocimientos técnicos ni especializados para crear, mantener, actualizar o personalizar los contenidos de un sitio Web [Mambo, 2004].

- Drupal.

Drupal es una plataforma dinámica para la construcción de sitios Web que permite a un individuo o una comunidad de usuarios publicar, manejar y organizar una variedad de contenido. Integra muchas características populares de los CMS, weblogs, herramientas de colaboración y comunidad de discusión, todo en un solo paquete fácil de utilizar. Como software de código abierto desarrollado y mantenido por una comunidad, Drupal es libre para descargarlo de Internet y usarlo [Drupal, 2007].

- OpenCMS.

OpenCms está basado en Java y en tecnología XML. Se trata de una aplicación CMS con características tales como Entorno de trabajo basado en navegador web, Gestión de activos, Sistemas de gestión de usuarios y permisos integrados, Publicación de contenidos basada en proyectos, Gestión de Workflow y tareas, Editor WYSIWYG, Soporte a la internacionalización, Versionado del contenido, Mecanismos de plantillas JSP y XML, Soporte Multi-idioma, Sistema de Ayuda Online, Publicación dinámica y estática de contenidos, Personalización, Sistemas de cacheo integrados, Mecanismo modular para las extensiones, Sistema de programación de trabajos, Mecanismo de Sincronización, Importación y Exportación de Contenidos, e Integración con el servidor de aplicaciones [OpenCms, 2006].

➤ **Wiki.**

Si bien es cierto que se puede utilizar un CMS para hacer gestión documental, existen alternativas especializadas que probablemente son más apropiadas. En el caso del wiki, sus aplicaciones están más orientadas a la gestión documental, que exige mucha más organización y ordenación, y son sumamente colaborativas.

Básicamente un wiki es un almacén de páginas web donde todos los usuarios pueden escribir y modificar los contenidos. Una buena cualidad es la de poder incorporar lo documentado anteriormente, haciéndolo a la vez más dinámico, y ágil de captar, debido a su capacidad de poder ‘colgar’ esa misma documentación, adaptándola para una mejor comprensión, sin a cambio mucho esfuerzo para de una forma inmediata poder seguir la labor que se está efectuando en ese momento. El wiki, da la posibilidad de ver de una forma muy rápida desde cualquier sitio, obtener esa información, detectar si el procedimiento es el apropiado, y si no, poder tomar medidas [Colaborativo, 2006].

➤ **GIT-DOC.**

GIT-DOC es la integración de las soluciones que tradicionalmente se habían desarrollado de forma independiente para satisfacer las diferentes demandas de la gestión documental, que principalmente son: la captura e indexación de documentos, la recuperación de la información, la gestión de contenidos, y la automatización de flujos de trabajo. Se ha desarrollado utilizando los estándares tecnológicos de facto en el mercado lo que permite la integración y conexión con otros sistemas y aplicaciones corporativas, logrando que los diferentes procesos de negocio se integren independientemente de las plataformas en las que se soportan.

GIT-DOC es un sistema de gestión documental integral, completamente adaptable a las necesidades específicas de cada compañía y de cada proyecto de gestión documental, pudiéndose adaptar a cualquier requerimiento, desde la automatización de un proceso de trabajo específico, a la evolución de una organización tradicional hacia una “empresa sin papeles”.

Es lo suficientemente flexible y escalable, de tal modo que permite la integración de nuevas funcionalidades de acuerdo a la evolución de las necesidades futuras de la empresa y a los cambios tecnológicos que se produzcan [GIT-DOC, 2007].

Características de esta herramienta:

1. Parametrización del sistema: la implantación comienza con el profundo análisis de las necesidades específicas de cada cliente, creando un sistema a la medida sin modificar los códigos fuente.
2. Captura de información en cualquier tipo de soporte: papel, documentos ofimáticos, correo electrónico, fax, audio, vídeo.
3. Indexación de la información obtenida con múltiples técnicas: video grabación, OCR, código de barras.

4. Consulta de la información mediante atributos estructurados, texto libre o la combinación de ambos, y presentación en múltiples formatos.
5. Creación y distribución automática de documentos.

➤ **Extensión de Gestión de Documentos para Microsoft Exchange (DME).**

Es un sistema de administración empresarial que tiene la capacidad de almacenar cualquier tipo de documentos, desde el momento mismo de su creación hasta que sea necesaria su destrucción, asegurando que toda la información estará disponible durante ese lapso para los integrantes de la organización que necesiten tener acceso a ella.

DME cuenta con el soporte del servidor Microsoft Exchange Server, que sirve de repositorio de toda la información de la empresa, y al que complementa con los datos adicionales de configuración, acceso y permisos que se hayan especificado para esos documentos. Además, DME cuenta con una interfaz totalmente integrada a Windows, se adapta a las aplicaciones de Microsoft Office y también provee una interfaz web, que permite realizar búsquedas, recuperar y guardar documentos on-line para que todos en su empresa dispongan de los datos que requieran [DME, 2007].

Características de esta herramienta:

1. Realizar búsquedas bajo criterios que usted mismo define.
2. Otorgar permisos a los usuarios sobre sus documentos, independientemente de la plataforma que utilicen.
3. Manejar distintas versiones de un mismo documento.
4. Llevar un historial de los accesos y modificaciones a sus documentos.

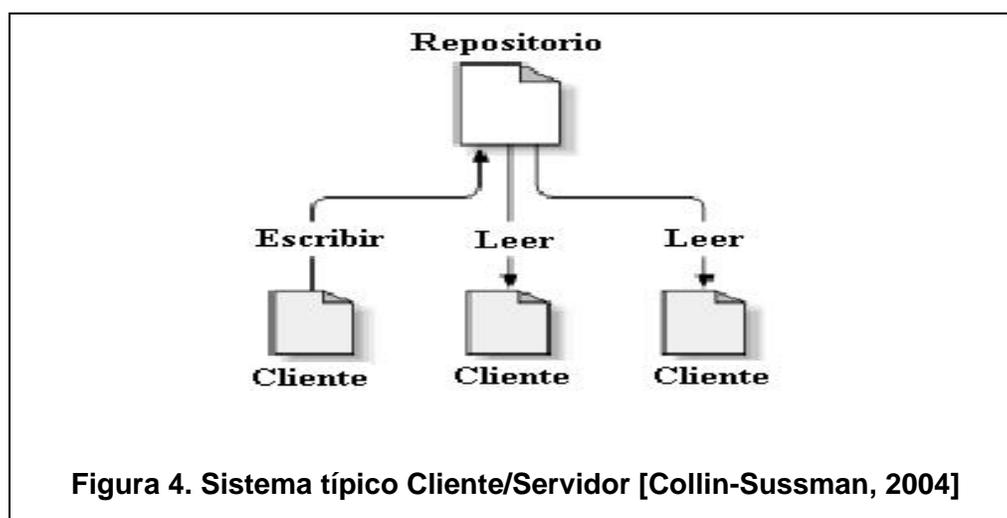
➤ **Subversion (SVN).**

Subversion es un sistema de control de versiones libre y de código fuente abierto, maneja ficheros y directorios a través del tiempo. Hay un árbol de ficheros en un

repositorio central. El repositorio es como un servidor de ficheros ordinario, excepto porque recuerda todos los cambios hechos a sus ficheros y directorios. Ésto le permite recuperar versiones antiguas de sus datos, o examinar el historial de cambios de los mismos.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintos ordenadores. A cierto nivel, la capacidad para que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Si se ha hecho un cambio incorrecto a los datos, simplemente se deshace ese cambio [Collins-Sussman, 2004].

Subversion es un sistema centralizado para compartir información. La parte principal de Subversion es el repositorio, el cual es un almacén central de datos. El repositorio guarda información en forma de árbol de archivos, una típica jerarquía de archivos y directorios. Cualquier número de clientes puede conectarse al repositorio y luego leer o escribir en esos archivos. Al escribir datos, un cliente pone a disposición de otros la información; al leer datos, el cliente recibe información de otros. (Ver figura 4)



La figura da la idea que se tiene un sistema de ficheros compartidos en el cual solamente se puede leer y escribir, esto no es lo básico del SVN, lo que lo hace especial es que se puede acceder a todas las versiones que se han creado anteriormente para un documento.

Características de esta herramienta:

1. Versionado de directorios.

Subversion implementa un sistema de ficheros versionado virtual que sigue los cambios sobre árboles de directorios completos a través del tiempo. Ambos, ficheros y directorios, se encuentran bajo el control de versiones.

2. Verdadero historial de versiones.

Con Subversion, se puede añadir, borrar, copiar, y renombrar ficheros y directorios. Y cada fichero nuevo añadido comienza con un historial nuevo, limpio y completamente suyo.

3. Envíos atómicos.

Una colección cualquiera de modificaciones o bien entra por completo al repositorio, o bien no lo hace en absoluto. Esto permite a los desarrolladores construir y enviar los cambios como fragmentos lógicos e impide que ocurran problemas cuando sólo una parte de los cambios enviados lo hace con éxito.

4. Versionado de metadatos.

Cada fichero y directorio tiene un conjunto de propiedades asociado a él. Se puede crear y almacenar cualquier par arbitrario de clave/valor que se desee. Las propiedades son versionadas a través del tiempo, al igual que el contenido de los ficheros.

5. Manipulación consistente de datos.

Subversion expresa las diferencias del fichero usando un algoritmo de diferenciación binario, que funciona idénticamente con ficheros de texto y ficheros binarios. Ambos tipos de ficheros son almacenados igualmente comprimidos en el repositorio, y las diferencias son transmitidas en ambas direcciones a través de la red.

1.7 Selección de la herramienta.

De acuerdo a las características que presenta SVN y las necesidades del proceso de GD de los proyectos productivos de la facultad 3, se selecciona esta herramienta.

El núcleo de cualquier sistema de gestión documental es el repositorio de documentos que permite gestionar y acceder a la información fácilmente. Con Subversion se pueden ir guardando los archivos en un repositorio. Cada vez que se hace algún cambio el sistema asigna una versión a este. Si se trabaja en equipo cualquier miembro del grupo podrá bajar los últimos archivos, y subir nuevos cambios de acuerdo a los permisos que posea. Así todo se mantiene más organizado y se da una máxima trazabilidad al proyecto.

En caso que se haya trabajado en un archivo por un determinado espacio de tiempo, cambiándolo varias veces, pero por alguna razón se necesite el archivo original nuevamente, solo hay que bajar del servidor la versión que se necesite. Con Subversion no se necesita tener en el ordenador 20 versiones distintas del mismo archivo, solo es necesaria la versión en la que se trabaja actualmente, y cuando se necesite una versión anterior, solo se pide del servidor.

1.8 Aspectos a tener en cuenta para trabajar con SVN.

Esta sección tatará aspectos que se deben tener en cuenta a la hora de comenzar el trabajo con la herramienta SVN.

Clientes para la interacción con SVN

Existen clientes que facilitan la interacción con la herramienta SVN. Muchos son multiplataforma: JSVN, RapidSVN, QSVN, etc. Otros corren sobre un sistema específico: WebSVN (Web), GSNV (Linux), SVNX (Mac), TortoiseSVN (Windows).

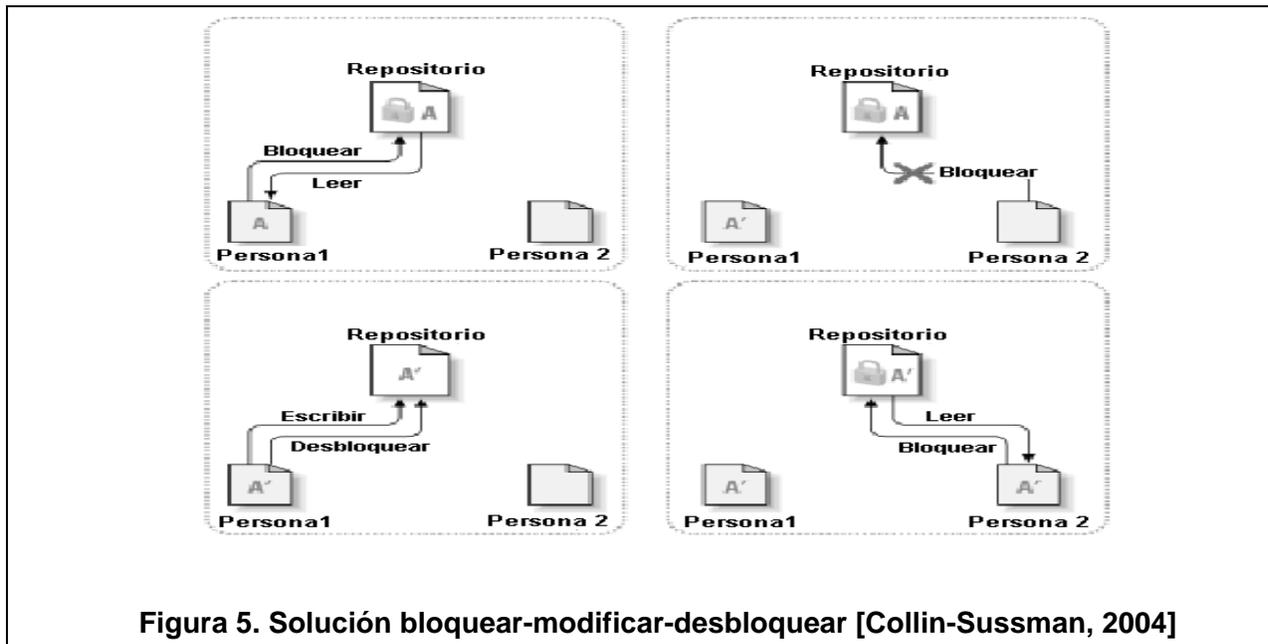
El cliente escogido en esta investigación es TortoiseSVN, el cual es un cliente gráfico integrado al explorador de carpetas de Windows, convirtiendo el mismo explorador en un cliente Subversion. Es realmente rápido y cómodo de usar. Además puede ser usado sin un entorno de desarrollo. Es software libre liberado según la licencia GNU GPL. Este programa tiene varias opciones para poder interactuar con SVN.

Compartir archivos

Para compartir información a los usuarios, y la vez impedir que estos sobrescriban accidentalmente los cambios de los demás en el repositorio, SVN presenta dos soluciones:

Solución Bloquear-modificar- desbloquear.

El repositorio sólo permite a una persona modificar un archivo al mismo tiempo. Persona1 debe bloquear primero el archivo para luego empezar a hacerle cambios. Bloquear un archivo se parece mucho a pedir prestado un libro de la biblioteca; si Persona1 ha bloqueado el archivo, entonces Persona2 no puede hacerle cambios. Por consiguiente, si esta última intenta bloquear el archivo, el repositorio rechazará la petición. Todo lo que puede hacer es leer el archivo y esperar a que Persona1 termine sus cambios y deshaga el bloqueo. Tras desbloquear Persona1 el archivo, Persona2 puede aprovechar su turno bloqueando y editando el archivo. La figura 5 demuestra esta sencilla solución.

