

Ministerio de Educación Superior



## **Trabajo para optar por el título de Ingeniero Informático**

**Tema:** Sistema Integrado para la Gestión Estadística (SIGE). Rol de Gestión de Configuración y Cambios.

**Título:** La Gestión de Configuración y Cambios en el proyecto productivo para la automatización de la Oficina Nacional de Estadística (ONE)

### **Autores:**

Yenei Yolanda Samada Lorié

Daily Hernández Segura

### **Tutor:**

Ing. Ridosbey Milián Iglesias

### **Cotutor:**

Dr.C Pedro Y. Piñeiro Pérez

Ciudad de la Habana, mayo de 2007.

“Año del 49 aniversario de la Revolución”

***“Los sueños y la perseverancia son una poderosa combinación”***

**WILLIAM LONGGOOD**

## **Dedicatoria**

### **De Daily:**

El presente trabajo está dedicado a Dios y en especial a mi madre Ileana Segura Alfonso, por creer en mí, por su apoyo, su amor, su incondicionalidad, por ser la autora intelectual de este sueño y posibilitar su realización, a esa persona a la que le debo lo que soy y lo que seré, le dedico este trabajo.

Dedico este trabajo además a la revolución y a Fidel por haberme permitido estudiar y desarrollarme profesionalmente.

A mis familiares y amigos por su constante preocupación y ayuda, sobre todo a mi papá y a mi tía Laime.

### **De Yenei:**

Dedico especialmente este trabajo que tanto esfuerzo nos ha costado, a mis padres por ser un ejemplo en mi vida. A ellos les debo todo, a sus enseñanzas e infinito amor...

A mis abuelos por darme fuerzas y principalmente por confiar siempre en mí.

A los profesores por sus valiosas enseñanzas, en especial a la vice-decana Yvonne Caridad Collada Peña por su valioso cariño y su incondicional amistad.

A mis familiares y amigos por estar siempre ahí cuando los necesité...

A todo el que de una forma u otra ha contribuido a que este trabajo salga bien.

## Agradecimientos

### De Daily:

Le agradezco al Señor mi Dios que esta en los cielos el haberme permitido llegar hasta donde estoy, a la Revolución y a Fidel, deseando que esté totalmente restablecido para que pueda participar con nosotros sus estudiantes en nuestra graduación, gracias por las oportunidades inigualables de las que he podido disfrutar, como estudiar, alcanzar un título universitario, pertenecer a esta institución de excelencia y poder crecer profesionalmente de forma gratuita.

Le agradezco a todos mis profesores por la ayuda que me han brindado; Pascual Verdecia Vicet mi amigo y guía, a Pedro Piñeiro, por confiar en mi, por su ayuda en el presente trabajo al igual que mi tutor, Ridosbey Milian.

A mis amigos les doy las gracias por acompañarme siempre, a ti Dinella por la amistad y el apoyo incondicional que me ha brindado en todo momento.

A mi familia, por la cual soy hoy una mejor persona, a mi papá por creer en este sueño y a Tito por apoyarme, gracias.

Le agradezco, no con palabras sino con el infinito amor que le profesó, a mi mamá, la persona a la que le debo lo que soy, mis logros, mis metas, mis triunfos, el poder estudiar, el estar graduándome de profesional, le agradezco por estar siempre ahí, por apoyarme, por ayudarme, por consolarme, por tener fe en mí, por guiar mis pasos al crecer. El presente trabajo y la carrera que estoy a punto de obtener no se hubieran podido realizar sin ella, que siempre está conmigo, a ti mami gracias de todo corazón.

### De Yenei:

Quiero agradecer especialmente a nuestro tutor Ridosbey Milián Iglesias por preocuparse tanto por el trabajo y darnos lo mejor de sí para que todo salga bien.

A Pedro Y. Piñeiro por ser nuestro consultante más cercano y compartir con nosotras sus experiencias y valiosos conocimientos.

A la profesora Eugenia Muñoz Lodos por su consejo siempre certero y oportuno.

A mis padres por la seguridad de poder siempre contar con ellos para lo que sea...

A toda mi familia, por confiar siempre en mí.

A Ileana Segura por su apoyo incondicional y sobre todo por considerarme una hija más.

A todos los profesores, ellos saben que a pesar de sus regaños, siempre se los agradecemos y al final, sus esfuerzos son bien recompensados. A todos ellos mi más sincero respeto e infinito agradecimiento.

A los muchachos del proyecto ONE, por su valiosa ayuda, en especial a Armando Robert Lobo y Henry Cruz Mulet.

A todo el que de una forma u otra ha contribuido a que este trabajo salga bien.

A todos, muchas gracias...

## **Declaración de Autoría**

Declaramos que somos los únicos autores del presente trabajo y por este medio autorizamos a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo cuando estime pertinente.

Para que así conste, firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del 2007.

---

Daily Hernández Segura  
Autor

---

Yenei Yolanda Samada Lorié  
Autor

---

Ridosbey Milián Iglesias  
Tutor

## Opinión del cliente del Trabajo de Diploma

El Trabajo de Diploma titulado “La Gestión de Configuración y Cambios en el proyecto productivo para la automatización de la Oficina Nacional de Estadística (ONE)”, fue desarrollado en la Universidad de las Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente   X  

Parcialmente en un     %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

1. Definición e implantación de un marco de apoyo a las actividades de desarrollo técnico del proyecto, en lo que refiere a gestión de la configuración y cambios, marco de trabajo que definió herramientas y procedimientos que son de vital importancia para el avance del proyecto.

Para tener una idea más exacta: uno de los principales riesgos que pueden afectar al proyecto es el cambio en los requisitos del cliente, provocado por los cambio transformacionales que sufre el negocio o como resultado de decisiones del propio cliente, con una correcta definición de procesos y procedimientos para la gestión de cambios se puede minimizar considerablemente el impacto catastrófico que podría representar este riesgo, lo que demuestra la importancia de la disciplina que se desarrolla en el presente Trabajo de Diploma.

2. Establecimiento de un estándar de expediente de proyecto, en conformidad con lo dictado por la metodología y adaptado a nuestro proceso de desarrollo, permitiendo de esta forma que la información que se genera del software este siempre organizada y actualizada.

De manera genera, en lo que respecta al proyecto el impacto es muy positivo teniendo en cuenta que ha sido llevada a cabo una adecuada gestión de la configuración y control de cambio como disciplina de apoyo al proceso de desarrollo técnico.

Y para que así conste, se firma la presente a los   8   días del mes de   mayo   del año   2007  

  Armando Robert Lobo  

Cliente

  Líder del proyecto ONE  

Responsabilidad

\_\_\_\_\_  
Firma

# Opinión del Tutor

Título: La Gestión de Configuración y Cambios en el proyecto productivo para la automatización de la Oficina Nacional de Estadística (ONE)

Autores: Daily Hernández Segura

Yenei Yolanda Samada Lorié

Tutor: Ridosbey Milián Iglesias.

El tutor del presente Trabajo de Diploma considera que durante su ejecución los estudiantes mostraron las cualidades que a continuación se detallan:

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante Yenei Yolanda Samada Lorié mostró una alta independencia en la ejecución de cada una de las tareas que le fueron asignadas, mostrando gran creatividad y originalidad en sus ideas, además demostró ser una persona muy responsable y trabajadora. Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue la calificación de **5 puntos**.

Por su parte la estudiante Daily Hernández Segura se desempeñó con alto grado de independencia durante todo el desarrollo del trabajo, además en sus ideas se ponían de manifiesto la originalidad, claridad y creatividad. Su responsabilidad, dedicación y laboriosidad fueron de gran importancia para la culminación de este trabajo de diploma. Por todo lo anteriormente planteado considero que el estudiante está en condiciones de ejercer como Ingeniero Informático; y propongo que se le otorgue la calificación de **5 puntos**.

El trabajo realizado contiene un gran valor productivo mostrando calidad en la documentación; los resultados obtenidos son buenos aunque en el futuro la aplicación mas profunda de este trabajo, en otros proyectos puede traer resultados mucho mejores.



Por todo lo anteriormente expresado considero que los estudiantes se encuentran aptos para ejercer como Ingenieros Informáticos; y propongo que se le otorgue al Trabajo de Diploma la calificación de **5 puntos**

---

Firma

---

Fecha

El presente trabajo tiene como base para su realización, la necesidad de hacer un estudio detallado de las funciones y actividades que se deben llevar a cabo en la Gestión de Configuración y Cambios (GCC) en un equipo de desarrollo de software, para la correcta aplicación de las mismas. El principal objetivo es desarrollar una estrategia que permita aplicar un modelo para el establecimiento de una adecuada GCC al proyecto productivo para la automatización de la Oficina Nacional de Estadística (ONE) y de este modo contribuir al logro de la integridad y funcionalidad del producto final, por lo que el tema que será tratado a lo largo del trabajo se basará en las actividades, herramientas y procedimientos que soporta el rol de GCC así como los resultados que se van obteniendo con la aplicación de dicho modelo.

En el capítulo 1 se mostrarán resultados obtenidos de la realización de un estudio crítico y valorativo del estado del tema, las tendencias técnicas y tecnológicas, a nivel de Universidad, de país e internacionalmente. También se definirán términos operacionales de suma importancia para el presente trabajo.

En el capítulo 2 se desarrolla la estrategia aplicando un modelo que describe procesos y procedimientos necesarios que tributen a la puesta en práctica de la GCC en el proyecto ONE y que sirve de base para la obtención de los resultados esperados de acuerdo a las necesidades específicas del mismo.

En el capítulo 3 se hace un análisis de los resultados del proyecto una vez concluido este el trabajo de investigación y luego de comenzar a aplicarse el modelo desarrollado y detallado en el capítulo 2.

The present work responds to the need of a detailed study of the functions and activities that should be carried out in the Software Configuration and Changes Management (SCChM) in a team of software development. The main objective is to develop a strategy that allows us take the first steps in the process of the application of a model to the productive project for the automation of the National Office of Statistics (NOS), in order to achieve of the integrity and functionality of the final product. The treatment of the topic is based on activities, functions and tools that support the role of SCChM. The results of the work are present in the mentioned productive project.

A critical analysis of the state of the art in chapter 1 is offered as well as its techniques and technological tendencies within the University, nationally and internationally. The operational terms of utmost importance are also presented in this topic.

In chapter 2 is developed the strategy applying a model of the process and procedures are described which are necessary for the implementation of the SCChM in the project. It also serves as the base to obtain the good results.

The analysis of the results of the project is presented in chapter 3. Here a detailed explanation of the model and the procedures described the chapter 2 are offered.

Introducción .....	1
Capítulo1 Fundamentación Teórica .....	10
1.1 Conceptos básicos de la Gestión de Configuración del Software.....	10
1.1.1 ¿Qué se entiende por Gestión de Configuración del Software? .....	10
1.2 Herramientas para la Gestión de la Configuración .....	11
1.2.1 Herramienta para la Gestión de la Configuración SYNERGY/CM .....	12
1.2.2 Herramienta para el manejo de repositorios. Fisheye.....	13
1.3 Conceptos adicionales presentes en la disciplina GCS .....	15
1.4 Tendencias actuales del uso del rol de GCS.....	16
1.5 Control de versiones .....	20
1.5.1 Herramienta para el control de versiones Concurrent Versions System (CVS).....	22
1.5.2 Herramienta para el control de versiones Visual Source Safe (VSS) .....	23
1.5.3 Herramienta para el control de versiones Subversion (SVN).....	24
1.6 Situación de la GCC del proyecto en cuestión .....	27
Capítulo 2 Desarrollo del Modelo a aplicar .....	31
2.1 Organización del expediente del proyecto.....	31
2.1.1 Organización del equipo de trabajo por módulos del proyecto ONE .....	35
2.2 Modelo para la Gestión de Configuración de Software y Cambios .....	36
2.2.1 Definición y establecimiento de la Configuración .....	37
2.2.1.1 ECS seleccionados .....	38
2.2.1.2 Relaciones entre los ECS .....	40
2.2.1.3 Análisis a profundidad de las relaciones (dependencias y asociaciones).....	47
2.2.1.4 Establecimiento del Esquema de Identificación.....	49
2.2.1.5 Procedimiento para establecer las líneas base .....	50
2.2.1.6 Definición y establecimiento de las líneas base .....	51
2.2.2 Gestión y control de los cambios en la configuración. ....	52
2.2.2.1 El Control de los Cambios.....	52

2.2.2.2	Proceso formal para la realización de cambios .....	54
2.2.2.3	Procedimiento para la solicitud de cambios.....	57
2.2.2.4	Procedimiento para aprobar o rechazar las solicitudes de cambios .....	59
2.2.2.5	Procedimiento para el seguimiento de los cambios .....	59
2.2.3	Procedimiento para el Control de Versiones .....	60
2.2.3.1	SVN, herramienta para el control de versiones utilizada en el proyecto ONE. ....	62
2.2.4	Generación de informes de estado.....	64
2.2.4.1	Registros .....	65
2.2.4.2	Informes .....	68
2.2.5	Auditoría de la Configuración del Software .....	69
Capítulo 3	Análisis de resultados .....	71
3.1	Expediente de proyecto .....	71
3.2	ECS .....	71
3.2.1	Avance del proyecto según la GCS.....	72
Conclusiones	.....	79
Recomendaciones	.....	80
Bibliografía Consultada	.....	81
Bibliografía Citada	.....	83
Anexos	.....	84
Anexo 1	Artefactos seleccionados para el desarrollo del Proyecto Productivo.....	84
Anexo 2	Orden de Cambio de la Configuración (OCC).....	104

La Oficina Nacional de Estadística (ONE) es la institución encargada de llevar a cabo el control de todas las estadísticas que se mueven en el país y que son de vital importancia para la planificación y control de la economía, de los servicios, etc. algunas de estas estadísticas por así citarlas pueden ser: la producción de cierta empresa en un día, la cantidad de alumnos que asistieron a una escuela en un día, la cantidad de personas que nacieron en una provincia o en el país en un tiempo determinado, en general todo tipo de estadística que circula en el país. La ONE, debe brindarle a la dirección del país periódicamente reportes sobre todas estas cuestiones. Para lograr este objetivo la ONE cuenta con la Oficina Municipal de Estadística (OME) ubicada en cada municipio del país que le brinda la información a la Oficina Territorial de Estadística (OTE) que se encuentra en cada provincia y estas a su vez son las encargadas de proporcionar la información a la ONE, creando así un flujo de información en ambos sentidos (ascendente y descendente).

Para automatizar eficientemente las actividades que se llevan cabo en cualquiera de estas oficinas se hace necesaria la creación de un sistema integral para la gestión estadística capaz de procesar toda la información con la que cuentan y que sustituya al existente (Microset) por encontrarse obsoleto y no cubrir las necesidades actuales de los clientes.

El logro de un producto íntegro y funcional depende en gran medida de la implantación de una buena GCC por lo que en lo adelante el tema que será tratado estará encaminado a definir actividades, funciones y herramientas que soporta la GCC para aplicarlas al proyecto productivo encargado de llevar a cabo la elaboración del software.

De manera general se entiende como GCC a una disciplina que comprende tareas y actividades encaminadas a lograr la organización del proyecto y el control de las diferentes versiones por las que va pasando el producto que se desea lograr.

Las ideas principales de estas actividades son:

1. Identificar los Elementos de Configuración del Software (ECS) que forman parte del sistema y controlar los cambios que estos sufren.

2. Documentar, probar, reportar y registrar periódicamente el estado en que se encuentran estos elementos y las distintas solicitudes de cambios que se van generando.

El resultado esperado con el cumplimiento de estas actividades es garantizar la calidad, visibilidad e integridad del producto final y de los diferentes ECS que lo conforman. Todas estas tareas y actividades forman parte de un flujo de trabajo de soporte de la metodología RUP que comienza con el nacimiento del proyecto y culmina con su retiro.

### **Situación problemática**

Actualmente en la Universidad de las Ciencias Informáticas (UCI) y en el país, en la mayoría de los casos, no se aplican correctamente las tareas y funciones que según la metodología utilizada se definen para llevar a cabo una buena GCC. Este aspecto dificulta la gestión de la información y el control de versiones de los diferentes ECS del proyecto.

Además es preciso establecer un adecuado conjunto de procedimientos que garanticen la calidad de la ejecución del proceso y evite la dependencia absoluta de factores humanos. Estas buenas prácticas permiten atenuar los riesgos del proyecto.

No contar además con una buena política de gestión de cambios internos y externos, poseer pocas plantillas (algunas en inglés) y artefactos que sustenten una buena organización del proceso y de la información del software así como la comunicación con el cliente, puede provocar insatisfacciones y atrasos.

La GCC es parte de la base de la organización del proyecto y el desarrollo en equipo, elementos esenciales para el desarrollo de productos de software.

### **Problema Científico**

La no aplicación de las técnicas de Gestión de la Configuración tales como: la correcta aplicación de las tareas y funciones para llevar a cabo una buena GCC, procedimientos que garanticen la calidad de la ejecución del proceso de informatización de la ONE, una buena política de gestión de cambios internos y externos, la falta de plantillas y artefactos para la gestión y documentación; afecta a la gestión de información del proyecto y el control de versiones de los diferentes elementos de configuración, a la calidad del proceso, a la organización y aceptación del producto.

## **Hipótesis**

Si se lleva a cabo una buena GCC: procedimientos que garanticen la calidad de la ejecución del proceso de informatización de la ONE, una buena política de gestión de cambios internos y externos así como la introducción de plantillas y artefactos para la gestión y documentación, el uso adecuado de herramientas de control de versiones, entonces es posible lograr una adecuada gestión de la información del proyecto, el control de versiones de los diferentes elementos de configuración, calidad en el proceso de desarrollo del software, una buena organización del trabajo y la aceptación del producto final.

## **Variables**

### **Variables independientes:**

- Buena política de cambios internos y externos.
- Introducción de plantillas y artefactos para la gestión y documentación.
- Uso adecuado de las herramientas de control de versiones.

### **Variables dependientes:**

- Adecuada gestión de la información del proyecto.
- Calidad en el proceso de desarrollo de software.
- Buena organización del trabajo.
- Aceptación del producto final.

## **Indicadores:**

- Elaboración de las plantillas para la creación de artefactos.
- Uso de la herramienta para el control de versiones por parte de los miembros del proyecto.
- Dominio y uso de la herramienta para el control de versiones por parte de los miembros del proyecto.
- Dominio y uso de la herramienta para el control de versiones por parte de los responsables del rol de GCC.
- Documentación y control de los cambios a los ECS.
- Satisfacción de los clientes.
- Cantidad de elementos de configuración en desarrollo.



- Cantidad de elementos de configuración en revisión.
- Cantidad de elementos de configuración en mejoras de revisión.
- Cantidad de elementos de configuración terminados.
- Cantidad de no conformidades o solicitudes de cambios realizadas por los clientes
- Cantidad de solicitudes de cambios en espera de ser revisadas
- Cantidad de solicitudes de cambios en desarrollo
- Cantidad de solicitudes de cambios en revisión
- Cantidad de solicitudes de cambios cerradas
- Situación del Expediente del proyecto
- Evaluación de los Procedimientos llevados a cabo en el proyecto

### **Objeto de estudio**

Proceso de investigación, comportamiento y aplicación de la GCC en la Universidad.

### **Campo de acción**

Proceso de investigación, comportamiento y aplicación de la GCC en el proyecto productivo de desarrollo e informatización de la ONE.

### **Objetivo**

Desarrollar una estrategia que permita aplicar un modelo para el establecimiento de una adecuada GCC al proyecto productivo para la automatización de la ONE.

### **Objetivos específicos**

1. Desarrollar un modelo de GCC para aplicar al proyecto de la ONE, teniendo como base el modelo definido por la IEEE para la correcta aplicación de la misma. Dicho modelo será adaptado a las necesidades específicas del proyecto.
2. Comenzar a aplicar el modelo para la GCC en el desarrollo del sistema para la automatización de la ONE.
3. Analizar los resultados luego de la aplicación del modelo para la GCC.

## **Tareas a desarrollar para cumplir los objetivos**

1. Hacer un estudio del estado del arte acerca de las diferentes tendencias de uso y aplicación de la GCC y sus herramientas.
2. Establecer y aprobar el Plan de Gestión de la Configuración, artefacto en el cual quedan claras las especificaciones o productos revisados y aprobados formalmente que sirven como base para el desarrollo posterior y el cual solo puede ser modificado a través de procedimientos formales de control de cambios.
3. Estudio detallado para captar las unidades físicas y/o lógicas producidas o adquiridas, que por sus características son distinguibles de las demás y cuya evolución interesa administrar, distinguiendo así los ECS.
4. Identificar la configuración del software en puntos dados en el tiempo, lo que se conoce actualmente como líneas bases de la configuración del software.
5. Mantener la integridad y trazabilidad de la configuración a través del ciclo de vida del software utilizando herramientas para el control de sus versiones.
6. Estudio comparativo de las herramientas para el control de versiones existentes hasta el momento para identificar la candidata a ser utilizada.
7. Establecer y aprobar el Plan de Gestión de cambios, artefacto que rige formalmente los procedimientos para realizar un cambio y que registrará todo este proceso.
8. Controlar sistemáticamente los cambios en la configuración.
9. Establecimiento de artefactos entregables y plantillas correspondientes que rigen la documentación.
10. Realizar auditorías y revisiones para probar el trabajo que se desempeña y obtener un producto de calidad.
11. Informar a todo el personal de trabajo y clientes sobre el seguimiento del estado de los distintos ECS, de los cambios, versiones, etc., que se realizan o establecen en el proyecto que son de suma importancia conocer, llevado a cabo mediante informes de estado.

