

*Universidad de las Ciencias Informáticas*

*Facultad 6*



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

*Título: Estrategia de Aseguramiento de Calidad para el proyecto SyGMe*

*Autora: Lilliam Menéndez Campo.*

*Tutora: Ing. Dayanis Elvia Alcantara Rabí.*

*22 de junio de 2011*

*La Habana*







## *Resumen*

La Universidad de las Ciencias Informáticas (UCI) es una de las empresas dedicadas a la producción de software. Su principal objetivo es que sus productos cumplan con los requisitos y la calidad deseada por el cliente. Esto implica que se deben llevar a cabo actividades para el aseguramiento de la calidad de sus procesos de desarrollo y de ahí surge la necesidad de adaptar métricas, normas y estándares internacionales de ingeniería de software para alcanzar estas metas.

El presente trabajo que lleva por título Estrategia de Aseguramiento de la Calidad para el proyecto SyGMe cuenta(Sistema de Gestión y Publicación de Metadatos Geográficos) con un estudio sobre calidad de software, especificándose temas como aseguramiento de la calidad, estándares y modelos de calidad, métricas y pruebas de software. Se realizó una descripción de la situación de dicho proyecto y se definió una estrategia de aseguramiento de calidad de sus productos. La aplicación de la misma permitió detectar y corregir las principales dificultades que afectan la calidad del producto final del proyecto de SyGMe.

Palabras Claves: calidad, Aseguramiento de la calidad, Metadatos.

Introducción.....	1
Capítulo 1: Fundamentación Teórica.....	5
1.1 Introducción al capítulo.....	5
1.2 Calidad.....	5
1.2.1 Calidad de Software.....	6
1.2.2 Calidad a nivel de Proceso.....	7
1.2.3 Calidad a nivel de producto.....	8
1.3 Calidad de software a nivel internacional, en Cuba y en la UCI.....	8
1.4 Gestión de la calidad.....	11
1.4.1 Planificación de la calidad.....	12
1.4.2 Control de la calidad.....	13
1.4.3 Aseguramiento de la calidad.....	14
1.4.4 Mejora de la calidad.....	15
1.5 Modelos y Estándares de calidad.....	16
1.5.1 Estándar ISO 9000/ 9000-3.....	17
1.5.2 Modelo de Madurez de Capacidad Integrado (CMMI).....	20
1.5.3 Otros modelos y estándares de calidad.....	23
1.6 Métricas de software.....	27
1.7 Surgimiento de los Metadatos.....	28
1.7.1 Normas internacionales para los metadatos geográficos.....	29
1.8 Conclusiones parciales.....	30
Capítulo 2: Propuesta de Solución.....	31
2.1 Introducción al capítulo.....	31
2.2 Estructura Organizativa del proyecto.....	31
2.3 Descripción de Roles.....	32
2.4 Estrategia de Aseguramiento de la Calidad.....	33
2.5 Plan de aseguramiento de Calidad.....	34
2.5.1 Propósito.....	34

2.5.2 Alcance.....	34
2.5.3 Definiciones, Acrónimos y Abreviaturas.....	34
2.5.4 Referencias.....	35
2.5.5 Resumen .....	36
2.5.6 Objetivos de calidad .....	36
2.5.7 Gestión del Aseguramiento de la Calidad.....	36
2.5.8 Documentación .....	38
2.5.9 Métricas .....	38
2.5.10 Estándares y Guías .....	41
2.5.11 Plan de Revisiones y Auditorías .....	41
2.5.12 Pruebas y Evaluación .....	44
2.5.12.1 Pruebas a realizar en el proyecto SyGMe .....	47
2.5.13 Herramientas, Técnicas y Metodologías.....	48
2.5.14 Gestión de Configuración .....	49
2.5.15 Registro de calidad .....	49
2.5.16 Entrenamiento .....	49
2.6 Conclusiones parciales .....	50
Capítulo 3: Validación de la Propuesta.....	51
3.1 Introducción al capítulo .....	51
3.2 Validación de la estrategia SQA.....	51
3.3 Revisiones .....	54
3.4 Métricas.....	55
3.5 Entrenamiento de los involucrados en el proyecto. ....	56
3.6 Conclusiones parciales .....	58
Conclusiones Generales .....	59
Recomendaciones .....	60
Bibliografía .....	62
Trabajos citados .....	65

### **Introducción**

En el mundo, el desarrollo de sistemas de software es cada vez más creciente, un producto con calidad se traduce en grandes ganancias y permite mayor competencia en el mercado.

El éxito de la industria del software hoy, depende en gran medida del aseguramiento de la Calidad de sus productos. El término calidad de software está siendo ampliamente utilizado, ya que para el desarrollo de sistemas informáticos, lo principal es que el mismo cumpla con las expectativas por la cual fue concebido, es decir, que tenga calidad, y para alcanzarla, debe estar libres de errores y que su interacción con el cliente sea lo más sencilla posible.

Existen varias razones que justifican por qué el aseguramiento de la calidad es crítico para la pervivencia de las empresas: es un factor competitivo, es esencial para el comercio internacional, reduce las pérdidas producidas por la no calidad, mantiene a los clientes e incrementa los beneficios y fundamentalmente es el sello distintivo en el mundo.

En Cuba se vienen desarrollando soluciones informáticas para el beneficio de toda la sociedad y para la exportación a otros países, principalmente en Latinoamérica, de ahí la importancia de concretar esfuerzos e ideas conjuntas entre centros investigativos, empresas y universidades, para la estandarización y mejora de los procesos de ingeniería de software y la creación de sistemas.

La Universidad de las Ciencias Informáticas (UCI), es el centro donde se desarrolla la mayor cantidad de software de Cuba, esto determina la necesidad de adaptar métricas, normas y estándares internacionales de Ingeniería de Software, para lograr que sus productos cumplan con los requisitos deseados por el cliente y las reglas de calidad establecidas, es de vital importancia que profesores y estudiantes responsables de asegurar la calidad en los proyectos, tenga en conocimiento necesario para lograr resultados satisfactorios.

La universidad se encuentra estructurada por varios centros de desarrollo, uno de estos es GEYSED, el cual pertenece a la facultad 6. En uno de sus departamentos, en este caso, Geoinformática, se cuenta con una plataforma que lleva por nombre GeneSIG, donde se utiliza como apoyo al desarrollo de aplicaciones



de Sistemas de Información Geográfica (SIG) en entornos web con tecnologías libres, el cual tiene integrado el catálogo de mapas LiberMaps.

El mismo maneja la seguridad sobre datos y funcionalidades, permite realizar la edición dinámica de los objetos de un mapa y poseer una estructura organizativa para estos, pero no cuenta con una documentación organizada de los datos contenidos en ellos. La solución a este problema la constituyen los metadatos geográficos que no es más que una descripción de datos geoespaciales específicos, que permite distinguirlos a través de aspectos como son: calidad, actualización, autor, entre otros.

Surge la necesidad de implantar un sistema informático que permita gestionar los metadatos geográficos de LiberMaps, y a su vez nace el proyecto SyGMe (Sistema de Gestión y Publicación de Metadatos Geográficos), que se encuentra en estos momentos en desarrollo, pues no solo se necesita que la información esté documentada y cumpla con las funcionalidades, sino que todo el proceso de gestión de metadatos geográficos se realice con la máxima calidad.

En estos momentos la calidad del proyecto se realiza de forma empírica, no existen planes que garanticen el aseguramiento de la misma, ni cronogramas de pruebas o validación de artefactos que se generan. De aquí que el **problema científico** sea ¿Cómo lograr que el proceso de desarrollo del proyecto SyGMe cumpla con los requisitos de calidad?

Por lo antes expuesto el **objeto de estudio** de este trabajo es el proceso de aseguramiento de la calidad del software y se define como **campo de acción** el proceso de aseguramiento de la calidad en el proyecto SyGMe.

Como **objetivo general** se plantea definir una estrategia para el aseguramiento de la calidad en el proyecto SyGMe.

Para dar solución al problema y cumplir con el objetivo trazado se proponen las siguientes **tareas de la investigación**:

- Sintetizar las principales características de los modelos y estándares que existen internacionalmente para el proceso de aseguramiento de la calidad de un software.
- Caracterizar las métricas que se utilizan en el proceso de evaluación del software.

- Realizar un plan de Aseguramiento de la Calidad.
- Realizar una propuesta de entrenamiento para el equipo de desarrollo del proyecto SyGMe.
- Organizar correctamente el trabajo de los recursos humanos del proyecto.
- Validar la estrategia propuesta.

Como **idea a defenderse** plantea que: con la definición correcta de una estrategia para el aseguramiento de la calidad de software se logrará el cumplimiento de las metas trazadas en el proyecto, así como la entrega del proyecto SyGMe en tiempo y con los requisitos de calidad deseados.

Los métodos científicos utilizados para el desarrollo de la investigación:

### Métodos Teóricos

**Analítico-Sintético:** permite identificar toda la información que aborda el tema estudiado y sintetizar los elementos más importantes y de mayor utilidad para esta investigación.

**Histórico-Lógico:** se utiliza con el fin de recopilar la información que se posee hasta el momento del fenómeno analizado, permitiendo comprender la historia, la evolución y la lógica del mismo.

### Métodos Empíricos

**Observación:** se utiliza durante todo el proceso de investigación, para identificar todo lo relacionado con el dominio del problema.

En la elaboración de la solución que será propuesta de este trabajo, se empleará el método Inductivo-Deductivo, este permitirá arribar a proposiciones generales a partir de estudio de hechos aislados e inferir casos particulares a través de un razonamiento lógico.

El presente trabajo consta de 3 capítulos:

**Capítulo 1. Fundamentación Teórica:** se realiza una investigación sobre los metadatos y se especifica sobre el término calidad, cómo se puede asegurar y cuáles son las normas y estándares de calidad que existen y sus mejores prácticas.

**Capítulo 2. Propuesta y descripción de la solución:** se realiza la propuesta donde se define la estrategia de aseguramiento de la calidad del proyecto SyGMe.

**Capítulo 3: Validación de la propuesta:** se evalúa la estrategia de aseguramiento de la calidad del proyecto SyGMe mediante los resultados obtenidos en la aplicación de la misma.

## Capítulo 1: Fundamentación Teórica

### 1.1 Introducción al capítulo

La calidad de software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente (Pressman, 2002).

En este capítulo se expondrá aspectos sobre los metadatos, calidad, calidad de software, calidad en Cuba y la UCI, y se abordará temas que sirvan de base teórica para la investigación, los cuales están presentes en el desarrollo de este trabajo.

### 1.2 Calidad

El término puede ser ambiguo e incluso subjetivo porque, como la belleza, la calidad depende de quien la observa. Es necesario definir el concepto con claridad, ya que si la calidad no puede ser definida, no puede ser medida; y donde la calidad no puede ser medida entonces no puede ser controlada. A continuación se ofrecen algunas definiciones utilizadas en la actualidad.

Calidad según la Norma ISO 9000-2000 es cumplir con las necesidades y expectativas del cliente, lo que conlleva medir la satisfacción del cliente. Grado en que un conjunto de características inherentes cumple con unos requisitos.

La calidad como resultado de la interacción de dos dimensiones: dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece). (SCALONE, 2006)

Calidad es la menor pérdida posible para la sociedad. (Miranda, et al., 2006)

La calidad es "La adecuación para el uso satisfaciendo las necesidades del cliente. (SCALONE, 2006)

Desarrollar, diseñar, manufacturar y mantener un producto de calidad que sea el más económico, el útil y siempre satisfactorio para el consumidor. (Miranda, et al., 2006)

Philip Crosby desarrolló un concepto denominado los "Absolutos de la calidad total", cuyos principios son:

- La calidad se define como cumplimiento de requisitos.
- El sistema de calidad es la prevención.
- El estándar de realización es cero defectos.
- La medida de la calidad es el precio del incumplimiento.

Por lo antes expuesto se puede concluir que la calidad es un proceso o servicio que cuenta con las habilidades explícitas o implícitas, que debe cumplir un producto para satisfacer las necesidades de un cliente.

### **1.2.1 Calidad de Software**

La calidad de software es un término muy utilizado en la industria del software, debido a esto, muchos autores lo han descrito de varias maneras:

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.(Vega Lebrún, et al., 2008)

La Calidad del Software es el conjunto de cualidades que la caracterizan y que determinan su utilidad y existencia, plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad.(SCALONE, 2006)

“La calidad del software se refiere a: “Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización”.(Pressman, 2002)

Luego de analizar las diferentes definiciones se puede afirmar que la calidad del software es impuesta por los requisitos con que debe cumplir un producto o servicio, además se enfatiza que debe existir una metodología, que sirva de guía para el desarrollo de software, para que exista una correlación con los criterios de desarrollo definidos, para que el producto tenga la calidad requerida.

### 1.2.2 Calidad a nivel de Proceso

Las empresas pueden implementar la calidad en una doble vertiente: a nivel proceso y a nivel producto, teniendo en cuenta la estrategia y los objetivos de la misma. Se puede pensar en poder aplicar y/o integrar modelos o estándares diferentes consiguiendo así objetivos mucho más globales, teniendo en cuenta al mismo tiempo las adopciones de CMMI e ISO 9001:2000.

El proceso software objeto de mejora según el *Software Engineering Institute*<sup>4</sup> es el conjunto de actividades, métodos, prácticas y transformaciones que la persona usa para desarrollar y mantener software y los productos de trabajo asociados (planes de proyecto, diseño de documentos, código, pruebas y manuales de usuario). Ahora bien, según establece la norma ISO/IEC 15504, es el proceso o conjunto de procesos usados por una organización o proyecto, para planificar, gestionar, ejecutar, monitorizar, controlar y mejorar sus actividades software relacionadas.

El objetivo final del modelo de procesos es lograr una representación clara de los procesos reales de desarrollo, con la cual poder trabajar para planificar las mejoras a incluir en cada uno de esos procesos. Si la representación conceptual del proceso es buena, los análisis de estos procesos sobre el papel permitirán a la empresa la posibilidad de automatizarlos, controlar su eficiencia, comprobar las interacciones con otros procesos, y ofrecer a la Dirección una nueva fuente de información, como puede ser la información actual del estado de cada proceso en cualquier momento, el significado que debe darse a cada uno de los puntos de decisión, etc.