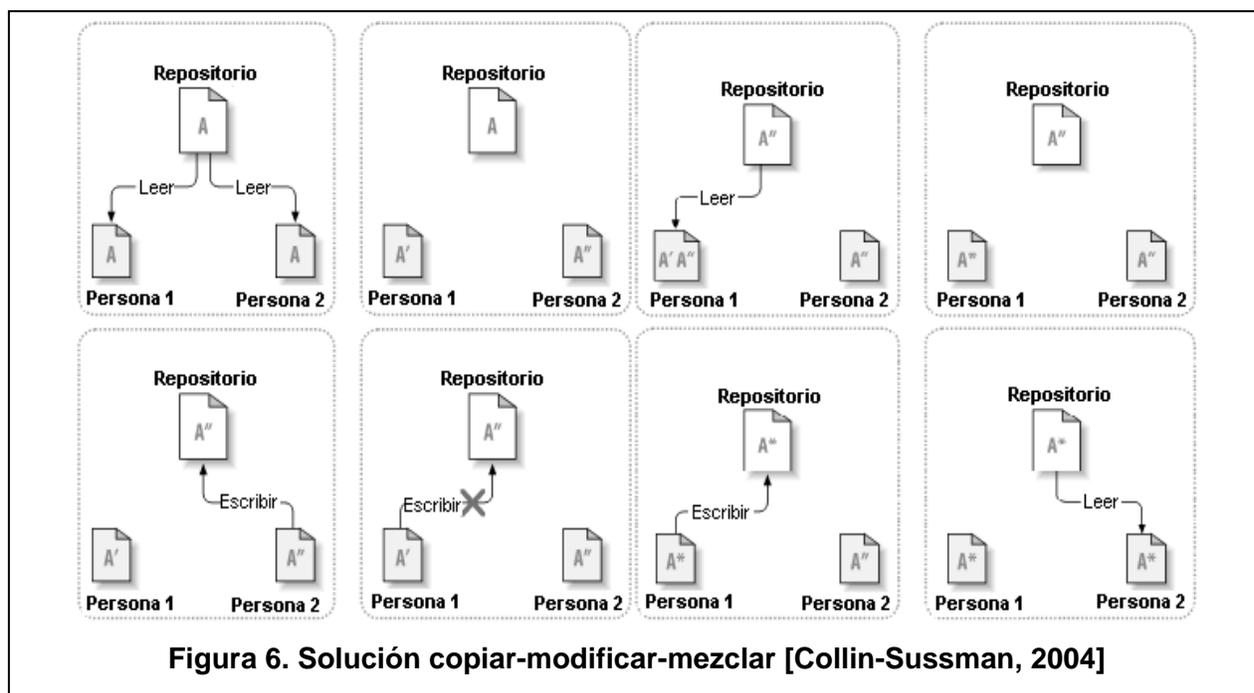


Solución Copiar-modificar-mezclar.

Subversion utiliza un modelo del tipo copiar-modificar-mezclar como alternativa al bloqueo. En este modelo, el cliente de cada usuario se conecta al repositorio del proyecto y crea una copia de trabajo personal: una réplica local de los archivos y directorios del repositorio. Los usuarios pueden entonces trabajar en paralelo, modificando sus copias privadas. Finalmente, todas las copias privadas se combinan o mezclan en una nueva versión final. El sistema de control de versiones a menudo ayuda con la mezcla, pero en última instancia es un ser humano el responsable de hacer que esto suceda correctamente.

La solución copiar-modificar-mezclar puede sonar un tanto caótica, pero en la práctica funciona extremadamente bien. Los usuarios pueden trabajar en paralelo, sin tener que esperarse el uno al otro. Cuando trabajan en los mismos archivos, sucede que la mayoría de sus cambios concurrentes no se solapan en absoluto; los conflictos son poco frecuentes. El tiempo que toma resolver los conflictos es mucho menor que el tiempo perdido por un sistema de bloqueos. Por tal motivo se recomienda utilizar esta última

solución, por que la primera es un tanto restrictiva y a menudo se convierte en un obstáculo para los usuarios. La figura 6 muestra este proceso.



Copias de trabajo

Una copia de trabajo de Subversion es un árbol de directorios corriente de su sistema de archivos local, conteniendo una colección de archivos. Se pueden editar estos archivos del modo que se prefiera y si se trata de archivos de código fuente, se podrá compilar su programa a partir de ellos de la manera habitual. La copia de trabajo es un área de trabajo privada: Subversion nunca incorporará los cambios de otra gente o pondrá a disposición de otros sus cambios hasta que usted le indique explícitamente que lo haga.

Tras hacer algunos cambios a los archivos en la copia de trabajo privada y verificar que funcionan correctamente, Subversion le proporciona comandos para publicar sus cambios al resto de personas que trabajan en el proyecto, escribiendo en el repositorio. Si las demás personas publican sus propios cambios, Subversion le proporciona

comandos para mezclar estos cambios en su directorio de trabajo, leyendo del repositorio. Una persona específica puede conseguir una copia de trabajo, de acuerdo a los permisos que posee, si ejecuta un *check out* de algún subárbol del repositorio.

Comandos Update y Commit

Para tener el proyecto actualizado, se puede pedir a Subversion que proceda a actualizar la copia de trabajo, usando para ello el comando update. Esto incorporará los cambios hechos en la copia de trabajo privada, así como otros cambios consignados desde que esta hizo el check out.

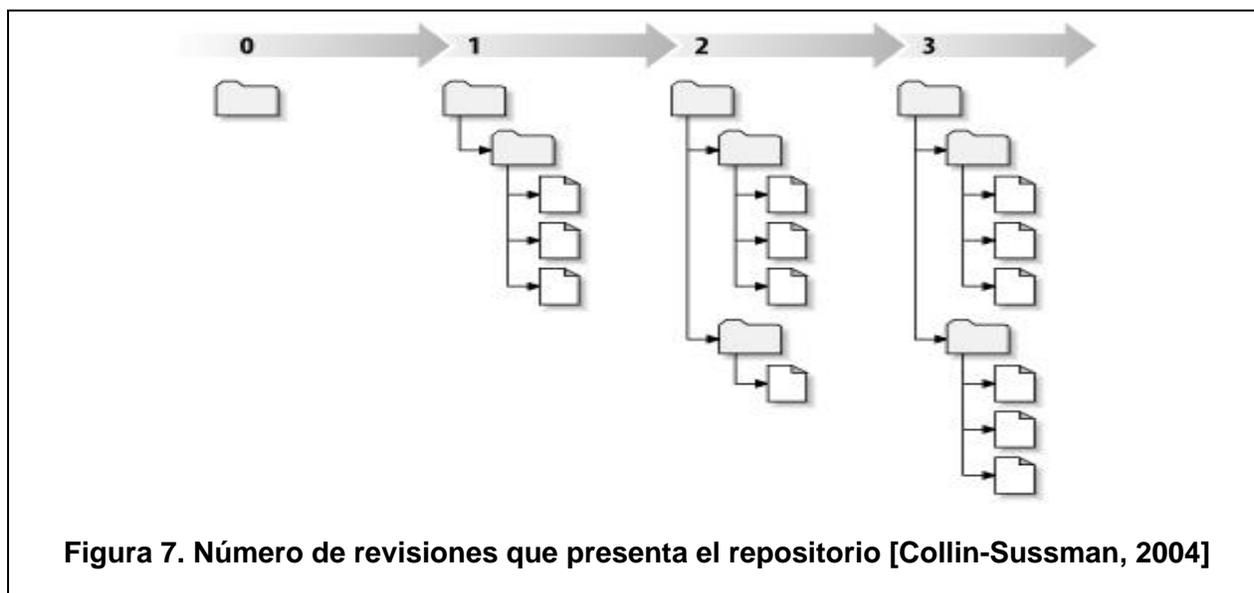
Después de actualizar la copia de trabajo privada, se procede a utilizar el comando commit para publicar los cambios realizados por una persona específica al resto del equipo de trabajo.

Por tanto cuando se realiza un cambio en cualquier archivo del repositorio de SVN, primeramente se debe ejecutar el comando Update, para actualizar la información, y luego el Commit para compartir la misma con el resto de los integrantes del proyecto.

Revisiones

En el repositorio, cada cambio es tratado como una transacción atómica: o bien se realizan todos los cambios, o no se realiza ninguno. Subversion trata de conservar esta atomicidad para hacer frente a posibles fallos del programa, fallos del sistema, problemas con la red, y otras acciones del usuario. Cada vez que el repositorio acepta un envío, éste da lugar a un nuevo estado del árbol de ficheros llamado revisión. A cada revisión se le asigna un número natural único, una unidad mayor que el número de la revisión anterior. La revisión inicial de un repositorio recién creado se numera con el cero, y consiste únicamente en un directorio raíz vacío.

La figura 7 ilustra una forma interesante de ver el repositorio: un arreglo de números de revisión, comenzando por el 0, que se extiende de izquierda a derecha; cada número de revisión tiene un árbol de ficheros colgando debajo de él, y cada árbol es una instantánea del aspecto del repositorio tras cada envío.



A diferencia de muchos otros sistemas que controlan versiones, los números de revisión de Subversion se aplican a árboles enteros, no a ficheros individuales. Cada número de revisión selecciona un árbol completo, un estado particular del repositorio tras algún cambio publicado.

1.9 Conclusiones.

Luego de haber analizado la Gestión Documental de Proyectos de Software, se arriba a la conclusión que es de vital importancia gestionar la documentación generada en un proyecto de producción de software, lo cual contribuye a su organización y realización exitosa. Después de valorar las herramientas que soportan el proceso de Gestión Documental, se seleccionó SVN porque se ajusta a las necesidades de la facultad donde se desarrolla la investigación.

CAPÍTULO 2: MONTAJE DE LA PROPUESTA DE EXPEDIENTE.

2.1 Introducción.

En este capítulo se describe la propuesta de expediente de proyecto elaborada a partir del análisis realizado en el capítulo anterior. También se detalla el proceso de instalación y configuración de SVN en función del prototipo de expediente propuesto.

2.2 Propuesta de expediente de proyecto.

Con el objetivo de garantizar una estructura estándar para la gestión de la documentación generada durante el desarrollo de un software, este trabajo propone un expediente de proyecto. La estructura que presenta está dividida en 4 módulos principales (Ver figura 8). Estos módulos a su vez se dividieron en submódulos, y estos últimos contienen las plantillas con la descripción de los artefactos generados durante este proceso, asegurando una organización detallada y mayor seguridad.



Figura 8. Módulos principales del expediente de proyecto

Estructura detallada de la propuesta (Ver figura 9).

1. Gestión de Proyecto: se guardan las actas de reuniones, acuerdos, planificaciones del proyecto y, la documentación resultante de la comunicación interna entre los miembros del proyecto y la comunicación externa entre los desarrolladores y los clientes. Se almacena lo referente a la aplicación de métricas, datos y gráficas que muestran la evolución del proyecto. Se guardan además los datos relacionados con los recursos humanos, la capacitación, los roles, los recursos materiales y el control económico del proyecto. Esta carpeta ayuda a organizar la información correspondiente al personal y a la gestión.



Gestión

Contiene la documentación perteneciente al flujo de trabajo Gestión de Proyecto. Tiene como propósito principal lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto acorde a los requisitos de los clientes y los usuarios. La gestión de proyectos proporciona herramientas necesarias para sistematizar el trabajo y asegurar que los riesgos que se puedan correr estén calculados. El principal artefacto generado en esta etapa de trabajo es el Plan de Desarrollo del Software. Además se realizan otras guías para la administración del proyecto y sus actividades. Básicamente cada plan debe incluir: las actividades, los responsables del desarrollo de las mismas y la fecha de cumplimiento. Los roles que intervienen en este flujo de trabajo son: Líder de Proyecto y Planificador de Proyecto.

➤ Plan de Mitigación de Riesgos: esta plantilla contiene lo correspondiente a cómo administrar los riesgos del proyecto. Interviene en su realización el Líder de Proyecto.

Contiene la siguiente estructura:

- Breve descripción del proyecto.
- Resumen de cantidad total de riesgos implicada en el proyecto.

- Tareas de la administración en caso de ser materializado el riesgo.
 - Listado de personas responsables de administrar los riesgos.
 - Presupuesto disponible para mitigar los riesgos.
 - Listado de herramientas y técnicas que serán utilizadas en caso de materializarse el riesgo.
- Plan de Aceptación del Producto: esta plantilla almacena la información relacionada con el proceso de aceptación de un producto. Interviene en su realización el Líder de Proyecto. Contiene la siguiente estructura:
- Responsabilidades del cliente y del equipo de desarrollo en la preparación y ejecución de las actividades de aceptación del producto.
 - Tareas a realizar en el proceso de aceptación del producto.
 - Requerimientos que debe presentar el producto para ser aceptado.
 - Procedimientos a desarrollar para identificar y resolver problemas durante las actividades de aceptación del producto.
 - Planes para el ambiente de aceptación del producto.
 - Evaluación individual realizada a un artefacto, especificando tipo de evaluación y objetivos de la misma.
 - Listado de herramientas, técnicas y metodologías que deben ser utilizadas al realizar las actividades de aceptación del producto.
- Plan de Resolución de Problemas: esta plantilla guarda la información concerniente a cómo se da solución a un problema. Interviene en su realización el Líder de Proyecto. Contiene la siguiente estructura:
- Breve descripción de las tareas de resolución de problemas, ejecutadas en caso que este se materialice. Se especifican los siguientes procedimientos: Divulgación de problemas, Análisis de problemas, y Acciones correctivas

apropiadas. Estos procedimientos se diferencian para diversas categorías de problemas.

- Listado de grupos o individuos específicos encargados del análisis y resolución de cada categoría de problemas, identificando tareas y responsabilidades de cada uno.
- Listado de herramientas y técnicas a utilizar para almacenar la información del problema, analizarlo, y seguir su estado.
- Detalles del producto: defectos, ediciones, no conformidades, etc.

➤ Plan de Desarrollo del Software: esta plantilla acumula toda la información perteneciente al proceso de desarrollo del software. Interviene en su realización el Líder de Proyecto. Contiene la siguiente estructura:

- Resumen del proyecto, especificando propósito, objetivos y alcance del mismo.
- Listado de las suposiciones en las que se basa el plan y cualquier contrariedad.
- Listado de artefactos que se crean durante el proyecto y su fecha de creación.
- Evolución del Plan de Desarrollo de Software a través de una tabla con las diferentes versiones propuestas, los criterios para la revisión y reedición no programada del plan.
- Descripción de la organización del proyecto: Estructura, donde se especifica la organización del equipo de proyecto, incluyendo la administración; Interfaces externas, muestra cómo interactúa el proyecto con los grupos externos; Roles y responsabilidades, identifica los responsables de cada una de las disciplinas, de los detalles del flujo de trabajo, y de los procesos de soporte.