## **Metodología utilizada**

Para llevar a cabo las tareas especificadas para la GCC utilizamos la metodología que define el Proceso Unificado de Desarrollo (RUP) con la herramienta Rational Rose Enterprise Edition.

RUP es una metodología que define Quién (Trabajadores) debe hacer Qué (Artefactos), Cuando (Flujos de Trabajo) y Cómo (Actividades) debe hacerlo.

Cuenta con 8 Flujos de Trabajo entre los cuales se encuentra el de Gestión de Configuración y Cambios que se considera un Flujo de trabajo de soporte (Ver figura 1), para el cual se encuentran bien definidas actividades (Ver figura 2) y artefactos (Ver figura 3), las cuales se tendrán en cuenta para ser aplicadas al proyecto.

Los aspectos definitorios de RUP, y que lo hacen uno de los más completos, se pueden resumir en tres características esenciales: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos para documentar la situación del proyecto, convirtiéndose de esta manera en una de las metodologías más importantes para alcanzar un grado de mayor calidad en el desarrollo del software.

RUP es la metodología que más se utiliza en la UCI y con la cual los equipos de desarrollo se encuentran más familiarizados por ser fácil de usar y por tener todas las actividades, flujos de trabajo y fases bien definidos.

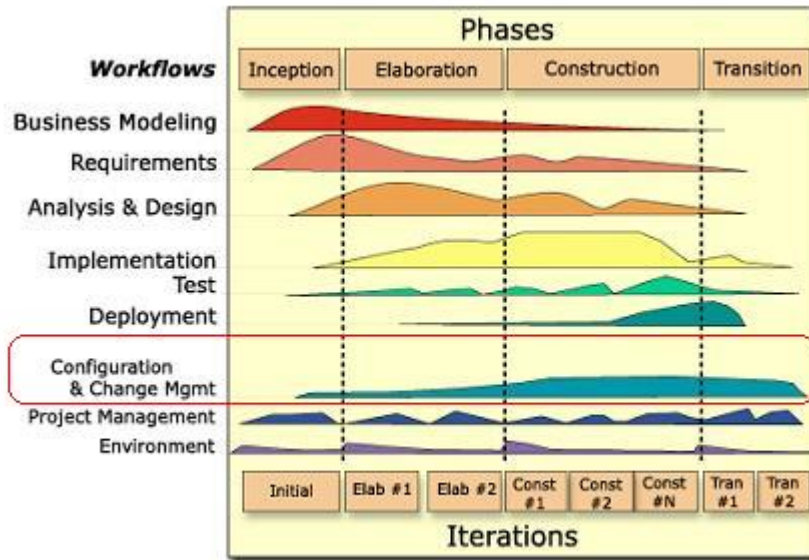


Figura 1. Flujos de trabajo, fases e iteraciones de la metodología RUP

### Configuration & Change Management: Activity Overview

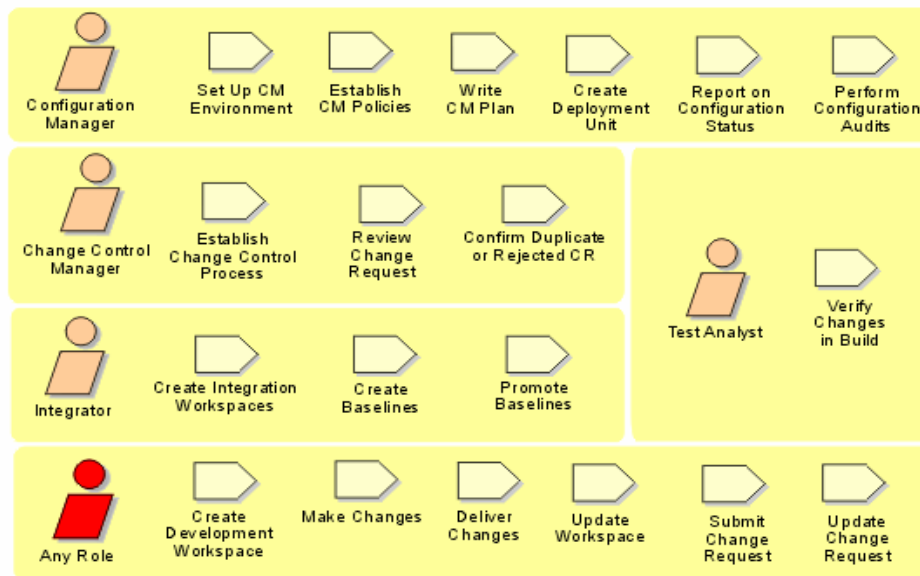


Figura 2. Actividades que define RUP para la Gestión de Configuración y Cambios

## Configuration & Change Management: Artifact Overview

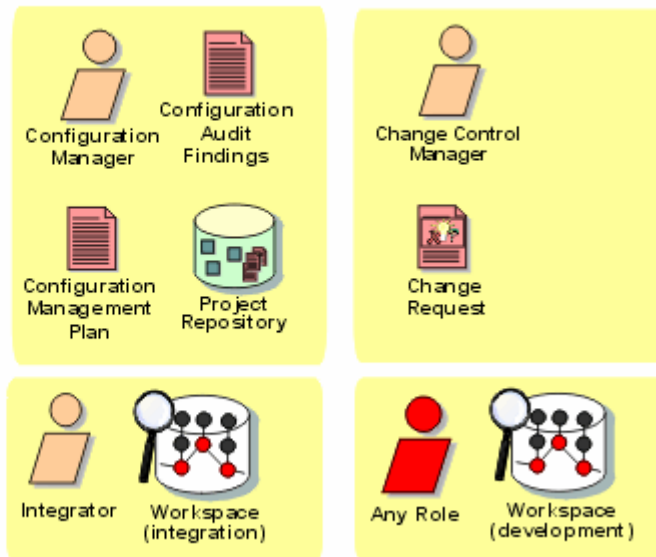


Figura 3. Artefactos principales que define RUP para la Gestión de Configuración y Cambios

### Estrategia de investigación

Para el desarrollo de la investigación se utilizaron las estrategias de investigación exploratoria y la explicativa.

Exploratoria porque se exploraron diferentes técnicas y tendencias en el estudio de la GCS así como herramientas y métodos para la aplicación del rol en el proyecto con el objetivo de desarrollar un producto eficiente y de calidad. Teniendo en cuenta además la arquitectura propuesta y las herramientas de desarrollo preseleccionadas para el desarrollo del sistema de gestión de información estadística de la ONE.

Se utilizó la explicativa ya que se determinaron los distintos problemas y causas que producen una mala y/o poca aplicación de la GCS, de la organización y documentación de un proyecto, a los cuales se les pretende dar solución con la realización de este trabajo ajustándonos a las características específicas del producto a desarrollar.

## **Métodos teóricos: Histórico \_ lógico, Hipotético \_ deductivo, sistémico**

### **Método histórico\_ lógico**

La GCS posee una estrecha relación con el producto y el proceso de desarrollo. En la primera parte de la investigación se desarrolló un estudio del estado del arte de la problemática analizada; se revisaron las diferentes tendencias y estrategias para la gestión de configuración. Además se analizan experiencias anteriores en el desarrollo de sistemas estadísticos y de sistemas para la gestión de información de la oficina nacional de estadística.

### **Método hipotético\_ deductivo**

La investigación que se realizó sigue un método hipotético \_ deductivo porque a partir del problema concreto al que se pretende dar solución, o sea el poco uso de la GCS en nuestro país y en la Universidad, se plantearon objetivos específicos, tareas e hipótesis que serán resueltas con el desarrollo de la investigación siguiendo métodos científicamente fundamentados, procedimientos y técnicas.

### **Método sistémico**

Se planteó el problema como un todo, donde las informaciones obtenidas, la dinámica de estudio de los distintos esquemas y uso de la GCS y cada uno de los problemas que se resolvieron, métodos y herramientas desarrolladas para facilitar el trabajo, el desempeño correcto de la GCS, el análisis del proceso de evolución del software se funden en un sistema sostenible e integral, en el que cada elemento esta estrechamente relacionado con los demás.

### **Método Empírico**

Conjuntamente con los métodos teóricos y las estrategias de investigación también se siguieron métodos empíricos basando la investigación en la observación y la experimentación.

A partir del estudio y aplicación de la GCS al proyecto del que somos parte, se observó y analizó la evolución y desarrollo del producto así como el manejo de la organización y el ambiente de trabajo, se registró el comportamiento de dichos elementos de forma favorable o no favorable y se definieron los eventos y sucesos más significativos.

Mediante la aplicación de la GCS fueron seleccionados y desarrollados una serie de procedimientos y artefactos para organizar y gestionar adecuadamente las versiones de los diferentes elementos de configuración del proyecto.

### 1.1 Conceptos básicos de la Gestión de Configuración del Software.

#### 1.1.1 ¿Qué se entiende por Gestión de Configuración del Software?

La Gestión de Configuración del Software (GCS) es una disciplina en progreso. Actualmente la demanda de software con calidad está creciendo cada vez más, como necesidad tanto de personas con sólidos conocimientos y experiencia en la GCS como para personas que recién comienzan su estudio.

Debido a la importancia que se le confiere a la GCS dentro del proceso de desarrollo del software y para suerte nuestra han sido varias las personas que han dedicado gran parte de su tiempo y esfuerzo a estudiar su progreso. Algunos de los criterios emitidos por estas personas se muestran a continuación:

1. *“La Gestión de Configuración del Software es uno de los procesos clave para toda organización dedicada a la Ingeniería del Software, ya que posibilita una mejor organización del desarrollo y mantenimiento, consiguiendo la visibilidad del producto y facilitando el resto de procesos de producción” [1]*
2. *“La gestión de configuración del software es una actividad de protección que se aplica a lo largo de todo el proceso del software. La GCS identifica, controla, audita e informa de las modificaciones que invariablemente se dan al desarrollar el software...” [2]*
3. *“Configuration management is unique identification, controlled storage, change control, and status reporting of selected intermediate works products during the life of software” [3]*
4. *“Gestión de Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.”[4]*

5. *“El arte de coordinar el desarrollo de software para minimizar la confusión, se denomina Gestión de Configuración. La Gestión de Configuración es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores.” [5]*

En Cuba, la GCS juega un papel protagónico en el éxito de cualquier proyecto de desarrollo de software. Es justo agregar que la labor que realiza cualquier persona que lleve a cabo la GCS es tan importante como compleja. La complejidad está dada por la cantidad de componentes a controlar, los cuales aumentan conforme el proyecto se desarrolla. Pero ciertamente el problema se incrementa cuando a medida que aumentan en número los componentes directamente aumentan también los cambios por los que pasan.

Para asumir estos cambios hay que estar bien preparados pues ellos son inevitables y además de alguna manera están justificados pues a medida que avanza el software se conoce más el problema y se cuenta con más base para resolverlo.

## **1.2 Herramientas para la Gestión de la Configuración**

En el mundo actualmente existen varios sistemas que gestionan automáticamente muchas de las funciones que se llevan a cabo por la GCS. Uno de los sistemas sacados recientemente al mercado es el Synergy que tiene un módulo de trabajo para la Gestión de la Configuración (Synergy/CM) y otro para la Gestión de los Cambios (Synergy/Change).

Una herramienta muy poderosa para gestionar todos los cambios que se realizan en el repositorio de control de versiones es FishEye, la cual será abordada posteriormente por las facilidades que brinda para el manejo del repositorio.



### 1.2.1 Herramienta para la Gestión de la Configuración SYNERGY/CM

Synergy es un Software que posibilita automáticamente gestionar todo lo que tiene que ver con la configuración del Software y brinda muchas facilidades pero no se ha podido probar por ser privativo y costar una gran cantidad de dinero. A pesar de todo se entiende de gran importancia mostrar sus características:

*“... SYNERGY/CM, una solución de gestión de la configuración basada en tareas, permite a los equipos de desarrollo de software trabajar más rápida y fácilmente aumentando la comunicación y la colaboración. SYNERGY/CM acelera la gestión de releases y los procesos de gestión de builds, maximiza la eficacia de los recursos de desarrollo limitados y une los equipos de desarrollo distribuidos. SYNERGY/CM proporciona un eficaz repositorio distribuido y un enfoque probado de workflow orientado a equipos para el desarrollo de software.”*[6]. Este sistema controla la mayoría de las actividades que tienen que ver con la GCS pero no incluye las tareas para la Gestión de Cambios (GCa). Para automatizar las actividades de GCa existe otro sistema, el “SYNERGY/Change”.

*“... SYNERGY/Change es una solución de gestión del cambio totalmente integrada y basada en Web para el seguimiento e informe de peticiones, que:*

- *Aumenta la calidad reduciendo el riesgo de implementar cambios no deseados o no autorizados*
- *Proporciona mayor responsabilidad con el seguimiento exhaustivo de todas las peticiones de cambio*

*SYNERGY/Change simplifica el proceso de gestión del cambio y permite a las organizaciones responder de forma sistemática las peticiones de cambio desde fuentes internas y externas.*[6]

Estos sistemas que automatizan la GCS y la GCa incluyen algunas funciones adicionales que vienen a complementar el trabajo que concierne al rol de GCS, lo que nos lleva a ampliar la concepción que hasta el momento se tenía de la GCS, estas actividades se enumeran a continuación:

- **“Construcción:** *Consiste en coordinar eficientemente la relación, o sea el enlazado de los componentes del software que se quiere lograr.*

- **Control del Trabajo en Equipo:** Consiste en la gestión y control de las relaciones e interacciones entre los diferentes roles que conforman el equipo de desarrollo de software a la hora de trabajar con elementos comunes.
- **Control de Versiones:** Consiste en mantener un registro histórico de las diferentes versiones por las que pasan los componentes de un producto, que permita la recuperación en cualquier momento de cualquiera de ellas.
- **Gestión de Problemas:** Consiste en realizar un seguimiento de la evolución de los problemas que afectan al producto.” [6]

## 1.2.2 Herramienta para el manejo de repositorios. Fisheye

Fisheye es muy fácil de usar si se tiene en cuenta que facilita en gran medida el manejo de toda la información que es almacenada en el repositorio mostrando gráficos, tablas de la información y trazas del historial de los cambios (Ver figura 4). El propósito fundamental de esta herramienta es el de mejorar el entendimiento de toda la información que a cambios se refiere.

Fisheye proporciona un grupo de características diseñadas para darle una comprensión mejor del estado actual del contenido del repositorio, y cómo ha cambiado en un cierto plazo. Las visualizaciones se hacen a pedido del usuario por lo que tiene siempre información actual.

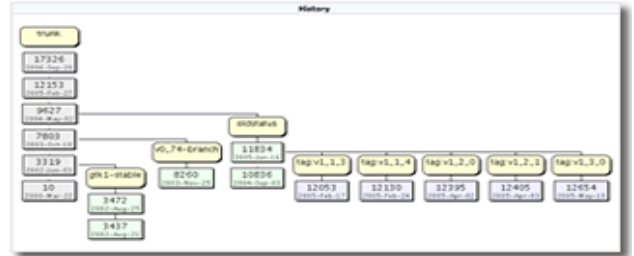


Figura 4. Ejemplo de gráficos, tablas y trazas de la información sobre los cambios que se puede extraer del repositorio

## Facilidades de análisis

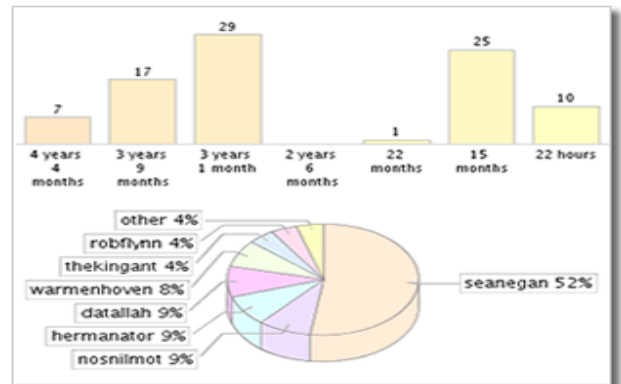
### 1. Tablas de revisiones históricas:

- Vista resumida de las revisiones históricas mostrado a través de ramas
- Se pueden acceder a la información de una revisión o una rama específica



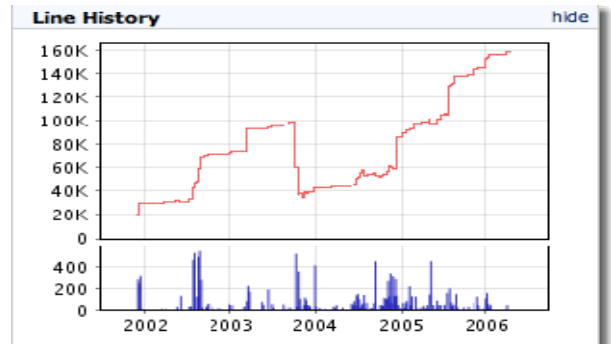
### 2. Información sobre los cambios

- Age histogram muestra una descripción rápida de cómo los archivos han cambiado en un cierto período dando estadísticas precisas del estado de los cambios.



### 3. Tablas Linecount

- Linecount actúa sobre los nodos del repositorio, incluyendo los archivos individuales, brindando información general del contenido del repositorio.
- Los datos de Linecount se actualizan en tiempo real



### 1.3 Conceptos adicionales presentes en la disciplina GCS

- **Configuración:** *Las características funcionales y físicas de una versión específica de hardware y elementos de software que combinados de acuerdo a procedimientos de construcción específicos cumplen un propósito particular.[4]*
- **Configuración del software:** Conjunto de toda la información valiosa y productos utilizados o producidos durante el ciclo de vida de un software. Es por tanto quien define los ECS.
- **Elementos de configuración de software (ECS):** Se considera una unidad física y/o lógica, producida o adquirida, que por sus características se distingue de las demás y cuya evolución interesa administrar. Constituyen ECS los documentos, prototipos de interfaces, códigos fuentes, código objeto y ejecutables, estándares, etc.

Entre los ECS deben establecerse relaciones que delimiten hasta que punto hay dependencia entre ellos. Estas relaciones pueden ser:

- **Relaciones de dependencia:** Cuando los ECS dependen de otros elementos, o de varios requisitos o funcionalidades, que nos facilitan el seguimiento y control de dichos elementos como una unidad integrada.
  - **Relaciones de equivalencia:** Existen varios ECS almacenados en distintas partes y estos responden al mismo ECS, o sea cuando se trata de varias copias del mismo elemento pero en distintas localizaciones.
  - **Relaciones de composición:** Se establecen estas relaciones cuando un ECS esta compuesto por otros artefactos, otros elementos, otros módulos o por otros ECS.
  - **Relaciones funcionales:** Se establecen cuando varios ECS responden o comparten la misma funcionalidad.
- 
- **Artefacto:** *Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad y es candidata a ser tenida en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento. [7].*

- **Variantes:** Son aquellas versiones de los ECS que en un momento dado dan cumplimiento de forma simultánea a requisitos que debe cumplir el Software y poseen características diferentes entre ellas.
- **Versión:** *Conjunto de artefactos relativamente completo y consistente, entregado a un usuario interno o externo; entrega de tal conjunto.*[7] Una versión es una instancia de un ECS.
- **Línea base:** *Conjunto de artefactos revisados y aprobados que representa un punto de acuerdo para la posterior evolución y desarrollo, y solamente puede ser modificado a través de un procedimiento formal, como la gestión de cambios y configuraciones.*[7]
- **Herramientas para el control de versiones:** Son herramientas que se encargan de identificar las distintas versiones por las que pasa cada elemento, de mantener la información de la relación que existe entre ellos y de su almacenamiento en un repositorio.
- **Repositorio:** Es la entidad en la que se archivan los ECS de un proyecto. Opcionalmente se puede identificar un segmento especial en el que se almacenarán los elementos reutilizables. Todo lo que se almacena en el repositorio debe estar adecuadamente identificado y catalogado, para facilitar su recuperación en caso de necesidad. Forman parte indisoluble de los sistemas para el control de versiones. Al repositorio no se puede acceder directamente, sino que son actualizados mediante copias de trabajo hechas por los clientes, sobre esas copias realiza los cambios y una vez terminados el repositorio se actualiza.

## 1.4 Tendencias actuales del uso del rol de GCS

### Oficina Nacional de Estadísticas (ONE)

La ONE no es una empresa productora de Software pero debido al creciente volumen de información que se hacía necesario manejar en la empresa y el avance de las tecnologías se plantea la necesidad de desarrollar un sistema integral que automatizara y gestionara el creciente número de información y que facilitara el manejo de la misma. Es así como surgen sistemas como el VALEDI y el SUPRE en Santiago de Cuba, en Villa Clara se desarrolló el VALTV; en La Habana se desarrolló el SISTEMA MONITOR orientado a determinada información, que trabajaba con programas estadísticos teniendo como base la

información que se le pasaba. Luego aparece el SETAMOT que comenzó a presentar limitaciones y es así como se desarrolla la primera versión del MICROSET, que en su versión actual sirve como base para la realización del sistema automático para la ONE (SIGE).