La mejora del producto final pasa, según estos modelos, por la mejora de los procesos que llevan a su creación. La adopción del modelo o metodología adecuados podrá realizar esta mejora con una correcta implantación, dotando implícitamente al producto final de una calidad manifiesta. Entre los modelos o estándares a nivel de proceso se pueden mencionar muchos, pero los más utilizados son: ISO 9001:2000 (ISO 90003:2004), ISO/IEC 15504, CMMI, TickIT, ISO 20000, Bootstrap, SwTQM (basado en EFQM y CMMI), etc.(SCALONE, 2006).

### 1.2.3 Calidad a nivel de producto

Dentro del conjunto de modelos y metodologías de mejora de la calidad, existen un conjunto de modelos que sitúan el foco de la mejora no en el producto final obtenido, sino aplicando un modelo de pruebas en el ciclo de vida del software sobre los productos obtenidos.

La estrategia en este caso se basa en el establecimiento de varios puntos de testeo y verificación sobre todas las etapas del ciclo de vida (requisitos, análisis, diseño y construcción), además de la última etapa de pruebas finales. Las técnicas a emplear suelen ser las siguientes:

- ✓ Detección temprana de errores mediante la anticipación del testing.
- ✓ Utilización de técnicas especializadas, como los test estáticos, dinámicos, pruebas unitarias, pruebas de caja negra y caja blanca, etc.
- ✓ Incorporación a los equipos de trabajo de grupos especializados en testing.

Estas aproximaciones se engloban en modelos especializados, como puede ser el caso de *Testing Process Improvement (TPI)*, desarrollado por Sogeti, y certificable bajo la evaluación TMAP.

Entre los modelos o estándares a nivel de producto se pueden mencionar TPI/TMAP, el Modelo de Boehm, el Modelo de Gilb, el Modelo de Dromey, la norma ISO 9126-1, el Modelo de McCall, WebQEM, ISO 25000, Portal Quality Model (PQM) y otros. Mientras que los modelos a nivel de proceso son independientes de la tecnología, los modelos y estándares a nivel de producto surgen o se actualizan de acuerdo a la evolución tecnológica. (Informática, 2000)

### 1.3 Calidad de software a nivel internacional, en Cuba y en la UCI

La certificación de la calidad en el desarrollo de software ha adquirido una gran importancia en el mundo durante los últimos años, surgen modelos e iniciativas que tratan de ofrecer a las organizaciones nuevas metodologías buscando la mejora en la calidad y la excelencia en sus productos. Muchos son los modelos que se han definido en el mercado para la obtención de este fenómeno en sus muchas acepciones.

En la actualidad la región europea y América del Norte son los primeros en el mundo, en la disponibilidad, uso y desarrollo en las nuevas tecnologías, la zona asiática apunta a un ligero incremento en este sentido,

pero debido a su alta población los indicadores muestran que aun no son suficientes los resultados obtenidos, la India ha venido emergiendo como una gran potencia en la producción en los últimos tiempos.

Para el desarrollo de cualquier software siempre está presente la calidad del mismo, debido a esto el aseguramiento y control de este proceso nace como una evolución natural, la cual resulta limitada y poco eficaz para prevenir la aparición de defectos.

La industria mundial de desarrollo de software en los últimos años se ha preocupado por mejorar sus capacidades en función de la calidad. Para poder distinguirse en el mercado las empresas están invirtiendo en la mejora de los procesos de desarrollo. Las experiencias exitosas de la Ingeniería de Software han propiciado que se definan varios modelos que sirvan de guía para las mejoras y unifican los criterios de evaluación de las empresas. Las normas ISO serie 9000, el modelo estadounidense conocido como CMM (Capability Maturity Model), el BOOTSTRAP (Estándar Europeo para Evaluación y Mejoras de Procesos de Desarrollo de Software) y la norma ISO 15504, conocida como SPICE, (Software Process Improvement and Capability Determination) son los ejemplos más reconocidos de estos modelos.

En el ámbito de este auge productivo de sistemas informáticos, la calidad se ha convertido en un reto, en Cuba también es una tarea de gran prioridad debido a la perspectiva económica que posee, ya que garantiza un grupo importante de actividades del país.

En el país se vienen desarrollando algunos avances sobre el desarrollo de técnicas para lograr la calidad del software. Desde la década de los 90, la Empresa de Producción y Desarrollo de Software y Calidad (SOFCAL) desarrolló un sistema de documentación y de control de calidad nacional. Existen otras empresas como son Desoft y Softel que tienen sus propios sistemas para garantizar la calidad de sus productos, aunque no poseen una estrategia de aseguramiento de la calidad definida en común, trayendo consigo que los procesos sean inestables e inmaduros.

La Universidad de las Ciencias Informáticas (UCI), tiene entre sus principales objetivos la vinculación estudio-trabajo, lo cual ha fomentado el desarrollo y aplicación de modelos para el desarrollo de la calidad de software en los productos realizados en el centro. El Centro para la Excelencia en el Desarrollo de Proyectos Tecnológicos (CALISOFT) es una organización enfocada a contribuir al desarrollo de la industria cubana de software facilitando la implementación de las mejores prácticas en el proceso de



desarrollo y/o mantenimiento de software. Es la entidad responsable de la evaluación y certificación de productos, procesos y organizaciones según normas nacionales e internacionales y de la asesoría, adiestramiento y formación continua de los especialistas en el país en los temas de calidad e ingeniería de software.

Sus principales funciones son:

- Definir, implantar y monitorear el Proceso de Medición y Análisis aplicando a los Proyectos de Producción de Software en correspondencia con la madurez del proceso productivo de la UCI.
- Lograr que el 100% de los Proyectos de Software desarrollados en la Universidad, realicen mediciones durante su proceso de desarrollo, registren las medidas y definan indicadores que controlen la planeación, seguimiento, control y calidad de sus Procesos y Productos.
- Capacitar al personal vinculado a la Producción sobre la aplicación en sus áreas del Proceso de Medición y Análisis.
- Promover el interés y hacer notar la necesidad de un proceso de Medición en sus Proyectos Productivos.

Este grupo brinda diferentes servicios a los Proyectos Productivos de la Universidad, entre los que se encuentran:

- Asesoría en los temas de Medición y Análisis al personal involucrado en dicho proceso.
- Definir y asesorar a los Proyectos Productivos en la utilización de los Métodos de Estimación ajustados a partir de las medidas recopiladas y según su ciclo de vida.

Desde 2009, la universidad se encuentra inmersa en el Programa de Mejora de Proceso, después de 2 años de arduo trabajo, se espera lograr aprobar satisfactoriamente es SCAMPI (método de evaluación del modelo de mejora de proceso CMMI) logrará alcanzar el nivel 2 de madurez del modelo de mejora de proceso CMMI y se convertirá en la primera empresa cubana en alcanzar dicha certificación. CMMI es un modelo de mejora de procesos que agrupa un conjunto de buenas prácticas que abarcan el ciclo de vida de un producto desde la concepción hasta la entrega y mantenimiento. Su propósito es ayudar a las organizaciones a mejorar sus procesos de desarrollo y el mantenimiento de sus productos.

En el nivel 2 de madurez se espera que los procesos de la organización sean definidos y administrados los recursos sean los adecuados, las responsabilidades sean asignadas, la dirección tenga visibilidad objetiva de los procesos y que se definan las actividades para la administración de los proyectos.

### **1.4 Gestión de la calidad**

Según la ISO QS/900 la Gestión de la Calidad, es el conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implantan por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad.

De gestión de la calidad se considera importante hacer referencia al concepto expresado por la ISO:2000 y que es el siguiente: “actividades coordinadas para dirigir y controlar una organización en lo relativo a la calidad”.(Vega Lebrún, et al., 2008)

La gestión de la calidad opera a todo lo largo del sistema de la calidad. Estos conceptos se pueden extender a todas las partes de una organización.

Conceptualmente, la gestión de la calidad es el conjunto de actividades de la función general de la dirección que determinan la política de la calidad, los objetivos y las responsabilidades y se lleva a cabo, tal como ya ha sido mencionado, por medios tales como la planificación de la calidad, la inspección, el control de la calidad, el aseguramiento de la calidad y el mejoramiento de la calidad, en el marco del sistema de la calidad.

La gestión de la calidad es responsabilidad de todos los niveles de dirección, pero debe ser conducida por el más alto nivel de la dirección. Su implementación involucra a todos los miembros de la organización y toma en cuenta los aspectos económicos. Como signo distintivo debe garantizar la participación activa y consciente de todo el personal.

Se concluye que la gestión de la calidad contribuye a organizar todo el trabajo durante el proceso de desarrollo de un producto con calidad, por lo cual no se debe descuidar ninguno de sus aspectos.

### 1.4.1 Planificación de la calidad

La planificación de la calidad es la parte de la gestión de la calidad enfocada al establecimiento de los objetivos de la calidad y a la especificación de los procesos operativos necesarios y de los recursos relacionados para cumplir los objetivos de calidad.(ISO, 2001)

Esta se encuentra asociada a las actividades que establecen los objetivos y los requisitos para la calidad, así como los requisitos para la aplicación de los elementos del sistema de la calidad y abarca:

- La planificación del producto: identificación, clasificación y ponderación de las características relativas a la calidad, así como el establecimiento de los objetivos, de los requisitos para la calidad y de las restricciones.
- La planificación administrativa y operativa: preparación de la aplicación del sistema de la calidad, incluida la organización y la programación.
- La preparación de planes de la calidad y el establecimiento de disposiciones para el mejoramiento de la calidad.

Este proceso se estructura en seis pasos:

1. Verificación del objetivo.
2. Identificación de los clientes.
3. Determinación de las necesidades de los clientes.
4. Desarrollo del producto. (bienes y servicios).
5. Desarrollo del proceso.
6. Transferencia a las operaciones diarias.

Los factores que determinan el Modelo o Estándar de Calidad de Software a elegir son:

- La complejidad del proceso de diseño.
- La madurez del diseño.
- La complejidad del proceso de producción.
- Las características del producto o servicio.
- La seguridad del producto o servicio.

- Económico.

El objetivo es modelar y remodelar los negocios y productos de la empresa, de manera que se combinen para producir un desarrollo y utilidades satisfactorias. Los aspectos a considerar en la Planificación de la Calidad de Software son: modelos/estándares de calidad de software a utilizar, costos de la calidad de software, recursos humanos y materiales necesarios.

### **1.4.2 Control de la calidad**

El Control de la Calidad del Software son las técnicas y/o actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos fundamentales:

- Mantener bajo control un proceso.
- Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.

Está formado por actividades que permiten evaluar la calidad de los productos de software desarrollados. El aspecto a considerar en el Control de la Calidad del Software es la “Prueba del Software”. (MINGUET MELIÁN, 2003)

Según Pressman el control de calidad es una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados. Este incluye un bucle de realimentación del proceso que creó el producto. La combinación de medición y realimentación permite afinar el proceso cuando los productos de trabajo creados fallan al cumplir sus especificaciones. Este enfoque ve el control de calidad como parte del proceso de fabricación.

Se llama prueba de software, al proceso en el que se ejecuta un sistema con el objetivo de detectar fallos. La prueba exitosa es aquella que descubre defectos.

Una estrategia tradicional de prueba del software debe incluir pruebas que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas que validen las principales funciones del sistema frente a los requisitos del cliente.

Cuando se construye un software a medida para un cliente, se llevan a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. Estas las realiza el usuario final en lugar del responsable del desarrollo de sistema.

Cualquier producto de ingeniería puede probarse de dos formas: prueba de caja negra y prueba de caja blanca.

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Para este tipo de prueba, se deben definir todos los caminos lógicos y desarrollar casos de prueba que ejerciten la lógica del programa.

De manera general todas estas son actividades a implementar para evaluar la calidad de los productos desarrollados.

### **1.4.3 Aseguramiento de la calidad**

La función de aseguramiento de la calidad tiene como finalidad primaria el determinar si las necesidades de los usuarios están siendo satisfechas adecuadamente. El aseguramiento de la calidad está basado en la aplicación de un sistema documental del trabajo, en el que se establecen las reglas claras, fijas y objetivas, se incluyen todos los aspectos relacionados al proceso operativo. Este proceso comienza por el diseño, seguido de la planeación, producción, presentación, distribución de las técnicas estadísticas de control del proceso y la capacitación del personal.

La complejidad del sistema de aseguramiento de calidad radica en que a lo largo de todo el proceso operativo se puede tener un estricto control sobre la correcta aplicación de las reglas y especificaciones técnicas establecidas, los métodos y las filosofías de calidad. Este control sobre el proceso operativo permite evaluar el desempeño del trabajo por medio de la obtención de datos confiables.

No existe aseguramiento de la calidad sin el dominio de la misma, los beneficios que se pueden obtener como resultado de aplicar los procesos de esta actividad son muchos y variados, algunos de estos son:

- Se detectan problemas rápidamente es posible identificar problemas en tempranas etapas del desarrollo de productos de software, ayudando al desarrollador a corregirlos inmediatamente y poder avanzar con más rapidez.
- Se crean y se siguen estándares de trabajo, con apoyo del proceso de aseguramiento de calidad, se pueden establecer estándares tan diversos como son los de codificación o de documentación, los cuales apoyan a uniformizar y consolidar el proceso de desarrollo.
- Se verifica que los objetivos individuales vayan acordes con los objetivos de la organización, Se busca y se recomienda que los requerimientos expuestos por usuarios finales estén alineados con los objetivos globales de la empresa, facilitando así el logro de los mismos y la integración total de los usuarios a la organización.
- Se recomiendan métodos para realizar el trabajo, Las prácticas de aseguramiento de calidad, como son muy robustas ya que aplican técnicas muy completas de medición, pueden proponer en un momento dado qué métodos se ajustan más a la naturaleza del producto a ser desarrollado, teniendo como efecto final que el producto tenga más posibilidades de ser un producto con calidad.
- Se evita incurrir en costos innecesarios, Como un efecto generalizado de algunos de los puntos mencionados con anterioridad, la práctica de procesos de aseguramiento de calidad lleva a las organizaciones a evitar costos no deseados como pueden ser todos aquellos ocasionados por mantenimiento correctivo.
- Se planea la calidad, está claro que el concepto de calidad no es algo que se da de una manera automática e impredeciblemente. Es algo que se busca. Por lo mismo, se debe de planear, construir e implantar en el producto.