- Administración del proceso: Estimación del proyecto, Plan de proyecto, Plan de Iteración, Control y monitoreo del proyecto, Plan de administración de riesgos, y Plan de cierre_fuera.
- Plan de procesos técnicos: Caso de Desarrollo; Métodos, Herramientas y Técnicas; Plan de infraestructura; y Plan de aceptación del producto.
- Plan del proceso de soporte: Plan de administración de la configuración, Plan de evaluación, Plan de documentación, Plan de aseguramiento de la calidad, Plan de resolución de problemas, Plan de administración de Subcontratistas, y Plan de mejoramiento del proceso.
- Planes adicionales.

➤ Cronograma: esta plantilla contiene un itinerario de cómo se va a desarrollar el proyecto. Es un esquema donde se distribuye y organiza en forma de secuencia temporal el conjunto de experiencias y actividades diseñadas a lo largo del proyecto. Interviene en su realización el Planificador de Proyecto. Contiene la siguiente estructura:

- Actividades.
- Tiempo estimado que se dedicará a cada actividad.
- Herramienta a utilizar en cada actividad.



Actas de reuniones y decisiones

Se guardan las actas de reuniones realizadas con el objetivo de planificar o estimar algún punto del proyecto; en estas actas queda plasmada cualquier decisión tomada en pos de maximizar la probabilidad de consecución de resultados a tiempo, dentro del presupuesto y con la calidad esperada. La documentación que se almacena en esta carpeta no tiene un formato específico para guardarla, no hay una plantilla para las actas de reuniones, se hace un documento que recoja todos los puntos discutidos en la reunión y los acuerdos tomados.



Comunicación

La planificación del proyecto resulta inútil si no se establece una comunicación adecuada entre los integrantes del equipo de proyecto. Cada miembro del equipo necesita conocer sus responsabilidades. Además de que todo lo que se planifique también debe ser comunicado al cliente para ver si está de acuerdo con todo lo estimado: tiempo, costo, etc.; debido a que puede darse el caso que después de empezado el proyecto el cliente no esté de acuerdo con el coste que tiene desarrollar el producto y esto provoca conflictos entre ambas partes. Por tal motivo debe existir una buena comunicación entre los miembros del equipo de desarrollo, y a su vez, de estos con los clientes, para que la negociación se desarrolle de la mejor forma y se logre un entendimiento común por ambas partes de lo que se quiere lograr en realidad. De aquí que esta carpeta se divida en dos a su vez:



Comunicación Interna

Contiene la documentación que pertenece al proceso de comunicación entre los miembros del equipo de desarrollo, puntualizando los responsables de cada actividad dentro del proyecto.



Comunicación Externa

Guarda la documentación que corresponde al proceso de comunicación que se establece entre el equipo de desarrollo y los clientes, donde se incluye el contrato que se realiza antes de comenzar la elaboración del producto, previendo de esta forma un mal entendido entre ambas partes.



Recursos Humanos

Se almacena la información relacionada con los integrantes del proyecto, donde se incluyen los siguientes campos: nombre, rol, nivel, laboratorio, etc. La plantilla tiene

formato Excel, pudiendo ser actualizada en cualquier momento del proceso de desarrollo del software.



Recursos Materiales

Se guarda la información relacionada con los medios de cómputo asignados al proyecto, donde se incluyen los siguientes campos: número de PC, cantidad de RAM, motherboard, monitor, etc. La plantilla tiene formato Excel, pudiendo ser actualizada en cualquier momento del proceso de desarrollo del software.



Planificación

Contiene la información perteneciente al proceso de planificación de un proyecto software. Para poder finalizar con éxito un proyecto se necesita tener un control riguroso sobre el tiempo, las personas y los imprevistos que puedan surgir, como por ejemplo cambios en el software. Antes de todo es necesario planificar con cuidado las tareas y luego organizarlas en orden de prioridad. No se trata de una etapa independiente abordable en un momento concreto del ciclo de vida del proyecto. Es decir, no se puede hablar de un antes y un después al proceso de planificación, puesto que según avance el proyecto será necesario modificar tareas, reasignar recursos, etc. Se debe tener claro que si bien sí se puede hablar de una "etapa de planificación", llamada así porque aglutina la mayor parte de los esfuerzos para planificar todas las variables que se darán cita, cada vez que se intenta prever un comportamiento futuro y se toman las medidas necesarias se está planificando.

➤ **Planificación del Proyecto:** esta plantilla almacena la información necesaria para llevar a cabo la planificación de un proyecto. Interviene en su realización el Planificador de Proyecto. Contiene la siguiente estructura:

- Breve resumen del proyecto especificando objetivos, alcance, características, beneficios que aportará, restricciones que pudiera tener.

- Breve resumen de la metodología a utilizar en el desarrollo del proyecto.
- Organización del equipo de proyecto y de los recursos humanos.
- Herramientas que se usan de manera intensiva durante el desarrollo del proyecto.
- Se describen cómo se controlarán los cambios y el proceso de actualización del Plan del Proyecto.
- Lista de tareas que serán necesarias para el proyecto, dividiendo las tareas en subtareas, hasta lograr el detalle suficiente para reflejar los riesgos y hacer estimaciones de tiempo razonables.
- Listado de entregables del proyecto, con la fecha de entrega en cada caso.
- Calendario del proyecto.
- Listado de los mayores riesgos para el proyecto, planteando qué hacer para resolver o prevenir cada riesgo. En el caso de que no se vaya a hacer nada para mitigar el riesgo debe quedar especificado.
- Planeación de Dependencias del Proyecto, en este apartado se reflejan las respuestas dadas a varias preguntas que ayudan a comprender si el proyecto depende o no de otro.

2. Desarrollo Técnico: se encuentra la documentación, ficheros fuentes y artefactos de software que se van generando durante el desarrollo del proyecto en los diferentes flujos de trabajo de RUP y que constituyen el producto de software como tal.



Modelación del Negocio

Se encuentran las plantillas de los artefactos generados durante este flujo, quedando registrado todo el proceso de modelamiento del negocio de un producto de software. Con la documentación que se genera en este flujo de trabajo definido por RUP se está

asegurando que el producto sea útil; conseguir que encaje de la mejor forma posible en la organización y tener un marco común para los desarrolladores, los clientes y los usuarios finales. Se usará una extensión de UML para modelar el negocio.

En concreto se guarda el modelo de casos de uso de negocio y el modelo de objetos de negocio. Con el primero se captura qué se hace y quién lo hace. Con el segundo se ve cómo se hace. La gran ventaja de modelar el negocio es que se tiene una mejor visión del sistema que se construirá. Los roles que intervienen en este flujo de trabajo son: Analista de procesos del Negocio, Diseñador del Negocio y el Revisor del Modelo del Negocio.

➤ **Modelo de Casos de Uso del Negocio:** en esta plantilla se recoge toda la información referente al modelamiento del negocio. Intervienen en su realización el Analista de procesos del Negocio, el Diseñador del Negocio y el Revisor del Modelo del Negocio. Contiene la siguiente estructura:

- Breve descripción del negocio que permita ilustrar y ubicar de manera general el módulo del negocio que se va a describir.
- Especificación de los actores del negocio con una descripción simple asociada.
- Especificación de los trabajadores del negocio con una descripción simple asociada.
- Representación del diagrama de casos de uso del negocio.
- Especificación de cada caso de uso con su diagrama de actividades correspondiente.
- Modelo de objetos del negocio.

- Especificación de las reglas del negocio: identificarlas dentro del negocio, evaluar si son relevantes dentro del campo de acción que se está modelando e implementarlas en la propuesta de solución.

➤ Documento Visión del Negocio: el propósito de este documento es reunir, analizar y definir necesidades y características del futuro sistema, se acuerda cuál es el problema a resolver, quiénes son los involucrados y qué necesidades tienen. Interviene en su realización el Analista de procesos del Negocio.

➤ Glosario de Términos del Negocio: en este documento se almacena la terminología propia del dominio del negocio, y que dependiendo del proyecto pueden ser términos muy especializados. Además puede usarse para definir un diccionario informal de tipos de datos. Interviene en su realización el Analista de procesos del Negocio.



Requerimientos

Se encuentran almacenadas todas las plantillas de los artefactos generados durante el flujo de trabajo Requerimientos, el cual establece lo que tiene que hacer exactamente el sistema que se quiere construir. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifican. Como parte de los requisitos de usabilidad, se diseña la interfaz gráfica de usuario. Se construyen diagramas de casos de uso para los requisitos funcionales, los no funcionales se describen textualmente en especificaciones suplementarias. Además hay que gestionar los cambios en los requisitos a lo largo de todo el proceso, crear la Visión de Sistema y el Glosario de Términos. Los roles que intervienen en este flujo de trabajo son: Analista de Sistemas, Arquitecto, Especificador de caso de uso y Diseñador de Interfaces de usuario.

➤ Modelo de Casos de Uso del Sistema: en esta plantilla se describen los casos de uso y los actores del sistema; y se representan las relaciones entre ambos. Intervienen en su realización el Especificador de caso de uso y el Analista de sistema. Contiene la siguiente estructura:

- Especificación de los actores del sistema con una descripción simple asociada.
- Representación del diagrama de Casos de Uso del Sistema.
- Descripción detallada de cada caso de uso en función de acción del actor y respuesta del sistema.

➤ Especificación de Requisitos: esta plantilla contiene los requisitos del sistema, tanto los funcionales como los no funcionales. De manera general para ambos tipos de requisitos se sigue el mismo modo de representación. Interviene en su realización el Especificador de caso de uso. Contiene la siguiente estructura:

- Breve descripción del requisito.
- Precedencia: caso de uso del negocio con el que se relaciona.
- Seguimiento: caso de uso del sistema que implementa el requerimiento.

➤ Documento Visión del Sistema: el propósito de este documento es reunir, analizar y definir necesidades y características del sistema de alto nivel. En principio, para un proyecto pequeño, no es necesario tantos apartados si se puede escribir un texto breve y claro que describa lo mismo. Sin embargo, es importante analizar todas las secciones para no olvidar nada. Interviene en su realización el Analista de sistema.

➤ Glosario de Términos del Sistema: en este documento se recoge el vocabulario propio del dominio del sistema, que dependiendo del proyecto pueden ser términos

muy especializados. Además puede usarse para definir un diccionario informal de tipos de datos. Interviene en su realización el Analista de sistema.

➤ Descripción de la Arquitectura: en esta plantilla se define una arquitectura candidata, se crea un esquema inicial de la arquitectura del sistema. Interviene en su realización el Arquitecto del proyecto.

➤ Prototipo de Interfaz de Usuario: esta plantilla recoge la información inicial de cómo va a ser la interfaz de usuario. Interviene en su realización el Diseñador de Interfaz de usuario del proyecto. Contiene la siguiente estructura:

- Descripción del prototipo de interfaz.
- Modelo del prototipo.
- Descripción de herramientas utilizadas en la confección del prototipo.



Análisis y Diseño

Se encuentra guardada la documentación creada durante el flujo de trabajo análisis y diseño. Aquí se traducen los requisitos a una especificación que describe cómo implementar el sistema. Si el sistema usa una base de datos, hay que diseñarla también, obteniendo un Modelo de Datos. El resultado final más importante de este flujo de trabajo será el Modelo de Diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas. Otro producto importante de este flujo es la documentación de la arquitectura software, que captura varias visiones arquitectónicas del sistema. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva.



Consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Los roles que intervienen en el análisis son: Arquitecto, Ingeniero de casos de uso e Ingeniero de componentes.

➤ Modelo de Análisis: en esta plantilla se hace un resumen del análisis del sistema. Intervienen en su realización el Ingeniero de casos de uso, el Ingeniero de componentes y el Arquitecto. Contiene la siguiente estructura:

- Breve descripción del análisis, incluyendo trabajadores y actividades.
- Identificación y descripción de las clases del análisis de cada CU.
- Representación del diagrama de clases de análisis.
- Clases identificadas en paquetes correspondientes a los casos de uso definidos.
- Representación del diagrama de colaboración y el diagrama de secuencia.

➤ Descripción de la Arquitectura (vista Modelo de Análisis): es una actualización del documento confeccionado en el flujo de trabajo requerimientos, dando una vista lógica del modelo de análisis. Interviene en su realización el Arquitecto del proyecto. Contiene la siguiente estructura:

- Representación de los elementos de análisis más importantes para la arquitectura del sistema.
- Descripción de las clases más importantes y su organización en paquetes.
- Descripción de las relaciones de casos de uso más importantes.



El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos, debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. Los roles que intervienen en el diseño son: Arquitecto, Ingeniero de casos de uso, Ingeniero de componentes y Diseñador de Base de Datos.

➤ Modelo de Diseño: se dispone de una plantilla para este producto; es sencillamente la estructuración de los distintos diagramas y modelos que se tengan referentes a la parte de diseño del sistema. Intervienen en su realización el Ingeniero de casos de uso y el Arquitecto. Contiene la siguiente estructura:

- Breve descripción del diseño que permita ilustrar y ubicar de manera general el módulo del diseño que se va a describir.
- Especificación de los trabajadores del diseño con una descripción simple asociada.
- Descripción de los casos de uso del diseño.
- Descripción de las clases del diseño.
- Representación del diagrama de clases de diseño.
- Creación de subsistemas de diseño.
- Interfaz.

➤ Descripción de la Arquitectura (vista Modelo de Diseño): se actualiza el documento de Descripción de la Arquitectura que se creó en el flujo de trabajo Requerimientos, dando una vista lógica del modelo de diseño. Interviene en su realización el Arquitecto del proyecto. Contiene la siguiente estructura:

- Representación de los elementos de diseño más importantes para la arquitectura del sistema.
 - Descripción de las clases más importantes y su organización en paquetes y subsistemas.
 - Descripción de las relaciones de casos de uso más importantes.
- Modelo de Datos: en esta plantilla se describen las clases persistentes del diseño, que son las que pasan a formar parte de este modelo, el cual se puede utilizar como un buen comienzo para el diseño de la base de datos. Interviene en su realización el Diseñador de Base de Datos. Contiene la siguiente estructura:
- Definición de las clases persistentes.
 - Descripción del modelo de datos.
 - Representación del diagrama de modelo de datos.
- Modelo de Despliegue: en esta plantilla se describen los componentes que se generan a partir de las necesidades de recursos físicos del producto que se está construyendo. También se muestran las relaciones existentes entre los nodos del sistema. Intervienen en su realización el Ingeniero de componentes y el Arquitecto. Contiene la siguiente estructura:
- Representación del diagrama de despliegue.
 - Descripción de los recursos a utilizar.
 - Descripción de las relaciones de comunicación.