La información acerca de los software parecidos salidos al mercado y la estructuración del sistema lo llevaron a cabo los desarrolladores mediante reuniones o simplemente pidiendo ayuda a personas con un poco más de experiencia en el tema, de alta calificación y que de cierto modo tenían que ver con las ideas que se estaban llevando a cabo, en otras palabras hacían algo parecido a lo que se le llama hoy estudio del arte de la situación de la Gestión de Configuración solo que en ese tiempo no se concebía con ese nombre, aunque se contaba con una estructura de equipo parecida a la que se tiene en la actualidad en el proyecto.

Los principales problemas que se presentaron en el desarrollo y puesta en práctica del software en años anteriores, eran los cambios que se realizaban a ciertos requisitos o a partes del sistema, los cuales no eran informados por lo tanto no se tenían en cuenta las afectaciones que provocaban a otros módulos o partes del sistema, situación que provocaba inconformidades por parte de los clientes y/o que se hiciera compleja la gestión de la información debido al volumen cada vez más creciente de los datos recopilados, casi toda quedaba sin documentar, situación que daba al traste con que los productos finales no llegaran a tener nunca las funcionalidades para las cuales supuestamente fueron creados, aunque hay que destacar que además de la no correcta o nula gestión de los cambios, el avance tecnológico jugó un papel fundamental en los atrasos de los proyectos de desarrollo de software y el hecho de que no existía el rol de GCS independientemente de que se realizaban tareas y funciones de la misma pero se omitían otras también importantes como el seguimiento y control de versiones además estas tareas se realizaban de forma manual.

## **Procyon**

Procyon es una empresa que radica en la UCI. En entrevista realizada a Javier Ramón García se obtuvieron los siguientes datos:

Procyon es una empresa que se dedica a la elaboración de software para teléfonos celulares, centrándose en tres áreas fundamentales:

- 1- Software que se ejecuta dentro de dispositivos móviles, como juegos.
- 2- Producción de otros tipos de contenidos como imágenes, animaciones, sonidos, temas (skins) para teléfonos. Estos contenidos también necesitan adaptaciones para abarcar la mayor cantidad de modelos, contando también con software con ese propósito.
- 3- software usado por operadores de telefonía celular o "agregadores" de contenidos.

En esta empresa la GCS se utiliza principalmente para el control de versiones con la herramienta SVN y el Trac. Utilizan para la elaboración del software, el lenguaje Java con la framework Enterprise Java Beans, aunque cuenta con proyectos que utilizan Visual Studio como herramienta de desarrollo por las poderosas facilidades que brinda.

Desde el comienzo de Procyon decidieron usar el SVN, por reconocer las ventajas que presenta esta herramienta y además por entender las limitaciones presentadas por CVS.

En Procyon cuentan con más de 50 repositorios. La documentación se guarda en los propios repositorios y mucha información relacionada con el proyecto se guarda en la Wiki del Trac del proyecto correspondiente. Cada proyecto cuenta con su repositorio y su sitio Trac asociado.

El trac lo utilizan para gestionar y darle seguimiento a los defectos, las tareas pendientes, las mejoras que se pueden hacer, etc. Utilizando principalmente los "tickets", los cuales están asociados a hitos, que son precisamente las fechas de entrega de los entregables, valga la redundancia.

Cuentan con sistemas de builds. Un build es una construcción de entregables que se realiza desde las fuentes. Es un proceso automatizado y que ocurre en un servidor, el servidor de builds. En ese servidor el ambiente es controlado. Esto permite garantizar que los entregables oficiales producidos en ese servidor se archiven y sean identificados correctamente mediante firmas digitales. Asegura que todas las aseveraciones sobre la calidad de un entregable se corresponden con ese y no con otro entregable.

Su sistema de build incluye la posibilidad de ejecutarse también en la máquina de un programador o cualquier otro integrante del proyecto. En ese caso se garantiza que el ambiente de la máquina de un programador sea igual al ambiente de trabajo de otro programador, disminuyendo los problemas de configuración debido a diferencias. De hecho la instalación del software necesario, y su configuración corre a cuenta de esa parte del sistema de build. Y eso no es distinto de lo que ocurre en el propio servidor de builds.

## **Softel**

Softel es otra de las empresas productoras de software que radica en la UCI.

Según entrevista realizada a la compañera: Lucy Cruz Águila, especialista de la Dirección de Desarrollo en dicha empresa, resulta indispensable la GCS por lo importante de la tarea.

Actualmente trabajan con la herramienta para el control de versiones VSS, aunque están analizando la posibilidad de migrar paulatinamente a utilizar el Clear Case por la necesidad real que tienen de:

1. Integrar el ClearQuest (herramienta de gestión de cambio que se utiliza en la empresa) con una herramienta de control de versiones, que pudiera ser el VSS; pero consideran que les va a ser más útil migrar al ClearCase teniendo en cuenta sus beneficios y la política o estrategia empresarial de utilizar herramientas de forma integrada.
2. Dar continuidad a la vida del producto una vez instalado al cliente: En la empresa tienen la concepción de que cuando un software es bueno no debe morir.
3. No hacer al software dependiente a factores humanos: Actualmente existe mucha inestabilidad laboral y en ocasiones no se cuenta con las personas encargadas de algunas tareas y al prescindir de dichas personas el software puede quedar “en el aire”, sin embargo si se cuenta con una documentación concreta, actualizada y disponible del producto, los problemas se minimizan.

El trabajo con la herramienta la consideran indispensable, con el VSS gestionan todo lo que tiene que ver con código, documentación e incluso informaciones adicionales que son de su interés almacenar en el repositorio.

En la entrevista se recogió, que debido a la experiencia acumulada luego de largos años de trabajo, las personas que trabajan con el VSS tienen pleno conocimiento de su utilización y transmiten sus



conocimientos con el objetivo de lograr productividad . Hasta el momento no han presentado ningún problema a la hora de trabajar con dicha herramienta, sobre todo porque cuentan con políticas internas de la empresa que posibilita la seguridad e integridad de la información que manejan, dígase la utilización de discos espejos, políticas de salvas automáticas y salvas mensuales de la información, cuyas salvas son guardadas en la bóveda de Segurmática para la protección de dicha información.

Para la correcta realización de las actividades de GCS en la empresa sesiona por proyectos los CCC los cuales cuentan en la empresa con procesos y procedimientos definidos, los cuales son llevados al pie de la letra, pero no solo eso, sino que son enriquecidos o actualizados una vez que detectan que no están funcionando bien por lo que han llegado a alcanzar un alto grado de perfeccionamiento en este sentido.

A modo de conclusión se puede decir que:

Las empresas mencionadas anteriormente tienen diferentes formas de llevar a cabo la GCS, por ejemplo: Procyon aplica la GCS con conocimientos empíricos, técnicos, o sea, no tienen definidos procedimientos, no cuentan con un Comité de Control de Cambios para gestionar los cambios (este comité se encarga de analizar los cambios y si estos se aceptan o no), no tienen establecidos ECS, etc, sin embargo aplican cosas novedosas que hacen más sencillo el trabajo, tal es el caso del sistema Build, que como se explicó anteriormente, es un software creado por ellos que automatiza muchas de las actividades de la GCS.

Softel por su parte aplica la GCS de una forma teórica, o sea, cuenta con procedimientos bien definidos, poseen un equipo estructurado donde cada persona es responsable de las tareas que se le asignan, controlan los cambios mediante la herramienta VSS.

La ONE, al no ser una empresa productora de software no aplica la GCS.

## **1.5 Control de versiones**

Para gestionar los pequeños cambios en la información de un proyecto (tildes, letras, cambios de párrafos, entre otros que no necesitan de todo un procedimiento de cambio por su simpleza) se hace uso de herramientas para el control de las versiones, mediante la cual se puede dar seguimiento a la documentación y a los componentes que son de importancia para el proyecto, velando así la calidad y la correcta gestión de la información.

Se llama control de versiones a la gestión de versiones (revisiones) de todos los elementos de configuración que forman la línea base de un producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, junto a las posibles especializaciones realizadas para algún cliente en específico.

El control de versiones ha sido utilizado por los programadores y la industria informática como un estándar para controlar las distintas versiones del código fuente, generalmente después que se crea un software se emplea un determinado tiempo para hacer y deshacer pequeños cambios hasta lograr el producto deseado.

El control de versiones se puede llevar a cabo de forma manual, pero con el avance de las tecnologías y el desarrollo de sistemas digitales es recomendable la utilización de herramientas que agilicen y faciliten la gestión. Existen herramientas para el control de versiones que se encargan de la identificación de las distintas versiones de cada elemento, de mantener información acerca de las relaciones que hay entre ellas y de su almacenamiento en un repositorio. Una de las tareas primordiales del responsable del rol de GCC es la de configurar y administrar la herramienta escogida para el control de las versiones por las que van pasando los ECS que son de interés para la empresa o proyecto en cuestión.

Para almacenar las versiones de los ECS en el repositorio existen varias formas. Una posibilidad es almacenar únicamente la última versión de cada uno de los ECS del sistema, pero esto puede plantear numerosos problemas ya que no se tomarían en cuenta las otras versiones y en caso de que se necesite esa información, no estaría disponible.

La estructura general de los sistemas de control de versiones se basa en un repositorio, que es donde se guarda toda la información que se gestiona en el proyecto. El repositorio contiene los datos de todas las versiones, o sea la fecha de creado, modificado, quien realizó cambios de todos los elementos gestionados (la historia de las versiones)

*“Las herramientas de control de versiones o gestión de revisiones ayudan a crear, identificar y almacenar nuevas versiones, al mismo tiempo que permite mantener las versiones anteriores”.* [8] Puede ser

conveniente guardar versiones anteriores como punto de seguridad, al cual poder volver en caso de que un cambio efectuado no nos lleve al resultado deseado. También puede ser conveniente guardar versiones antiguas para poder reutilizar elementos que aparecían en ellas pero que fueron desechados en un momento dado.

Las herramientas de control de versiones utilizan maneras más eficientes para almacenar las versiones, para disminuir la cantidad de espacio requerido. Normalmente estas herramientas no almacenan físicamente todas las versiones, sino solamente una de ellas, que puede ser la última. Sin embargo, nos permiten recuperar cualquier otra versión. Para ello guardan también toda la historia de cambios que han ocurrido sobre el elemento y que lo han hecho pasar de una versión a otra.

Algunas de las herramientas de control de versiones más conocidas son:

- Concurrent Versions System (CVS)
- Visual Source Safe (VSS)
- Subversion (SVN).

### **1.5.1 Herramienta para el control de versiones Concurrent Versions System (CVS)**

CVS es un sistema para el control de versiones que se ha hecho muy popular en el mundo del software libre pues es muy difundido por sus desarrolladores bajo la licencia GPL. Es empleado en proyectos comerciales y prácticamente en todos los proyectos de código abierto.

Es un software Open-Source que permite gestionar los cambios realizados sobre códigos fuentes de cualquier archivo en varias plataformas (C, C++, Java, etc).

#### **Principales Características:**

1. Utiliza una arquitectura cliente-servidor en la cual el servidor es el encargado de guardar las últimas versiones mientras que los clientes se conectan a ese servidor para obtener una copia local del proyecto completo, sobre la cual trabajan y es la que luego suben al repositorio después de haber efectuado los cambios.
2. Una de sus características esenciales es que aunque el servidor utilice un sistema operativo libre, los clientes pueden trabajar en cualquier sistema operativo.

3. Tiene por característica que varios clientes pueden realizar copias al mismo tiempo, lo que puede provocar que existan problemas a la hora de ingresar las modificaciones sobre la misma línea en un archivo, de existir problemas el servidor deniega la segunda operación e informa del problema, que debe ser resuelto manualmente. De no existir problemas entonces los números de los archivos implicados se incrementan automáticamente.
4. Permite a los clientes tener acceso a diferentes versiones de archivo, solicitar la historia de los cambios además de consultar un número de revisión determinado.
5. No requiere que cada versión de los archivos sea guardada sino que se mantiene en un repositorio desde donde son administradas las versiones.

### **Limitaciones:**

- A los archivos gestionados en los repositorios no se le pueden cambiar los nombres. Si se desea renombrar tienen que ser eliminados y luego ser subidos con el nuevo nombre situación esta que provoca más trabajo a los clientes, además de que con este procedimiento de eliminar y añadir cada vez que se desea cambiar nombres, se pierde el historial de las revisiones.
- Para crear copias de seguridad es necesario interrumpir el acceso al repositorio.
- Las transacciones de datos no son atómicas, o sea, cuando se suben un conjunto de datos se van subiendo uno a uno, lo que provoca que no se pueda asegurar la integridad de la información y además provoca que todos los ficheros no se etiqueten con el mismo número de revisión en el repositorio.

### **1.5.2 Herramienta para el control de versiones Visual Source Safe (VSS)**

- VSS es la herramienta para el control de versiones de un proyecto de desarrollo de software de Microsoft. Está estructurada por una arquitectura cliente-servidor y además está compuesta por dos módulos, uno de cliente y otro de administración.
- Es una de las herramientas que mejor integración tiene con el resto de las herramientas de desarrollo como Visual Studio .NET y en alguna medida con Visual Studio 6.
- Permite almacenar varias versiones de un archivo así como llevar un control de los cambios realizados a los mismos.

- Esta herramienta permite administrar cualquier tipo de archivos (texto, gráficos, sonido o video, etc.) y la almacena en un BD la cual permite que los cambios que se le realicen a los archivos se guarden de manera que puedan ser recuperadas en cualquier momento las versiones anteriores.
- Lo commits no son atómicos.

### **Limitaciones:**

Consta con una característica que va en contra de un buen sistema de versiones o sea el trabajo en paralelo. La forma de trabajo del VSS obliga al cliente a bloquear un archivo al hacer "Check Out" antes de poder trabajar con él lo que provoca que mientras el archivo este bloqueado ningún otro usuario puede trabajar con él.

### **1.5.3 Herramienta para el control de versiones Subversion (SVN)**

- Es una herramienta para el control de versiones que ha venido a solucionar los problemas y limitaciones presentados por CVS.
- Existen versiones para Windows y cualquier otro sistema operativo.
- Es un software libre que facilita la administración de un conjunto de ficheros.
- Es muy fácil de usar pues cada usuario puede acceder a él para realizar modificaciones desde un ordenador que forme parte de una red local o tenga conexión a Internet, además una instalación básica solo requiere conocimientos a nivel de usuario del sistema operativo.
- Es bastante seguro. El acceso al servidor se realiza mediante contraseñas. Permite además realizar copias de seguridad de toda la información.
- Consume pocos recursos.
- Los commits son atómicos.
- Permite trabajar directamente sobre el repositorio, sin necesidad de hacer copia local.
- La numeración de versiones es global, o sea, con SVN los números de versión son para todo el repositorio.
- Fuerte integración con Apache puesto que corre bajo licencia del mismo: Esto permite controlar el repositorio definiendo controles de acceso avanzados. Además permite navegación vía Web para consultar el repositorio.
- Facilidad a la hora de eliminar y cambiar nombres a los elementos de configuración de software.

- Copias fáciles sobre ramificaciones: Independientemente del número de ramificaciones creadas mantiene un árbol diferencial de cambios. Las ramificaciones permiten que varios elementos pueden ser modificados en paralelo pues parten del mismo Trunk. Las ramificaciones son utilizadas solamente para corregir errores o problemas con versiones anteriores del producto, o sea, en caso de que ya se cuente con un release (1) del producto y se planea crear un nuevo release (2). .(Ver figura 5)
- Copias diferenciales de archivos binarios: Es capaz de mantener un control diferencial sobre cualquier archivo binario del repositorio reduciendo así el consumo de espacio.

## **Funcionamiento**

- SVN está compuesto por una arquitectura cliente-servidor en la que los servidores contienen una copia de la información a compartir mientras que los usuarios usan el cliente para conectarse al servidor y descargar la información. Cuando un usuario realiza cambios en ficheros, lo sube al servidor para que otros puedan verlo también.
- El trabajo con el servidor puede hacerse mediante comandos. Además de lo básicos (check out, add, commit, update) cuenta con algunos que brindan nuevas funcionalidades con respecto al CVS (copy, move, merge, reolve, medir, propget, proplit, propedit, revert, switch, etc).

Por las facilidades que brinda esta herramienta es la escogida para llevar el control de las versiones de los elementos de configuración del proyecto ONE.

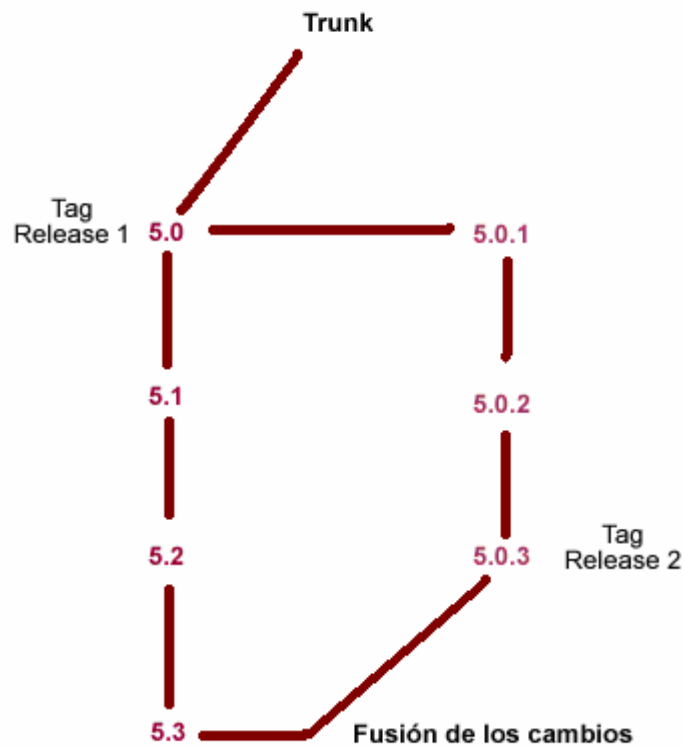


Figura 5. Ejemplifica 2 ramificaciones que parten de un mismo trunk. La parte izquierda representa el producto al cual se le realizarán los posibles cambios. La parte derecha representa el desarrollo del producto hacia una nueva versión. Una vez obtenido el release 2 serán incorporados los cambios hechos al release 1.

### Leyenda

**Trunk:** significa tronco, o sea, el trabajo que se realiza, cuerpo del árbol que representa el repositorio.

**Tag:** etiqueta, características que se presentan.

**Release:** entregables (objetos, ECS, el producto).

**Tag release:** producto listo para su lanzamiento o para otra versión.

## 1.6 Situación de la GCC del proyecto en cuestión

La situación del proyecto será evaluada partiendo de hacer un análisis crítico del mismo a partir de la aplicación de la GCC y los errores que la gestión actual está provocando teniendo en cuenta algunos indicadores de calidad que son de interés monitorear. Los indicadores son detallados a continuación.

- **Elaboración de las plantillas para la creación de artefactos**

La situación del proyecto en el indicador a medir se puede catalogar de mala si se tiene en cuenta que las pocas plantillas existentes, que son las generadas por RUP se encuentran en inglés, situación que provoca que el rendimiento de los integrantes del proyecto no sea el óptimo, pues se le dan varias interpretaciones al contenido de las plantillas. Para contrarrestar esta situación la solución podría ser traducir la totalidad de las mismas, es importante destacar que no se cuenta con la cantidad de plantillas necesarias para documentar la totalidad de los artefactos que se necesitan, lo que provoca que tengan que ser elaboradas una gran cantidad de ellas por parte de los responsables del rol de GCC. Además las plantillas existentes carecen de un formato acorde a las necesidades de organización y presentación que se requieren.

- **Situación del Expediente del proyecto**

La organización del proyecto es regular pues por razones de desconocimiento, no existe un expediente de proyecto, la organización de la documentación del proyecto no se encuentra en la forma más óptima. Falta por definir el expediente, como va a estar estructurado el mismo, como va a estar distribuida la información que contendrá cada carpeta miembro del expediente y precisar algunos otros detalles organizativos respecto al proyecto.

- **Dominio y uso de la herramienta para el control de versiones**

El uso de la herramienta para el control de versiones seleccionada en el proyecto es crítica, los miembros del proyecto, con algunas excepciones, desconocen el trabajo con la parte cliente de la herramienta por lo que el uso de la herramienta es mínimo, situación que provoca que la información no se encuentre actualizada.

Al analizar el dominio de la herramienta por parte de los administradores se puede evaluar de regular si se tiene en cuenta que se cuenta con cierto conocimiento de su uso pero el hecho de que el servidor



de la herramienta se encuentre con sistema operativo Debian de Linux y debido a la poca experiencia con la que se cuenta, genera dificultades a la hora de administrarlo sobre todo a la hora de subir algunos datos.