### **1.4.4 Mejora de la calidad**

La mejora de la calidad del software contribuye, por medio de las mediciones, a los análisis de los datos y auditorías, a efectuar mejoras en la calidad del software.

Según la Norma ISO 9000:2000, la mejora de la calidad es la parte de la gestión de la calidad orientada a aumentar la capacidad de cumplir con los requisitos de la calidad. Los requisitos pueden estar relacionados con cualquier aspecto tal como la eficacia, la eficiencia o la trazabilidad.

Una auditoria de calidad tiene como objetivo mostrar la situación real para aportar confianza y destacar las áreas que pueden afectar adversamente esa confianza. Otro objetivo consiste en suministrar una evaluación objetiva de los productos y procesos para corroborar la conformidad con los estándares, las guías, las especificaciones y los procedimientos.

Las razones para realizar una auditoría son:

- Establecer el estado del proyecto.
- Verificar la capacidad de realizar o continuar un trabajo específico.
- Verificar qué elementos aplicables del programa o plan de aseguramiento de la calidad han sido desarrollados y documentados.
- Verificar la adherencia de esos elementos con el programa o plan de aseguramiento de la calidad.

### **1.5 Modelos y Estándares de calidad**

Un reto para la industria del software es desarrollar las estrategias que permitan un posicionamiento y un reconocimiento internacional con productos competitivos de exportación, lo que requiere de la elección e implantación del modelo o estándar de calidad indicado. La búsqueda de reconocimiento internacional de calidad, permitirá enfrentar los mercados con mayores posibilidades de éxito y abrirá las puertas a que otras empresas en el país, adquieran el compromiso hacia la incorporación de dichos modelos y estándares de calidad de software.

Los modelos de calidad son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad.

Los estándares de calidad son aquellos que permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad.

La ventaja de los Modelos y/o Estándares de Calidad es que la calidad se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar. Estos ayudan también a

comprender las relaciones que existen entre las diferentes características de un producto de software, una desventaja es que aún no ha sido demostrada la validez absoluta de ninguno de estos Modelos o Estándares. Las conexiones que se establecen entre características, atributos y métricas se derivan de la experiencia. Esto originó que existan múltiples Modelos y Estándares de Calidad.

Nivel de Calidad	Modelo de Calidad del Software	Estándares de la Calidad del Software
Procesos	CMMI	ISO 90003
	TickIT	ISO 12207
	Bootstrap	ISO 15504 (SPICE)
	Personal SW Process (PSP)	IEEE / EIA 12207
	Team SW Process (TSP)	ISO 20000
	Practical SW Measurement(PSM)	ITIL
	Six Sigma for Software	Cobit 4.0

Tabla 1: Modelos/Estándares de calidad del software a nivel de proceso.(Zibert van Gricken, 2004)

### 1.5.1 Estándar ISO 9000/ 9000-3

Las normas son reglas metodológicas y modelos adaptados por organizaciones nacionales e internacionales que se dedican al establecimiento de estándares. Estas organizaciones se apoyan en foros de expertos que se suelen constituir en comités muy calificados. La Organización Internacional para la Estandarización (ISO), propone gran cantidad de normativas en numerosos campos tecnológicos.

La norma ISO 9000 define un sistema de calidad como: “La estructura de organización, de responsabilidades, de actividades, de recursos y de procedimientos que se establecen para llevar a cabo la gestión de la calidad”.(ISO, 2004)



El sistema de calidad debe ser concordante con los objetivos de calidad de la empresa, definidos en la política de calidad de la misma. La alta dirección debe expresar dicha política, planificarla estratégicamente, asignar recursos y elegir los planes y evaluaciones sistemáticos, por tanto la alta dirección debe iniciar, desarrollar, implementar y actualizar periódicamente el sistema de calidad.

Los elementos escritos son los documentos en los que se describe el sistema, los procedimientos, etc. Los principales documentos son el Manual de Calidad, los Procedimientos y los Registros de datos sobre calidad.

La familia de la Norma 9000 es un conjunto de documentos sobre las buenas prácticas de los sistemas de gestión de la calidad. Al igual que estándares ya comentado hace hincapié en que requisitos debe cumplir el sistema de calidad de una empresa sin establecer en cómo deben cumplirse.

En este trabajo se basará principalmente en el estudio de la ISO 9000-3:2004, la cual provee una guía para las organizaciones respecto de la aplicación de ISO/IEC 9001:2000 en la adquisición, suministro, desarrollo, operación y mantenimiento de software y servicios de soporte. Esta norma no agrega o cambia los requerimientos de ISO/IEC 9001:2000. Las guías de ISO 90003:2004 no tienen el propósito de ser utilizadas como criterio de evaluación en una certificación de SGC (Sistema de Gestión de la Calidad).

La aplicación de ISO 90003:2004 es apropiada para un software que:

- Forma parte de un contrato comercial con otra organización.
- Es un producto disponible para un sector del mercado.
- Es usado para soportar los procesos de una organización.
- Está relacionado a servicios de software.

La Norma cuenta con 5 capítulos que especifican actividades que deben ser consideradas cuando se implemente el SGC. Los capítulos son:

- Sistema de Gestión de la Calidad.
- Responsabilidad de la Dirección.
- Gestión de los Recursos.
- Realización del Producto.

- Medida, Análisis y Mejora.

Algunos de los beneficios que se obtienen con la certificación de la Norma ISO 9000-3 son:

- Mejor documentación de los sistemas.
- Cambio cultural positivo.
- Incremento en la eficiencia y la productividad.
- Mayor percepción de la calidad.
- Se amplía la satisfacción del cliente.
- Se reducen las auditorías de calidad.
- Agiliza el tiempo de desarrollo de un sistema.

La norma ISO 9000-3 consta de varias secciones, algunas de ellas son:

- Administración de la responsabilidad.
- Sistemas de calidad.
- Auditorías internas de calidad.
- Revisión de contratos.
- Control de documentos y datos.
- Planificación del diseño y el desarrollo.
- Inspección y pruebas.
- Control, calibración y mantenimiento de los equipos de inspección, medición y pruebas utilizados por la empresa.
- Control del Producto no conforme.
- Acciones correctivas y preventivas.
- Control de los registros de calidad.
- Capacitación.
- Estadísticas.

La ISO 9000-3 proporciona una guía útil que sirve para detectar y corregir una serie de problemas de los productos de software, consiguiendo tras su aplicación una mejora en la calidad de los mismos.

### 1.5.2 Modelo de Madurez de Capacidad Integrado (CMMI)

El modelo CMMI tiene el propósito de proporcionar una guía unificada para la mejora de múltiples disciplinas tales como Ingeniería del Software, Ingeniería de Sistemas, Desarrollo Integrado de Producto y Procesos, y la Fuente Proveedora, entre otras. Actualmente los documentos que forman el modelo CMMI son:

- CMMI for development, versión 1.2.: es la continuación y mejora de modelo CMMI v.1.1 y ha sido realizado gracias al “concepto de constelaciones CMMI”. Estos son un conjunto de componentes básicos que pueden ser aumentados por material adicional para proveer modelos de aplicación específicas con un alto contenido común. Con esta nueva descripción, CMMI-DEV no es más que el modelo CMMI para el desarrollo, que anteriormente se especificaba por completo en el modelo CMMI v.1.1.
- CMMI Acquisition Module v.1.1.: define una serie de prácticas (a añadir al modelo básico de CMMI) con respecto a proyectos de adquisición. Estas “mejores prácticas” se enfocan en las actividades llevadas a cabo por profesionales en el marco de la oficina de programas de adquisición.

El modelo CMMI se puede representar de dos formas, el continuo y el escalonado. La representación continua consta de cinco niveles (comenzando por nivel 0), el cual indica el nivel de capacidad. La representación escalonada cuenta de cinco niveles que representa la madurez organizacional, ambas son equivalentes.(CMMI, 2006)

El modelo cuenta con áreas de procesos, las cuáles se clasifican en categorías:

- Ingeniería.
- Gestión de Proyecto.
- Gestión de Proceso.
- Soporte.

La ingeniería da soporte a las actividades del ciclo de vida de desarrollo del producto, desde el desarrollo inicial de requisitos a la transición al uso operacional.

Área de Procesos	Descripción
Desarrollo de Requisitos(RD)	Recopila y armoniza las necesidades de los participantes y las traducen en requisitos del producto.
Gestión de Requisitos(RM)	Asegura que los requisitos acordados son comprendidos y gestionados.
Solución Técnica(TS)	Convierte requisitos en arquitectura del producto, diseño y desarrollo.
Integración del Producto(SI)	Combina los componentes de producto y asegura las interfaces.
Verificación (VER)	Asegura que el producto cumpla con las especificaciones.
Validación(VAL)	Asegura que el producto cumple con el uso propuesto cuando se sitúa en el entorno propuesto.

**Tabla 2:** Áreas de procesos de Ingeniería (Elaboración propia).

La gestión de proyecto cubre las actividades relacionadas con la planificación, seguimiento y control del proyecto, proporciona mecanismos para establecer, mantener, monitorizar acuerdos con clientes y proveedores. Facilita mecanismos para establecer y mantener un entorno de colaboración entre equipos, Suministra un método común para gestionar el proyecto cuantitativamente y anticipándose al los problemas.

Área de Procesos	Descripción
Planificación de Proyecto(PP)	Desarrolla y mantiene el plan de proyecto, implica a los participantes y obtiene compromiso con el plan.
Seguimiento y control de Proyecto(PMC)	Monitoriza las actividades y toma acciones correctivas incluyendo re-planificación.
Gestión Integrada de Proyecto(IPM)	Adapta los procesos organizativos al proyecto, y establece la visión compartida del proyecto.

Desarrollo de Equipo Integrado(IT)	Identifica y organiza a los participantes en los equipos colaborativos y desarrolla la visión compartida alineada con la visión compartida del proyecto y la organización.
Gestión de Riesgos(RSKM)	Desarrolla e implementa una estrategia proactiva para identificar y evaluar, priorizar y manejar riesgos del programa.
Gestión de Proyectos Cuantitativa(QPM)	Recopila métricas de producto y proyecto, y analiza los resultados para identificar oportunidades de mejora.
Gestión de Acuerdos con Proveedores(SAM)	Gestiona la adquisición de productos de proveedores para los que existe un acuerdo formal.

**Tabla 3:** Áreas de procesos de Gestión de Proyecto (Elaboración propia).

La gestión de procesos contiene las prácticas relacionadas con la implementación de un programa de mejora de procesos. Proporciona la capacidad para documentar y compartir las mejores prácticas, los activos de proceso y aprendizaje, suministra capacidad de conseguir objetivos cuantitativos de calidad y rendimiento del proceso.

Área de Procesos	Descripción
Enfoque en el Proceso Organizativo(OPF)	Ayudar a la organización a establecer y mantener la comprensión de sus procesos e identificar, planificar, coordinar e implementar la mejora.
Definición del Proceso Organizativo(OPD)	Establece y mantiene el conjunto de procesos organizativos estándares y los activos de soporte.
Formación Organizativa(OT)	Identifica las actividades formativas estratégicas y tácticas entre los proyectos y grupos de soporte.
Rendimiento del Proceso Organizativo(OPP)	Deriva objetivos cuantitativos de calidad y rendimiento de proceso de los objetivos de negocio de la organización.
Innovación y Despliegue	Selecciona y despliega mejoras incrementales para mejorar la habilidad

Organizativo(OID)	de la organización para alcanzar los objetivos de calidad y rendimiento del proceso.
-------------------	--

**Tabla 4:** Áreas de procesos de Gestión de Proceso (Elaboración propia).

El soporte proporciona los procesos esenciales para soportar el desarrollo y mantenimiento del producto, soporta el establecimiento y mantenimiento de un entorno de trabajo que estimula la integración y gestiona al personal para permitir y premiar comportamientos integradores. Facilita funciones de soporte usadas para todas las áreas de procesos durante el desarrollo del producto.

Área de Procesos	Descripción
Medición y Análisis(MA)	Establece un programa de métricas para proveer resultados objetivos que puedan ser usados para tomar decisiones informadas y tomar acciones correctivas.
Gestión de Configuración(CM)	Establece y mantiene la integridad de los productos de trabajo.
Aseguramiento de la Calidad de Proceso y Producto(PPQA)	Proporciona prácticas para evaluar objetivamente procesos, productos y servicios.
Análisis y Resolución de Decisiones(DAR)	Proporciona un proceso estructurado de toma de decisiones que asegura que las alternativas se comparan con criterios objetivos, y se elige la mejor alternativa.
Análisis y Resolución Causal (CAR)	Identifica las causas de defectos y otros problemas, y toma acciones para prevenir que ocurran en el futuro.
Entorno Organizativo para la Integración(OEI)	Establece en entorno para la implementación de equipos integrados.

**Tabla 5:** Áreas de procesos de Soporte (Elaboración propia).

### 1.5.3 Otros modelos y estándares de calidad

#### TickIT

La BSI (British Standards Institution) es la asociación británica miembro de ISO, la cual elaboró una guía llamada TickIT para aplicar la norma ISO 9001 e ISO 9000-3 en el software. Esta guía ya va por la versión 5.0, la cual considera las normas ISO/IEC 12207 y la ISO 9000-3. BSI es el mayor cuerpo de certificación independiente del mundo. BSI ha dado el servicio de certificación a empresas de software desde 1979.()

Los objetivos principales de TickIT son, además de desarrollar un sistema de certificación aceptable en el mercado, estimular a los desarrolladores de software a implementar sistemas de calidad, dando la dirección y guías necesarias para tal efecto. El objetivo de la certificación es demostrar que las prácticas necesarias para asegurar la calidad durante el desarrollo de software existen y son verificables. En general, el modelo permite certificar cualquier tipo de proyecto a través de una estructura más flexible.