Implementación

Se encuentra archivada la información que genera el flujo de trabajo implementación. Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y

demás. El resultado final es un sistema ejecutable. Los roles que intervienen en este flujo de trabajo son: Arquitecto, Integrador de sistemas e Ingeniero de componentes.

➤ Modelo de implementación: esta plantilla esboza los componentes claves del modelo correspondiente. Describe la forma de implantar los elementos del modelo de diseño, y la organización de los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación en los lenguajes de programación utilizados. Intervienen en su realización el Ingeniero de componentes y el Arquitecto. Contiene la siguiente estructura:

- Especificación de los trabajadores con una breve descripción asociada.
- Descripción del flujo de trabajo.
- Definición de componentes, entre ellos los que serán reutilizados: clasificación de componentes, definición de relaciones entre componentes, y representación del diagrama de actividades.
- Definición de subsistemas de implementación.
- Representación de un diagrama por cada subsistema de implementación.
- Representación del diagrama de despliegue.

➤ Descripción de la Arquitectura (vista Modelo de Implementación): se actualiza el documento de Descripción de la Arquitectura que se creó en el flujo de trabajo Requerimientos, dando una vista lógica del modelo de implementación. Interviene en su realización el Arquitecto del proyecto. Contiene la siguiente estructura:

- Descripción de la descomposición del software en capas y subsistemas de implementación.
- Vista de la trazabilidad de los elementos de diseño de la vista lógica para la implementación.

- Modelo de Despliegue: en esta plantilla se refina el modelo de despliegue creado en el flujo de trabajo análisis y diseño. Intervienen en su realización el Ingeniero de componentes y el Arquitecto.

- Plan de Integración: en esta plantilla se tratan las construcciones y subsistemas que van a estar dentro de estas. Interviene en su realización el Integrador de sistemas del proyecto. Contiene la siguiente estructura:
 - Lista de los casos de uso que se están construyendo.
 - Lista de los subsistemas de implementación que se pretenden sumar, el orden en el que se enumeran debe coincidir con el orden que se seguirá para su incorporación o construcción.
 - Plan concreto de construcciones donde se especifican qué construcciones se van a crear y qué subsistemas van a formar parte de la misma, así como los criterios de evaluación.



Están todas las plantillas que recogen la información generada durante el flujo de trabajo de prueba. En esta parte se evalúa la calidad del producto que se está desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida. El objetivo de las pruebas no es asegurar la calidad, pero sí evaluarla, y proporcionar una realimentación a tiempo, de forma que las cuestiones de calidad puedan resolverse de manera efectiva en tiempo y coste. Los principales aspectos a ser evaluados en un producto software son la Fiabilidad: resistente a fallos, la Funcionalidad: hace lo que debe, y el Rendimiento: lleva a cabo su trabajo de manera efectiva. Las pruebas pueden hacerse a diferentes niveles dependiendo del objetivo de las mismas: Prueba de unidad, Prueba de sistema y Prueba de aceptación. Los roles que intervienen en

este flujo de trabajo son: Ingeniero de pruebas, Ingeniero de pruebas de integración, Ingeniero de pruebas de sistema e Ingeniero de componentes.

➤ **Modelo de Prueba:** esta plantilla es una representación de lo que será probado y cómo debe hacerse. Intervienen en su realización el Ingeniero de pruebas, el Ingeniero de pruebas de integración y el Ingeniero de pruebas de sistema. Contiene la siguiente estructura:

- Breve descripción del modelo de pruebas, incluyendo trabajadores y sus actividades correspondientes.
- Descripción de las estrategias de prueba que se van a utilizar.
- Listado de ideas de prueba.
- Listado de datos de prueba.
- Clasificación de las pruebas por componentes.
- Tipos de prueba.
- Descripción de aplicaciones o herramientas automáticas que sirvan para probar los diferentes componentes.

➤ **Plan de Prueba:** esta plantilla contiene información sobre los objetivos generales y específicos de las pruebas en el proyecto, así como las estrategias y recursos con que se dotará a esta tarea. RUP diferencia entre un Plan de Prueba global, donde se describen los objetivos y mecanismos que se van a utilizar para el proyecto en general, así como un Plan de Prueba específico para cada iteración donde se especifica qué elementos se deben probar, cuáles son los objetivos que se persiguen con esas pruebas, y la aproximación a utilizar para conseguir esos objetivos. Incluye también una estimación de los recursos necesarios para llevarlos a cabo. Intervienen en su realización el Ingeniero de componentes y el Ingeniero de pruebas. Contiene la siguiente estructura:

- Listado de los requerimientos de prueba.
- Definición de la estrategia de prueba, especificando tipo de prueba y herramienta a utilizar.
- Recursos de acuerdo a los roles del sistema.
- Artefactos.

➤ **Casos de Prueba:** esta plantilla presenta la información correspondiente a los casos de prueba del producto. Contiene una breve descripción de los casos de prueba que se realizan, incluyendo entradas, resultados y un resumen de los mismos. Interviene en su realización el Ingeniero de componentes del proyecto.

➤ **Procedimientos de Prueba:** esta plantilla contiene una descripción de los procedimientos que se siguen a la hora de realizar una prueba, detallando el o los procedimientos que se utilizaron para hacer una prueba. Interviene en su realización el Ingeniero de pruebas del proyecto.

➤ **Resumen de Evaluación de Prueba:** esta plantilla es un resumen de la información que se obtuvo a través de la ejecución de las pruebas. Interviene en su realización el Ingeniero de pruebas del proyecto. Contiene la siguiente estructura:

- Resultado de las pruebas.
- Resultado de las pruebas basadas en requerimientos.
- Resultado de las pruebas basadas en código.
- Acciones sugeridas a realizar basándose en una evaluación de los resultados obtenidos.
- Diagramas que visualizan los resultados de la prueba o medidas claves de la prueba.



Despliegue

Contiene la documentación que se genera del flujo de trabajo despliegue. Se tiene como objetivo producir con éxito versiones del producto y distribuirlo a los usuarios. Los roles que intervienen en este flujo de trabajo son: Administrador de despliegue y Revisor técnico.

➤ Plan de Despliegue: en esta plantilla se describen los pasos a seguir para llevar a cabo el despliegue del producto. Interviene en su realización el Administrador de despliegue. Contiene la siguiente estructura:

- Planificación del despliegue donde se especifican responsabilidades e itinerario.
- Enumeración de fuentes y recursos requeridos para realizar la actividad prevista del despliegue.
- Adiestramiento describiendo el plan y las entradas para entrenar a los usuarios finales con la utilización del producto.

➤ Notas del Release: esta plantilla recoge la información que se obtiene de una versión. Interviene en su realización el Revisor Técnico. Contiene la siguiente estructura:

- Breve descripción de la versión que se presenta, incluyendo características de la misma.
- Listado de productos compatibles con esta versión.
- Mejoras previas a la liberación del producto.
- Descripción de errores o defectos encontrados en esta versión y el remedio para estos, en caso de que existan.

➤ **Manual de Usuario:** esta plantilla guarda la información necesaria para la utilización del producto. Incluye de forma detallada la explicación de todas sus prestaciones. Durante la elaboración del documento debe indicarse con claridad donde procede, y las medidas de seguridad que debe tomar el usuario durante la explotación del producto. Interviene en su realización el Administrador de despliegue. Contiene la siguiente estructura:

- **Instalación:** incluye las indicaciones indispensables para desensamblar y ensamblar el producto y sus partes componentes.
- **Generalidades:** incluye cualquier información general que facilite la comprensión del resto del documento y que por su volumen y complejidad no se incluya en la introducción.
- **Forma de utilización:** incluye todas las indicaciones requeridas para utilizar el producto de forma óptima y segura.
- **Fallos de funcionamiento y métodos de corrección:** incluye la relación de los posibles fallos de funcionamiento del producto y los métodos de corrección que están al alcance del usuario.
- **Accesorios e insumos:** incluye en forma de tabla, la relación completa de accesorios e insumos que se suministran con el producto.
- **Indicaciones para el mantenimiento técnico:** incluye la relación y periodicidad de todos los trabajos de mantenimiento técnico que puede requerir el equipo. Se describen solamente aquellos que puede realizar el operador.
- **Glosario:** incluye una relación en orden alfabético de los términos cuya descripción se considere necesaria para una mejor comprensión del documento.
- **Anexos:** su numeración es mediante letras A, B, C, etc. Los anexos pueden ser varios o ninguno. En general incluyen tablas, dibujos, fotos, esquemas y cualquier otra información complementaria.

3. Sistema de Calidad: se encuentra la información relacionada con el plan de calidad y las listas de chequeo para cada artefacto generado durante el desarrollo del producto. Contiene además, la información relacionada con las auditorías e inspecciones de calidad y la seguridad del software. En general el sistema de calidad contribuye a la aplicación de los principios de calidad total en el proyecto.



Seguridad del Software

Contiene la información relacionada con la seguridad del software a través de la plantilla Lista de Riesgos. Esta parte garantiza la calidad del software centrándose en la identificación y evaluación de los riesgos potenciales que pueden producir un impacto negativo y hacer que falle el sistema completo. Inicialmente, se identifican los riesgos y se clasifican por su importancia y su grado de riesgo. Cuando se han identificado los mismos se utilizan técnicas de análisis para asignar su gravedad y probabilidad de ocurrencia; para que sea efectivo se tiene que analizar el software en el contexto del sistema completo. Una vez que se han identificado y analizado los riesgos, se pueden especificar los requisitos relacionados con la seguridad del software. El rol que interviene es: Líder de Proyecto.

➤ Lista de Riesgos: esta plantilla guarda lo relacionado con los riesgos del proyecto que se desarrolla en el momento actual. Interviene en su realización el Líder de Proyecto. Contiene la siguiente estructura:

- Listado de riesgos, detallando el tipo de riesgo, el impacto que este implica en el proyecto, y los efectos que provoca cada uno.
- Gestión de riesgos, estimando la probabilidad de ocurrencia.



Inspecciones

La información relacionada con el proceso de inspección o revisión se encuentra almacenada en documentos con un formato específico. Las inspecciones son una

actividad de garantía de calidad del software llevada a cabo por los ingenieros del software. Sus objetivos son: descubrir errores en la función, la lógica o la implementación de cualquier representación del software; verificar que el software bajo revisión alcanza sus requisitos; garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos; conseguir un software desarrollado de forma uniforme; y hacer que los proyectos sean más manejables. Las inspecciones sirven como campo de entrenamiento, permitiendo que los ingenieros más jóvenes puedan observar los diferentes enfoques de análisis, diseño e implementación del software; también sirven para promover la seguridad y continuidad. Interviene en su realización el Administrador de Calidad.



Listas de Chequeo

Contiene las plantillas de los chequeos hechos en los flujos de trabajo de un proyecto software. En estos documentos se acumula la información que se obtuvo de cada chequeo con el fin de tener una medida de cómo se encuentra el sistema en un momento determinado. El rol que interviene es: Administrador de calidad.

- Lista de Chequeo de CU.
- Lista de Chequeo de Productos.
- Lista de Chequeo Modelo de CU.
- Lista de Chequeo del Modelo de Negocio.
- Lista de Chequeo de Requerimientos.
- Lista de Chequeo de la Interfaz Windows.
- Lista de Chequeo para el Especialista de la Aplicación.
- Lista de Chequeo para la etapa de Levantamiento de Requisitos.



Procedimientos

Acumula la documentación que pertenece a los procedimientos utilizados durante el proceso de calidad del software. Los procedimientos describen, de forma clara y precisa, los pasos que se deben seguir para empezar, desarrollar y ultimar una actividad, incluyendo entre otras consideraciones, los elementos técnicos que deben emplearse, las condiciones que se requieren, el alcance y las limitaciones fijadas, y el personal que participa. Intervienen en su realización el Administrador de Calidad.

- Procedimiento de Certificación: la certificación de productos de software se realiza fuera del proceso productivo y abarca desde la entrega de la documentación del producto terminado, hasta la conformación del informe final como resultado de las pruebas de laboratorio. El proceso de certificación no evalúa a los productos semielaborados, sino a los terminados. Solo se abarca la verificación funcional (las pruebas de caja negra). No se incluye la verificación estructural ya que se necesitaría la disponibilidad del código fuente.

- Procedimiento de Evaluación: este procedimiento especifica los pasos a seguir para evaluar un producto o servicio. Contiene la siguiente estructura:
 - Explicación a los propietarios del producto de cómo se llevará a cabo el proceso de evaluación.
 - Reunión con los inspectores que intervendrán en la evaluación.
 - Distribución de los perfiles de inspección entre los inspectores.
 - Evaluación del producto.
 - Notificación de los resultados de la evaluación al propietario del producto o servicio.



Auditorías

Son inspecciones que se realizan al proceso de desarrollo del software en cualquiera de las etapas de construcción, con el objetivo de verificar que el mismo se esté llevando a cabo de acuerdo a los parámetros exigidos por el cliente. Es un examen crítico que se realiza con el fin de evaluar la eficacia y eficiencia de una parte del producto o la totalidad del mismo. La información que se recoge de estas auditorías queda guardada en un documento con un formato determinado. Interviene en su realización el Administrador de Calidad.



Estándares

Esto se define con el objetivo de lograr uniformidad en un aspecto determinado del proceso de producción de software. En la facultad que se está desarrollando la investigación se ha definido un estándar para el proceso de codificación del software. Se esboza para los lenguajes de programación Java y CSharp porque son los utilizados en los proyectos productivos de la facultad 3.

➤ Estándar de Codificación (Java y C#): este documento explica una propuesta de trabajo para los estándares de codificación a utilizar por los desarrolladores de software de la facultad 3 de la UCI. Los lenguajes para los cuales está dirigido el estándar son: Java y CSharp. Contiene la siguiente estructura:

- Organización de los ficheros.
- Indentación.
- Comentarios.
- Declaraciones.
- Espacios en blanco.
- Convenciones de declaración.

- Prácticas de programación.
- Ejemplos de código.



Plan de Calidad

Es un documento que gestiona el sistema de calidad de un proyecto, y especifica la política a seguir para lograr los objetivos de aseguramiento de la calidad. En el se describen la política de calidad de un proyecto, la estructura organizacional, la misión de todo elemento involucrado en el logro de la calidad, etc. El rol que interviene es: Administrador de Calidad.