- **Documentación y control de los cambios a los ECS.**

La documentación de los cambios ha presentado dificultades pues existe cierto desconocimiento por parte de los miembros del proyecto de los procedimientos formales que deben llevarse a cabo para realizar cualquier cambio, además del desconocimiento con respecto a la solicitud de cambio que se hace necesario llenar y enviar para pedir un cambio a un determinado ECS, la cual debe ser aprobada para luego pasar a su ejecución, por lo que en determinadas ocasiones se han pasado por alto ciertos cambios cuya connotación no es mucha pero no han sido documentados y afectan a otros componentes. Los cambios son de suma importancia pero también lo es el conocimiento de los cambios que se realicen por parte del personal de desarrollo, los jefes y clientes (para ello se llevan a cabo informes de estado, documentación de los cambios así como el seguimiento de los mismos y auditorias de comprobación).

- **Satisfacción de los clientes**

La satisfacción de los clientes no ha sido la mejor pues debido al desconocimiento por parte de los miembros del proyecto del uso del SVN, la falta de plantillas y un formato para las mismas se han presentado problemas a la hora de presentar y almacenar la información.

- **Avance del proyecto con respecto a la Configuración del software**

Para medir el avance del proyecto con respecto a la GCS es necesario administrar y controlar los ECS, actualmente en el proyecto ONE no se tienen en cuenta la evolución de los mismos pues debido a la falta de estrategia para la evaluación, la falta de orientación para el control y seguimiento de los ECS y el poco uso de indicadores o métricas necesarias para medir y contabilizar el desarrollo del proyecto, se han presentado problemas de estancamiento y de tiempo, o sea, se utiliza más tiempo del establecido a la hora de desarrollar las iteraciones.

- **Cantidad de elementos de configuración en desarrollo.**

La cantidad de ECS en desarrollo es desconocida debido a que no se realiza ningún proceso o tarea para contabilizar dichos ECS.

- **Cantidad de elementos de configuración en revisión.**

La cantidad de ECS en revisión es desconocida debido a que no se realiza ningún proceso de revisión o análisis de los mismos.

- **Cantidad de elementos de configuración en mejoras de revisión.**

La cantidad de ECS en mejoras de revisión es incierta debido al desconocimiento sobre los ECS que deben haber pasado el proceso de revisión y que se le hayan aplicado mejoras.

- **Cantidad de elementos de configuración terminados.**

Existe desconocimiento sobre la cantidad de ECS terminados y listos debido a la poca comunicación entre los miembros del equipo de trabajo.

- **Cantidad de no conformidades o solicitudes de cambios realizadas por los clientes**

No se posee un mecanismo para controlar y documentar la cantidad de no conformidades o solicitudes de cambios realizadas por los clientes por lo que se puede incurrir en problemas como: Software que no cubre las expectativas o que no cumple todas las funcionalidades establecidas, incomodidad del cliente al no ser atendida una no conformidad, etc.

- **Cantidad de solicitudes de cambios en espera de ser revisadas**

Hasta el momento no se ha contabilizado los cambios realizados por lo que no se cuenta con solicitudes de cambio en espera de ser revisadas.

- **Cantidad de solicitudes de cambios en desarrollo**

No existe conocimiento sobre la cantidad de solicitudes de cambio en desarrollo debido a que se desconoce la cantidad de cambios solicitados.

- **Cantidad de solicitudes de cambios en revisión**

La cantidad de solicitudes de cambio en revisión es nula

- **Cantidad de solicitudes de cambios cerradas**

Debido a la falta de un procedimiento bien definido para la solicitud de cambios, no se lleva un control de la cantidad de solicitudes de cambios cerradas.

- **Evaluación de los Procedimientos llevados a cabo en el proyecto**

En estos momentos no se cuenta con procesos de evaluación de los procedimientos, existen pocos procedimientos documentados debido a la falta de experiencia y en muchas ocasiones por ser un poco incómodo documentar todas las actividades que se realizan para concretar un hecho.

### 2.1 Organización del expediente del proyecto

La organización de la documentación del proyecto está establecida en 5 carpetas fundamentales (Ver figura 6):

- Carpeta de Desarrollo técnico
- Carpeta de Gestión de Configuración
- Carpeta de Gestión de Proyecto
- Carpeta de Sistema de Calidad
- Carpeta de Seguridad Informática

En esta carpeta se recoge toda la documentación necesaria e imprescindible para tener bien controlada la información que es manejada por todo el equipo de desarrollo. A continuación se detallan cada una de estas carpetas y la organización de la información que regulan.

**Carpeta de Desarrollo técnico:** Contiene una carpeta de arquitectura donde se almacenan los componentes reutilizables y aprobados para su utilización, los documentos que soportan la descripción detallada y diseño de la arquitectura del sistema, métodos, técnicas y estructura de la misma.

Contiene una carpeta para la base de datos (SIGE BD) donde se almacena la información de la misma como: tablas, modelos de datos y de diseño, diagramas de clases, etc.

Contiene además una carpeta para cada uno de los módulos existentes en el proyecto: Módulo Entrada de Datos (SIGE MED), Módulo Generador de Modelos (SIGE MGM), Módulo Generador de Reportes (SIGE MGR), Módulo de Registro y Clasificadores (SIGE MRC), Módulo de Encuestas Periódicas (SIGE MEP), las cuales a su vez están compuestas por carpetas que delimitan la documentación entregable en cada flujo de trabajo del proyecto, estas son:

1. Modelo de Negocio: Almacena toda la información referente al negocio y que sirven de entendimiento entre los clientes y los desarrolladores: Visión del negocio, Estado del arte del producto a desarrollar, Ámbito del proyecto y del producto, Propuesta del prototipo inicial, etc.

2. **Requerimientos:** Se detalla como transcurre el levantamiento de requisitos necesarios para la construcción del software, se genera información como: Plan de gestión de requisitos, Especificaciones de casos de usos del sistema, Especificación de requisitos de Software, etc.
3. **Análisis & Diseño:** Se hace un análisis del sistema que suavice la entrada al diseño, asegurando que cumpla las restricciones y funcionalidades impuestas por los requerimientos. Para ello se genera información como: Modelo de análisis, de diseño y de despliegue, Descripción de la arquitectura con diferentes vistas, Diseño de investigación, Descripción de los casos de uso de análisis, etc.
4. **Implementación:** Esta compuesta por la documentación que incluye los ejecutables, prototipos funcionales, se implementa la estructura definida en el diseño. Incluye: Modelo de implementación, Plan de integración de construcciones, Descripción de la arquitectura, etc.
5. **Prueba:** Es donde se detectan los errores y defectos a lo largo del ciclo de vida del Software. Incluye: Plan de pruebas, Procedimientos de Pruebas, Informe de investigación, etc.
6. **Despliegue:** Es donde se generan los release y se desarrollan actividades con fines de entregar el producto a los clientes y de darle mantenimiento. Incluye: Cuentas de materiales, Plan de despliegue y Notas de los release

**Carpeta Gestión de Configuración:** Contiene lo referente al plan de configuración, el control de versiones, los Elementos de Configuración de Software. Contiene una carpeta de plantillas técnicas conformada por 11 carpetas que almacena y administra todos los documentos que regulan el proceso de desarrollo de software, o sea, las plantillas de artefactos que rigen la información que debe ser documentada por parte del equipo de desarrollo del proyecto ONE. (Ver anexo 1). Incluye los 6 flujos de trabajos ingenieriles, los de calidad y soporte. Contiene una carpeta para los procedimientos que rigen las actividades más importantes del proyecto como pueden ser: procedimientos de control de versiones, de control de cambios, de captura de requisitos, de planificación, procedimientos de selección de los ECS etc.) Y otra carpeta de patrones que podemos utilizar para resolver problemas que se presenten, describir la arquitectura que implementamos, o sea que le dan soporte a la arquitectura, la programación, la gestión de la configuración, etc.

**Carpeta Gestión de proyecto:** Es donde se almacenan todas las actas de reuniones, ambiente de desarrollo del proyecto, acuerdos, decisiones y toda la documentación resultante de la comunicación interna entre los miembros del proyecto y la comunicación externa entre los desarrolladores y los clientes. Se almacena todo lo referente a la aplicación de las métricas, planificaciones del proyecto, los datos y gráficas que muestran la evolución del proyecto. Contiene una carpeta para documentar las investigaciones paralelas sobre temas de interés. Se guardan además todos los datos referentes a los recursos humanos, capacitación, roles y los recursos materiales y el control económico del proyecto. Esta carpeta ayuda a organizar la información referente al personal y a la gestión.

**Carpeta de Seguridad Informática:** almacena todo el plan de seguridad del proyecto, las políticas de protección contra programas malignos, la política de confidencialidad de la información, principios de programación segura que se establezcan, plan de pruebas contra ataques y otros elementos que permiten trabajar en un entorno seguro.

**Carpeta de Sistema de calidad:** Esta carpeta recoge toda la información relacionada con el plan de calidad, las listas de chequeos para cada rol, las listas de comprobación para cada artefacto generado durante el desarrollo del producto. Contiene además una carpeta donde se recoge la información relacionada con las auditorías e inspecciones de calidad, estándares de codificación, criterios de medida para evaluaciones, procedimientos que se deben seguir. En general el sistema de calidad contribuye a la aplicación de los principios de calidad total en el proyecto.

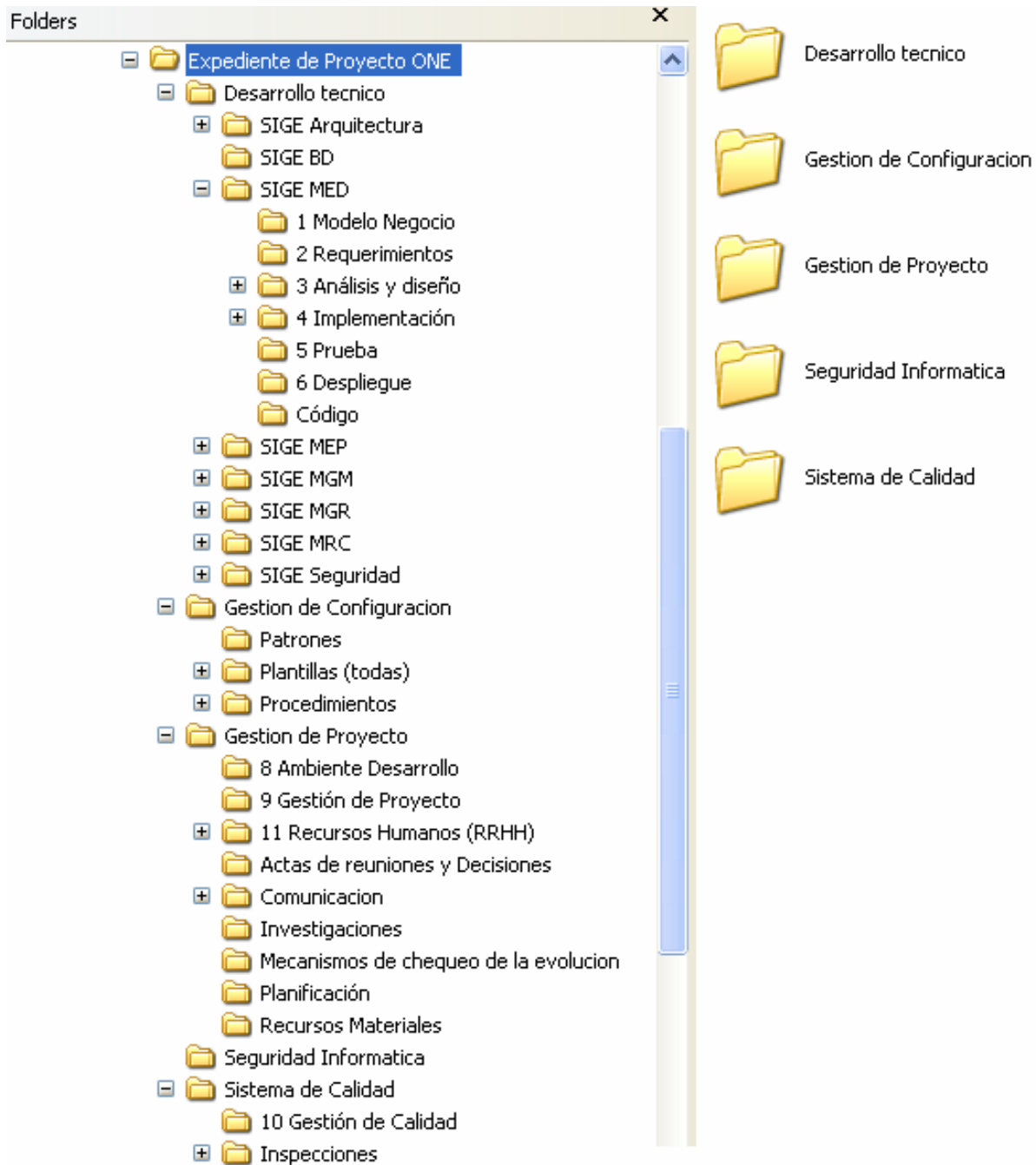


Figura 6. Organización del expediente del proyecto

### **2.1.1 Organización del equipo de trabajo por módulos del proyecto ONE**

Por estrategias de organización del proyecto, se determinó una estructura para el equipo de desarrollo con vista a asignarle, al responsable de cada rol, actividades y tareas necesarias que deben realizar para tener un mejor control de la información y el ambiente en que se desarrolla. La estructura es la siguiente:

1. Jefe de Módulo
2. Analista
3. Arquitecto
4. Técnico de pruebas
5. Diseñador
6. Programador

El Jefe de Módulo y el analista además de sus actividades ordinarias, son los encargados de controlar y subir la información del producto al SVN, de esta forma se mejora la utilización de la herramienta para el control de versiones y se evitan algunos riesgos, sobre todo de pérdida de la información. El resto del equipo trabaja en diferentes actividades y tareas que rigen su rol.

El jefe de módulo puede desempeñar cualquier rol, en el caso del proyecto ONE, generalmente juegan el papel de analista y diseñador.

#### **Actividades que realizan los desarrolladores**

Arquitecto: Esta encargado de establecer la arquitectura que va a regir el modulo, priorizar los casos de uso mas importantes, diseñar todo lo que se va a programar

Programador: Es el encargado de seguir paso a paso el modelo de diseño, llevándolo al lenguaje de programación, o sea, llevar a líneas de código todo el análisis y diseño del producto a realizar.

Diseñador: Es el encargado de desarrollar la estructura del producto, o sea, tiene la responsabilidad de diseñar como va a quedar el software, de detallar todas las funcionalidades y aspecto que debe tener el producto en desarrollo.



Técnico de pruebas: Es el responsable de la realización y aplicación de todas las pruebas necesarias con el objetivo de obtener un producto que cumpla con todas las funcionalidades establecidas por el cliente y con calidad. Es el encargado de diseñar los Casos de uso de prueba, el modelo de pruebas, el plan de pruebas y aplicarlos.

Analista: Es el que se encarga del desarrollo de los artefactos del producto (ECS) y su documentación, o sea, se establecen los ECS y ellos son los responsables de realizarlos. Se encargan de gestionar el negocio (modelar el negocio con los clientes), el levantamiento de requisitos, el prototipo de interfaz de usuario, entre otras actividades que se llevan a cabo mediante reuniones y entrevistas con el cliente. Su principal objetivo es llevar a un lenguaje común los criterios de los desarrolladores y de los clientes para luego pasar a la realización del software.

## **2.2 Modelo para la Gestión de Configuración de Software y Cambios**

Para la aplicación del rol de GCC al proyecto ONE se hace necesario elaborar una estrategia que permita llevar a cabo el proceso de GCC en el mismo, para la obtención de resultados satisfactorios.

La estrategia consiste en redefinir el modelo estándar establecido por IEEE para la GCS, de forma tal que se adecue a las necesidades del proyecto, haciéndolo más completo. El modelo que se propone constará con 4 sub-capítulos cuya generalización se muestra a continuación:

1. Definición y establecimiento de la Configuración: Su objetivo principal es el de definir como va a estar estructurado el producto, sus componentes y de que tipo de estos son, y determinar de alguna manera la forma de distinguirlos y hacerlos únicos e identificables.
2. Gestión y control de Cambios en la Configuración: Consiste en controlar las diferentes versiones por las que va pasando un producto así como los cambios que se producen en él a lo largo de su ciclo de vida.
3. Generación de Informes de Estado: Consiste en reportar el estado en que se encuentran los ECS y los cambios por los que ha pasado, generando informes de evolución del producto
4. Inspecciones en la Configuración: Su objetivo es el de valorar que el producto cumpla con las solicitudes de los clientes y que su funcionamiento sea completo como completo deben estar también sus componentes.

### **2.2.1 Definición y establecimiento de la Configuración**

En este sub-capítulo se pretende analizar más detalladamente una tarea de gran importancia en la GCS. Esta tarea consiste en identificar y asignar nombres significativos y consistentes a todos y cada uno de los elementos que forman parte del producto software, es decir, a cada uno de los ECS, en cada fase de su desarrollo.

Los ECS son aquellos componentes que forman parte de la Gestión de Configuración del Software, son su soporte, así como son la base del software en construcción, o sea son todos los elementos fundamentales que brindan una visión del producto. Hay que tener en cuenta el número de ECS que se va a definir: si son demasiados, la documentación, especificación y descripción de las funciones se hace inmanejable; si son pocos, el software carece de información suficiente que lo sustente; se deben escoger los necesarios y más importantes.

Los ECS también pueden ser denominados objetos o componentes, se considera que cada ECS puede ser un objeto manejable, configurable, sujeto a cualquier tipo de cambios o versiones que puedan ocurrir.

#### **¿Cómo se identifican los ECS?**

Para dar respuesta a la interrogante, se llevó a cabo un proceso de identificación sustentado en procedimientos.

El procedimiento de selección de los ECS consta de los siguientes pasos:

- Establecer responsables que van a ejecutar la actividad.
- Establecer estados de las versiones de los ECS. (Establecer tantos estados como sean necesarios)
- Definir las actividades que rigen este documento como:
  - 1) Identificar los ECS y documentarlos como se sugiere más adelante, adicionar al artefacto ECS con la siguiente información:
    - A. Identificador del ECS
    - B. Descripción del ECS
    - C. Autor(es) del ECS
    - D. Listado de elementos dependientes del ECS
    - E. Listado de los módulos que lo usan
    - F. Tipo de ECS
    - G. Localizaciones

Los ECS se podrán definir basándose en varios criterios:

- *“Utilización múltiple: Número de elementos de su mismo nivel o niveles superiores que lo utilizan.*
  - *Criticidad: Gravedad del impacto de un fallo en dicho componente.*
  - *Número de personas implicadas en su mantenimiento.*
  - *Complejidad de su interfaz: Las interfaces de un EC con otros deberían ser simples. Hay que minimizar el acoplamiento entre ECS.*
  - *Singularidad del componente con respecto al resto.*
  - *Reutilización: Si el componente se va a diseñar especialmente para ser reutilizado.*
  - *Si se trata de elementos reutilizados.*
  - *Tipo de tecnología: Si el componente incorpora nuevas tecnologías no utilizadas en otros componentes”.[8]*
- 
- Se determinan y aprueban los ECS y automáticamente pasan a ser controlados por la Gestión de configuración de Software.
  - Se determinan y documentan las relaciones que pueden existir entre los ECS.
  - Se almacena en el repositorio la versión de los ECS definidos.
  - Se hace público el artefacto de ECS incluyendo las relaciones establecidas y se pasa a la distribución del mismo.

Cuando se van a seleccionar los ECS es necesario analizarlos y organizarlos por categoría para mitigar el impacto de los cambios en los mismos.

#### **2.2.1.1 ECS seleccionados**

- Plan de Gestión de Requerimientos.
- Especificaciones de requisitos de Software.
- Especificaciones de Casos de Uso del Sistema.
- Documento Visión del Sistema.
- Prototipo Interfaz de Usuario.
- Modelo de Análisis.

- Realización de Casos de Uso del análisis.
- Modelo de Diseño.
- Realización de Casos de Uso de Diseño.
- Modelo de Datos.
- Diseño de la Base de datos.
- Descripción de la Arquitectura (vista modelo de diseño).
- Modelo de Despliegue.
- Descripción de la Arquitectura (vista modelo de despliegue).
- Modelo de Implementación.
- Descripción de la Arquitectura (vista modelo de implementación).
- Plan de Integración de construcciones.
- Modelo de Pruebas.
- Casos de Uso de Prueba.
- Resumen de evaluación de Prueba.
- Plan de Prueba.
- Plan de Despliegue.
- Plan de Gestión de la Configuración.
- Elementos de Configuración de Software.
- Plan de Gestión de Cambios.
- Control de versiones.
- Preceptos de la Programación.
- Preceptos del modelado de Casos de Uso.
- Preceptos de Diseño.
- Preceptos de Prueba.
- Plan de Iteración.
- Plan de Aceptación del Producto.
- Plan de Administración de Riesgos.
- Plan de desarrollo del Software.
- Plan de Gestión de la Calidad.
- El código fuente del programa.