TickIT tiene como meta principal estimular a los desarrolladores de sistemas de software a pensar acerca de:

- Qué calidad existe en el contexto de los procesos de desarrollo de software.
- Cómo puede ser lograda la calidad.
- Cómo los sistemas de información de la calidad pueden ser mejorados en forma continua.

La guía de TickIT provee el enlace necesario para que la conformación (o no) del sistema auditado respecto al modelo TickIT, pueda ser expresada en función de los criterios de la ISO 9001, logrando así la aplicación de esta última al desarrollo de software. Los requerimientos de experiencia y conocimientos que se piden a los auditores hacen posible la aplicación del modelo, suponiendo que la experiencia y conocimientos de los auditores no se vean obsoletos frente a prácticas y técnicas nuevas dentro del medio de desarrollo de software.

### **Modelo Bootstrap**

Este proyecto fue creado por la Comisión Europea como parte del programa ESPRIT(ESPRIT 5441 BOOTSTRAP: A European Assessment Method to Improve Software Development). La administración y el mantenimiento del programa Bootstrap corresponden al Grupo Europeo de Interés Económico del Instituto Bootstrap (Bootstrap Institute European Economic Interest Group, BI EEIG) de Milán, Italia. El interés principal del programa Bootstrap es evaluar y mejorar la capacidad de las Unidades Productoras de Software (SPU, Software Producing Units).(Bicego, et al., June 1995.)

Mediante esta metodología se tratará la mejora de procesos de software. ISO/IEC TR15504 define un proceso como un grupo de actividades interrelacionadas donde una entrada se transforma en una salida. Se podría decir que la mejora de procesos es en parte mejor que la reingeniería. Esta metodología mediante prácticas, herramientas y estándares de calidad internacional; mide, evalúa y propone mejoras al proceso de desarrollo de SW que siguen las Unidades de Producción de Software (UPS) de las empresas.

El programa Bootstrap combina las normas ISO 9000, las normas europeas para la Ingeniería de Software y el Modelo de Madurez de la Capacidad CMM para sentar una base con la cual evaluar y dar asesoría. La metodología Bootstrap engloba tanto la evaluación para establecer el diagnóstico de un proceso para desarrollo de software, el cual incluye la organización, los métodos y la capacidad de ingeniería, las herramientas y la tecnología, como la creación de un plan de acción que defina los pasos, los detalles de la implantación y los marcos temporales para que la organización aumente su capacidad de entrega de productos y servicios de calidad. El resultado de la evaluación es un perfil basado en el instrumento de evaluación de Bootstrap que añade una segunda dimensión a los niveles del CMM: el atributo de la calidad del proceso. Se pretende que mediante el programa Bootstrap se identifiquen los atributos de un proyecto de una organización que desarrolle software y que se asignen todas las preguntas del cuestionario a los atributos de la calidad del proceso así como a los niveles de madurez.

### **ISO/IEC 9001:2000**

Esta Norma Internacional pueden utilizarla partes internas y externas, incluyendo organismos de certificación, para evaluar la capacidad de la organización para cumplir los requisitos del cliente, los reglamentarios y los propios de la organización. En el desarrollo de esta Norma Internacional se han tenido en cuenta los principios de gestión de la calidad enunciados en las Normas ISO 9000 e ISO 9004.

La adopción de un SGC debería ser una decisión estratégica de la organización. El diseño y la implementación del SGC de una organización están influenciados por diferentes necesidades, objetivos particulares, productos suministrados, procesos empleados; y tamaño y estructura de la organización. No es el propósito de esta Norma Internacional proporcionar uniformidad en la estructura de los SGC o en la documentación.



ISO/IEC 9001:2000 promueve la adopción de un enfoque basado en procesos cuando se desarrolla, implementa y mejora la eficacia de un SGC, para aumentar la satisfacción del cliente mediante el cumplimiento de sus requisitos. Para que una organización funcione de manera eficaz, tiene que identificar y gestionar numerosas actividades relacionadas entre sí. Una actividad que utiliza recursos, y que se gestiona con el fin de permitir que los elementos de entrada se transformen en resultados, se puede considerar como un proceso. Frecuentemente el resultado de un proceso constituye directamente el elemento de entrada del siguiente proceso.

Un enfoque que se utiliza dentro de un SGC enfatiza la importancia de:

- La comprensión y el cumplimiento de los requisitos.
- La necesidad de considerar los procesos en términos que aporten valor.
- La obtención de resultados del desempeño y eficacia del proceso
- La mejora continua de los procesos con base en mediciones objetivas.

### **Personal Software Process (PSP)**

El Personal Software Process (PSP) es un proceso de software definido y medido diseñado para ser usado por medio de un Ingeniero de Software individual. El PSP fue desarrollado por Watts Humphrey y tiene como objetivo guiar el planeamiento y desarrollo de los módulos de software o pequeños programas; y es adaptable a otras tareas del personal. Es una tecnología de SEI (Software Engineering Institute) que trae disciplina a las prácticas de los Ingenieros de Software, mejorando la calidad del producto, aumentando los costos y reduciendo el tiempo del ciclo de desarrollo del software. (Malevski, 1995)

El PSP está basado en los principios de mejoramiento del proceso. CMM está enfocado respecto del mejoramiento de la capacidad organizacional. El enfoque de PSP es el Ingeniero individual. Para fomentar el mejoramiento a nivel personal, PSP ofrece la administración y control del proceso al Ingeniero de Software. Con PSP los Ingenieros desarrollan software usando una propuesta estructurada y disciplinada. Los Ingenieros se ocupan de:

- Seguir un proceso definido.
- Planificar, medir y seguir su trabajo.

- Administrar la calidad del producto.
- Aplicar aspectos cuantitativos para mejorar los procesos de trabajo personales.

### **1.6 Métricas de software**

Se refiere a aquella aplicación continua de técnicas basadas en la medida de los procesos de desarrollo del software, para producir una información de gestión significativa al mismo tiempo que se mejoran aquellos procesos y productos, se denominan métricas de software. “Un método y una escala cuantitativos que pueden ser usados para determinar el valor que toma cierta características en un producto de software concreto”.

“Una función que toma como entrada cierta información del software que se está midiendo, y que devuelve como salida un valor numérico sencillo, el cual es interpretada, como el grado en que el producto de software posee un atributo dado que afecta a su calidad”.(SCALONE, 2006)

Las métricas existen a nivel de proyecto, proceso y producto.

Se afirma que para que las métricas sean útiles deben ser:

- Cuantificables: deben basarse en hechos, no en opiniones.
- Independientes: los recursos no deben poder ser alterados por los miembros que las apliquen o utilicen.
- Explicable: debe documentarse información acerca de la métrica y de su uso.
- Precisas: debe de conocerse un nivel de tolerancia, permitido cuando se mide.

Las métricas, entre ellas se encuentran las métricas para el aseguramiento y control de calidad. Las mismas se clasifican según los criterios de:

- Complejidad: son métricas que definen la medición de la complejidad: volumen, tamaño, anidaciones, y configuración.
- Calidad: son métricas que definen la calidad del software: exactitud, estructuración o modularidad, pruebas, mantenimiento.

- Competencia: son métricas que intentan valorar o medir las actividades de productividad de los programadores con respecto a su certeza, rapidez, eficiencia y competencia.
- Desempeño: son métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware.
- Estilizadas: son métricas de experimentación y de preferencia: estilo de código, convenciones, limitaciones, etc.

Según el contexto donde se aplican:

- Las métricas de proceso: recopilan de todos los proyectos, y durante un largo periodo de tiempo. Caracterizado por control y ejecución del proyecto y medición del tiempo de las fases.
- Las métricas de proyecto: permite evaluar el estado del proyecto y permite seguir la pista de los riesgos.
- Las métricas del producto: se centra en las características del software y no en cómo fue producido, también son productos los artefactos, documentos, modelos, y componentes que conforman el software, se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad y el esfuerzo.

### **1.7 Surgimiento de los Metadatos**

El surgimiento de los Sistemas de Información Geográfica (SIG) hace más de tres décadas fue un paso trascendental en el uso y manejo de la información geográfica, permiten el procesamiento, acceso y análisis de forma automática. Estos sistemas posibilitan la gestión y visualización de este tipo de información.

El avance en el desarrollo de los SIG sumado a los alcances de la ciencia en otras ramas hace que el número, complejidad, calidad y diversidad del conjunto de datos geográficos que se hace necesario manejar crezca de forma vertiginosa; por lo que se hace imprescindible la creación de métodos que compacten todos los elementos característicos de estos datos para garantizar una eficiente manipulación.

Se necesita una estructura bien organizada que permita documentar fácilmente la información existente y la que se cree en un futuro; de forma tal que se puedan hacer búsquedas en dichas estructura, publicar

características fundamentales de esta y hacer sobre ella análisis sin gran dificultad. Los metadatos vinieron a dar respuesta a este problema.

El concepto “Metadato” no surgió recientemente en las Ciencias de la información. Se plantea que el término fue formalmente empleado por Jack Myers en la década de los 60 con el objetivo de describir conjuntos de datos.(2008)

Un Metadato es un conjunto de información que identifica diferentes aspectos relacionados a grupos de datos o a datos específicos y permite conocer características de esos datos que los particularizan dentro de un conjunto. Describe aspectos de los datos espaciales como son: calidad, actualización, referencia espacial, autor, entre otras. Constituyen información sobre la forma y el contenido de los recursos informativos. La primera y más extendida a nivel mundial definición que se le dio al término Metadato fue: “Datos que describen datos”.

### **1.7.1 Normas internacionales para los metadatos geográficos**

Muchas veces la información no viene acompañada de datos que la representen como su autor, modo de identificación, fechas, entre otras características. Por ello se hace necesaria una estructura bien organizada que permita su documentación, la cual está basada en la utilización de metadatos. Existen estándares internacionales que definen los elementos fundamentales que deben contener los metadatos geográficos.

Para la creación de metadatos la ISO a través de su familia ISO 19100 ha definido normas o estándares con este fin.

La ISO 19115 es la norma apropiada si se quiere trabajar en el campo de los metadatos relacionados con datos y servicios geográficos. La misma constituye la normativa que se sigue en la investigación por ser mundialmente utilizada, además es una norma genérica que cubre la totalidad de los datos necesarios para cualquier recurso que se describa.

Esta norma brinda una guía y define un conjunto común de términos, conceptos y procedimientos que se aplican a los metadatos. Los elementos de metadatos describen la información sobre la identificación, la

extensión, la calidad, el modelo espacial y temporal, la referencia espacial y la distribución de los datos geográficos.

Esta norma tiene aplicaciones como:

La catalogación de conjuntos de datos, actividades de *clearinghouse*, y la descripción completa de conjuntos de datos.

Diferentes niveles de información: conjuntos de datos geográficos, series de conjunto de datos, fenómenos geográficos individuales, propiedades de los fenómenos y otros.

Existen diferentes Aplicaciones da la Norma ISO 19115 para metadatos geográficos como son:

ISO 19139: Metadatos. Esquema de implementación XML.

ISO 19115-2: Metadatos para imágenes y datos en malla.

Perfiles de aplicación de la norma ISO 19115.

- ISO 19115 NEM (Núcleo Español de Metadatos).
- ISO 19115 INSPIRE.
- ISO 19115 WISE (Water Information System for Europe).

### 1.8 Conclusiones parciales

En este capítulo se trataron los principales conceptos asociados al tema de calidad, los conceptos metadatos geográficos, así como las normas y estándares internacionales de calidad para guiar el proceso y producto de software, como también se trataron las normas y estándares utilizados en la universidad y la importancia que tienen su utilización. Los temas tratados en este capítulo permiten una mejor interpretación, y comprensión sobre la importancia que tiene crear una estrategia de aseguramiento de la calidad para el proyecto SyGMe, que sea acorde a las actividades a realizar en el proyecto.

## **Capítulo 2: Propuesta de Solución**

### **2.1 Introducción al capítulo**

Para la realización de una estrategia de Aseguramiento de la Calidad es necesario conocer la situación en la que se encuentra el proyecto, en cuanto a saber de qué forma se supervisa la calidad de producto, y el proceso dentro del proyecto, de aquí la importancia conocer, los recursos humanos con que se cuentan en el proyecto, objetivos, herramientas que se utilizan para el desarrollo de metadatos.

En este capítulo se propone un plan para el aseguramiento de la calidad del proyecto SyGMe, con el objetivo de proporcionar una guía para garantizar la calidad del software. Se realiza un análisis y descripción de las actividades para el Aseguramiento de la Calidad.

### **2.2 Estructura Organizativa del proyecto**

El equipo de desarrollo del proyecto SyGMe está compuesto por profesionales y estudiantes de la Facultad 6 de la universidad. La estructura actual del proyecto está compuesta por: un jefe de proyecto que cuenta con un analista principal y un arquitecto. También cuenta con un equipo de desarrolladores formado básicamente por 7, 5 estudiantes de quinto año y 2 estudiantes de cuarto año, además cuenta con 4 estudiantes de tercer año en proceso de adiestramiento. Este proyecto cuenta actualmente con 8 computadoras ubicadas en el laboratorio 806 del área de UCI FAR.

El proyecto SyGMe realiza un ciclo de vida siguiendo la metodología de Proceso Unificado de Rational (RUP), sus fases de desarrollo están divididas en: Inicio, Elaboración, Construcción y Transición. Se encuentra estructurado por 6 módulos, los cuales son: el Módulo Central o Núcleo, el Módulo de Publicación, Módulo de Agenda de Contacto, Módulo de Tesoros, Módulo de Gestionar Metadatos por la Normas DublinCore, el Módulo Gestionar Metadatos por la Norma ISO 19115.