➤ Plan de Aseguramiento de la Calidad: esta plantilla ofrece los pasos a seguir con el objetivo de medir la calidad de un proyecto durante todo su desarrollo. Para poder asegurar la calidad de un producto hay que estar midiendo la calidad durante todo el proceso de realización del mismo. Interviene en su realización el Administrador de Calidad. Contiene la siguiente estructura:

- Resumen con todos los aspectos que contiene el plan.
- Especificación de los objetivos de calidad.
- Gestión, se describe la estructura de la organización especificando cada uno de los responsables de la calidad y las tareas a realizar.
- Listado de los documentos utilizados por el Plan de Calidad.
- Estándares y guías utilizados por el Plan de Calidad.
- Plan de revisiones y auditorías, se describe brevemente cada tipo de revisión y auditoría que se llevará a cabo en el proyecto, detallando el cronograma de estas. Se listan los grupos específicos o individuos involucrados en cada una de las actividades de auditoría y revisión identificadas. Se describen los procedimientos para informar y manejar problemas identificados durante las revisiones y auditorías del proyecto. Se

describen las herramientas, técnicas o metodologías específicas que serán usadas para llevar a cabo las actividades de revisión y de auditoría identificadas en este plan.

- Pruebas y evaluación, hace referencia al Plan de Pruebas.
- Listado de todas las herramientas, técnicas y metodologías utilizadas en las actividades del Plan de Calidad.
- Resolución de Problema y Acción Correctiva, hace referencia al Plan de Resolución de Problemas.
- Gestión de configuración, hace referencia al Plan de Gestión de Configuración.
- Descripción de varios registros de calidad que se mantendrán durante el proyecto.
- Listado de actividades de entrenamiento necesarias para que el equipo de proyecto ejecute las actividades del Plan de Aseguramiento de la Calidad.

4. Gestión de Cofiguración: se encuentra lo referente al plan de configuración, el control de versiones, los elementos de configuración de software, los procedimientos para el trabajo y las plantillas. En ella se encuentran los documentos que regulan el proceso de desarrollo de software.



Configuración

Contiene la documentación generada durante el flujo Gestión de Configuración. La finalidad de este flujo de trabajo es mantener la integridad de los artefactos que se crean en el proceso, así como mantener la información del proceso evolutivo que han seguido. La información que aquí se almacena recoge todo el proceso de control de cambios y mantenimiento de la integridad de los productos que encierra el proyecto, que incluye: identificar los elementos configurables, restringir los cambios en los elementos configurables, auditar los cambios hechos a estos elementos y, definir y

mantener las configuraciones de estos elementos. Los roles que intervienen en este flujo de trabajo son: Gestor de Configuración y Gestor de Cambios.

➤ Elementos de Configuración de Software (ECS): esta plantilla regula los elementos de configuración de un sistema. Los mismos aparecen agrupados por categorías para lograr una mejor organización y entendimiento. Interviene en su realización el Gestor de Configuración.

➤ Control de Versiones: esta plantilla contiene un resumen de las versiones hechas del producto software que se está desarrollando, describiendo la herramienta que se utiliza para controlar este proceso y cómo queda organizada la información al aplicarla. Interviene en su realización el Gestor de Cambios.

➤ Documento de Gestión de Cambios: esta plantilla se emite a partir de la solicitud de cambio de los reportes enviada por el J'de Proyecto por cualquier vía. Constituye el análisis realizado a los reportes entregados, e incluye sugerencias del equipo de desarrollo para la solución del cambio propuesto. Se habla de los elementos de configuración que van a ser afectados por el cambio y se explica en detalles la solución que propone el equipo de desarrollo. Interviene en su realización el Gestor de Cambios.

➤ Plan de Gestión de Configuración: esta plantilla establece y documenta los requisitos, políticas, estándares y procedimientos para la gestión de la configuración de los elementos software. Interviene en su realización el Gestor de Configuración. Contiene la siguiente estructura:

- Organización, Responsabilidades e Interfaz.
- Herramientas, entorno e infraestructura.

- Identificación de los ECS, estableciendo las líneas bases.
- Proceso de solicitud y aceptación de cambios, especificando el proceso de selección de los miembros del Comité de Control de Cambios (CCB).
- Almacén del proyecto y procedimiento de liberación de ECS.
- Reportes y auditorías, especificando responsables de esta actividad.
- Hitos.
- Listado de actividades de entrenamiento, necesarias para que el equipo de proyecto ejecute las actividades del Plan de Gestión de Configuración.



Procedimientos

Almacena la documentación que corresponde a los procedimientos utilizados durante el desarrollo del software. Los procedimientos son documentos complementarios, donde se describe cómo, quién y cuándo deben realizarse las funciones previstas. Se encuentran organizados en 3 categorías, de acuerdo a como lo definió la facultad donde se está desarrollando esta investigación: Procedimientos de Gestión, Procedimientos de Ingeniería y Procedimientos de Soporte.



Procedimientos de Gestión

Se encuentran los procedimientos de Gestión del Proyecto.

- Procedimiento para la selección e incorporación y formación de estudiantes a proyectos: este procedimiento especifica los parámetros que necesita un estudiante para formar parte de un proyecto, así como el proceso de capacitación que se lleva a cabo con los estudiantes seleccionados.



Procedimientos de Ingeniería

Contiene los procedimientos del desarrollo técnico de un proyecto.

- Procedimiento que define la metodología a seguir en los proyectos de la facultad. Artefactos de RUP a desarrollar: este procedimiento se refiere a la metodología de desarrollo, que en el caso de la facultad 3 es RUP la que se utiliza en todos los proyectos productivos de la misma, estableciendo los artefactos a desarrollar en cada flujo de trabajo.

- Procedimientos que define el desarrollo técnico del software: aparecen los procedimientos relacionados con el proceso de desarrollo del software: Levantamiento de Requisitos y, Análisis y Diseño.



Procedimientos de Soporte

Se encuentran los procedimientos de soporte del proyecto.

- Procedimiento para el Control de Versiones: este procedimiento especifica punto por punto cómo se lleva a cabo el proceso de control de versiones en un software. Contiene la siguiente estructura:
 - Definición de los responsables de esta actividad.
 - Definición de los posibles estados de las versiones de los componentes.
 - Notificación del problema.
 - Bloqueo en el repositorio de la última versión del ECS.
 - Entrega al desarrollador correspondiente para su modificación.
 - Bloqueo del componente en la biblioteca del proyecto.
 - Liberación del componente en la biblioteca del proyecto.
 - Entrega al responsable de configuración la nueva versión de ECS.
 - Se almacena la nueva versión en el repositorio y se marca como libre.

➤ Procedimiento para la Gestión de los Cambios: este procedimiento describe cómo se deben realizar los cambios durante el desarrollo de un software. Contiene la siguiente estructura:

- Definición de los responsables de esta actividad.
- Definición de los posibles estados de las solicitudes de cambio.
- Notificación del problema.
- Almacenamiento del problema con un código consecutivo de problemas.
- Clasificación de la solicitud recibida según el tipo.
- Si es un cambio se genera una Solicitud de Cambio (SC) y se analiza.
- Se decide si se acepta el cambio.
- Creación y asignación de una Orden de Trabajo (OT) asociada a la SC.
- Si no es un cambio, en la elaboración de la orden se analiza el defecto o mejora y se le asigna al equipo encargado de resolverlo.
- Asignación de la OT, aquí se planifica el cambio, se estima el tiempo de duración, complejidad y prioridad.
- Envío del cambio ejecutado al probador asignado en la OT.
- Prueba del cambio realizado.
- Notificación del cambio resuelto y distribución de la nueva versión.

➤ Procedimiento para la identificación de los ECS: este procedimiento describe paso a paso cómo se identifican los ECS. Contiene la siguiente estructura:

- Definición de los responsables de esta actividad.
- Definición de los posibles estados de las versiones de los ECS.
- Identificación y documentación de los ECS.
- Especificación de los ECS y cuándo pasan al control de configuración.
- Establecimiento de las relaciones que van a tener los ECS.
- Ubicación de la versión del ECS en el repositorio del proyecto.

- Publicación y distribución de este documento.
- Procedimiento para la planificación de la Gestión de Configuración de Software (GCS): este procedimiento describe los pasos a seguir en la planificación de la Gestión de Configuración de un software. Contiene la siguiente estructura:
- Descripción de los aspectos generales del Plan de Gestión de Configuración.
 - Identificación de las tareas de coordinación y gestión necesarias para llevar a cabo las actividades de Control de Configuración.
 - Identificación de los momentos para la realización de actividades de control de configuración que se desarrollarán en el proyecto.
 - Se establecen los procedimientos que se utilizarán para el control de versiones.



Plantillas

Contiene las plantillas del módulo Desarrollo Técnico, y las plantillas del flujo de trabajo Ambiente.



Modelación del Negocio (plantillas del flujo de trabajo Modelación del Negocio)



Requerimientos (plantillas del flujo de trabajo Requerimientos)



Análisis y Diseño (plantillas del flujo de trabajo Análisis y Diseño)



Implementación (plantillas del flujo de trabajo Implementación)



Prueba (plantillas del flujo de trabajo Prueba)



Despliegue (plantillas del flujo de trabajo Despliegue)



Gestión de Proyectos (plantillas del flujo de trabajo Gestión de Proyectos)



Gestión de Configuración (plantillas del flujo de trabajo Gestión de Configuración)



Ambiente

Se encuentra la documentación relacionada con el flujo de trabajo Ambiente o Entorno, como también se nombra en algunas bibliografías. La finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso unificado que se va a seguir. El principal artefacto que se genera en este flujo de trabajo es el Caso de Desarrollo. Además se tendrán que definir las guías para los distintos aspectos del proceso, como pueden ser el modelado del negocio y los casos de uso, el diseño, la programación. Los roles que intervienen son: Ingeniero de Procesos y el Administrador de Sistema.

- **Caso de Desarrollo:** esta plantilla especifica en concreto para el proyecto actual cómo se aplicará el proceso, qué productos se van a utilizar y cómo van a ser utilizados. Interviene en su realización el Ingeniero de Procesos.
- **Preceptos del Modelado del Negocio:** esta plantilla está dividida en secciones que explican como se lleva a cabo el proceso de modelado del negocio. Contiene la siguiente estructura:

- Preceptos generales del modelado del negocio.
 - Cómo describir un caso de uso del negocio.
 - Preceptos generales del modelo de objetos.
 - Cómo describir los trabajadores del negocio.
 - Cómo describir las entidades del negocio.
 - Estereotipos UML utilizados por RUP para la representación gráfica del modelo de negocio.
- Preceptos del Modelado de Casos de Uso: la plantilla está dividida en secciones que explican cómo se realiza la descripción de cada caso de uso del negocio. Contiene la siguiente estructura:
- Preceptos generales del modelado de casos de uso.
 - Cómo describir un caso de uso.
 - Estereotipos UML utilizados en la realización del proyecto.
- Preceptos de Programación: esta plantilla contiene lo referente a la programación del sistema. Contiene la siguiente estructura:
- Organización del código y el estilo.
 - Comentarios.
 - Nombramientos.
 - Declaración.
 - Expresiones y declaraciones.
 - Gestión de la memoria.
 - Gestión de error y excepciones.
- Preceptos de Diseño: esta plantilla contiene lo correspondiente al diseño del software que se está desarrollando.

- Preceptos generales de Diseño e Implementación.
- Preceptos del diseño de la Base de Datos.
- Preceptos del diseño de la Arquitectura.
- Preceptos del Mecanismo.
- Descripción textual de los estereotipos UML utilizados.

Fuera de esta estructura de módulos y submódulos descrita anteriormente se encuentra una carpeta llamada Herramientas_Desarrollo.



Herramientas_Desarrollo

Esta carpeta contiene los instaladores de las herramientas utilizadas para el desarrollo del software, así como el Sistema Operativo sobre el cual se implementó este desarrollo.

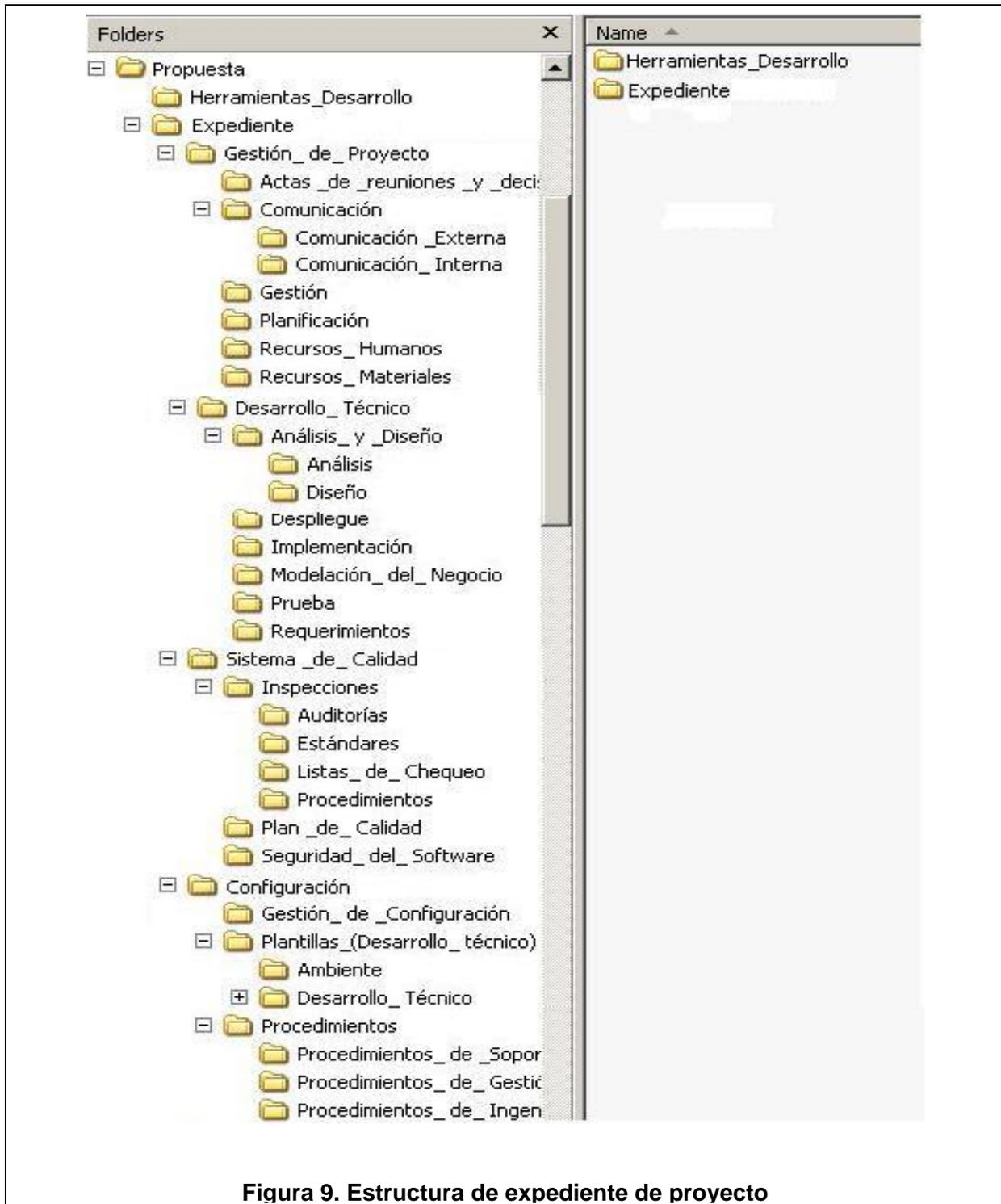


Figura 9. Estructura de expediente de proyecto

2.3 Instalación de SVN.

La instalación de SVN es extremadamente fácil.