- Código objeto y ejecutable.
- Manual de usuario.

### **2.2.1.2 Relaciones entre los ECS**

Entre los ECS se pueden establecer varias relaciones: de composición, de funcionalidad, de dependencia, entre otras que han sido definidas con anterioridad en este trabajo. Las relaciones existentes entre los ECS serán descritas y representadas claramente mediante un grafo. El propósito e importancia de describir y representar mediante un grafo estas relaciones es el de dar visibilidad e informar al equipo de trabajo sobre las mismas (se refiere a las relaciones) y tratar de minorizar el impacto de los cambios a los ECS, o sea cuando exista la necesidad de un cambio en algún ECS, los desarrolladores puedan ser capaces de identificar haciendo uso de este grafo, los elementos relacionados a dicho ECS y que por consiguiente se afectan con la posibilidad del cambio.

Se hace necesario establecer todas las relaciones posibles entre los ECS sin caer en una dependencia total, para ello es importante regirse por el principio de bajo acoplamiento, la intención es representar la menor cantidad de dependencias posibles sin omitir alguna.

Para comenzar con la organización, los ECS seleccionados se agruparon en 4 carpetas, divididas por flujos de trabajo, esta estructura brinda la posibilidad de un mejor manejo de los datos cuando se cuenta con una cantidad considerable de ECS, como es el caso.

La distribución por carpetas se describe a continuación:

#### **1. Carpeta de los Flujos ingenieriles:**

En esta carpeta se guardan las relaciones existentes entre los ECS pertenecientes a los distintos flujos ingenieriles definidos por RUP, incluyendo al flujo de soporte Ambiente de Desarrollo. Estos serían:

- Los ECS definidos en el levantamiento de requisitos.
- Los ECS definidos en el flujo de Análisis & Diseño.
- Los ECS definidos en el flujo de Implementación.
- Los ECS definidos e el flujo de Prueba.
- Los ECS definidos en el Despliegue.
- Los ECS definidos en el flujo Ambiente de desarrollo

#### **2. Carpeta de Gestión de Configuración de Software**

En esta carpeta se va a almacenar las relaciones referentes a los ECS definidos en la Gestión de Configuración del Software.

### 3. La carpeta de Gestión de Proyecto

Contiene las relaciones establecidas entre los ECS que intervienen en el proceso de desarrollo del Proyecto.

### 4. La carpeta de Calidad

Contiene el Plan de Gestión de la Calidad que se aplica a todos los componentes, ECS y objetos identificados y desarrollados en el proyecto.

Estas carpetas poseen relaciones entre ellas, relaciones de dependencias , etc., producto a los ECS almacenados en su interior y teniendo en cuenta que la calidad, la gestión de la configuración y la gestión de proyectos por ser flujos de trabajos de soporte se encuentran presentes en todo el ciclo de vida del proyecto, de ahí las relaciones en ambos sentidos. (Ver figura 7).

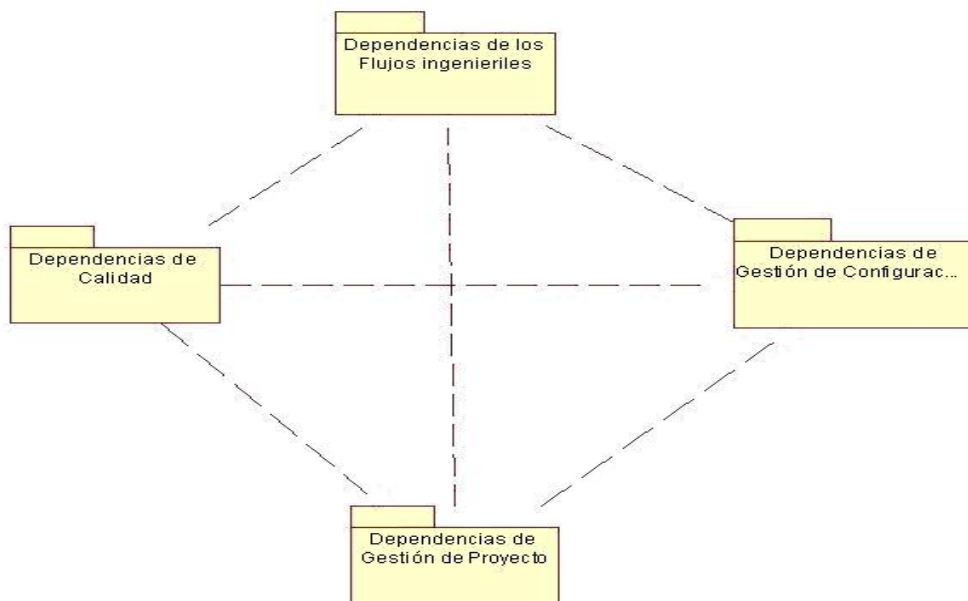


Figura 7. Distribución de los ECS y sus relaciones en carpetas funcionales.

Como se ha mencionado con anterioridad, entre los ECS seleccionados se establecieron relaciones de dependencia (Ver figura 8). La dependencia se describe de varias formas, una primera definición de

dependencia puede darse cuando un ECS contiene información de otro componente o puede servir de base para el surgimiento de otro ECS; por otro lado un ECS puede estar nombrado en un elemento y explicado en otro, varios ECS pueden estar implicados en la creación de un objeto, etc.

Las dependencias que se identificaron teniendo en cuenta los flujos de trabajos ingenieriles son las siguientes:

El Plan de Gestión de Requerimientos es el encargado de administrar y gestionar los requisitos de software capturados en el levantamiento, así como de controlar los responsables del flujo.

Del Plan de Gestión de Requerimientos se originan:

- Las Especificaciones de requisitos de Software
- Las Especificaciones de Casos de Uso del Sistema

La creación de estos documentos es guiada por los Preceptos para el modelado de Casos de Uso por lo que entre los primeros y el segundo se establece una relación de dependencia.

En base a estos 2 últimos documentos se puede crear el Prototipo de Interfaz de Usuario con su respectivo Manual de usuario (esto ocurre para cada módulo con que cuenta el proyecto) .El prototipo a su vez posibilita hacer una descripción global del sistema mediante el Documento Visión del Sistema.

Las Especificaciones de requisitos de Software y las Especificaciones de Casos de Uso del Sistema dan lugar al flujo de trabajo de análisis. Aunque no es obligatorio, se estableció como ECS la elaboración del Modelo de Análisis, en el cual se describen los casos de uso del análisis que son especificados en el documento de Realización de Casos de Uso del análisis, estos ECS suavizan la entrada al Diseño.

Para la elaboración del Modelo de Diseño y la realización de los Casos de Uso del mismo se toma como entrada el Modelo de Análisis y la realización de Casos de Uso del análisis, luego es elaborado el Modelo de datos con las clases persistentes definidas en la Realización de Casos de Uso del diseño, de este Modelo de datos depende el Diseño de la Base de Datos, este último junto con el Modelo de Diseño están

guiados por los Preceptos de diseño y detallados en la Descripción de la Arquitectura (Vista modelo de diseño).

Para desarrollar el Modelo de Implementación utilizamos como base el Modelo de Diseño y la Realización de Casos de Uso de diseño.

El Modelo de Implementación depende del Modelo de Despliegue pues contiene un diagrama de despliegue para la distribución física de los recursos, de este último surge la descripción de la Arquitectura (Vista modelo de despliegue) la cual sirve de referencia para la conformación del Plan de Despliegue.

El Modelo de Implementación esta relacionado además con el Plan de integración de construcciones que se encarga de integrar los componentes implementados y es descrito mediante el documento de Descripción de la Arquitectura (Vista modelo de implementación). Las implementaciones relacionadas con los Preceptos de la Programación dan lugar al código fuente del programa, al código objeto y ejecutables que forman parte del software, estos dependen de las pruebas que se les apliquen para ser aceptados, para la detección de errores, para comprobar su funcionalidad y correspondencia con los requisitos, etc.

Las pruebas son detalladas en el Plan de Prueba, en el Modelo de Prueba que surge del plan y en los Casos de Uso de Prueba que se aplican. Las pruebas no solo se aplican al código sino en todos los ECS, como la calidad, la GCC y la Gestión de Proyecto.



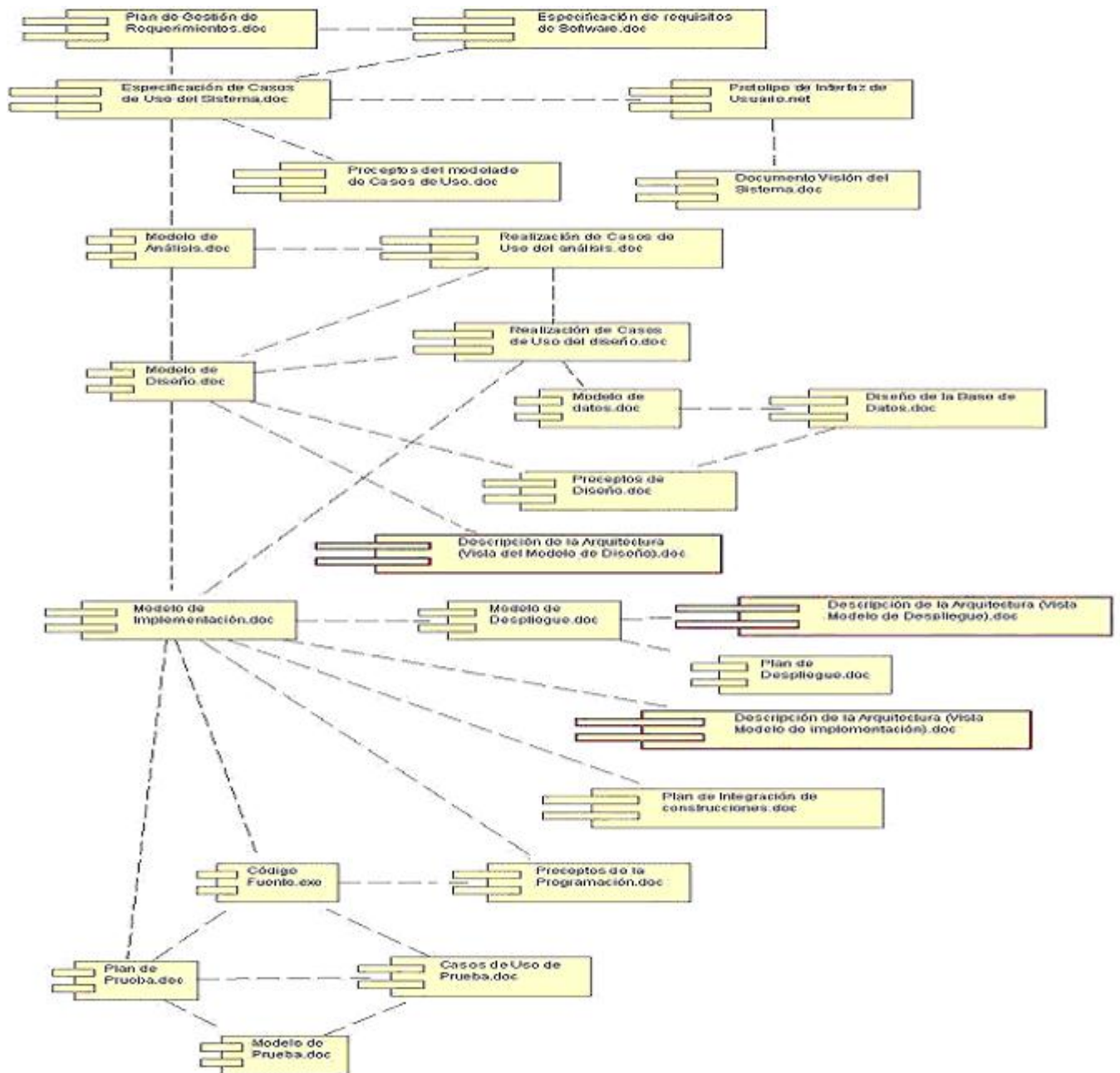


Figura 8. Descripción de las dependencias entre los ECS del paquete de Flujos ingenieriles incluyendo el Flujo de soporte Ambiente de Desarrollo.

Para continuar se tendrá en cuenta los ECS que se encuentran en el paquete de Gestión de Proyecto, estos elementos regulan y ayudan al control del trabajo que se realiza con los ECS y demás documentos, actividades y tareas que se lleven a cabo por el equipo de desarrollo. (Ver figura 9)

Lo primero que se hace es establecer el Plan de Iteraciones que orienta el trabajo por iteraciones y por consiguiente indica la documentación que hay que llenar, los ECS que surgen, a los que hay que darle refinamiento y seguimiento, este plan esta relacionado con el Plan de desarrollo del Software, que describe como se debe construir el producto, este a su vez esta relacionado con el Plan de Aceptación del Producto que establece las pautas para que el producto sea aceptado y el Plan de Administración de Riesgos que se establece para controlar los riesgos relacionados con el Software.

Estos planes dependen unos de otros, sin un plan de desarrollo de software no hay producto que pueda ser aceptado, sin un plan de iteraciones el trabajo es incoherente por lo tanto el software puede estancarse y sin un buen control de los riesgos no se puede asegurar la integridad del producto.

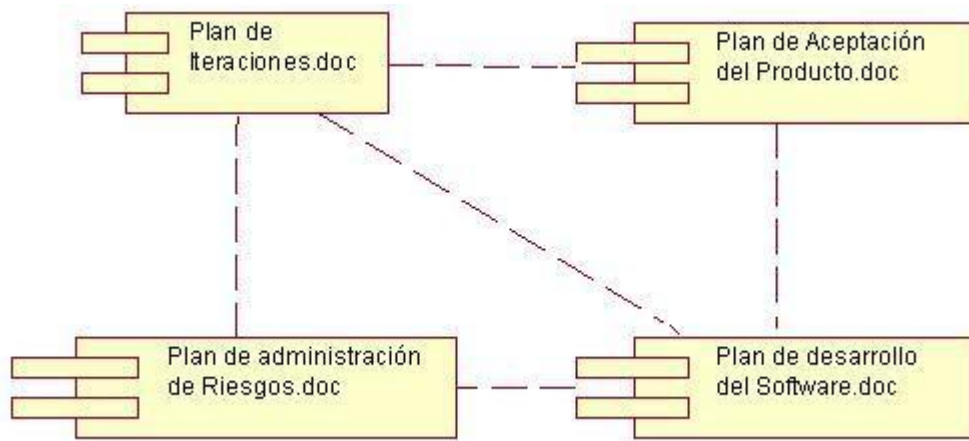


Figura 9. Descripción de las relaciones de dependencias entre los ECS seleccionados en Gestión de Proyecto.

En la carpeta de Gestión de Configuración de Software las dependencias entre los ECS determinados se presentan de la siguiente forma: (Ver figura 10):

Se establece el Plan de Gestión de Configuración del Software, este menciona los ECS y con ellos como entrada se establece la línea base de la configuración del proyecto. El Plan de Gestión de Configuración del Software recoge además información de la herramienta de control de versiones y permite gestionar los cambios realizados o por realizar a los ECS, entre otras actividades. Este plan depende para su gestión del documento de Elementos de Configuración de Software, estos elementos pueden sufrir cambios a lo largo de todo el ciclo de desarrollo del proyecto por lo que se hace necesario establecer un Plan para la Gestión de los Cambios.

Un ECS no solo puede cambiar sino que también puede contar con varias versiones , para controlar las distintas versiones se hace uso de una herramienta para el control de versiones, en el caso del proyecto ONE, se utiliza Subversión, la información de las versiones y de los cambios van a dar al Plan de GCS.

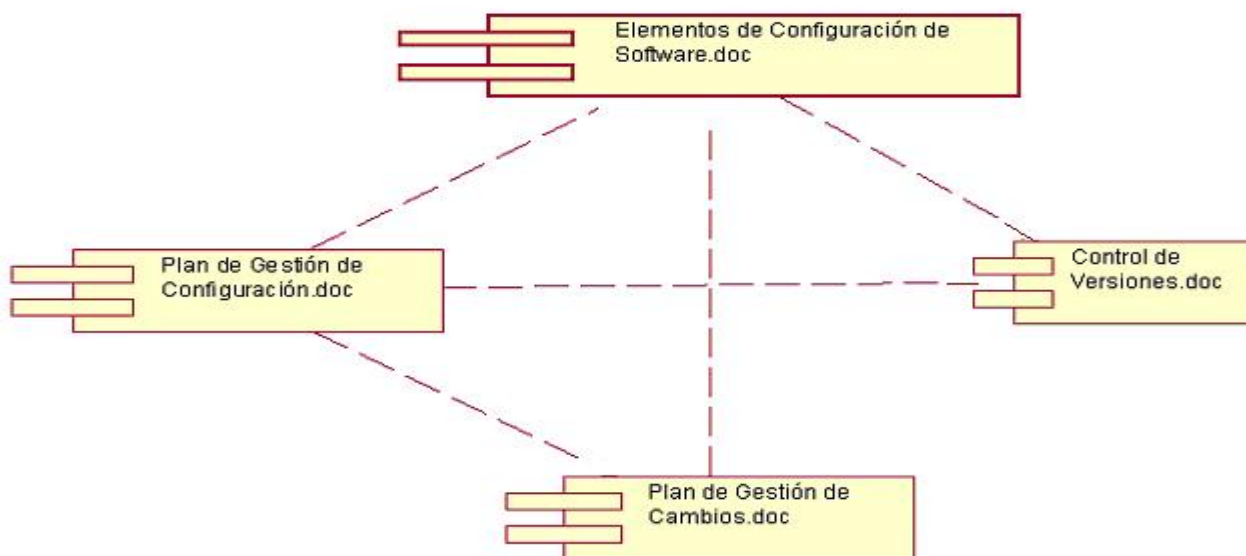


Figura 10. Descripción de las relaciones que se establecen entre los distintos ECS determinados en la Gestión de Configuración del Software.

La carpeta de Calidad contiene el Plan de Gestión de la Calidad que se va a estar relacionado con todos los ECS existentes incluyendo además objetos, documentos, componentes surgidos a lo largo del desarrollo del proyecto ya que se trabaja en base de obtener un producto con el mas alto nivel de Calidad reduciendo los costos. Todos los ECS existentes en esta carpeta poseen una relación de dependencia a este ECS aunque si ocurre un cambio el Plan de Gestión de la Calidad no sufre afectaciones, solo pudiera darse el caso si sucede al revés. (Ver figura 11)

El Plan de Gestión de Calidad se aplica a cada flujo, a cada proceso, a cada construcción, elemento, diseño, descripción, etc.

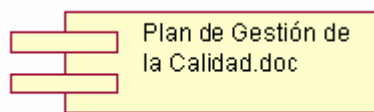


Figura 11. ECS seleccionado para el control de Calidad en todo el proyecto.

### **2.2.1.3 Análisis a profundidad de las relaciones (dependencias y asociaciones)**

El objetivo de hacer un análisis más profundo de las relaciones que se establecen entre los ECS específicamente las de dependencias y asociación es la de tener un poco más controlado el impacto de los cambios que se producen entre los ECS, si se tiene en cuenta que se ha establecido una estructura jerárquica donde se muestran los ECS principales, que dan entrada a los ECS de los flujos de trabajos posteriores. De dichos ECS dependen otros que llamaremos de apoyo que son tan importantes como los principales con la diferencia que los segundos surgen a partir de los primeros.

La distribución de los ECS se establece en forma de árbol de manera que un cambio en la raíz provocaría afectaciones en las ramas correspondientes.