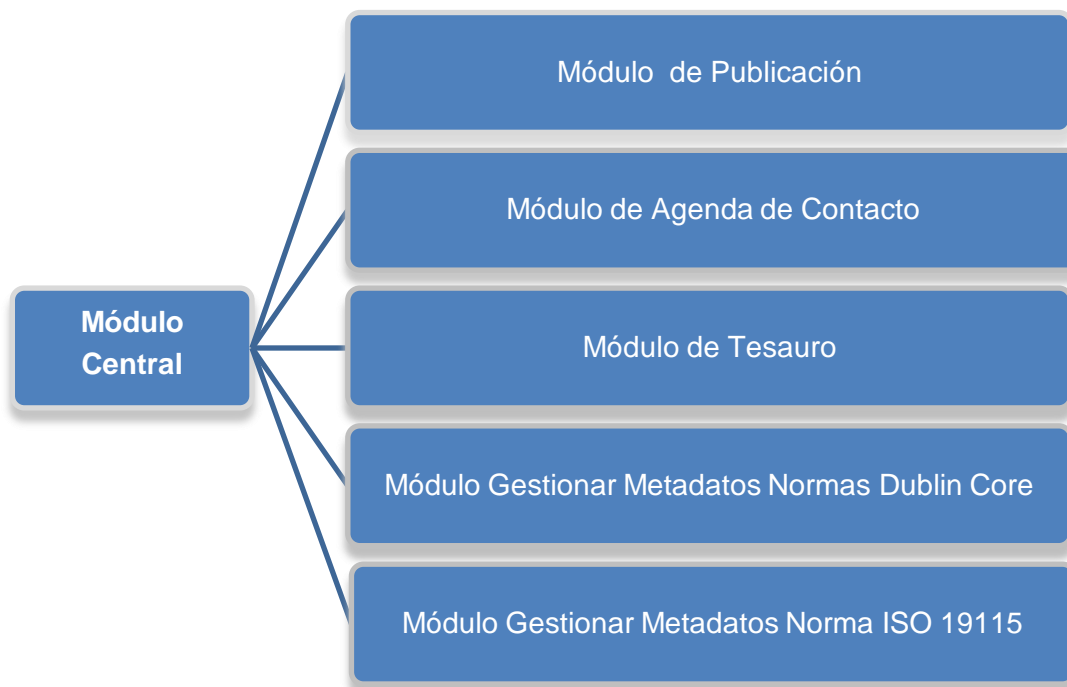


Figura1: Estructura por módulos del proyecto. (Elaboración Propia)

## 2.3 Descripción de Roles

En el proyecto están definidos los siguientes roles, con sus respectivas funcionalidades, pero se propone que se redefinan los roles, para una mejor distribución del equipo de trabajo.

Rol	Responsabilidades
Jefe de proyecto	Define la organización y estructura (líneas de trabajo) del proyecto. Gestiona y asigna recursos humanos y de otro tipo. Establece los horarios de trabajo del proyecto. Establece las estrategias de desarrollo del proyecto. Define, planifica, asigna y controla las tareas del proyecto. Coordina las interacciones con los clientes. Define el Plan de Capacitación y evaluación del personal. Planifica y realiza reuniones de información y control del proyecto. Realiza talleres y consejos técnicos con todos los miembros del proyecto.

	Participa en la selección del personal del proyecto.
Analista	Encargado de elaborar el análisis del modelado del software. Define los artefactos que se deben realizar para el análisis del software. Construye los artefactos para el análisis del software.
Arquitecto	Encargado de definir la arquitectura de la información. Define y establece la línea base de la arquitectura. Encargado de establecer los patrones de diseño y los estándares y restricciones de la implementación.
Desarrolladores	Encargado de la programación de los servicios Define las funcionalidades del software.
Administrador de Calidad	Encargado de cumplir la normas de calidad establecida en todo momento. Revisa la documentación y el software.

**Tabla 6:** Roles y responsabilidades (Elaboración propia).

En el proyecto actualmente, la organización por roles no es óptima ya que el proyecto no cuenta con la cantidad de personal necesario para realizar cada uno de los roles definidos, lo cual trae como problema que las actividades de los roles que han sido definidos, tengan que ser ejercidas por otros miembros del proyecto, que están realizando sus propias tareas, lo cual entorpece el trabajo y ocasiona pérdidas respecto al tiempo. El proyecto tiene definido el rol de administrador de calidad pero al no contar con el suficiente personal, este rol tiene que desarrollarlo otra persona que a su vez, ya desempeña otro rol.

## 2.4 Estrategia de Aseguramiento de la Calidad

La estrategia de aseguramiento de la calidad para el proyecto SyGMe está conformada por las siguientes actividades:

- Organizar los Recursos Humanos.
- Plan de Aseguramiento de la Calidad.
- Entrenamiento de los miembros del proyecto SyGMe.



## 2.5 Plan de aseguramiento de Calidad

El presente plan de aseguramiento de la calidad tiene como propósito realizar una planificación de las actividades que respondan al aseguramiento y control de la calidad del proyecto SyGMe, con el objetivo de alcanzar la certificación del nivel 2 de CMMI en la que está inmersa la Universidad y con ello alcanzar mayor capacidad, madurez y eficiencia en el centro.

### 2.5.1 Propósito

El propósito que persigue la elaboración de un Plan de Aseguramiento de Calidad, es describir de forma general, como se asegura la calidad del producto, a través de herramientas y procesos para alcanzar los objetivos de calidad del proyecto. Se convierte en una guía obligatoria para todos los miembros del proyecto que responden por la calidad del producto. En conclusión se propone el uso de un conjunto de documentos que servirán para asesorar el buen funcionamiento del proyecto SyGMe.

### 2.5.2 Alcance

El presente refleja el plan que realiza el proyecto SyGMe, perteneciente al centro GEYSED, de la facultad 6, para el proceso de calidad de software.

### 2.5.3 Definiciones, Acrónimos y Abreviaturas

La presente temática incluye un listado de definiciones, acrónimos y abreviaturas utilizadas durante la elaboración del plan, con el objetivo de que los miembros del proyecto, y cualquiera que esté interesado en esta propuesta, la usen como guía y así puedan entender con facilidad todo lo planteado.

Término	Descripción
<b>SQA</b>	Aseguramiento de la Calidad del Software
<b>CMMI</b>	Modelo de Capacidad y Madurez Integrado
<b>ISO</b>	Organización Internacional para la Estandarización
<b>IEEE</b>	Instituto de Ingenieros Electricistas y Electrónicos
<b>Auditoría</b>	Evaluación independiente de un producto SW, proceso SW, o conjunto de procesos SW, para evaluar el cumplimiento de especificaciones estándares,

	acuerdos contractuales y otros criterios.
<b>Revisión</b>	Un proceso o reunión, durante el cual un producto SW se presenta al personal del proyecto, gerentes, usuarios, clientes, representantes de usuarios u otras partes interesadas para comentarios o aprobación.
<b>RTF</b>	Revisión Técnica Formal: Evaluación sistemática de un producto SW por un equipo calificado de personal que examina la adecuación de un producto SW a su uso previsto e identifica discrepancias respecto a especificaciones o estándares. También pueden proporcionar recomendaciones de alternativas y examinar varias alternativas.
<b>Prueba</b>	Examen visual de un producto SW para detectar e identificar anomalías de SW, incluyendo errores y desviaciones de estándares y especificaciones.
<b>Lista de Chequeo</b>	Lista de aspectos en forma de preguntas utilizados para revisar o inspeccionar temas de interés.
<b>Métrica</b>	Medidas que se le hacen al software para indicar su calidad, Ej.: productividad de las personas, estimación de tamaño o duración.
<b>LCS</b>	Lineamientos de Calidad del Software.

**Tabla 7:** Términos y descripción (Elaboración propia).

#### 2.5.4 Referencias

Documentos del expediente de proyecto SyGMe.

### 2.5.5 Resumen

El documento consta de varias secciones y anexos que en conjunto se integran para formar lo que se entiende por Plan de Aseguramiento de la Calidad. En las primeras secciones se hace alusión al propósito, alcance y objetivos de calidad del proyecto, a continuación se describe la estructura de los miembros del equipo de calidad del proyecto SyGMe y de forma más detallada las tareas y responsabilidades correspondientes a cada uno de ellos. Se muestra además, una tabla donde se proponen los estándares y guías a usar durante las diferentes etapas del proyecto. Posteriormente se abordan las estrategias, técnicas y herramientas a utilizar en las Revisiones, Auditorías y Pruebas para cada proceso de desarrollo del software definido por los miembros del proyecto. El resto del plan identifica las herramientas y métodos que soportan las actividades y tareas de aseguramiento de la calidad de forma general.

### 2.5.6 Objetivos de calidad

Los objetivos de la calidad para el proyecto SyGMe son:

Establecer un plan de SQA para el control de la calidad del proyecto en desarrollo SyGMe.

Asegurar que el proceso de Calidad durante el proceso de desarrollo se cumpla, de acuerdo a las exigencias del plan y a los Lineamientos de Calidad del Software en la UCI.

Verificar que se cumplan los requisitos establecidos por el cliente.

Garantizar las revisiones durante todo el ciclo de vida del proyecto, para chequear la documentación del proyecto, con el fin de que el producto esté bien documentado y libre de errores.

### 2.5.7 Gestión del Aseguramiento de la Calidad

#### 2.5.7.1 Organización

El proyecto SyGMe no tiene definido un equipo de calidad, por no contar con la cantidad de personal necesario en el proyecto, lo que trae como limitante que la calidad de proyecto sea llevada a cabo solamente por un miembro de proyecto, el mismo sería el encargado de realizar todas las tareas del

administrador de calidad, lo cual trae como consecuencia que existan dificultades con la realización de algunas tareas y responsabilidades asociadas a este rol.

Con la aplicación de la estrategia de aseguramiento de la calidad y para un mejor funcionamiento del proyecto se propone la creación de un Equipo de Calidad, los roles que se proponen son el administrador de calidad, el diseñador de pruebas, el revisor técnico, y el probador.

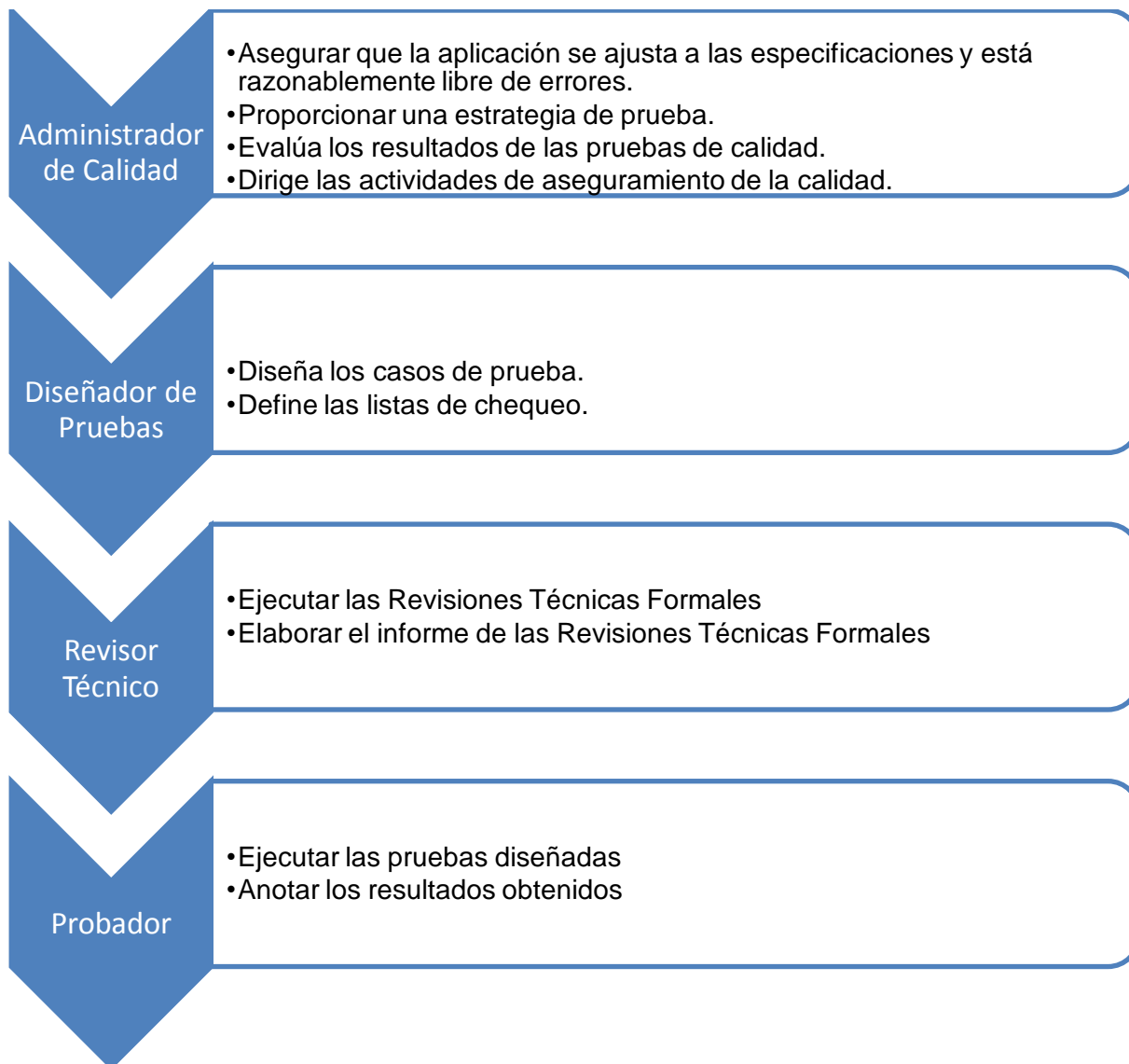


Figura 2: Roles del Equipo de Calidad

### 2.5.8 Documentación

Para la elaboración de este plan se utilizaron una serie de documentos, en esta sección debe aparecer un listado de los documentos utilizados en el Plan de Aseguramiento de la Calidad son: de roles y responsabilidades, plan de administración de configuración del software.