Requisitos para la instalación de SVN:

- ✓ Servidor Apache
- ✓ Subversion
- ✓ TortoiseSVN

Servidor Apache

La configuración más flexible de todas las instalaciones de servidor posibles para Subversion es la que se basa en Apache. Aunque es un poco más complicada de preparar, ofrece beneficios que otros servidores no pueden dar:

1. WebDAV: el servidor de Subversion basado en Apache utiliza el protocolo WebDAV que es utilizado por otros programas. Por ejemplo, se podría montar un repositorio como una Carpeta Web en el explorador de Windows y luego acceder a ella como cualquier otra carpeta en el sistema de ficheros.
2. Navegando por el repositorio: puede apuntar el navegador a la URL del repositorio y navegar por los contenidos sin tener un cliente de Subversion. Esto da acceso a los datos a un mayor círculo de usuarios.
3. Autenticación: se puede utilizar cualquier mecanismo de autenticación que Apache soporte, incluyendo SSPI y LDAP.
4. Seguridad: dado que Apache es muy estable y seguro, automáticamente obtendrá la misma seguridad para el repositorio. Esto incluye la encriptación SSL.

Una vez que se tenga el instalador de Apache2 se hace doble click en él y le guiará a través del proceso de instalación. Hay que asegurarse que ha introducido la URL del servidor correctamente, si no tiene un nombre DNS para el servidor se introduce la

dirección IP. Se recomienda que se instale apache para todos los usuarios, en Puerto 80, como un Servicio. Si ya se tiene Internet Information Server (IIS) u otro programa ejecutándose que escuche en el puerto 80 la instalación puede fallar. Si esto ocurre, se va al directorio de Archivos de Programa\Apache Group\Apache2\conf y se localiza el fichero httpd.conf. Se edita dicho fichero para cambiar Listen 80 por un puerto libre, por ejemplo, Listen 58, que es el utilizado en este caso. Luego se reinicia la instalación, esta vez debería terminar sin problemas. Ahora se comprueba si el servidor web Apache funciona correctamente apuntando desde el navegador web a la dirección `http://localhost/`, debería aparecer un sitio web preconfigurado.

SVN

No hay mayor complicación en su instalación, pues solo se da doble click en el instalador, y se acepta todo hasta que finalice. Si todo salió bien, se puede abrir el command prompt (dando click en el botón Inicio de windows, luego en Ejecutar, y en la ventana que aparece se escribe 'cmd' y aceptar) y escribir el siguiente comando: `svn help`, y saldrá una lista de parámetros que se pueden usar con 'svn' que es el comando básico para todo lo relacionado con subversion.

1. Si el instalador de Subversion reconoce que ha instalado Apache, se habrá casi terminado. Si no puede encontrar un servidor de Apache entonces se tendrán que realizar algunos pasos adicionales.
2. Utilizando el explorador de Windows, se va al directorio de instalación de Subversion, normalmente `C:\Archivos de programa\Subversion`, y se buscan los ficheros `/bin/mod_dav_svn.so` y `mod_authz_svn.so`. Se copian estos ficheros al directorio de módulos de Apache, normalmente `C:\Archivos de programa\Apache Group\Apache2\modules`.
3. Se copia el fichero `/bin/libdb43.dll` desde el directorio de instalación de Subversion al directorio de módulos de Apache.

4. Se edita el fichero de configuración de Apache, normalmente C:\Archivos de Programa\Apache Group\Apache2\conf\httpd.conf con un editor de texto como el Bloc de Notas y se hacen los siguientes cambios:

Se quitan los comentarios '#' en las siguientes líneas:

```
#LoadModule dav_fs_module modules/mod_dav_fs.so
```

```
#LoadModule dav_module modules/mod_dav.so
```

Se añade las dos líneas siguientes al final de la sección LoadModule.

```
LoadModule dav_svn_module modules/mod_dav_svn.so
```

```
LoadModule authz_svn_module modules/mod_authz_svn.so
```

TortoiseSVN

TortoiseSVN es una aplicación que se integra con windows explorer para manejar los archivos. Realmente, lo único que hace es darle una interfaz gráfica a los comandos que se pueden utilizar desde el command prompt, pero así también se hace más fácil de utilizar.

Su instalación es igual de sencilla. Se da doble click en el instalador, y se acepta todo hasta que finalice el proceso de instalación. Una vez terminado, se abre el windows explorer, y se da click derecho sobre cualquier archivo o carpeta, y veras que aparece una opción para TortoiseSVN, con un icono de tortuga, que a su vez presenta un submenú con las posibles opciones. (Ver figura 10)

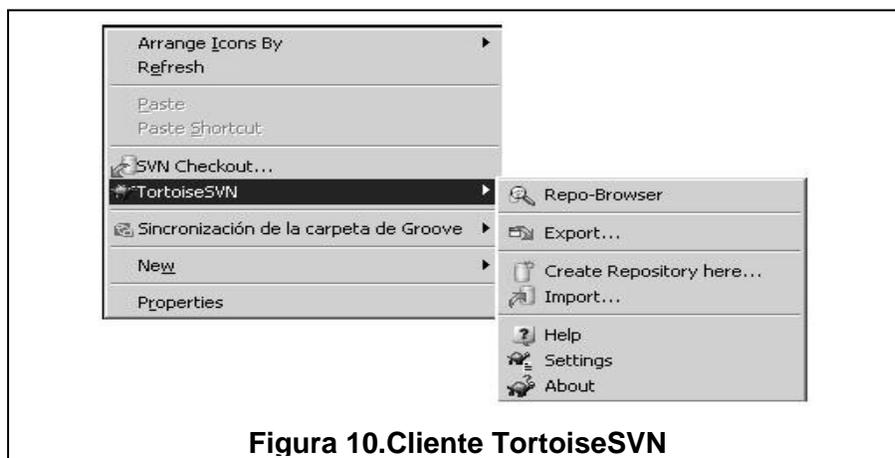


Figura 10. Cliente TortoiseSVN

2.4 Configuración de la herramienta para montar el expediente de proyecto.

Ahora ya se ha preparado Apache y Subversion, pero Apache aún no sabe cómo manejar los clientes de Subversion como TortoiseSVN. Para que Apache sepa qué URL debe utilizarse para los repositorios de Subversion se debe editar el fichero de configuración de Apache, normalmente está en C:\Archivos de programa\Apache Group\Apache2\conf\httpd.conf, con cualquier editor de texto que desee, por ejemplo el Bloc de Notas:

1. Al final del fichero Config se añaden las siguientes líneas:

```
<Location /rep>
  DAV svn
  SVNParentPath D:\Proyectos\repositorios
  AuthzSVNAccessFile D:\Proyectos\config\access.txt
  AuthType Basic
  AuthName "Repositorios"
  AuthUserFile D:\Proyectos\config\passwd.txt
  Require valid-user
</Location>
```

Esto configura el Apache de forma que todos sus repositorios de Subversion están físicamente localizados bajo D:\SVN. Los repositorios se sirven al mundo exterior desde la URL: <http://MiServidor/svn/>. El acceso es restringido a los usuarios/contraseñas listados en el fichero passwd.

2. Para crear el fichero passwd, se abre el Símbolo del sistema o línea de comandos de nuevo, se cambia la carpeta Apache2, normalmente C:\Archivos de programa\Apache Group\Apache2 y se crea el fichero introduciendo

```
bin\htpasswd -c passwd <nombreusuario>
```

- Para llenar el fichero passwd se adaptó un programa hecho por el proyecto SIGIA en Visual Studio 2003, de modo contrario este proceso se tendría que realizar a

mano. Este sistema introduce de forma automática los nuevos usuarios con su contraseña de acceso y los ubica en el grupo deseado, también elimina usuarios, y cambia la contraseña de estos. Permite además la creación de nuevos grupos de trabajo, la eliminación de estos y de repositorios, siempre y cuando la persona que ejecute esta acción esté autorizada para hacerlo. (Ver figura 11)

El fichero passwd contiene la siguiente estructura:

```
[users]
```

```
# listado de usuarios y password
```

```
usuario = pass
```

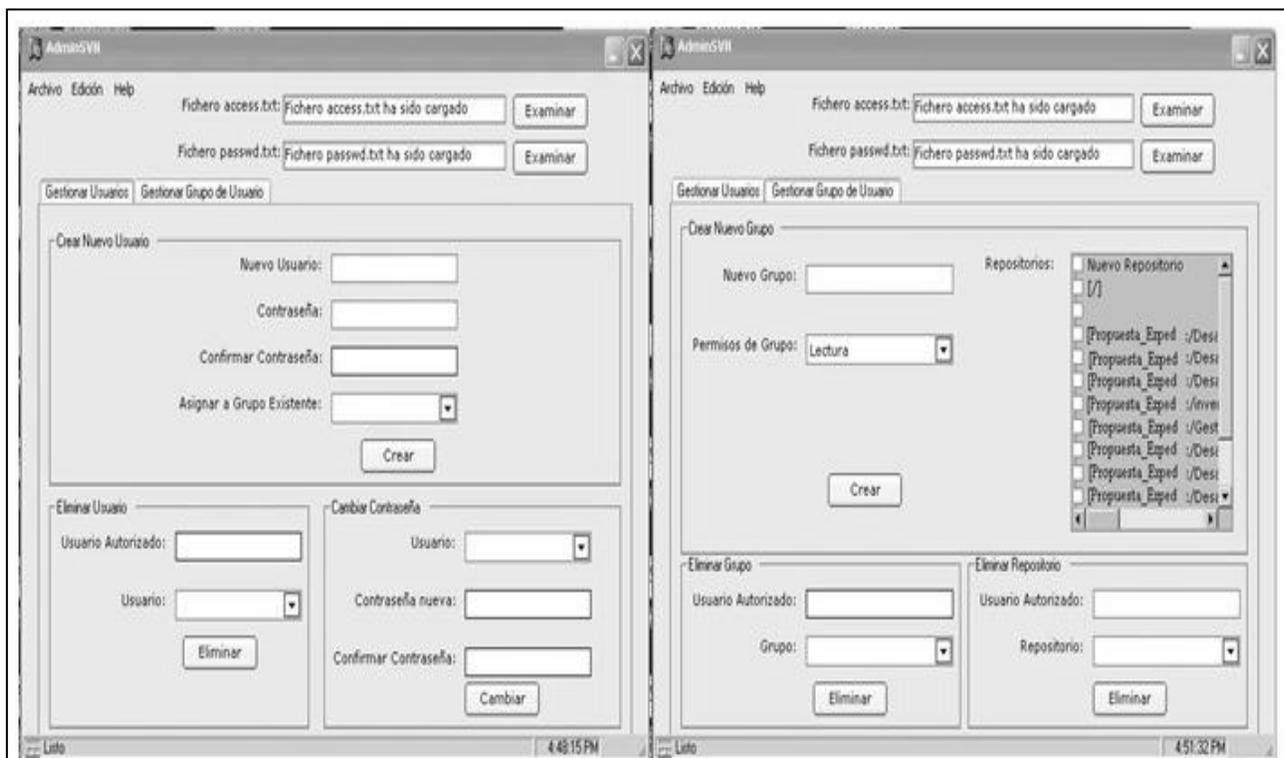


Figura 11. Programa para llenar el fichero passwd (SIGIA 2006)

3. Se reinicia el servicio de Apache de nuevo.
4. Se pone en el navegador <http://MiServidor/svn/MiNuevoRepositorio> como dirección URL, donde MiNuevoRepositorio es el nombre del repositorio de Subversion que se creó antes. Si todo ha ido bien se debería ver una ventana preguntando por un usuario y una contraseña (Ver figura 12), y luego se podrá ver los contenidos del repositorio. (Ver figura 13)

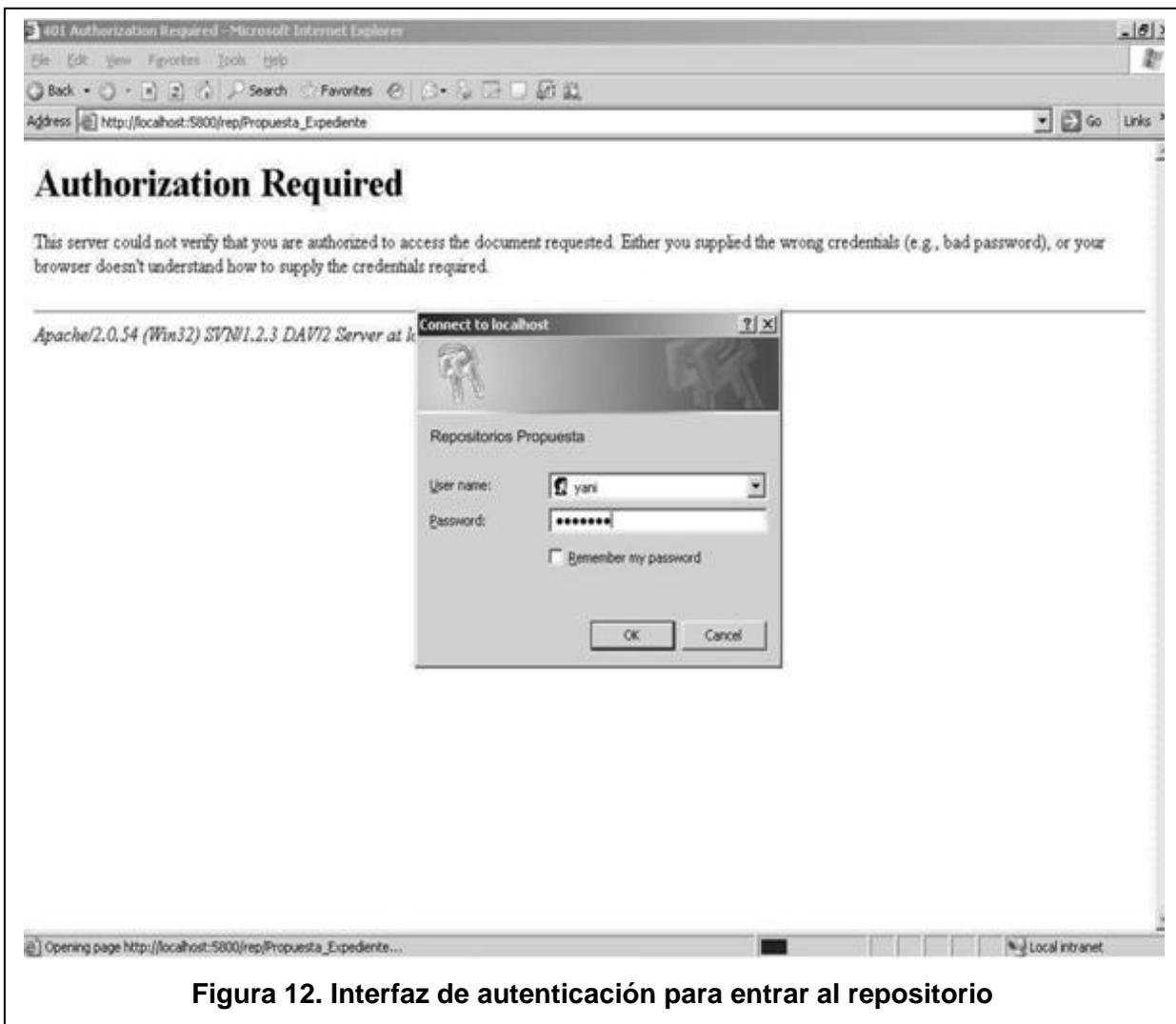


Figura 12. Interfaz de autenticación para entrar al repositorio

Revision 1: /

- Gestión de Proyecto/
 - Actas de reuniones y decisiones/
 - Comunicación/
 - Gestión /
 - Planificación/
 - Recursos Humanos/
 - Recursos Materiales/
- Desarrollo Técnico/
 - Análisis y Diseño/
 - Despliegue/
 - Implementación/
 - Modelación del Negocio/
 - Prueba/
 - Requerimientos/
- Sistema de Calidad/
 - Inspecciones/
 - Plan de Calidad/
 - Seguridad del Software/
- Gestión de Configuración/
 - Configuración/
 - Plantillas (Desarrollo Técnico)/
 - Procedimientos/

Powered by Subversion version 1.2.3 (r15833).