De acuerdo a las relaciones establecidas anteriormente las especificaciones se harán siguiendo esa organización pero aún más clara, o sea, se define un árbol para los flujos de trabajos ingenieriles incluyendo el flujo de soporte Ambiente de desarrollo (Ver figura 12) y otro para el resto de los flujos de soporte (Gestión de Configuración del Software, Gestión de Proyectos, Gestión de Calidad). (Ver figura 13)

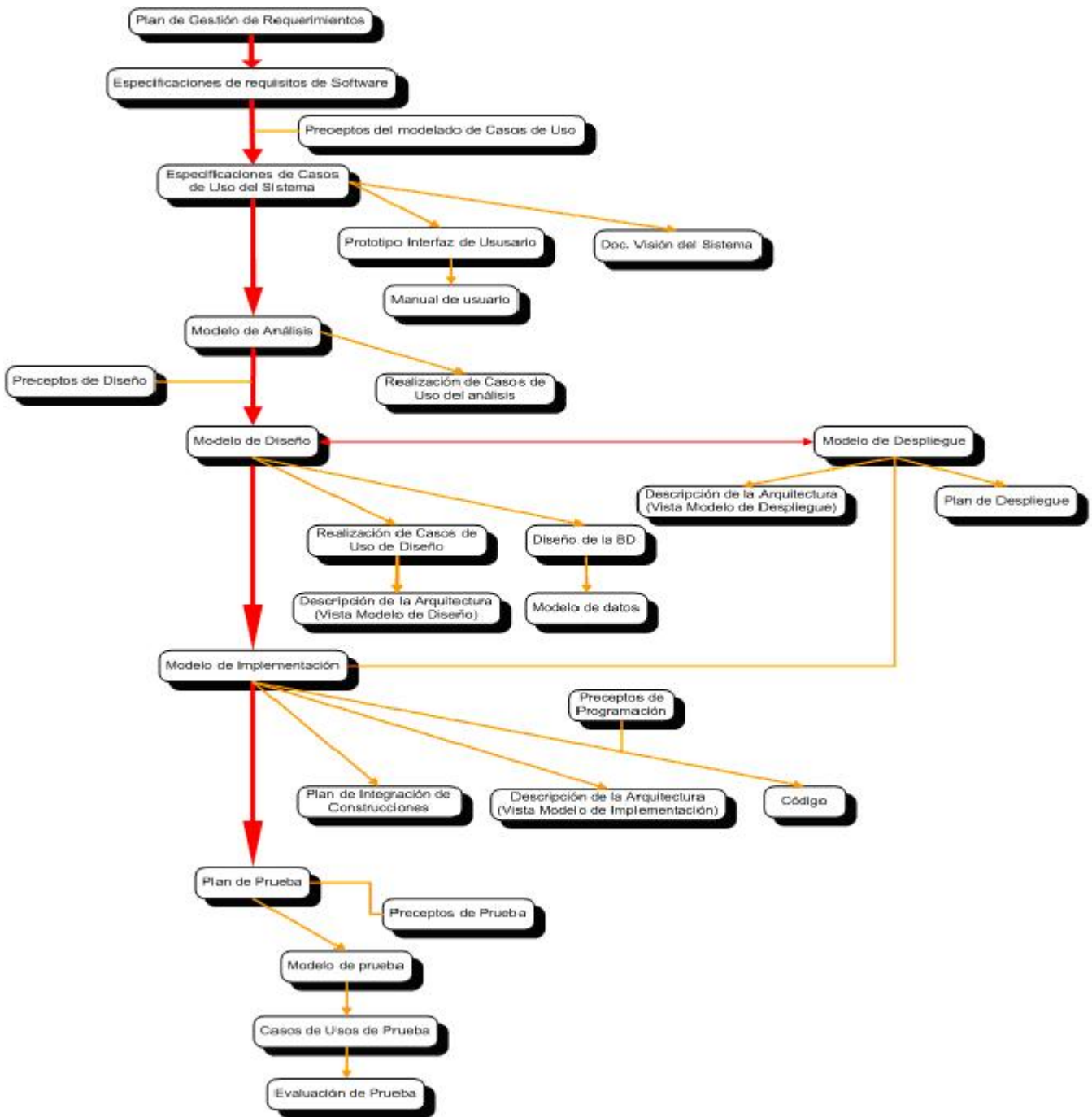


Figura 12. Dependencias y asociaciones entre los ECS de los flujos de trabajos ingenieriles incluyendo al de soporte Ambiente de desarrollo, para un mejor control de las afectaciones que algunos ECS provocan en otros

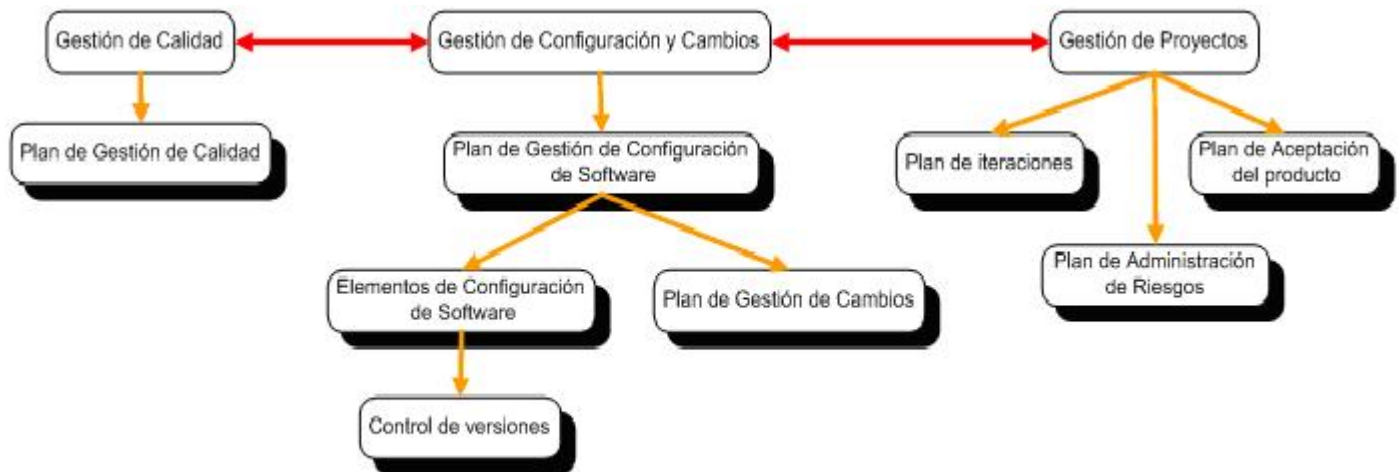


Figura 13. Dependencias y asociaciones entre los ECS de los flujos de trabajo de soporte (Gestión de Configuración del Software, Gestión de Proyectos, Gestión de Calidad).

### Leyenda

↔ Representa los procesos concurrentes o sea que son llevados a cabo al mismo tiempo.

○ Representa los flujos de trabajo de soporte: Gestión de Configuración del Software, Gestión de Proyectos, Gestión de Calidad.

◐ Representa los distintos ECS.

**Asociación:** Es la relación que se establece cuando un ECS influye en la creación de otro.

— Representa las asociaciones entre ECS.

**Dependencia:** Es la relación que se establece entre los ECS cuando dependen unos de otros.

→ Representa las dependencias entre los ECS principales de los flujos de trabajo involucrados.

→ Representa las dependencias entre los ECS generados a partir de los principales.

#### 2.2.1.4 Establecimiento del Esquema de Identificación

El esquema que se establece para la identificación de los ECS es el siguiente:

[ONE-EC] \_ [Módulo] \_ [Nombre] \_ [# Versión]

Donde:

- ONE-EC: Especifica el proyecto al que pertenece y la abreviatura de Elemento de Configuración. Acompañará a cada ECS.
- Módulo: Nombre del módulo al cual pertenece el ECS.
- Nombre: Nombre del ECS y por consiguiente su identificación.
- # Versión: Numero de la versión en la que se encuentra el ECS.

La identificación establecida para los módulos es la siguiente:

1. GM: Generador de modelos
2. ED: Entrada de datos
3. RC: Registros y clasificadores
4. BD: Base de datos
5. GR: Generador de reportes
6. EP: Encuestas Periódicas

Los posibles identificadores de los ECS pueden ser:

- BD: Base de datos
- Doc: Textos
- Zip: Ficheros compactados
- F: Fuentes
- R: Ficheros del Rational
- Exe: Ejecutable
- Obj: Objetos
- N: Ficheros .Net
- Dll: Librerías

Los números de las versiones comienzan por la 1.0 e incrementará según se desarrolle el producto.

### **2.2.1.5 Procedimiento para establecer las líneas base**

Para establecer la línea base del proyecto, es necesario tener en cuenta los siguientes pasos:

- Establecer responsables que van a ejecutar la actividad.
- Establecer fecha de elaboración.
- Establecer estrategia a seguir para la implantación de la línea base.
- Verificar la terminación de la iteración.
- Actividades a desarrollar

1. Establecer estado de las versiones de los ECS.
  2. Identificar ECS involucrados en el proceso.
  3. Aprobar ECS para la línea base.
  4. Aclarar iteración a la que pertenece la línea base.
  5. Verificar y establecer fase de realización.
  6. Elaborar la línea base.
- Termina el procedimiento con la distribución del informe de línea base a todos los miembros del equipo.

#### **2.2.1.6 Definición y establecimiento de las líneas base**

Una línea base está formada por ECS que han sido revisados y aprobados por los miembros de la dirección del proyecto y que van a servir como base para el posterior desarrollo del Software. Una vez establecida la línea base solo podrá ser cambiada mediante un proceso formal de cambios.

Se puede definir además una línea base como *"...un punto de referencia en el proceso de desarrollo del software que queda marcado por la aprobación de uno o varios Elementos de Configuración del Software, mediante una revisión técnica formal."*[8]

Se establece en la primera iteración como línea base, la versión 1.0 del producto en la fase de elaboración/construcción que incluye los siguientes ECS: Documento Visión, Documento de especificaciones de casos de usos, Documento de diseño del sistema, Documento Arquitectura, ejecutable del producto, modelo de datos y modelo físico de la base de datos.

Esta línea base es una versión piloto que se hizo necesaria para la primera presentación a los clientes, mostrando la estructura concebida por el equipo de desarrollo en el levantamiento de requisitos realizado en agosto del 2006, fue una representación rápida y sencilla basada principalmente en la estructura del Microset (software que se utiliza como base y guía).

Luego de realizarse un refinamiento de los requisitos, la concepción del software cambia de forma radical. Se estableció una nueva estructura de desarrollo para el proyecto basada en módulos, se detallaron las funcionalidades que deben tener dichos módulos así como la forma de integrarlos en un solo sistema que



abarque todos los productos construidos por los mismos, se incluyeron nuevos módulos, otros que se crearon pasaron a ser independientes.

Para la iteración 2, se establece como línea base la versión 1.1 del producto SIGE en la fase de elaboración/construcción que incluye los siguientes ECS: Documento Visión del Sistema, Documento de especificaciones de casos de usos, Modelo de Análisis, Realización de Casos de Uso del análisis, Modelo de Diseño, Realización de Casos de Uso del diseño, Modelo de Datos, Descripción de la Arquitectura (Vista modelo de diseño), Modelo de Despliegue, Descripción de la Arquitectura (Vista modelo de despliegue), Modelo de Implementación, Descripción de la Arquitectura (Vista modelo de Implementación), Plan de Integración de construcciones, Modelo de Pruebas, Casos de Uso de Prueba, Plan de Prueba, Plan de despliegue, diseño de la BD, Modelo físico de la BD, Prototipo de Interfaz de Usuario, ejecutable del producto y manual de usuario.

## **2.2.2 Gestión y control de los cambios en la configuración.**

### **2.2.2.1 El Control de los Cambios**

El control de los cambios es un proceso que se considera como el más importante en la GCC, es casi evidente que en un gran proyecto el cambio no controlado lleva rápidamente al caos. Para cualquier proyecto el proceso de control de cambios combina los procedimientos humanos y las herramientas que proporcionan el mecanismo para el control de los cambios.

*“El control de cambios es una actividad procedimental que asegura la calidad y la consistencia a medida que se realizan cambios en los objetos de la configuración” [2]*

Para la realización de cambios es necesario contar con personas que tengan cierto conocimiento de las actividades que se realizan en el proyecto para poder tomar decisiones y decidir hasta que punto es factible la realización del cambio. Estas personas (puede ser una o varias de acuerdo a la complejidad del proyecto) formarán parte del Comité de Control de Cambios (CCC). El CCC generalmente es el encargado de determinar las prioridades de las peticiones de cambios y el estado en que se encuentran dichas peticiones.

Estados en que se pueden encontrar las peticiones

- Aprobado: es cuando se analiza una solicitud de cambio y se aprueba.
- Rechazado: es cuando el cambio solicitado no cumple las funciones necesarias, afecta a ECS importantes u otros elementos que provoca que no sea rentable y que sea rechazado.
- Pendiente de aprobación: son aquellos cambios que no son tan importantes y que se analizarán luego de pasados los cambios críticos.

Por estrategias del proyecto se decidió que los integrantes del CCC son:

Henry Cruz Mulet que por ser el jefe de proyecto será el encargado de:

- Verificar la correcta documentación de los cambios.
- Verificar que los cambios sean informados al equipo de desarrollo.
- Verificar el uso de la herramienta de control de versiones por todo el personal.
- Participar en la toma de decisiones que en relación a los cambios se realicen.

Armando Robert Lobo responsable principal de la arquitectura del sistema en el proyecto, será el encargado de:

- Realizar una valoración técnica de las solicitudes de cambio:
  - Determinar el impacto de los cambios en el resto de los componentes del sistema.
  - Análisis del coste de cada cambio.
- Participar en la toma de decisiones que en relación a los cambios se realicen.

Daily Hernández Segura y Yenei Y. Samada Lorié, responsables de la GCC serán las encargadas de:

- Recibir, priorizar y clasificar las solicitudes de cambios.
- Aprobar o rechazar cambios que no requieran pasar por el CCC.
- Controlar que los cambios sean realizados en las versiones adecuadas.
- Verificar que los cambios son seguidos correctamente por parte de los involucrados en los mismos.
- Informar y hacer seguimiento de las solicitudes de cambio aprobadas.
- Reportar el estado de los cambios.
- Realizar auditorías y revisiones a los ECS luego de los cambios y antes de su liberación.
- Informar de los cambios y distribuir la nueva versión de los ECS que forman parte del producto.

Además de estas personas participarán otras que serán llamados “invitados” que pueden ser los jefes de los módulos involucrados en los cambios. Estos invitados participarán también en el proceso, para estar al tanto de los cambios que a sus respectivos módulos concierne.

En el proyecto están previstos 2 tipos de cambios:

1. Cambios internos: Son las solicitudes de cambios realizadas por el equipo de desarrollo, o sea los desarrolladores.
2. Los cambios externos: Son las peticiones de cambios realizadas por los clientes.

Estos cambios tanto internos como externos pueden clasificarse en:

- Cambios formales: Son los que solo pueden realizarse mediante un procedimiento formal, que deben ser analizados y aprobados por el CCC.
- Cambios no formales: Son los que por su sencillez no requieren de un proceso formal, pues no tiene gran connotación para la organización y visibilidad del proyecto.
- Cambios a nivel de proyecto: Son cambios que pueden afectar a una línea base establecida o pueden ser también grandes cambios que se realizan internamente y que son de interés documentar. Estos cambios deben ser evaluados, analizados para decidir si se utiliza un proceso de cambio formal o no, aunque en algunos casos no precisa de un proceso formal de cambio. Estos cambios requieren de las decisiones del CCC para su ejecución.

#### **2.2.2.2 Proceso formal para la realización de cambios**

Para realizar cambios sobre líneas bases de la GC es necesario llevar acabo un procedimiento para su realización en el cual es valorado por parte del CCC el alcance del cambio, repercusión en el resto del sistema, y de acuerdo a las conclusiones a las que se llegan se decide o no proseguir con el cambio. Los pasos a seguir para la realización de cambios es la siguiente: (Ver Figura 14)

- 1) Se reconoce la necesidad de realización del cambio
- 2) Se presenta la solicitud de cambio, que puede venir provocada por un problema que se ha detectado o por un cambio en los requisitos.

- 3) Aprobación o rechazo inicial de la solicitud de cambio. El Comité de Control de Cambios es el responsable de esta actividad.
- 4) Clasificación y registro de la solicitud de cambio. Se almacena el problema con un código consecutivo de problemas. Se clasifica la solicitud recibida en: Defecto, Mejora, Error o Cambio.
- 5) Evaluación de la solicitud de cambio: Si ha sido aprobada, para calcular el esfuerzo técnico, los posibles efectos secundarios, el impacto global sobre otras funciones del sistema y el coste estimado del cambio. Como resultado se obtiene un Informe de Cambio y se continúa con el paso 5. Si no se aprueba la solicitud se pasa al paso 8.
- 6) Presentación del Informe de Cambio al Comité de Control de Cambios. Si se considera que el cambio es beneficioso se genera una Orden de Cambio de la Configuración (OCC), que describe el cambio a realizar, las restricciones que se deben respetar y los criterios de revisión y de auditoría. Esta OCC es asignada a alguno de los ingenieros de software para que se encargue de llevarlo a cabo. En este momento, el objeto a cambiar se da de baja en el repositorio.
- 7) Se realiza el cambio, entrando en un proceso de seguimiento y control.
- 8) Una vez finalizado el cambio, se certifica, mediante una revisión, que se ha efectuado correctamente el cambio y con ello se ha corregido el problema detectado o bien se han satisfecho los requisitos modificados. El objeto se devuelve al repositorio.
- 9) Se notifica el resultado al solicitante del cambio y se finaliza el procedimiento.

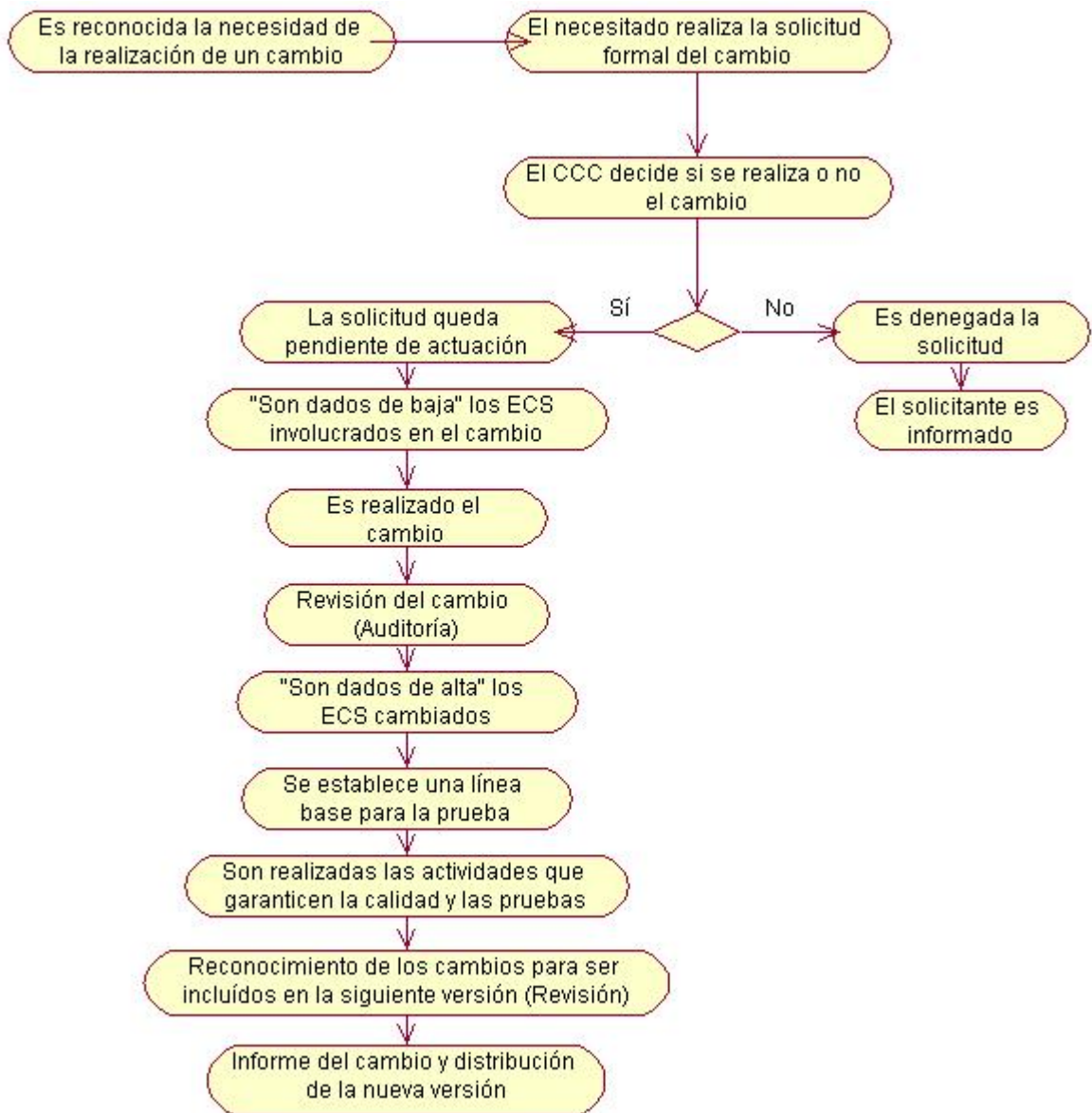


Figura 14. Proceso formal para a realización de cambios

James Bach en 1998 resumió magistralmente la realidad sobre el control de los cambios al decir que:  
*" El control del cambio es vital. Pero las fuerzas que lo hacen necesario también lo hacen molesto. Nos preocupamos por el cambio porque una diminuta perturbación en el código puede crear un gran fallo en el*

*producto. Pero también puede reparar un gran fallo o habilitar excelentes capacidades nuevas. Nos preocupamos por el cambio porque un desarrollador pícaro puede hacer fracasar el proyecto; sin embargo las brillantes ideas nacidas en la mente de estos pícaros, y un pesado proceso de control de cambio pueden disuadirle de hacer un trabajo creativo” [2]*

### **2.2.2.3 Procedimiento para la solicitud de cambios**

Para formular la solicitud formal de cambio se cuenta con una plantilla, en la cual se recoge la información necesaria para hacer formal la solicitud. Y en la cual se debe mostrar la siguiente información:

- Fecha de solicitud: [dd/mm/aa]
- Título del cambio
- ¿Quién emite la solicitud?
- Descripción de la propuesta
- Razones o motivos que provocan la solicitud de un cambio
- Facilidades o beneficios que ofrece el cambio
- Grado de importancia: (máxima, media,mínima)
- Documentos, ECS o parte del sistema que afecta
- Propuesta de solución al problema

Hasta el estado actual del proyecto no han sido presentadas solicitudes de cambios formales, solo a nivel de proyecto, también se ha dado el caso de no conformidades con respecto al levantamiento de los requerimientos. Los principales fueron:

1. La especificación de los requisitos de los 3 primeros módulos en desarrollo tuvieron problemas con el formato de la plantilla que debían documentar así como con los puntos que debían esclarecer dentro de la misma.
2. Problemas de comprensión al identificar los requisitos, el equipo encargado de hacer el levantamiento inicial de requisitos en algunos casos no comprendió las necesidades planteadas por los clientes.
3. Algunos de los requisitos seleccionados no fueron bien explicados en la documentación requerida para ello por lo que los clientes no entendían de que se trataban.