### 2.5.9 Métricas

“Las métricas internas les proporcionan a los usuarios, evaluadores, verificadores y diseñadores el beneficio de que ellos pueden evaluar la calidad de producto de software y dirigir la calidad desde las etapas más tempranas antes de que el producto de software devenga ejecutable”. (ISO, 2004)

“Las métricas externas les proporcionan a los usuarios evaluadores, verificadores y diseñadores el beneficio de que ellos pueden evaluar la calidad del producto de software durante las pruebas o el funcionamiento”. (ISO, 2004)

Para una correcta planificación de las métricas que se deben utilizar durante todo el ciclo de vida del proyecto, se debe elaborar un plan de mediciones que recoja toda esta información, no obstante a continuación se hace una propuesta de uso de métricas que pueden resultar de beneficio para el proyecto. Para una evaluación temprana del software en producción se utilizan métricas durante los primeros flujos de trabajo del ciclo de vida, lo cual servirá para comenzar con la aplicación de acciones correctivas desde etapas tempranas del desarrollo del software, además acelera el proceso de revisiones.

**Métricas a utilizar para evaluar los Requerimientos.** (Pressman, 2002)

Para evaluar la calidad de los requerimientos de un proyecto, se puede aplicar la siguiente métrica.

$$Q1 = \text{Nu1} / \text{NR}$$

Q1: Consistencia de la interpretación de los revisores

Nu1: Es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

$$\text{NR} = \text{NF} + \text{NNF}$$

Donde:

NR: Requisitos que hay en una especificación.

NF: Es el número de Requisitos Funcionales.

NNF: Es el número de Requisitos No Funcionales.

Si el valor de Q1 se acerca a 1, significa que los revisores han tenido interpretaciones parecidas y que las especificaciones tienen un alto grado de claridad, por lo que tendrían una buena calidad. Es conveniente utilizar esta métrica pues se detectarían errores en etapas tempranas del desarrollo lo que disminuiría el costo en tiempo y recursos, de los defectos encontrados durante las pruebas.

**Métrica a utilizar para verificar la completitud de la Implementación.**(ISO/IEC9126-3)A medida que se desarrolla el software, es conveniente hacerse la siguiente interrogante, ¿Cuán completa ha sido la implementación y cuál es su conformidad con la especificación de requisitos? Para dar respuesta a esta interrogante se debe ejecutar las pruebas de caja negra funcionales de acuerdo con la especificación de requisitos, contando el número de funciones perdidas detectadas y comparando el resultado con el número de funciones descritas en la especificación de requisitos. La fórmula que se debe utilizar es:

$$X = 1 - A/B$$

Donde:

A es el número de funciones perdidas detectadas en la evaluación.

B es el número de funciones descritas en especificación de requisitos.

El valor obtenido debe estar entre  $0 \leq X \leq 1$ , a mayor cercanía al 1 resultará mejor. El uso de esta métrica se propone para saber si el sistema implementado cumple con los requisitos especificados. Esta es una métrica de adecuación, valorada con un peso alto. Para evaluar la calidad del producto una vez generado algún tipo de ejecutable, se necesitan métricas también. Una propuesta de utilización de estas métricas se hace a continuación.

**Integridad de la descripción del producto:** Valora el grado en que los usuarios potenciales van a entender las capacidades del producto en cuanto a los factores relacionados con el aprendizaje y la educación una vez que se hayan leído la descripción del mismo.

$X = A / B$  Donde A - número de funciones comprendidas por el usuario en la descripción del producto. B - número total de funciones.

**Comprensibilidad de la función:** Permite hacer un análisis de si pueden o no los usuarios comprender lo que se requiere con cada una de las funciones, relacionados con el aprendizaje y la educación.

$$X = A / B$$

Donde: A - número de funciones que tienen el propósito de ser comprendidas por el usuario. B - número total de funciones.

**Métricas a utilizar durante las revisiones y pruebas.** Para realizar un análisis adecuado del proceso de revisiones es necesario utilizar algunas métricas que aporten una idea de cuan exitoso ha sido o debe ser en etapas futuras este proceso, para ello se propone la utilización de las siguientes métricas.

**Detección de errores:** Cuántos fallos fueron detectados en las revisiones al producto. Se define de la siguiente forma:  $X = A / B$  donde A es el número absoluto de defectos encontrados en las RTF y B es el número de errores estimados para esta revisión. El valor de esta métrica debe oscilar entre [0; 1].

Mientras más cercano a 1 mayor cantidad de fallos encontrados.

**Eficacia de la eliminación de defectos (EED):** Esta métrica se puede calcular mediante la fórmula:

$$EED = E / (E + D)$$

Donde:

E= es el número de errores encontrados antes de la entrega del software al usuario final.

D= es el número de defectos encontrados después de la entrega.

El valor ideal de EED es 1, simbolizando que no se han encontrado defectos en el software. De forma realista, D será mayor que cero, pero el valor de EED todavía se puede aproximar a 1 cuando E aumenta. En consecuencia cuando E aumenta es probable que el valor final de D disminuya. La utilización de esta métrica alienta a que el equipo de proyecto establezca técnicas para encontrar los errores posibles antes de su entrega. De forma general se puede utilizar esta métrica para evaluar la habilidad del equipo de

encontrar deficiencias que harán que se atrase el proyecto.  $EED = E_i / (E_i + E_{i+1})$  Donde:  $E_i$ : es el número de errores encontrados durante la actividad *iesimade*: ingeniería del software *i*.  $E_i + 1$ : es el número de errores encontrado durante la actividad de ingeniería del software (*i + 1*) que se puede seguir para llegar a errores que no se detectaron en la actividad *i*. Un objetivo de calidad del equipo de desarrollo es alcanzar un EED que se aproxime a 1, esto es, los errores se deberían filtrar antes de pasarse a la actividad siguiente.

### 2.5.10 Estándares y Guías

Se plasman en esta sección un listado de todos los estándares y guías utilizados por el plan de SQA, pues son los establecidos por el Centro de Calidad de Soluciones Tecnológicas (Calisoft) de la universidad. En el proyecto se puede definir por cuáles optar y la ubicación de los mismo.

Estándar	Comentarios
<b>RUP</b>	Para el proceso de desarrollo de software.
<b>CMMI</b>	Para el aseguramiento de la calidad, gestión de configuración, gestión de requisitos.
<b>Lineamientos de Calidad</b>	Guía a seguir por los proyectos de desarrollo de software atendiendo a cuáles son los artefactos que deben generar y procesos que deben realizar para aspirar a un producto final con calidad.
<b>NC-ISO 9000</b>	Para el Sistema de Administración de Calidad. Conceptos de Calidad.
<b>ISO 19011-2002</b>	Revisiones y Auditorías.

Tabla 8: Estándares utilizados (Elaboración propia).

### 2.5.11 Plan de Revisiones y Auditorías

#### 2.5.11.1 Tareas generales de Revisiones y Auditorías

El plan de revisiones y auditorías es la actividad más importante dentro del plan de aseguramiento de la calidad. El procedimiento para introducir las revisiones en el proceso de desarrollo de software es:



- En caso de que no esté conformado el equipo de Aseguramiento de Calidad del Software, conformarlo y definir el administrador de calidad, el cual se encargaría de las funciones correspondientes a este grupo.
- Definir roles adecuados para el grupo de desarrollo del proyecto, entre los que deben estar: Líder del proyecto, Desarrolladores y Administrador de Calidad.
- Ejecutar revisión del producto terminado que ha desarrollado el proyecto.
- Definir e introducir métricas que permitan analizar la efectividad de las revisiones.
- Introducción de herramientas automatizadas para planificación y seguimiento de las revisiones.

Las revisiones que se llevarán a cabo en el proyecto serán las Revisiones Técnicas Formales, más conocidas por sus siglas RTF, debido a su efectividad para detectar errores en un 75%.

Los objetivos de la RTF son:

Descubrir errores en la función, la lógica o la implementación de cualquier representación del software antes de que se conviertan en defectos.

Verificar que el software bajo revisión alcanza sus requisitos.

Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos.

Conseguir un software desarrollado de forma uniforme.

Hacer que los proyectos sean más manejables.

La RTF es realmente una clase de revisión que incluye recorridos, inspecciones, revisiones cíclicas y otro pequeño grupo de evaluaciones técnicas del software. Cada RTF se lleva a cabo mediante una reunión y sólo tendrá éxito si es bien planificada, controlada y atendida.

### **2.5.11.2 Revisiones a utilizar en el proceso de revisión y auditoría del proyecto SyGMe**

Las diferentes revisiones que se le realizan al software sirven para purificar las actividades de ingeniería del software, que suceden como resultado del análisis, el diseño, y la implementación.

- Revisión de los Requerimientos: Asegurar que el sistema tenga condiciones o capacidades para satisfacer las exigencias expuestas por el cliente (el artefacto revisado es la lista de requerimientos).
- Revisión de la Arquitectura: Garantizar un diseño preliminar (el artefacto revisado es la arquitectura).
- Revisión del Análisis y Diseño: Asegurar que el modelo se acerque lo más posible a la implementación para tener una idea más clara de lo que se quiere en el sistema (los artefactos revisados son modelo de casos de uso del sistema, actores que se relacionan con el modelo, clases del diseño, diagramas de secuencia, base de datos.)
- Revisión del proceso.
- Revisión Administrativa: Revisión de aprobación del proyecto, revisión de la planificación del proyecto y revisión del plan de iteración.
- Revisión post-mortem: Revisión de la aceptación de la iteración y revisión de la aceptación del proyecto.

### Principios de auditoría.

Para realizar una auditoría se deben tener varios principios como los que a continuación se plantean:

- El equipo auditor debe tener una conducta ética, donde la confidencialidad, la integridad y la confianza sean sus principales valores.
- Se debe revelar con veracidad y exactitud los hallazgos de la auditoría.
- Deben tener un sentido de responsabilidad elevado puesto que estos deben proceder con el debido cuidado, de acuerdo con la importancia de la tarea que desempeñan y la confianza depositada en ellos por el cliente de la auditoría y por otras partes interesadas.

- El equipo auditor debe ser completamente independiente en la entidad auditada. Los auditores deben mantener una actitud objetiva a lo largo del proceso de auditoría para asegurarse de que los hallazgos y conclusiones de la auditoría estarán basados sólo en la evidencia de la auditoría.

Las auditorías se realizan a nivel central, estas son conocidas como Auditorías a la actividad productiva y son coordinadas por Calisoft. El objetivo de estas es examinar y evaluar el grado de correspondencia entre los procedimientos, lineamientos y disposiciones establecidas y su aplicación a fin de determinar el grado de planificación, organización, dirección, control y si se han alcanzado las metas propuestas.

### 2.5.12 Pruebas y Evaluación

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. La creciente percepción del software como un elemento del sistema y la importancia de los costes asociados a un fallo del propio sistema, están motivando la creación de pruebas minuciosas y bien planificadas.

La prueba de software permite al desarrollador determinar si el producto generado satisface las especificaciones establecidas. Así mismo, una prueba de software permite detectar la presencia de errores que pudieran generar salidas o comportamientos inapropiados durante su ejecución. Las pruebas de software son una parte del proceso de aseguramiento de calidad; realizar pruebas a un sistema de información no significa necesariamente que el proceso de desarrollo esté asegurado y tampoco que de manera directa esté mejorando. Pero implementar un proceso de pruebas de software y más aún sostenerlo en el tiempo, es un buen inicio para más adelante aumentar el alcance y con base en las reflexiones al interior del equipo y de los hallazgos registrados en su producto, realizar un mejoramiento del proceso de desarrollo basado en los lineamientos del aseguramiento de calidad de software. Como guía para la aplicación de pruebas existe el plan de pruebas. En dicho plan se describe de forma específica cómo el equipo de aseguramiento de calidad realizará todas sus pruebas.

Los niveles de prueba son los escenarios donde se aplican las pruebas. Están estructurados de forma ascendente para lograr concordancia y organización en los procesos de pruebas. Todos deben realizarse con el objetivo de garantizar la calidad del software de forma continua e incremental.

Los niveles de pruebas son los siguientes:

- **Pruebas de Unidad:** Esta prueba se centra en el proceso de validación en los componentes o módulos, es decir, menores unidades del diseño de software. Verifican las interfaces de cada módulo con el objetivo de asegurar una fluidez total en todos los datos que interactúan con la unidad que se está probando. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos. La prueba de unidad siempre está orientada a caja blanca.
- **Pruebas de Integración:** Con la prueba de unidad se verifica cada módulo o componente por separado, pero es necesario comprobar que unidos también funcionen correctamente. Cuando se combinan todos pueden existir problemas debido a su interacción, provocando pérdida de datos en alguna interfaz. Si las funciones de cada módulo en conjunto llegan a transformar la función principal se producen problemas expuestos a ocurrir en las etapas posteriores del ciclo de vida del software. Las pruebas de integración se realizan durante la construcción del sistema con el objetivo de crear la estructura del programa comprobando la unión de todos los componentes mediante sus interfaces y asegurando que cumplen con las funcionalidades requeridas.
- **Pruebas de Validación:** Se desarrollan cuando se han terminado las pruebas de integración, cuando ya se han detectado y corregido los errores en la interfaz, la validación se consigue sí, el sistema cumple con los expectativas del cliente. Para la validación del software es necesaria la realización de pruebas de caja negra, demostrando la excelencia en las funcionalidades del sistema.
- **Pruebas de Sistema:** Se realizan durante la construcción del sistema trazándose como objetivo fundamental probar completamente el sistema verificando su integridad y funcionalidad, para de esta forma garantizar la calidad del software ante el cliente, aquí es donde se detecta la mayor cantidad de errores. Estas pruebas permiten ejercitar el sistema comprobando la integración de la información, el funcionamiento de las interfaces, entre otros aspectos. En este nivel se realizan pruebas como, pruebas de recuperación, pruebas de seguridad, pruebas de resistencias (stress), pruebas de usabilidad, pruebas de funcionalidad y pruebas de rendimiento.
- **Pruebas de Aceptación:** Las pruebas de aceptación permiten validar que el sistema cumpla con el funcionamiento esperado y deseado tanto por los clientes como por los desarrolladores del

software. El usuario del sistema es el encargado de definir estas pruebas, además de ejecutarlas y darles su aprobación final.

### Técnicas:

Prueba de Caja Blanca.

Prueba de Caja Negra.