Figura 13. Expediente montado sobre la herramienta SVN

5. Se crea un fichero de acceso de Subversion. Apache se asegurará que sólo los usuarios válidos pueden acceder la ruta /svn, y luego pasará el nombre de usuario al módulo de Subversion AuthzSVNAccessFile para que pueda forzar un acceso más

específico basado en las reglas que se especifican en el fichero access de Subversion.

2.5 Asociación de permisos con roles de RUP.

Para asignar los permisos de acuerdo a los roles que define RUP se crea el fichero Access, el cual contiene los usuarios involucrados en el proyecto y el área de acceso de estos, especificando si puede leer y/o escribir un artefacto de acuerdo al rol definido al principio del fichero .txt. (Ver Anexo 1)

2.6 Proceso de las copias de trabajo para seguir la pista del repositorio.

Para cada fichero de una copia de trabajo, Subversion registra dos datos esenciales en el área administrativa .svn/:

- Revisión en la que está basado el fichero de la copia de trabajo, esto se llama la revisión de trabajo del fichero.
- Una marca de tiempo con la fecha de la última actualización del fichero desde el repositorio.

Con esta información, y comunicándose con el repositorio, Subversion puede conocer en cuál de los cuatro estados siguientes se encuentra el fichero de la copia de trabajo:

1. Sin cambios y actualizado: el fichero no ha sido modificado en la copia de trabajo ni se ha enviado ningún cambio sobre ese fichero al repositorio desde su revisión de trabajo. Un svn commit de ese fichero no hará nada, y un svn update del fichero tampoco hará nada.
2. Modificado localmente y actualizado: el fichero ha sido modificado en la copia de trabajo pero no se ha enviado ningún cambio sobre ese fichero al repositorio desde su revisión base. Hay cambios locales que no han sido enviados al repositorio, por lo que

un svn commit del fichero publicará con éxito sus cambios, y un svn update del fichero no hará nada.

3. Sin cambios y desactualizado: el fichero no ha sido modificado en la copia de trabajo, pero sí en el repositorio. El fichero debería ser actualizado para sincronizarlo con la revisión pública. Un svn commit del fichero no hará nada, y un svn update del fichero introducirá los últimos cambios en su copia de trabajo.

4. Modificado localmente y desactualizado: el fichero ha sido modificado tanto en la copia de trabajo como en el repositorio. Un svn commit del fichero fallará dando un error de desactualizado. El fichero debe ser actualizado primero; un svn update intentará mezclar los cambios públicos con los cambios locales. Si Subversion no puede combinar los cambios de manera convincente automáticamente, dejará que sea el usuario el que resuelva el conflicto.

2.7 StatSVN: métricas sobre Subversion.

StatSVN es una herramienta que genera diferentes métricas, analizando los ficheros controlados por Subversion. StatSVN recupera la información de un repositorio de subversion y genera varias tablas y las gráficas que describen el desarrollo del proyecto.

Características de StatSVN:

- StatSVN es un software a código abierto, con licencia bajo LGPL, StatSVN está usando StatCVS versión 0.2.3+
- El código fuente esta disponible desde Subversion server.StatSVN, está escrito en Java
- StatSVN usa JFreeChart para generar gráficos

- El proyecto StatSVN fue iniciado por Jean-Philippe Daigle y mantenido actualmente por Jean-Philippe Daigle, Jason Kealey, Gunter Missbacher, y Benoit Xhenseval

Reportes que genera:

- Evolución de las líneas de código del proyecto: representa una tabla con las líneas de código del proyecto en función del tiempo de desarrollo.
- Líneas de código por programador: representa una tabla con los siguientes campos: autor, cambios, líneas de código, líneas cambiadas.
- Reloj de actividad: representada por dos tablas; la primera con las actividades por horas al día, y la segunda con las actividades por días de la semana.
- Actividad por autores: representa una tabla con las líneas de código modificadas y adicionadas por cada programador.
- Actividad del autor por módulo: representa una tabla que contiene las líneas de código por programador en cada módulo.
- Tamaño medio de fichero: representa una tabla que contiene las líneas de código por fichero contra la fecha de realización.
- Contador de fichero: representa una tabla con los ficheros contra la fecha de creación.
- Ficheros de mayor tamaño: representa una tabla con la dirección de ficheros de mayor tamaño y líneas de código que contiene cada uno.
- Ficheros con más revisiones: representa una tabla con la dirección de ficheros más revisados y números de revisiones hechas.
- Tamaño del directorio: representa una tabla con las líneas de código por directorio: Loc y fecha.
- Número de etiquetas del repositorio de LOC por versión.
- Árbol del repositorio con cuenta del fichero y las líneas del código.

- Mapa del repositorio con vista jerárquica dinámica del repositorio para los 30 días pasados.

Ejecutando StatSVN

StatSVN se ejecuta usando el comando `java -jar statsvn.jar`. Esto toma parte de dos líneas de comando.

```
java -jar statsvn.jar <logfile> <checked-out-module>
```

Por ejemplo:

```
java -jar /path/to/statsvn.jar /path/to/module/logfile.log /path/to/module
```

Este comando crea los reportes en el mismo directorio de usuario. Usted puede ahora apuntar el navegador al archivo `index.html`, donde se encuentra la tabla de contenidos del reporte.

En adición a los parámetros requeridos de las líneas de comando, se puede usar parámetros adicionales para deshacer los reportes y cambiar el comportamiento de StatSVN.

```
java -jar statsvn.jar [options] <logfile> <checked-out-module>
```

Opciones generales:

`-output-dir <directory>`

Especifica un directorio de salida. Los reportes se deben generar en este directorio. El directorio se crea si no existe.

`-include <include-pattern-list>`

`-exclude <exclude-pattern-list>`

Especifica filtros para excluir archivos del análisis. Un archivo será ignorado si cualquiera de los patrones no se incluye en la lista de patrones, o si cualquiera de estos patrones se excluye de la lista de patrones.

-cache-dir <directory>

-username <subversion_username>

Especifica un directorio caché para recibir información del servidor subversion. Estos datos serán salvados en el directorio. El directorio será creado si no existe. Por defecto, se usa el directorio actual del usuario. Especifica un nombre de usuario para ser usado cuando se conecta a un servidor subversion.

-password <subversion_username>

Especifica una contraseña para ser usada cuando se conecta a un servidor subversion.

-tags <tags-regularexpression>

Incluye información sobre tags de subversion en los reportes. Solo los tags que corresponden con la expresión regular son incluidos. Para incluir todos los tags, adicionar -tags '.*' a la línea de comando.

-no-developer <login-name>

Excluye el nombre de una cuenta subversion de todos los reportes de desarrollo. Esto es importante para reducir ruidos del administrador u otras cuentas. Múltiples cuentas pueden ser excluidas adicionando la opción en la línea de comando para cada cuenta.

2.8 Conclusiones.

Se obtuvo un expediente de proyecto para el trabajo en una Factoría de Software, el cual permitirá estandarizar la información generada en los proyectos productivos de la facultad 3. Además fue exitosa la implantación de este expediente sobre la herramienta Subversion, contribuyendo a elevar la calidad y productividad del producto final.

CAPÍTULO # 3: ANÁLISIS DE LOS RESULTADOS.

3.1 Introducción.

En este capítulo se presenta un análisis de los resultados de la investigación a través de criterios de especialistas en el tema. Para validar la propuesta de expediente desarrollada en este trabajo se entrevistaron a especialistas en el tema de la documentación, principalmente aquellos que se han desempeñado en el rol de Arquitecto de Información de algún proyecto. Estas entrevistas demuestran la importancia que tiene la documentación en un proyecto, y lo ventajoso de tener uniformidad en este sentido.

3.2 Criterios de especialistas.

Las preguntas formuladas a los especialistas fueron elaboradas con el objetivo de conocer su criterio acerca de la importancia que tiene la documentación en un proyecto, y la facilidad de uso de la propuesta desarrollada. A continuación se citan fragmentos de la conversación sostenida con los entrevistados.

Yudisbel Rojas Cruz: Lic. Información Científica y Bibliotecología

Área: Dirección Casa Authoring DVD

Cargo: Arquitecta de Información

¿Qué importancia usted le concede a la GD para el éxito de un proyecto?

Considero que la GD ocupa un lugar importante dentro del desarrollo de un proyecto, no solo de software, sino del tipo que sea. De esta forma quedan documentadas todas las actividades que se realizaron en el proyecto, pudiendo ser consultada o reutilizada esta información. La persona encargada de gestionar la información en un proyecto tiene

bajo su responsabilidad redactar el manual de usuario, así como todos los entregables del proyecto. También, junto al Gestor de Configuración define la herramienta que se utilizará como repositorio de información, y establece cómo se organizará y almacenará la misma dentro de la herramienta.

¿Qué beneficios le ofrecería nuestra propuesta soportada por la herramienta SVN?

De manera general la propuesta que presentan es completa y bien estructurada, caracterizándose por su ajuste a la metodología RUP. Debería ser más general de acuerdo a la metodología de desarrollo a utilizar, para que pueda ser usada por equipos de trabajo de software libre, aunque si esta propuesta es para la facultad 3, donde se utiliza RUP en todos los proyectos, esto no será impedimento alguno.

Sobre la herramienta les puedo decir que es muy utilizada en la Universidad; proyectos de gran importancia como Identidad y, Registros y Notarias la utilizan en su desarrollo. Pienso que se puede utilizar como repositorio de información de acuerdo a las necesidades del proyecto. Además tiene la ventaja de ser software libre, política hacia la cual se encamina la Universidad y el país en general.

¿Qué beneficios tiene que la propuesta de expediente desarrollada esté orientada a una Factoría de software?

Tener implementada una Factoría de Software es algo muy ventajoso. Es una estructura creada para el desarrollo de software, con procesos estandarizados, repetibles, y principalmente mejorable continuamente. Significa esfuerzos integrados por encima de proyectos individuales, para mejorar las operaciones relativas al software. Debe poseer un grupo de herramientas estandarizadas tanto para la construcción de software como

para la gestión y administración de proyectos, automatizando gran parte del trabajo. Reduce la cantidad de trabajo promoviendo la reutilización de componentes software y elevando la productividad.

Liuris Rodríguez Castilla: Lic. Información Científica y Bibliotecología

Área: Dirección Información de la UCI

Cargo: Especialista General

¿Qué importancia usted le concede a la GD para el éxito de un proyecto?

La GD es sin duda muy importante. Es la constancia de que el proyecto se llevó a cabo, indica un historial del proyecto. Gestiona tareas importantes como contratos, códigos, procedimientos. A medida que se desarrolla un proyecto se van cumpliendo actividades, las cuales quedan plasmadas en la documentación, sirviendo de base en caso de ser necesario consultar el estado de avance del proyecto. La gestión documental es algo más que la gestión y el acceso a los documentos. Debe asegurar que la información y el conocimiento se comparten. En definitiva, una correcta gestión documental contribuye a mejorar la eficiencia de un proyecto.

¿Qué beneficios le ofrecería nuestra propuesta soportada por la herramienta SVN?

Considero que la propuesta está organizada correctamente y contiene todo lo necesario para documentar un proyecto de forma eficiente. Solo que debería ser más general, de modo que sea aplicable a personas que trabajen con software libre, aunque si su aplicación es solo en la facultad 3, donde se utiliza RUP como metodología de desarrollo, pienso que no habrá inconvenientes.

Con respecto a la herramienta que utilizaron para montar el expediente propuesto, pienso que sí se puede utilizar en este proceso, va a ser un lugar centralizado donde consultar la información generada de acuerdo a los permisos que se establezcan. También tiene la ventaja de ser software libre, política hacia la cual está tratando de encaminarse la Universidad de las Ciencias Informáticas.

¿Qué ventajas tiene que la propuesta de expediente esté orientada a una Factoría de software?

El enfoque de factoría de software viene a formalizar todos los procesos y sus productos, trabajando en líneas de producción, con etapas y tareas perfectamente definidas para cada tipo de profesional involucrado en el proceso, yendo de la productividad en la línea de producción a las rutinas de control de la calidad. Se busca la especialización de los profesionales, para que cada uno garantice la productividad de la fase en la que está ocupado. Debe ser una fábrica en la cual las actividades de desarrollo sean predecibles, lo que implica el establecimiento de procedimientos y métodos que cuenten con herramientas que ayuden en su implantación.

Ramsés Delgado Martínez: Ing. Informático

Área: Dirección de Calidad del Software

Cargo: Director de Calidad

¿Qué importancia usted le concede a la GD para el éxito de un proyecto?

La gestión documental es de vital importancia en el desarrollo de un proyecto, porque documenta todas las acciones llevadas a cabo durante su desarrollo, permitiendo la consulta o reutilización de esta información en cualquier momento.

¿Qué beneficios le ofrecería nuestra propuesta soportada por la herramienta SVN?