La solicitud de cambio a nivel de proyecto estuvo dada porque en la primera iteración concluida y presentada al cliente en el módulo Generador de Modelos se hizo necesario algunos cambios y algunas nuevas funcionalidades para mejorar el sistema.

Cambios sugeridos por parte de los clientes:

1. El campo dedicado a introducir el código del Indicador debe permitir 11 dígitos en vez de 6 como permite actualmente.
2. Incorporar subsistema de seguridad a la aplicación para garantizar que cada uno de los modelos que se generan cuenten con la seguridad que el software requiere.

Cambios sugeridos por parte de los desarrolladores:

1. Cambio en la arquitectura del módulo, debido a problemas existentes en el diseño arquitectónico actual y la incorporación de nuevas capas (Capa de Acceso a Datos).
2. Cambio en la estructuras de las clases debido a incompatibilidad entre la BD y el diseño actual y para disminuir la dependencia entre un Módulo y otro (Módulo Generador de Modelos y Módulo Entrada de Datos)

Para el módulo Reportes y Clasificadores

Los cambios sugeridos por los clientes son los siguientes:

1. Orientar los clasificadores a los registros con el objetivo de que a la hora de clasificar un CI aparezcan los clasificadores y por ende las clasificaciones que realmente son necesarias.
2. Filtrar la información proveniente de la Base de Datos para no cargar demasiado la memoria.
3. Formalizar la utilización de los registros en cuanto al tratamiento de los establecimientos pertenecientes a Centros Informantes determinados.

Cambios sugeridos por los desarrolladores:

1. Cambio del color predominante en la aplicación: El cambio consistió de gris que estaba inicialmente a GradientInactiveCaption. Esto produjo que a todos los controles de usuario, controles y formularios tuvieran que modificar su color.
2. Creación de una nueva tabla en la Base de Datos para guardar los CI que sean dados de baja; de ahí que la lógica sea modificada en consecuencia con los cambios realizados en la BD.

3. Atender las actualizaciones o refrescamientos a nivel de Base de Datos respecto a la aplicación.

Para el módulo de Entrada de Datos

Los cambios sugeridos por los clientes son los siguientes:

1. Incorporar el procesamiento de la validación mediante los cuadros.

Cambios sugeridos por los desarrolladores:

1. Cambio del color predominante en la aplicación:
2. El cambio consistió de azul que estaba inicialmente a ButtonFace. Esto produjo que a todos los controles de usuario tuvieran que modificar su color.
3. Cambio en la estructuras de las clases debido a incompatibilidad entre la BD y el diseño actual y para disminuir la dependencia entre un Módulo y otro (MGM - MED).

#### **2.2.2.4 Procedimiento para aprobar o rechazar las solicitudes de cambios**

Para aprobar o rechazar las solicitudes de cambios es necesario tener en cuenta aspectos como:

- Tamaño
- Complejidad
- Costo
- Tiempo para su culminación
- Disponibilidad de recursos para la realización del cambio (humanos, materiales)
- Alcance
- Relación con otros cambios aprobados o en proceso de aprobación
- Afectaciones al resto de los componentes que conforman el sistema
- Influencias del cambio

#### **2.2.2.5 Procedimiento para el seguimiento de los cambios**

Una vez aceptados los cambios se debe hacer un seguimiento de la evolución de los problemas que lo provocaron, así como llevar un registro de las acciones efectuadas para resolver los problemas e informes acerca de los problemas, por lo que al procedimiento de seguimiento de los cambios usualmente se le llama también Gestión de Problemas.



A la hora de llevar a cabo el seguimiento de los cambios es necesario tener en cuenta los siguientes aspectos:

- Persona que lo solicitó
- Breve descripción del problema que lo provocó
- Módulos que afecta
- Prioridad
- Responsables de la realización del cambio
- Solución al problema

### **2.2.3 Procedimiento para el Control de Versiones**

En cada laboratorio de proyecto para la utilización del repositorio SVN, se cuenta con un responsable de mantener en funcionamiento dicho servidor que viene a funcionar como un administrador, el cual es el encargado de configurarlo y crear el espacio de trabajo de cada proyecto presente en el laboratorio.

En el servidor es creado lo siguiente:

- El repositorio SVN de cada proyecto
- El Trac para la gestión y planificación de cada proyecto
- El repositorio “especial” svn-meta para la administración del repositorio

Como se hace referencia existen 2 servidores el primero es donde se almacena la información y el segundo es uno que podemos llamar “especial” llamado svn-meta, el cual solo es usado para que los responsables de la GCC manejen los permisos de utilización del repositorio real de cada proyecto o sea cada responsable de la configuración es el encargado de gestionar los datos de los usuarios y sus permisos al repositorio real.

Las actividades a seguir por la GCC para el control de versiones con el SVN es la siguiente:

1. Creación de usuarios en el repositorio: Una vez creado el espacio de trabajo del proyecto se procede a crear los usuarios con sus respectivas contraseñas y permisos para que puedan comenzar a trabajar. Para administrar los usuarios y establecer las contraseñas y permisos es necesario realizarlo mediante los siguientes comandos básicos, por estar montado el servidor en el sistema operativo Debian de Linux:

- svn: para la interacción con el svn-meta.
- htpasswd: para crear y/o cambiar las contraseñas de los miembros del proyecto
- mc: para editar los ficheros dav\_svn.passwd y dav\_svn.authz. En el fichero dav\_svn.passwd es donde se almacenarán los usuarios y sus contraseñas, mientras que el dav\_svn.authz permite establecer los permisos a los usuarios.

La asignación de permisos a los usuarios se realiza teniendo en cuenta hasta que punto debe tener acceso a modificar qué información, aunque todos los usuarios tiene permisos de lectura que le permite además bajar cualquier información del repositorio, solo unos pocos (en este caso los cinco jefes de módulos y el analista de ese módulo) tienen permisos de escritura, por el hecho de ser ellos responsables de la integridad y confiabilidad de la información que se desea subir y solo en algunos módulos el analista tiene permiso de escritura también (existe un responsable definido por módulo para la manipulación de la información en el repositorio).

El mecanismo para administrar los usuarios, contraseñas y permisos es el siguiente:

a) Crear una copia local del svn-meta solo basta con ejecutar el comando svn co (check out) de esta manera: `svn co http://10.32.18.29:5901/svn-meta/ONE`. De esta manera se crea el directorio cd ONE dentro del cual aparecen los ficheros dav\_svn.passwd y dav\_svn.authz.

b) Se realizan las operaciones necesarias para la creación de los usuarios haciendo uso del comando htpasswd de la siguiente manera:

```
htpasswd -m <usuario del dominio>
```

c) Luego se establecen los permisos haciendo uso del fichero dav\_svn.authz de esta manera:

```
[Expediente de Proyecto/Desarrollo Técnico/SIGE MGM/]
```

```
<usuario del dominio>=rw si es de lectura y escritura
```

```
*=r para el resto de los miembros que no tienen permitido la lectura
```

Significa que un usuario del dominio X, miembro del proyecto tiene permisos de lectura y escritura sobre la carpeta Desarrollo técnico al módulo SIGE MGM y que todos tienen permiso de lectura.

d) Una vez que los ficheros están listos se ejecuta el comando `svn add dav_*`

e) Luego se ejecuta el comando `svn ci -m "<Comentario de lo realizado>"`

2. Instruir a los miembros del proyecto acerca del uso del SVN con la herramienta cliente para Windows Tortoise siempre teniendo en cuenta los pasos básicos:

- Hacer copia local del contenido del SVN haciendo un `svn checkout`
- Si esta ya existe haciendo `svn update`
- Subir el contenido al repositorio mediante `svn commit`

3. Garantizar que los miembros del proyecto usen el SVN

4. Administrar el Trac de SVN mediante el Trac-admin

### **2.2.3.1 SVN, herramienta para el control de versiones utilizada en el proyecto ONE.**

Para gestionar, controlar y garantizar la calidad de la información se utiliza la herramienta de control de versiones Subversión. Ofrece muchas características y aspectos gratos para el personal que lo utiliza. Esta herramienta esta diseñada para el trabajo sobre múltiples plataformas, mantiene servidor Linux aunque las aplicaciones se puedan desarrollar sobre plataformas Mac OS o Windows, entre otros:

- Posee alta eficiencia y elevado nivel de seguridad en el manejo de las versiones así como facilidades para su manejo.
- Es intuitivo y fácil de usar.

El repositorio de datos presenta la siguiente organización:

<http://10.32.18.29:5901/svn/ONE/>

[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico](http://10.32.18.29:5901/svn/ONE/Desarrollo_Tecnico)

[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico/ SIGE Arquitectura/](http://10.32.18.29:5901/svn/ONE/Desarrollo_Tecnico/SIGE_Arquitectura/)

[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico/ SIGE BD/](http://10.32.18.29:5901/svn/ONE/Desarrollo_Tecnico/SIGE_BD/)

[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico/ SIGE MED/](http://10.32.18.29:5901/svn/ONE/Desarrollo%20Tecnico/SIGE%20MED/)  
[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico/ SIGE MGM/](http://10.32.18.29:5901/svn/ONE/Desarrollo%20Tecnico/SIGE%20MGM/)  
[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico/ SIGE MGR/](http://10.32.18.29:5901/svn/ONE/Desarrollo%20Tecnico/SIGE%20MGR/)  
[http://10.32.18.29:5901/svn/ONE/Desarrollo Tecnico/ SIGE MRC/](http://10.32.18.29:5901/svn/ONE/Desarrollo%20Tecnico/SIGE%20MRC/)

[http://10.32.18.29:5901/svn/ONE/Gestion de Configuracion](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20Configuracion/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de Configuracion/Plantillas \(todas\)/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20Configuracion/Plantillas%20(todas)/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de Configuracion/Artefactos.doc/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20Configuracion/Artefactos.doc/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de Configuracion/Plantillas \(todas\)/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20Configuracion/Plantillas%20(todas)/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de Configuracion/Procedimientos/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20Configuracion/Procedimientos/)

[http://10.32.18.29:5901/svn/ONE/Gestion de proyecto](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20proyecto/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de proyecto /8 Ambiente Desarrollo/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20proyecto%20/8%20Ambiente%20Desarrollo/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de proyecto /11 Recursos Humanos \(RRHH\)/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20proyecto%20/11%20Recursos%20Humanos%20(RRHH)/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de proyecto /Recursos Materiales/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20proyecto%20/Recursos%20Materiales/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de proyecto /Investigaciones/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20proyecto%20/Investigaciones/)  
[http://10.32.18.29:5901/svn/ONE/Gestion de proyecto /Comunicacion/](http://10.32.18.29:5901/svn/ONE/Gestion%20de%20proyecto%20/Comunicacion/)

[http://10.32.18.29:5901/svn/ONE/Seguridad Informatica](http://10.32.18.29:5901/svn/ONE/Seguridad%20Informatica/)  
[http://10.32.18.29:5901/svn/ONE/Seguridad Informatica/Plan de Seguridad Informatica.doc/](http://10.32.18.29:5901/svn/ONE/Seguridad%20Informatica/Plan%20de%20Seguridad%20Informatica.doc/)

[http://10.32.18.29:5901/svn/ONE/Sistema de Calidad](http://10.32.18.29:5901/svn/ONE/Sistema%20de%20Calidad/)  
[http://10.32.18.29:5901/svn/ONE/Sistema de Calidad/Inspecciones/](http://10.32.18.29:5901/svn/ONE/Sistema%20de%20Calidad/Inspecciones/)  
[http://10.32.18.29:5901/svn/ONE/Sistema de Calidad/Estandares de codificacion/](http://10.32.18.29:5901/svn/ONE/Sistema%20de%20Calidad/Estandares%20de%20codificacion/)

El repositorio esta conformado por el Expediente de Proyecto, o sea todas las informaciones están organizadas como lo indica el expediente definido al inicio de este capítulo.

Nota: Se ha omitido el acento de algunas palabras para conservar el formato original que se le dio a la estructura del SVN pues el acento en algunos casos provoca problemas.

Para organizar mejor el trabajo y por necesidades de seguridad en la información que se genera se decidió que solo el Jefe de Modulo y el analista de ese módulo serán los encargados de subir al SVN la información del producto que sus módulos generan y controlar que dicha información sea la correcta, antes de que sea revisada formalmente mediante una RTF. El resto del equipo trabaja en diferentes actividades y tareas que rigen su rol.

### **¿Por qué SVN y no VSS?**

Es necesario aclarar que a pesar de estar elaborándose el software en Visual Studio. Net 2005 (VS 2005) y ser VSS la que más eficientemente se integra a dicha herramienta, se escogió el SVN para su utilización debido a las siguientes características:

- Es un producto que corre bajo licencia Apache, acorde a las políticas de la UCI de migrar paulatinamente a software libre, mientras que VSS es una herramienta propietaria.
- Se integra muy bien en los repositorios centrales en la UCI permitiendo de este modo una mejor integración con los servicios que estos brindan. Además se integra muy bien al Trac, herramienta que se utiliza para la gestión de proyectos.
- No existe ningún problema a la hora de integrar SVN con la herramienta de desarrollo VS 2005, pues existen herramientas que permiten dicha integración y facilitan el trabajo, tal es el caso del Anca.
- Es muy eficiente en cuanto al uso de espacio en disco y para el trabajo en paralelo sobre el mismo código, permitiendo el mezclado, el importe de repositorios, etc.

### **2.2.4 Generación de informes de estado**

La generación de informes de estado desempeña un importante papel en el éxito de cualquier proyecto de desarrollo de software. El objetivo principal de esta tarea es el de mantener informados a los participantes del proyecto el estado de la configuración del software y su evolución, del producto como tal, de los procedimientos, de las decisiones de cambios, reuniones, talleres, conferencias, etc. por lo que en alguna medida sirve de ayuda para mejorar la comunicación entre los integrantes del equipo incluso con los

clientes. De esta forma los integrantes del equipo tienen conocimiento de todas las actividades que se realizan y los clientes de la situación y desarrollo del software.

Esta tarea es muy importante si se tiene en cuenta que sirve para tener bien contabilizado lo que ocurre en el proyecto y cuando ocurre, por lo que evita que se trabaje con información duplicada, que se trabaje sobre ECS que han sufrido cambios y que no han sido informados, que se trabaje sobre información no actualizada o que se cometan los mismos errores reiteradamente si se analiza que sirve para determinar posibles fallos o errores en el sistema. Por otro lado permite identificar aquello que se hizo bien y que puede repetirse. Una de las características más importantes de los informes de estado es el hecho de que por tener documentado y contabilizado lo que ocurre en el proyecto permite aminorar los problemas que pueden presentarse si por alguna razón un integrante fundamental para el trabajo en el proyecto debe abandonar el mismo.

La actividad deberá tener en cuenta 2 tareas muy importantes que serán detallados posteriormente. Se está haciendo referencia a los Registros e Informes.

#### **2.2.4.1 Registros**

Es la tarea que permite llevar un control de toda la información relativa a lo que se desea monitorear. Los principales registros que se llevan acabo son:

1. Registro de los ECS: Es donde se almacena toda la información referente a los ECS como puede ser: tipo de ECS, relaciones con el resto de los elementos, Identificador del ECS, descripción, autores, etc.
2. Registro de las líneas bases: Es donde se captura la información relacionada con las líneas bases como puede ser:
  1. Fecha de establecimiento
  2. Versión y fase a la que pertenece
  3. ECS que la componen
3. Registro de solicitudes de mejoras o cambios: Incluye información que permite tener controlado las diferentes solicitudes de cambios realizadas en el proyecto. Esta puede ser:
  - Quien emite la solicitud
  - Resumen y contenido de la propuesta
  - Situación de la solicitud (Aplazada, aprobada, rechazada)

4. Registro de Cambios realizados: Contiene información referente a los cambios realizados como puede ser:

- Evaluación del cambio: Tiene que ver con el coste y el esfuerzo que genera el cambio.
- Impacto del cambio: Partes del sistema o componentes que afecta
- Historial del cambio: Número de solicitud al que pertenece, fecha de solicitado, fecha de aprobado, fecha de realizado, historial de las revisiones, número de incidencias del cambio, etc.

5. Registro de la documentación generada por el CCC: Es donde se almacena toda la información referente a las reuniones del CCC. Estas pueden ser:

- Fecha
- Relación de personal asistente
- Propósito
- Acuerdos tomados respecto a líneas bases cambiadas y aprobadas, análisis de las solicitudes de cambios,
- Resultados de las auditorias: Resume deficiencias detectadas y las acciones para solucionarlas.

6. Registro sobre modificaciones en la documentación: Es la información referente a las modificaciones a los documentos para lo cual no se necesita un procedimiento formal de cambio. Esta información al igual que los cambios formales de cambios es de interés documentarla y llevar un registro. Para ello se gestiona:

- Número de revisión a la que pertenece la modificación
- Resumen y contenido de la modificación
- Documentos modificados
- Fecha de inicio y fin de la modificación
- Responsable (s) de la modificación

### **Registro sobre modificación de la Arquitectura**

Al comienzo de la segunda iteración en el proceso de desarrollo del Software SIGE para la ONE, debido al refinamiento de los requisitos y la nueva estructura en módulos del proyecto, se realizó un análisis de la arquitectura establecida, esta actividad se hizo necesaria debido a que la misma no

cubría todos los aspectos y funciones necesarios para el trabajo del equipo de proyecto, o sea, limitaban el trabajo.

La arquitectura establecida es la de arquitectura en capas y tiene la siguiente distribución:

**Capa de presentación:**

1. Recoger información del usuario
2. Enviar esta información a la capa de servicios del negocio
3. Recoger los resultados de la capa de servicios del negocio
4. Presentar los resultados al usuario

**Capa de servicios del negocio:**

1. Recibir información de la capa de presentación
2. Interactuar con los servicios de datos para realizar la lógica de negocio de la aplicación
3. Enviar resultados a la capa de presentación

**Capa de acceso a datos:**

1. Almacenar datos
2. Recibir datos
3. Mantenimiento de los datos
4. Integridad de los datos

Luego del análisis de la aplicación de la arquitectura a cada módulo, se llegó a la conclusión de que eran necesarias algunas modificaciones y cambios no formales a dicha arquitectura, de manera tal que se resolvieran todas las limitaciones y se cubrieran todas las funcionalidades.

La nueva estructura queda de la siguiente forma:

Arquitectura en capas, se incluye una capa para un Web Server y otra de presentación entre la Capa de presentación y la lógica del negocio.

El trabajo de la arquitectura fue orientado a componentes y a controles visuales.

**Lo que facilita:**

El desenvolvimiento del equipo de desarrollo en un ambiente de desarrollo favorable, se estableció un plan para la construcción de la arquitectura así como la definición de su estructura, etc.



7. Registros de Incidencia: Es donde se almacena toda la información referente a la incidencia de cambios sobre un ECS.
  - Nombre del ECS
  - Código del ECS
  - Numero de Incidencia
  - Cambio propuesto
  - Evaluación de la situación
  - Conclusiones
8. Registros de Seguridad: Recoge la información referente a las directrices de seguridad que se establecen en el Plan de Seguridad.

#### **2.2.4.2 Informes**

Los informes que se determinan por el proyecto son:

- Informe del estado de los ECS: Permite dar a conocer los ECS que componen el expediente del proyecto.
- Informe de estado de las relaciones de los ECS: Permite dar a conocer el estado de las relaciones que se establecen entre los ECS.
- Informe de estado de los cambios: Permite resumir en un tiempo determinado el estado en que se encuentran las solicitudes de cambios registradas.
- Informe de los cambios realizados: Permite informar los cambios una vez realizados.
- Informe de las modificaciones: Permite informar las modificaciones hechas al producto durante su ciclo de vida.
- Informe del CCC: Permite dar a conocer todas las decisiones que se lleven a cabo con respecto a los cambios solicitados.
- Informe de línea base: Permite presentar a los desarrolladores la línea base del producto que se establece.
- Informe de cambios: Este informe permite dar a conocer a los miembros del proyecto los ECS a los que se le han realizado cambios.

## 2.2.5 Auditoría de la Configuración del Software

El proceso de auditorías se realiza mediante pruebas, estas tienen como objetivo evaluar el trabajo desempeñado por miembros del equipo de proyecto, principalmente los responsables de la GCS. La auditoría te muestra los errores, defectos que se puedan presentar así como las necesidades inmediatas a atender.