**Prueba de caja blanca:** Durante la aplicación de esta técnica se trabaja con el código del proyecto, este es separado por segmentos y se le asigna un número a cada instrucción con el objetivo de crear un grafo con estos números. Cada sentencia tiene una representación, con esta prueba se realiza la revisión del código detectando errores lógicos dentro de este. Una vez hecho esto se calcula la complejidad ciclomática del grafo formado con el objetivo de definir la cantidad de casos de pruebas que se van a hacer.

Con este método se determina cuáles son los casos de prueba a partir del código fuente del software y se utilizan las especificaciones para determinar el resultado esperado del caso. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto. La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y ciclos. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.

Dentro de técnica de caja blanca se encuentran las siguientes pruebas:

Prueba de camino básico.

Prueba de condición.

Prueba de flujo de datos.

Prueba de bucles.

**Prueba de Caja negra:** Las técnicas de pruebas de caja negra consisten en seleccionar los casos de prueba basándose en la funcionalidad esperada del mismo, es decir, en la especificación de lo que debe hacer el programa. El objetivo de estas pruebas es comprobar si existen errores en la funcionalidad del software, es decir se comprueban las interfaces del mismo para detectar defectos de función en ellas. Antes de realizar un programa conviene tener por escrito qué es lo que debe hacer el mismo (documento de especificación software), las pruebas de caja negra deben emplear esta documentación para seleccionar los casos de prueba. La prueba de todos los casos posibles es inviable, por lo que, al igual que para las pruebas de caja blanca, se debe realizar una selección de los casos que se van a utilizar.

Dentro de la técnica de caja negra se encuentran las siguientes pruebas:

Técnica de partición equivalencia.

Análisis de valores límites.

Prueba de comparación.

### 2.5.12.1 Pruebas a realizar en el proyecto SyGMe

Las pruebas que se proponen en este plan de aseguramiento de la calidad al proyecto SyGMe son pruebas de sistema, lo que permite probar el funcionamiento de producto en su totalidad, después de integrados sus módulos y a través de la interfaz de usuario. Se define este nivel debido a que se pretende probar la integridad y funcionalidad del sistema, tratando de encontrar la mayor cantidad de errores. Las técnicas de pruebas de dicho nivel permiten que el sistema se ejercite, además se comprueba la integración de la información y el funcionamiento de las interfaces.

Cada nivel de prueba contiene un conjunto de tipos de métodos y técnicas las cuales están diseñadas con el objetivo de encontrar posibles errores del software en su respectivo nivel. Dentro de los tipos de prueba que se realizan en el nivel de sistema, las escogidas fueron las siguientes:

**Pruebas Funcionales:** Se decidió realizar estas pruebas con el objetivo de verificar que el sistema tenga implementado adecuadamente cada uno de los requisitos establecidos por el cliente. Se escogió el método de Caja Negra, haciendo uso de la técnica de Partición de Equivalencia. Esta técnica se basa en una evaluación de las clases de equivalencia para una condición de entrada.

**Pruebas de carga y estrés:** Estas pruebas se ejecutan para poder conocer los límites de carga que soporta el sistema en una situación determinada. El proyecto SyGMe presenta gran cantidad de información y gran cantidad de usuarios interactúan con el sistema por lo que se considera importante determinar hasta qué punto puede cargarse de trabajo el software sin que ocurran fallas.

**Prueba de seguridad:** Estas pruebas presentan un alto nivel de importancia, ya que todo sistema computacional que maneje información sensible constituye en blanco de entradas indebidas o ilegales al mismo. Otra de las razones principales por la que escogimos las pruebas de seguridad es debido a que nunca se podrá construir una aplicación segura sin realizarse este tipo de pruebas. Mediante ellas se puede verificar que el sistema tenga mecanismos incorporados que protejan la aplicación de accesos impropios de los usuarios. Para que estas pruebas logren su objetivo, requieren de tiempo y recursos suficientes que permitan una correcta ejecución.

**Prueba de usabilidad:** Las pruebas de usabilidad permiten que un sistema sea funcionalmente correcto, eficiente, que para los usuarios sea fácil de aprender y recordar, además de presentar tolerancia a los errores. Estas pruebas fueron escogidas debido a que la usabilidad es un factor estratégico fundamental para conseguir el máximo aprovechamiento de recursos que brinda una aplicación. Un sistema usable reduce los costos de producción, mantenimiento y soporte, también es capaz de aumentar la productividad disminuyendo esfuerzos.

### 2.5.13 Herramientas, Técnicas y Metodologías

Para la realización de este plan se utilizaron una serie de herramientas, técnicas y metodologías:

Estándares y Normas de la IEEE, ISO, el modelo CMMI.

Plantillas del expediente de proyecto establecido por el proyecto para registrar la documentación del proyecto.

Para la realización de las revisiones se utilizaron las revisiones técnicas formales mencionadas en secciones anteriores y las listas de chequeo.

### 2.5.14 Gestión de Configuración

El proceso de gestión de configuración se aplica durante todo el ciclo de vida del sistema para controlar las modificaciones realizadas en cualquier módulo del proyecto y las versiones de los elementos que componen el sistema.

La Gestión de Configuración se realiza durante todas las actividades asociadas al desarrollo del sistema, y continúa registrando los cambios hasta que éste deja de utilizarse. En el desarrollo de software, los cambios son inevitables, debidos principalmente a modificaciones de requisitos y fallos. Normalmente, se trabaja en equipo por lo que es preciso llevar un control y registro de los cambios con el fin de reducir errores, y aumentar la calidad y la productividad.

### 2.5.15 Registro de calidad

Los registros de calidad serán almacenados en un repositorio establecido durante todo el ciclo de vida del proyecto, gestionándose los cambios a través de las políticas establecidas. Registros que se guardarán:

- Informe de Revisiones.
- Informe de Auditoría.
- Informe de No Conformidades.
- Casos de Pruebas.
- Plan de Pruebas.
- Listas de Chequeos.

### 2.5.16 Entrenamiento

Seguidamente se proponen un conjunto de actividades para capacitar al equipo de calidad las cuales deberán ser cumplidas luego de crearse el equipo de calidad:



- Curso de capacitación sobre los temas de aseguramiento de la calidad. Con este curso se pretende que el equipo de trabajo obtenga mayores conocimientos del tema en general y así lograr una nivelación entre todos los integrantes de este equipo.
- Investigación sobre las normas nacionales e internacionales de calidad. Se propone con este trabajo elevar los conocimientos del equipo al respecto y garantizar que todos los integrantes adquieran familiarización con las normas seleccionadas.
- Investigación detallada de los roles de RUP. El objetivo de esta tarea es identificar según cada rol las tareas que se deben realizar, así se podría distribuir mejor el trabajo y no sobrecargar a una sola persona con exceso de responsabilidades.
- Curso de capacitación para la confección de las listas de chequeos. Se procura que se vayan creando las herramientas y conocimientos necesarios para la realización de este tipo de trabajo.
- Curso de capacitación para mejorar el trabajo con las herramientas propuestas anteriormente en este documento, con el objetivo de emplearlas para contribuir al aseguramiento de la calidad del producto software desarrollado.
- Curso de capacitación para el diseño de casos de prueba que se está impartiendo por la Calisoft. La meta de esta actividad es proporcionar los conocimientos indispensables para realizar de forma satisfactoria este tipo de trabajo.
- Curso de auditoría que se está impartiendo por la Calisoft, de esta forma se confirma la estrategia trazada para el desarrollo de las auditorías en el plan, acorde con la establecida por la universidad.

### 2.6 Conclusiones parciales

En este capítulo se definió la estrategia de SQA para el proyecto SyGMe, explicando todas las actividades que se realizarán para darle cumplimiento al objetivo de este trabajo, como el Plan de Aseguramiento de la Calidad y la organización de los recursos humanos del proyecto, principalmente los involucrados en el SQA.

## Capítulo 3: Validación de la Propuesta

### 3.1 Introducción al capítulo

En este capítulo se validará la estrategia de aseguramiento de la calidad propuesta para el proyecto SyGMe, se describirán las actividades realizadas y la medida en la que se encuentran cumplidas las mismas.

El aseguramiento de calidad del software es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos dados de calidad.(Cueva, 1999)

### 3.2 Validación de la estrategia SQA

La estrategia de aseguramiento de la calidad para el proyecto SyGMe está conformada por las siguientes actividades:

- Organizar los Recursos Humanos.
- Plan de Aseguramiento de la Calidad.
- Entrenamiento de los miembros del proyecto SyGMe.

Después de un estudio sobre las necesidades de personal del proyecto se definieron los roles que a continuación se plantean:

Rol	Responsabilidad	Competencia
Líder del proyecto	Guía y líder del proyecto, responsable de definir, coordinar, facilitar y comunicar las tareas dentro del equipo de proyecto.	RUP. Tecnologías de desarrollo de software. Gestión de proyectos. Ingeniería de Software.

		Facilidades para la comunicación.
<b>Jefe de Módulo</b>	Dirige y Coordina el trabajo de su módulo. Define u orienta las tareas a cumplir por el resto de su equipo.	RUP. Tecnologías de desarrollo de software. Gestión de proyectos. Ingeniería de Software.
<b>Analista</b>	Encargado de elaborar el análisis del modelado del software, es quien define los artefactos que se deben realizar para el análisis del software y los construye.	Análisis de sistemas. RUP y UML. Herramienta de modelado visual (Visual Paradigm). Técnicas de recopilación de información.
<b>Planificador</b>	Encargado de elaborar el plan de proyecto y definir el conjunto de tareas que se deben realizar en cada fase del desarrollo del software, se encarga también del cumplimiento de las mismas, y hace un estimado del tiempo de desarrollo.	Herramientas de planificación de proyectos. Métricas de estimación. Ingeniería de Software.
<b>Arquitecto de Información</b>	Encargado de definir la arquitectura de la información, qué contenidos y la estructura en que se visualizará la información.	Ingeniería de Software. Facilidades para la comunicación.
<b>Implementador</b>	Encargado de la programación de los servicios y funcionalidades del software construido para la	Herramienta de desarrollo (Netbeans 6.8).

<b>Aplicaciones web</b>	web.	Lenguaje de programación PHP. Gestor de BD: PostgreSQL. Framework Symfony. ExtJS, framework de javascript.
<b>Programador de Bases de Datos</b>	Usa el diagrama entidad-relación para generar el diseño físico de la base de datos. Participa y colabora en la codificación de la capa de acceso a datos.	Herramienta de modelado visual (ErStudio). Modelo relacional. SQL.
<b>Calidad</b>	Se encarga de revisar la documentación y el software como tal, además, vela que se cumple en todo momento las normas de calidad establecidas.	RUP y UML. Calidad de Software. Ingeniería de Software.
<b>Gestor de Configuración</b>	Maneja, gestiona y asegura la infraestructura, la gestión documental y la información del Repositorio del Proyecto. Realiza salvadas periódicas de la información del Repositorio.	Subversión Configuración de servidores. SO GNU/Linux.
<b>Diseñador</b>	Encargado de diseñar la interfaz del Portal y de las demás aplicaciones a desarrollar.	Photoshop. GIMP. CCS. ExtJS
<b>Arquitecto</b>	Define y establece la línea base de la arquitectura. Establece los patrones de diseño y los estándares y restricciones de la	Herramientas de desarrollo de software. ( Netbeans). Lenguajes de programación.

	implementación.	(Javascript y PHP). Patrones de diseño. Estilos arquitectónicos. RUP y UML. Bases de datos. Protocolos de red. Programación Web (Apache HTTP, XML, Web Services). Facilidades para la comunicación.
--	-----------------	--

Tabla 9: Roles, responsabilidades y competencias propuestas para el proyecto SyGMe. (Elaboración Propia)

### 3.3 Revisiones

#### Revisiones al Expediente de Proyecto

Al proyecto SyGMe se le realizaron 2 revisiones a la documentación del expediente de proyecto que confecciona el grupo de ingeniería. Estas revisiones arrojaron muchas deficiencias en el proyecto, permitieron obtener las no conformidades que existían en los documentos. Entre las principales deficiencias encontradas está que faltan documentos en el expediente de proyecto.

Las recomendaciones realizadas, facilitan a los analistas, a corregir las no conformidades encontradas, comparando las mismas con revisiones anteriores. Se detectó en la primera iteración 34 no conformidades (NC) y en la segunda revisión se detectaron 21 NC, donde la evaluación obtenida en ambas ocasiones fue "Deficiente". De todo lo planteado anteriormente se puede concluir que los miembros del proyecto no corrigieron las NC encontradas en la primera revisión, debido a el mismo no cuenta con el personal suficiente para desempeñar todos los roles planteados en la estrategia.

Se recomienda que para un mejor funcionamiento del proyecto que se aplique correctamente la estrategia de aseguramiento de la calidad propuesta en este trabajo, donde lo primero es la organización de los

roles. En el proyecto están definidos, pero no existe el personal suficiente para desempeñarlos, siendo esto la principal deficiencia del proyecto ya que un miembro tiene que realizar las actividades de varios roles, ocasionando problemas que inciden en la calidad del producto final. Entre las principales deficiencias encontradas radica en la documentación del expediente de proyecto, los mismos no cuentan con todos los requisitos especificados en la lista de chequeo. En la mayoría de ellos existen errores ortográficos, las descripciones no están bien redactadas, los diagramas presentan irregularidades, los casos de uso tienen errores en las descripciones de los parámetros, faltan los índices de contenidos, ausencias de actualizaciones y modificaciones de los documentos, las páginas de algunos documentos no coinciden con las páginas reales de los mismos.

### 3.4 Métricas

De las métricas planteadas a utilizar durante las revisiones, se evaluó la métrica de detección de errores, donde A significa el número absoluto de defectos encontrados en las RTF y B el número de errores estimados para esta revisión. Como se planteó en el epígrafe anterior en la primera iteración se encontraron 34 NC, al aplicar esta métrica en la segunda revisión el valor de B será la cantidad de NC encontradas en la primera revisión, y el valor de A será la cantidad encontrada en la segunda iteración, es decir, 21 NC por tanto los valores quedaron planteados de la siguiente forma:

1 iteración	2 iteración
$DE = A/B$	$DE = A/B$
$DE = 34/34$	$DE = 21/34$
$DE = 1$	$DE = 0,61$

En la primera y la segunda iteración los resultados fueron altos porque mientras más cerca este el valor de 1 mayor cantidad de fallos se detectaron, lo cual demuestra el trabajo en el proyecto en esta revisiones estuvo mal, lo cual trajo como consecuencia que el proyecto en estas dos revisiones tuviera evaluaciones finales de las revisiones de deficiente.