La propuesta está bien estructurada, se acerca bastante a la que proponemos aquí en la IP, solo difiere en cuanto a organización. Nosotros nos apoyamos en CMMI como modelo de procesos para hacer la propuesta más general, aplicable también a proyectos que trabajen con software libre. Aunque si la propuesta es para la facultad 3, donde la metodología de desarrollo que utilizan los proyectos productivos es RUP, los artefactos que se generan corresponden a lo que plantea este proceso de desarrollo. En caso que se pretenda elevar esta propuesta a nivel de universidad para ser aplicada en todas las facultades de la UCI, es necesario hacerla más abarcadora, pudiendo tomar algunas de las artefactos que propone RUP pero sin centrarse en esta metodología.

3.3 Valoración de las entrevistas realizadas.

Con las entrevistas realizadas se obtuvieron diversas opiniones y recomendaciones de especialistas en el tema de la documentación.

Los entrevistados mencionaron de forma implícita varios parámetros, que miden los beneficios que se obtendrán al aplicar la propuesta desarrollada. Estos son:

- **Calidad:** los entrevistados concuerdan que la propuesta desarrollada ayuda a elevar la calidad del producto final, porque la información manejada estará mejor estructurada.
- **Productividad:** los especialistas coinciden en que la puesta en práctica de la propuesta permitirá reutilizar la información, en función de disminuir la cantidad de trabajo.
- **Organización:** los entrevistados afirman que la implementación de esta propuesta asegurará una mejor distribución de la documentación generada en los proyectos de producción de software.

- Uniformidad: los especialistas aseguran que la Factoría de Software promueve la utilización de procesos estandarizados y repetibles, lo cual hace que el trabajo se desarrolle unificadamente, y sea comprensible para todos los involucrados.

3.4 Conclusiones.

Las entrevistas realizadas a especialistas demostraron que la propuesta desarrollada está bien estructurada y brinda beneficios al proceso de gestión documental de la futura Factoría de Software en la facultad 3. La selección de los parámetros posibilitó conocer hasta que punto podrá ser efectiva la propuesta cuando se aplique.

CONCLUSIONES

Al terminar esta investigación se arriba a las siguientes conclusiones:

- La implantación de la propuesta de expediente de proyecto permitirá una mejor organización de la documentación, así como la estandarización de la información generada en los proyectos productivos.

- El empleo de la herramienta SVN asegura el manejo confiable y seguro de la información generada en los proyectos de producción de software, almacenando la documentación en un repositorio para que pueda ser accedida en el momento deseado de acuerdo a los permisos asignados, y controlando las versiones creadas durante el desarrollo del producto.

- Los criterios de los especialistas entrevistados evidencian la calidad de la propuesta realizada, demostrando la importancia que amerita esta temática en el ámbito de la producción de software.

- La implementación del expediente de proyecto orientado a una Factoría de Software, contribuirá a potenciar la calidad de desarrollo del producto final.

RECOMENDACIONES

Con el objetivo de continuar y mejorar la propuesta desarrollada en este trabajo, se recomienda lo siguiente:

- Profundizar aún más en los fundamentos de la gestión documental.

- Profundizar en el estudio de la herramienta SVN.

- Implementar la propuesta inmediatamente que entre en funcionamiento la Factoría de Software de la Facultad 3.

- Extender la propuesta realizada a niveles superiores, ajustándola a las necesidades específicas de cada proyecto.

REFERENCIAS BIBLIOGRÁFICAS

- *DME -Document Management Extension*. 2007]. Disponible en: http://www.bs.com.ar/bsweb/MENU/SUBMENU/Archivos/bs_053.pdf
- *Drupal 5.1 and 4.7.6 released*. 2007. [2007]. Disponible en: <http://drupal.org>
- *El nuevo Sistema de Gestión de Expedientes de Altia Consultores en la Universidad de A Coruña*. 2006. 14.
- *Gestión Documental*. 2005. [2007]. Disponible en: http://www.guellconsulting.com/servicios/internet/gestion_documental
- *Gestión Documental. e-doctimo: 4*.
- *IBM RATIONAL, S. C. Rational Unified Process [Help]*. 2003.
- *Introducción a Subversion* 2004. [2007]. Disponible en: <http://libertonia.escomposlinux.org/story/2004/5/19/142850/344>
- *New Summer Mambo templates released* 2004. [2007]. Disponible en: <http://www.mamboserver.com>
- *Project Description*. 2006. [2007]. Disponible en: <http://plone.org/products/plone>
- *PROYECTIS IV: DE LAS HERRAMIENTAS GRÁFICAS DE GESTIÓN DE PROYECTOS AL SPO* 2006.
- *RUP*. 2006. [2007]. Disponible en: <http://prug.solucionesracionales.com/node/12>
- *SISTEMA INTEGRADO DE GESTION DOCUMENTAL GIT-DOC: NUESTRO ENFOQUE*. 2007]. Disponible en: http://www.gitdoc.com/ctl_arch/image/infotecnica.pdf
- *StatSVN News*. 2007. [2007]. Disponible en: <http://www.statsvn.org>
- *TRABAJO DOCUMENTAL Y COLABORATIVO*. 2006. [2007]. Disponible en: http://www.csi.map.es/csi/tecniemap/tecniemap_2006/02T_PDF/trabajo%20documental.pdf
- *Welcome to the OpenCms Project*. 2006. [2007]. Disponible en: <http://www.opencms.org/opencms/en>

-
- ANTONIO, A. D. *LA GESTIÓN DE LA CONFIGURACIÓN DEL SOFTWARE*. 59 p.
 - BÁRCENAS, A. A. *III JORNADA DE GERENCIA DE PROYECTOS*, 2005. [2007]. Disponible en: <http://www.acis.org.co/index.php?id=357>
 - COLLINS-SUSSMAN, B. *Control de versiones con Subversion*. 2004. 286 p.
 - FERNÁNDEZ, E. *METODOLOGÍA DE EXPLOTACIÓN DE INFORMACIÓN GESTOR DE DOCUMENTOS*, 2005. [2007]. Disponible en: <http://www.fi.uba.ar/laboratorios/lsi/Rrtis-7-1-2005-01-07.pdf>
 - FOWLER, M. *La Nueva Metodología*, 2003. [2007]. Disponible en: <http://www.willydev.net/descargas/articulos/general/MetologiaXP.PDF>
 - GUERRERO, S. E. *Construcción de un Modelo de Trabajo de Gestión Documental en la Organización, en el Contexto de la Actividad Archivística*, 2006.
 - LETELIER. *Rational Unified Process (RUP)*, 2007]. Disponible en: <https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20a%20RUP.doc>
 - LLORET, O. *Gestión de la información vs. gestión del conocimiento*, 2004.
 - MOLPECERES, A. *Procesos de desarrollo: RUP, XP y FDD*, 2003. [2007]. Disponible en: http://www.javahispano.org/download/articulos/metodos_desarrollo.pdf
 - PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. 2002. 601 p.
 - RUESTA, C. B. *GESTIÓN DOCUMENTAL EN LAS EMPRESAS: UNA APROXIMACIÓN PRÁCTICA*, 2000. [2007]. Disponible en: <http://www.inforarea.es/Documentos/fesabid.pdf>
 - SAGI, D. J. B. *Introducción a la Gestión del Conocimiento, Gestión Documental y CRM*: 8.
 - SANTOS, N. G. *Avila distingue a Ciego*, 2004. 2007.
 - TUYA, J. *Plan General de Calidad Aplicable al Desarrollo de Equipos Lógicos*, 2004. [2007]. Disponible en: <http://www.di.uniovi.es/~tuya/is/descarga/gestion/CalidadPGC.x2.pdf>

- WERNER, F. G.-G. *La gestión documental y la norma ISO 15489:2001 Record Management*, 2005. 10.

BIBLIOGRAFÍA CONSULATADA

- *Control de código fuente con Subversion*. 2007]. Disponible en: <https://projects.warp.es/projects/inyourface/wiki/SVNTutorial>
- *Factoría de software, más allá de la fabricación industrial*. 2004. [2007]. Disponible en: <http://www.ibermatica.com/ibermatica/eventos/2004/factoriasoftware>
- *gestión documental*. 2005. [Disponible en: http://www.soluzion.com/htdocs/areas/consultoria/servicios/gestion_documental/gestion_documental_interior.shtml
- *Gestión documental*. 2004. [Disponible en: http://www.doc6.es/servicios/gestion_documental.html
- *Gestion Documental. Consultoría de Internet y Comunicaciones*. 2007]. Disponible en: <http://www.forodigital.es/gestiondocumental.htm>
- *Gestion Documental. Consultoría Inforarea*. 2007]. Disponible en: <http://www.inforarea.es/servicio4.htm>
- *Herramientas de software libre para la gestión de contenidos*. 2002. [Disponible en: <http://www.hipertext.net/web/pag258.htm>
- *Integración de sistemas/ Factoría de Software* 2006. [2007]. Disponible en: <http://www.profit.es/22facsof.html>
- *Metodología de Desarrollo de Software (MDS)*. 2005. [Disponible en: <http://www.reynox.com.ar/sap/metodologia.php>
- *Modelos de gestión documental*. 2006. [Disponible en: http://xribas.typepad.com/xavier_ribas/2006/09/modelos_de_gest.html
- *Por qué hacer Gestion Documental?* , 2007]. Disponible en: http://www.openfile.cl/gestion_documental.htm
- *PROYECTOS DE GESTIÓN DOCUMENTAL Y ARCHIVO ELECTRÓNICO*. 2004. [Disponible en: <http://www.abast.es/webabast/proyectos.html?webabast/gestdoc.html>
- *SISTEMA DE GESTION DOCUMENTAL: High Performance Solutions*. 2004. [2007]. Disponible en: <http://www.hps.es/index.php?p=prod&num=8>

-
- StatSVN News. 2007. [2007]. Disponible en: <http://www.statsvn.org/>
 - Subversion. 2006. [2007]. Disponible en: <http://subversion.tigris.org/>
 - TortoiseSVN. 2006. [2007]. Disponible en: <http://tortoisesvn.tigris.org/>
 - *Un cliente de Subversion gráfico: TortoiseSVN* 2007]. Disponible en: <http://mundogeek.net/archivos/2004/06/26/un-cliente-de-subversion-grfico-tortoisesvn/>
 - ARRIBAS, D. C. *Norma ISO 15489 / Especificación MoReq*, 2004. [Disponible en: <http://www.fesabid.org/federacion/gtrabajo/aenor/seminario101204/lcarnicer.pdf>
 - CABALLERO, R. G. *HERRAMIENTAS PARA LA GESTIÓN DE LOS DOCUMENTOS ELECTRÓNICOS EN LOS NUEVOS SERVICIOS DE INFORMACIÓN Y DOCUMENTACIÓN*, 1999. [Disponible en: <http://rayuela.uc3m.es/~bmartin/publicaciones/jornades1999.pdf>
 - D'ALÒS–MONER, A. *La gestión documental: aspectos previos a su implementación*, 2006. [Disponible en: http://eprints.rclis.org/archive/00009540/01/vol15_3.6.pdf
 - INCLÁN, M. D. *ACTUACIONES PARA LA IMPLANTACIÓN DE UN SISTEMA DE GESTIÓN DOCUMENTAL CORPORATIVA*:. Disponible en: http://eprints.rclis.org/archive/00007844/01/Ponencia_2_Maria_Inclan.pdf
 - JACOBSON I, B. G., RUMBAUGH J. *El Proceso Unificado de Desarrollo de Software*. Madrid, 2000. p.
 - MARIANO, J. A. *AUTOMATIZACIÓN DE LA GESTIÓN DE EXPEDIENTES ADMINISTRATIVOS*. Disponible en: <http://www.csae.map.es/csi/tecniemap/tecniemap2002/pdfs/c2.0-29.pdf>
 - ---. *AUTOMATIZACIÓN DE LA GESTIÓN DE EXPEDIENTES ADMINISTRATIVOS*, 2002. [2007]. Disponible en: <http://www.csae.map.es/csi/tecniemap/tecniemap2002/pdfs/c2.0-29.pdf>
 - MENDOZA, F. M. *METODOLOGÍA PARA ESTABLECER UN SISTEMA DE GESTIÓN DOCUMENTAL EN UNA ORGANIZACIÓN*, 2003.
 - RODRÍGUEZ, C. L. *Ejemplo de desarrollo de software utilizando la metodología RUP*, 2003. [Disponible en: <http://www.dsic.upv.es/asignaturas/facultad/lsi/ejemplorup/>

- RUESTA, C. B. *Gestión de documentos: enfoque en las organizaciones*, 2007. [Disponible en: <http://www.inforarea.es/Documentos/Anuario-ThinkEPI-2007-Bustelo-Gestion-documentos.pdf>
- ---. *Gestión documental y gestión de contenidos en las empresas: Estado del arte 2002 y perspectivas para 2003*, 2003. [2007]. Disponible en: http://www.inforarea.es/Documentos/IWE_estado_arte.pdf
- SUAREZ, R. G. *Subversion personal*, 2002. [2007]. Disponible en: http://cronopios.net/Traducciones/subversion_personal.html
- TRAMULLAS, J. *Herramientas de software libre para la gestión de contenidos*, 2005. [2007]. Disponible en: <http://www.hipertext.net/web/pag258.htm>

GLOSARIO DE TÉRMINOS Y SIGLAS

Administrador: Todo aquel usuario que tenga asignado derechos administrativos sobre algo.

Calidad de software: Es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Ciclo de vida de los documentos: Tiempo en que un documento tiene valor para algún tipo de grupo o interesado.

Código abierto: Es el término con el que se conoce al software distribuido y desarrollado libremente.

Documento electrónico: Combina diferentes unidades de información (texto, imágenes fijas o en movimiento, voz, gráficos, etc.); su contenido puede cambiar de soporte con el tiempo, permite establecer relaciones con otros documentos.

ENRON: En cuestión de quince años, pasó de ser una pequeña empresa de gas en Texas, a ser el sétimo grupo empresarial de mayor valor en Estados Unidos.

Extranet: Es una red privada virtual resultante de la interconexión de dos o más intranets que utiliza Internet como medio de transporte de la información entre sus nodos.

Intranet: Es una red de ordenadores de una red de área local (LAN) privada empresarial o educativa.

Ofimática: Referente a trabajo con las aplicaciones del paquete Office de Microsoft.

Permiso: Autorización de un usuario para realizar una acción.

Proceso: Programa de actividades que van a estar organizadas de forma lógica y ordenadas encaminadas a ofrecer un producto o brindar un servicio, contando siempre con entradas, transformaciones y salidas.

Servidor: Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

UML: Es el Lenguaje de Modelado Unificado Orientado a Objetos. Puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software.

WORLDCOM: La segunda empresa telefónica de larga distancia de Estados Unidos.