Es una actividad de aseguramiento de la calidad que ayuda a mantener la misma durante la realización de los cambios, el control de las versiones, el establecimiento de la línea base, los ECS y sus relaciones, etc. Para asegurar que los cambios han sido implementados correctamente se realizan actividades imprescindibles:

- Revisiones técnicas formales (RTF). Los objetivos de las RTF son los de determinar:
  - Posibles errores en la implementación o la lógica del sistema.
  - Compatibilidad del producto con requisitos funcionales.

De las RTF para la GCC es importante gestionar:

1. Qué fue revisado
2. Por quién fue revisado
3. Qué fue descubierto
4. Resumen de la revisión

- Auditorías de la configuración del software.

Para gestionar las auditorías se tendrán en cuenta los siguientes aspectos que son de interés analizar y que forman parte de un procedimiento para la realización de las auditorías, en dicho procedimiento es preciso comprobar que:

1. Se han cumplido las especificaciones descritas en la Orden de Cambio de la Configuración (OCC). Si se han incorporado cambios no especificados en dicha orden. (Ver Anexo 2)
2. Ha sido llevado a cabo una buena RTF para evaluar las correcciones técnicas.
3. Se han seguido como debe ser los procedimientos para solicitar, aprobar, realizar e informar los cambios.
4. Se han actualizado los ECS que se afectan con los cambios.

El procedimiento como tal para la realización de auditorías es el siguiente:

- Establecer responsables de la realización de la auditoría.
- Fecha de realización
- ECS a auditar
- Estado de los ECS a auditar (en desarrollo, en revisión, en mejoras de revisión, terminados, en proceso de cambios)
- Actividades del procedimiento
  1. Revisar la información
  2. Evaluar la información
  3. Enumerar errores y/o defectos encontrados
  4. Resumen de la auditoría
- El procedimiento termina con la distribución del informe de estado de auditoría de los ECS a la dirección del proyecto y a los miembros del equipo de desarrollo.

Los resultados obtenidos luego de la aplicación del modelo de GCC al proyecto ONE serán analizados en detalle en este capítulo. Para hacer el análisis se tendrán en cuenta los indicadores de avance y calidad definidos con anterioridad.

### 3.1 Expediente de proyecto

Teniendo en cuenta que el expediente de proyecto quedó aprobado y establecido en el proyecto los resultados alcanzados son notables:

- El expediente es público y puede ser consultado por cualquier persona incluso que no forme parte del proyecto en la dirección
- Es auditable en cualquier momento por los revisores técnicos y encargados de la calidad del proyecto.
- La herramienta que se utiliza para manejar el expediente es el SVN con el Trac ha demostrado que es muy estable y fácil de usar por lo que el expediente está disponible en cualquier momento para cualquier miembro del equipo de desarrollo de forma tal que puedan obtener en cualquier momento la información actualizada.

Por lo que el objetivo principal de su establecimiento, o sea, el de organizar, controlar y gestionar adecuadamente toda la información que genera el producto en su evolución, permita que tanto el equipo desarrollo como los clientes, puedan tener actualizada y al alcance de la mano en cualquier momento toda la documentación que necesitan, sin tener que depender de factores humanos para su conocimiento, fue cumplido en su totalidad.

### 3.2 ECS

Los resultados respecto a ECS son evidentes, si se tiene en cuenta que se definieron y establecieron correctamente todos los ECS que son de interés monitorear por la GCC hasta el momento, así como las relaciones que se establecen entre ellos, para mitigar las consecuencias de los cambios y las posibles afectaciones que pudieran ocasionar estos en el resto de los ECS. Para medir el avance en el trabajo se tuvieron en cuenta algunos indicadores:

### **3.2.1 Avance del proyecto según la GCS**

#### **▪ Avance del proyecto con respecto a la Configuración del software**

Para medir el avance del proyecto por parte de la GCS y teniendo como referencia lo antes expuesto en el Capítulo 1 referente a este indicador, se estableció una estrategia de medición, la misma involucra una serie de indicadores que representan el estado de los ECS involucrados en las iteraciones, esto nos permite controlar los ECS y posibilita la forma de medir el avance, comparando entre las iteraciones el nivel de desarrollo, de terminación de los ECS y de calidad. A continuación se muestran los siguientes resultados:

Los ECS terminados que se presentaron a los clientes en la primera iteración, Versión 1.0 fueron los siguientes:

- Documento visión del sistema
- Especificaciones de los requisitos de software (primer levantamiento)
- Especificaciones de Casos de Uso del sistema
- Documento de la Arquitectura del sistema
- Plan de Gestión de la Configuración de Software
- Plan de Gestión de Cambios
- Procedimiento de cambio
- Control de versiones
- Propuesta del primer prototipo (ejecutable del producto)
- Modelo de datos
- Modelo físico de la base de datos

Nota: En esta iteración no se contaba con indicadores para medir el desarrollo del proyecto por parte de la GCS por lo que se trataron de manera general.

Los resultados que se muestran a continuación pertenecen a la iteración 2, Versión 1.1.

#### **▪ Cantidad de elementos de configuración en desarrollo.**

De los 38 ECS determinados hasta el momento, 8 están en desarrollo lo que significa que un 21.1% de los ECS se encuentran en actual desarrollo por parte del equipo de desarrollo. Ellos son:

- 1) Modelo de Datos.
- 2) Diseño de la Base de datos.
- 3) Descripción de la Arquitectura (vista modelo de diseño).
- 4) Modelo de Pruebas.
- 5) Elementos de Configuración de Software.
- 6) Plan de desarrollo del Software.
- 7) Plan de Gestión de la Calidad.
- 8) El código fuente del programa.

▪ **Cantidad de elementos de configuración terminados.**

De los 38 ECS determinados hasta el momento, 20 están terminados y aprobados por los clientes lo que significa que un 52.6% de los ECS se encuentran terminados. Ellos son:

- 1) Plan de Gestión de Requerimientos.
- 2) Especificaciones de requisitos de Software.
- 3) Especificaciones de Casos de Uso del Sistema.
- 4) Documento Visión del Sistema.
- 5) Plan de Iteración.
- 6) Plan de Prueba.
- 7) Casos de Uso de Prueba.
- 8) Modelo de Implementación.
- 9) Realización de Casos de Uso de Diseño.
- 10) Realización de Casos de Uso del análisis.
- 11) Modelo de Diseño.
- 12) Modelo de Análisis.
- 13) Código objeto y ejecutable.
- 14) Plan de Gestión de la Configuración.
- 15) Plan de Gestión de Cambios.
- 16) Control de versiones.

- 17) Preceptos de la Programación.
- 18) Preceptos del modelado de Casos de Uso.
- 19) Preceptos de Diseño.
- 20) Preceptos de Prueba.

- **Cantidad de elementos de configuración en revisión.**

De los 38 ECS determinados hasta el momento, 2 están en proceso de revisión, lo que significa que un 5.3 % de los ECS se encuentran en revisión. Ellos son:

- 21) Prototipo Interfaz de Usuario.
- 22) Manual de usuario.

- **Cantidad de elementos de configuración en espera desarrollo**

El resto de los ECS se encuentran en espera de desarrollo. Representan un 21.1% del total de los ECS establecidos. Ellos son:

- 1) Modelo de Despliegue.
- 2) Descripción de la Arquitectura (vista modelo de despliegue).
- 3) Descripción de la Arquitectura (vista modelo de implementación).
- 4) Plan de Integración de construcciones.
- 5) Resumen de evaluación de Prueba.
- 6) Plan de Despliegue.
- 7) Plan de Aceptación del Producto.
- 8) Plan de Administración de Riesgos.

Teniendo en cuenta la información anterior, se ha realizado un análisis de los resultados obtenidos en cada iteración que se ha desarrollado, estos últimos resultados han sido valorado haciendo uso de indicadores para la evaluación del avance del proyecto.

Sírvase de aclaración que una iteración del proyecto ONE esta constituida de la siguiente forma: se define como iteración al conjunto de iteraciones definidas por RUP, o sea, cuando el producto pasa por todos los

flujos de trabajo dándole cumplimiento a todos los hitos establecidos por fase, obteniéndose un producto estable o un prototipo, le llamamos iteración del proyecto, las iteraciones están planificadas y según el tiempo de desarrollo poseen un límite de entrega de artefactos.

Se analizaron los resultados obtenidos en la 1ra iteración (Versión 1.0) y la 2da (Versión 1.1), actualmente se trabaja en la 3ra iteración (Versión 1.2).

En la Versión 1.0, basándonos en la arquitectura que en aquel momento estaba establecida (el trabajo se realizaba de manera global, general o sea no estaba dividida por módulos), se obtuvieron 11 ECS, la propuesta de un prototipo y un entregable y se estableció la primera línea base.

En la versión 1.1, con la ayuda de las modificaciones realizadas a la arquitectura (las cuales mejoran sobremanera el trabajo que desempeñan los desarrolladores), se obtuvieron 38 ECS por cada módulo (en diferentes estados), se presenta un prototipo funcional y bastante completo por cada módulo, se establece la segunda línea base de un producto más estable.

Los resultados del análisis se muestran en la figuras 15 y 15.a

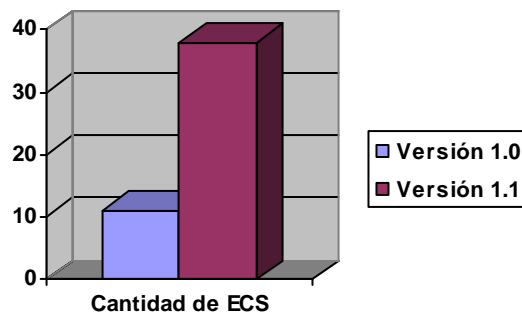


Figura 15. Gráfica de comparación entre los ECS de las versiones establecidas en cada iteración. Parámetro de comparación: Cantidad de ECS en números enteros.



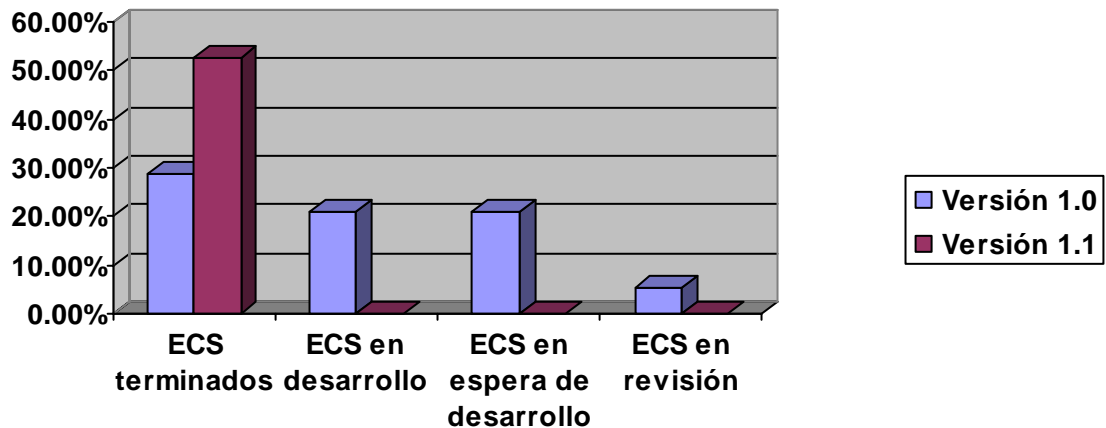


Figura 15.a. Gráfica de comparación entre los ECS de las versiones establecidas en cada iteración. Parámetros de comparación: Indicadores ECS terminados, ECS en desarrollo, ECS en espera de desarrollo, ECS en revisión evaluados en %.

- **Cantidad de no conformidades o solicitudes de cambios realizadas por los clientes**

Hasta el momento se cuenta con 3 no conformidades por parte de los clientes, así como con 3 solicitudes de cambios a nivel de proyecto.

- **Cantidad de solicitudes de cambios en revisión**

Las 7 solicitudes de cambios (las de los clientes y desarrolladores) presentadas se encuentran actualmente en espera de ser aceptadas para proceder al cambio.

- **Dominio y uso de la herramienta para el control de versiones**

El dominio de la herramienta para el control de versiones SVN por parte de los responsables de gestión de la configuración aún no es la más correcta aunque se ha incrementado el nivel de conocimiento adquirido en la práctica y con el apoyo de personal familiarizado con la herramienta así como del estudio basado en libros existentes y búsquedas en Internet.

Por parte del equipo de desarrollo se puede decir que el avance en cuanto a conocimiento y uso es muy poco si se tiene en cuenta que la herramienta no es utilizada regularmente por los desarrolladores pues en

la mayoría de los casos están adaptados a guardar la información en una carpeta creada por ellos mismos en su máquina, lo que provoca que en ocasiones la información del SVN no se encuentre actualizada pese al constante control e insistencia por parte de la dirección del proyecto a los desarrolladores de trabajar como debe ser y subir la documentación del proyecto generada.

Por lo que el dominio y uso de la herramienta para el control de versiones de manera general puede catalogarse de regular si se tiene en cuenta lo antes expuesto, actualmente se trabaja para el logro de mejores resultados.

- **Documentación y control de los cambios a los ECS.**

Actualmente se cuenta con un procedimiento formal para la realización de los cambios, discutido y aprobado por el CCC, que ha sido informado al equipo de desarrollo y llevado a cabo correctamente por el mismo.

- **Elaboración de las plantillas para la creación de artefactos**

De un total de 76 plantillas con las que se cuenta actualmente para documentar la información del proyecto, 35 fueron creadas por el equipo de gestión de configuración para cubrir las necesidades de documentación del proyecto, lo que significa un 46.05 % del total de plantillas. El resto de las plantillas, generadas por RUP, que se encontraban en inglés fueron traducidas y llevadas al formato (Ver figura 15) establecido por la gestión de configuración, de manera que se facilitó el trabajo del equipo de desarrollo contribuyendo de esta forma a la correcta documentación del proyecto.


	<b>Especificación de Requisitos</b>
<Nombre proyecto> <Módulo> Elaborado por: E-mail:	<p style="text-align: right;"><b>ONE</b></p> <p style="text-align: right;"><b>&lt;Módulo&gt;</b></p> <p style="text-align: center;"><b>Especificación de Requisitos</b></p> <p style="text-align: right;"><b>Versión &lt;x.x&gt;</b></p> <p style="text-align: right;">Especificación de Requisitos</p>

Figura 15. Ejemplo de formato dado a la plantilla de Especificación de Requisitos.

---

## Conclusiones

A partir de la investigación llevada a cabo para la realización de este trabajo de diploma se arribó a las siguientes conclusiones:

- Se estableció y aprobó satisfactoriamente el Plan de Gestión, artefacto que rige el trabajo de esta disciplina y que permite en gran medida la organización del trabajo.
- Se estableció y aprobó el Plan de Gestión de cambios, lo que permitió que se estructurara correctamente todo el proceso de control de cambios y sus correspondientes procedimientos.
- Fueron definidos todos los ECS cuya evolución es de interés monitorear por parte de la Gestión de la Configuración del Software.
- Se establecieron y aprobaron las líneas base correspondientes a cada fase del proyecto, tomando como guía los ECS previamente seleccionados.
- Se establecieron los artefactos entregables y plantillas que rigen la documentación del producto en desarrollo, lo que permitió que el software cuente con una información correctamente documentada y actualizada en el tiempo.
- Se logró hacer un aporte al mantenimiento y trazabilidad del software utilizando la herramienta para el control de versiones SVN, con el logro de resultados satisfactorios.

Lo anteriormente expuesto permitió que se le diera cumplimiento al objetivo general así como a los específicos del presente trabajo pues:

- Fue desarrollado íntegramente el modelo que permite establecer una adecuada GCC al proyecto productivo para el que trabajamos, por lo que en el mismo contará con una estrategia de aplicación de la GCC bien definida y organizada.
- La utilización del modelo propuesto contribuirá a organizar, gestionar y controlar en gran medida la información que se genera del software durante su ciclo de desarrollo, garantizando de esta manera calidad en el proceso de informatización de la ONE.
- El modelo comenzó a aplicarse al proyecto con el logro de buenos resultados.

---

## Recomendaciones

Para la completitud del presente trabajo se proponen las siguientes recomendaciones:

- Continuar aplicando el modelo desarrollado en el presente trabajo al proyecto ONE.
- Generalizar a toda la facultad las plantillas creadas por los responsables de GCC en el proyecto, pues se considera que a pesar de que aún se pueden seguir mejorando algunos detalles están bastante completas.
- Continuar mejorando el Expediente de proyecto: Crear nuevas plantillas para la documentación de la Calidad del proyecto así como las que tienen que ver con Recursos humanos, pues actualmente las que existen no cubren toda las necesidades de documentación de estas áreas.
- Utilizar la herramienta Fisheye para el manejo de la información que se almacena en un repositorio: Continuar profundizando en las facilidades que brinda y ponerlo en práctica en el proyecto.
- Establecer como política interna del proyecto que todos sus integrantes tengan pleno dominio del uso que debe dársele a la herramienta para el control de versiones SVN, pues actualmente los conocimientos de los integrantes del proyecto son mínimos, situación que dificulta en gran medida el trabajo de la GCC.
- Continuar perfeccionando los indicadores y métricas para medir la GCC en el proyecto que den una idea un poco más clara del avance de esta disciplina.

## **Bibliografía Consultada**

---

Javier Tuya, M.J.S.-C., Adenso Diaz, Gestión de la Configuración del Software en un proyecto de I+D. V Jornadas de Calidad del Software, 2000.

Hass, A.M.J., Configuration Management Principles and Practices. Addison Wesley. 2003.

Gestión de Configuración del Software [cited; Available from:  
<http://www.histaintl.com/soluciones/configuracion.htm>.

Babich, W., Software Configuration Management. 1986: Addison-Wesley.

Gestión de la Configuración para mejorar la comunicación y la colaboración 2006 [cited;  
Available from: <http://www.telelogic.es/products/synergy/synergycm/index.cfm>.

Gestión del Cambio para aumentar la calidad y la responsabilidad 2006.

Ivar Jacobson, G.B., James Rumbaugh, El Proceso Unificado de Modelado, ed. F. Varela. Vol. II. 2004, La Habana.

Antonio, A.d., La Gestión de la Configuración del Software. p. 15.

Pressman, R.S., Ingeniería del Software. Un enfoque práctico. 5ta edición ed. Vol. I. 2005. 156.

Gestión de la Configuración. [cited; Available from:  
<http://www.csi.map.es/csi/metrica3/gescon.pdf>.

Consideraciones sobre Gestión de la Configuración. 2005 [cited 2006; Available from:  
<http://www.calidaddelsoftware.com/modules.php?name=News&file=article&sid=66>.

Change control minimizes outages. 2006: p. 33.

Gestión de Software. 2006 [cited; Available from:  
<http://www.fing.edu.uy/inco/cursos/gestsoft/ppts/gs13.ppt>.

Bersoff, E.H. and A.M. Davis, Impacts of life cycle models on software configuration management. (technical). (technical), 1991. v34(n8): p. p104(15).

Burton, T. and H. Ernst, Configuration management can make a crucial difference. (Special Report: Software testing)(making a case for software configuration management tools). (Special

Report: Software testing)(making a case for software configuration management tools), 1993.  
v32(n9): p. p82(2).

Conradi, R. and B. Westfechtel, Version Models for Software Configuration Management. 1998.  
30(2): p. 232.

David Salischiker. Glosario de Términos ITIL. 2005 [cited; Available from: [http://www.itsmf-argentina.com.ar/IMG/pdf/ItilGlosario\\_v.01.pdf](http://www.itsmf-argentina.com.ar/IMG/pdf/ItilGlosario_v.01.pdf)

Navarro, A. Gestión de la configuración software. [Cited; Available from:  
[http://www.fdi.ucm.es/profesor/anavarro/10.\\_Gestion\\_de\\_la\\_configuracion\\_software.pdf](http://www.fdi.ucm.es/profesor/anavarro/10._Gestion_de_la_configuracion_software.pdf).

## **Bibliografía Citada**

---

1. Javier Tuya, M.J.S.-C., Adenso Diaz, *Gestión de la Configuración del Software en un proyecto de I+D*. V Jornadas de Calidad del Software, 2000.
2. Pressman, R.S., *Ingeniería del Software. Un enfoque práctico*. 5ta edición ed. Vol. I. 2005. 156-160.
3. Hass, A.M.J., *Configuration Management Principles and Practices*. Addison Wesley. 2003.
4. *Gestión de Configuración del Software* [cited; Available from: <http://www.histaintl.com/soluciones/configuracion.htm>].
5. Babich, W., *Software Configuration Management*. 1986: Addison-Wesley.
6. *Gestión de la Configuración para mejorar la comunicación y la colaboración* 2006 [cited; Available from: <http://www.telelogic.es/products/synergy/synergycm/index.cfm>].
7. Ivar Jacobson, G.B., James Rumbaugh, *El Proceso Unificado de Modelado*, ed. F. Varela. Vol. II. 2004, La Habana.
8. Antonio, A.d., *La Gestión de la Configuración del Software*. p. 15.