### 3.5 Entrenamiento de los involucrados en el proyecto.

Las actividades de entrenamiento se propusieron con el objetivo de elevar el conocimiento de todos los involucrados, y que de esta manera puedan contribuir con la excelencia de los productos que se desarrollan en el proyecto. Estas actividades fueron realizadas según necesidad del proyecto, entre las actividades cumplidas se encuentran:

Curso de capacitación sobre los temas de aseguramiento de la calidad. Con este curso se pretende que el equipo de trabajo obtenga mayores conocimientos del tema en general y así lograr una nivelación entre todos los integrantes de este equipo.

- Resultados: Se recibió un curso sobre PPQA, que es el área de proceso de nivel 2 de CMMI, para el aseguramiento de la calidad del proceso y del producto, luego los integrantes del proyecto recibieron 2 talleres sobre esta área de proceso, para su mejor entendimiento y profundidad sobre el tema, se realizó un juego didáctico para el intercambio de conocimiento sobre este tema, tan importante.

Aunque no se puede afirmar que todos se apoderaron del conocimiento sobre la importancia de asegurar la calidad a la hora de realizar un producto de software tanto en el proceso como en el producto, si quedaron aclaradas muchas dudas sobre la importancia que tiene para la universidad la aplicación de este modelo para la certificación de la calidad de los productos que se realicen en el centro.

Investigación sobre las normas nacionales e internacionales de calidad. Se propone con este trabajo elevar los conocimientos del equipo al respecto y garantizar que todos los integrantes adquieran familiarización con las normas seleccionadas.

- Resultados: Para la evaluación de esta actividad se impartió un curso sobre las principales normas internacionales, así como la aplicación de los estándares de calidad, y se mostraron ejemplos de cómo influye la aplicación de normas y modelos internacionales en la elaboración de productos y a su vez en la forma de desarrollarlos, demostrando la importancia que tiene para nuestra universidad la certificación del nivel 2 de CMMI, para la creación de software de alta calidad, y la importancia que tiene esto en la industria del software mundial.

Investigación detallada de los roles de RUP. El objetivo de esta tarea es identificar según cada rol las tareas que se deben realizar, así se podría distribuir mejor el trabajo y no sobrecargar a una sola persona con exceso de responsabilidades.

- Resultados: Para la evaluación de esta tarea la jefa de proyecto, le asignaron a cada uno de sus miembros, los roles que debían investigar, enfocándose en el rol que desempeñaba cada uno. Para esto cada grupo realizó un taller de Roles y Responsabilidades, permitiendo que de esta manera el equipo de trabajo se apoderara del conocimiento necesario para desempeñar su rol. Para lograr que cada cual realice solo las tareas que pertenecen a su rol y se sienta identificados con el buen desempeño de su papel dentro del proyecto.

Curso de capacitación para la confección de las listas de chequeos. Se procura que se vayan creando las herramientas y conocimientos necesarios para la realización de este tipo de trabajo.

- Resultados: Para este curso se realizó de forma conjunta un encuentro entre todos los integrantes del proyecto, ya que para el mismo, se orientó un trabajo investigativo sobre las distintas listas de chequeos que existen así como es la manera más apropiada de confeccionarlas. Se aclararon muchas dudas referentes al tema de la manera de redactar las deficiencias encontradas.

Curso de capacitación para mejorar el trabajo con las herramientas propuestas anteriormente en este documento, con el objetivo de emplearlas para contribuir al aseguramiento de la calidad del producto software desarrollado.

- Resultados: Se realizó un taller de Visual Paradigm, de PostGIS y de PHP, esto permitió que los miembros del proyecto pudieran tener un mejor desenvolvimiento en las herramientas con las que trabajan, aunque de PHP debió darse el curso como se había planificado, pues este es el Lenguaje de programación que utiliza en el proyecto para el desarrollo de todos sus productos.

Curso de capacitación para el diseño de casos de prueba que se está impartiendo por la Calisoft. La meta de esta actividad es proporcionar los conocimientos indispensables para realizar de forma satisfactoria este tipo de trabajo.



- **Resultados:** Se realizó un taller para el equipo de trabajo y de revisores, en el que se trataron todas las actividades que se deben realizar para lograr una revisión con la calidad requerida. Se le dio una capacitación a los probadores explicándole detalladamente cómo realizar la Prueba de Sistema. Aunque la prueba no se ha realizado, porque los DCP que se le entregó a los probadores para aplicar la prueba estaba mal elaborada.

### 3.6 Conclusiones parciales

Con la aplicación correcta de la estrategia de aseguramiento de la calidad propuesta para el proyecto SyGMe, se puede constatar que siguiendo todos los pasos de la misma, se puede lograr que el proyecto realice de forma eficiente, la preparación y capacitación de todos sus miembros. Después de la aplicación de las actividades propuestas para el SQA se logrará que el proyecto SyGMe, libere satisfactoriamente sus productos.

### Conclusiones Generales

El desarrollo de la investigación arrojó las siguientes conclusiones:

- ✓ Se cumplió el objetivo y las tareas trazadas en la investigación.
- ✓ A partir del análisis realizado sobre las principales características del aseguramiento de la calidad del software se conformó la propuesta de esta investigación, que agrupa todas las actividades de aseguramiento y control de la calidad que deben ser aplicadas en el proyecto SyGMe para alcanzar que el producto final cumpla con las expectativas del cliente.
- ✓ Con la aplicación de la estrategia, se logró organizar los recursos humanos, capacitar al personal y detectar las principales deficiencias mediante las revisiones realizadas, permitiendo una mejor organización del trabajo en el proyecto, aumentar el nivel de preparación de los profesionales y corregir las deficiencias identificadas.
- ✓ Se desarrolló una valoración de los resultados alcanzados una vez concluida la aplicación de la estrategia demostrando que estos fueron deficientes para el proyecto SyGMe.

### **Recomendaciones**

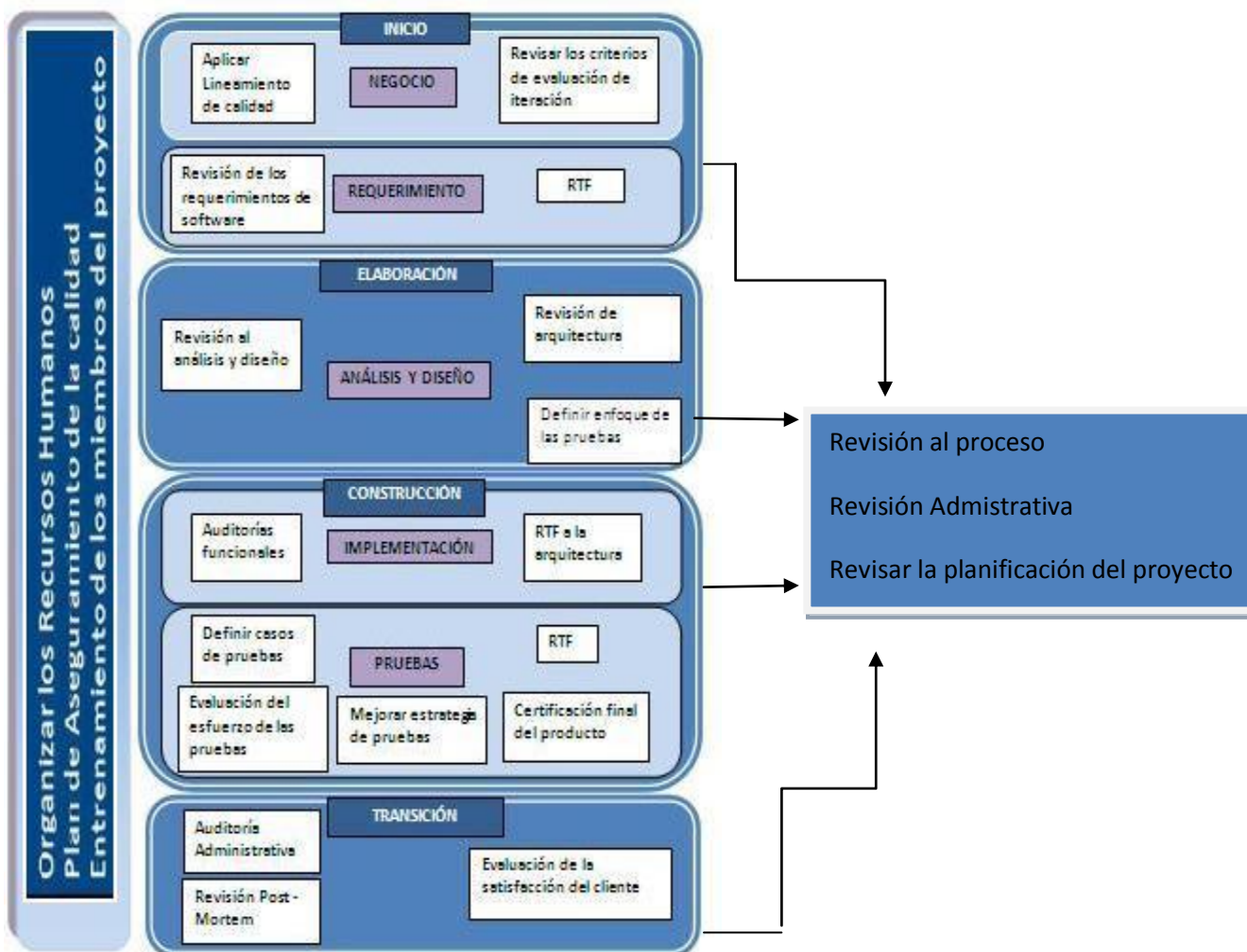
Elaborar el Plan de Prueba que agrupe todos los tipos de pruebas y casos de pruebas

Elaborar el Plan de Mediciones que agrupe todas las métricas necesarias para una medición efectiva de los elementos del proceso de desarrollo.

Estudiar las herramientas para realizar pruebas de caja blanca y utilizar las que sean más convenientes al proyecto.

Darle seguimiento a las actividades que plantea la estrategia de aseguramiento de la calidad mientras exista el proyecto SyGMe.

Anexo 1: Estrategia de Aseguramiento de Calidad para el Proyecto SyGMe



## Bibliografía

**Bicego, A., Krzanik, L. y Kuvaja, P. June 1995.** *Tutorial: BOOTSTRAP – Assessment and Improvement Methodology*. San Francisco USA : s.n., June 1995.

Calidad del Software. [En línea] [Citado el: 9 de 2 de 2011.] <http://www.calidaddelsoftware.com/>.

**Calisoft. 2009.** Libro de Proceso para PPQ. [En línea] 2009. [Citado el: 16 de 1 de 2011.] [http://calisoft.uci.cu/index.php?option=com\\_content&view=article&id=46&Itemid=27](http://calisoft.uci.cu/index.php?option=com_content&view=article&id=46&Itemid=27).

calisoft.uci.cu. *calisoft.uci.cu*. [En línea] [Citado el: 10 de noviembre de 2010.] [http://calisoft.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com\\_content&Itemid=24](http://calisoft.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com_content&Itemid=24).

**CMMI. 2006.** *CMMI for Development, Version 1.2*. 2006.

**Cueva, Juan Manuel Lovelle. 1999.** [www.uniovi.es](http://www.uniovi.es). *www.uniovi.es*. [En línea] 21 de Octubre de 1999. [Citado el: 15 de 1 de 2011.] <http://gidis.ing.unlpam.edu.ar>.

Department of Computer Science. [En línea] [Citado el: 9 de 2 de 2011.] <http://www.ise.gmu.edu/>.

ESI. [En línea] [Citado el: 10 de 2 de 2011.] <http://www.esi.es/>.

**Garcia, Romero.** *Autores de la gestión de la calidad*.

GestionPolis. *GestionPolis.com*. [En línea] [Citado el: 25 de octubre de 2010.] <http://www.gestiopolis.com/recursos2/documentos/fulldocs/ger/caltotalmemo.htm>.

inf.educ. *www.inf.udec.cl*. [En línea] [Citado el: 26 de noviembre de 2010.] <http://www.inf.udec.cl/publicaciones/edicion1/lmonsalve.html>.

**Informática, Departamento de Control de Calidad y Auditoría. 2000.** *Control de Calidad en los Sistemas*. 2000.

inteco. *inteco.es*. [En línea] [Citado el: 28 de noviembre de 2010.] [http://www.inteco.es/calidad\\_TIC/calidad\\_TIC/estudio\\_indicadores.pdf](http://www.inteco.es/calidad_TIC/calidad_TIC/estudio_indicadores.pdf).

**ISO. 2001.** *ISO 9000: "Sistema de Gestión de la Calidad"*. 2001.

—. **2004.** *ISO/IEC 90003: "Guía para la Aplicación de la ISO 9000 al Software de Computación"*. 2004.

- ISO/IEC9126-3.** ISO/IEC9126-3.Metricas Internas de la Calidad del Producto de Software.Estándares de Calidad. [En línea] [Citado el: 14 de 04 de 2011.] [http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso\\_9126\\_3](http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126_3).
- ITIL. *itil.co.uk*. [En línea] [Citado el: 10 de noviembre de 2010.] <http://www.itil.co.uk/>.
- Los sistemas de información geográficas SIG:Definición, características,estado actual y tendencias de desarrollo.* **Diaz Cisneros, Luís R y Candeaux, Rafael Duffatt. Julio de 1994.** Julio de 1994, Revista Internacional de Ciencias de la Tierra, págs. 100 -131.
- Malevski, Yoram. 1995.***Manual de gestión de la calidad total a la medida.* Guatemala : Piedra, 1995.
- 2008.** Mapping Interactivo. Revista Internacional de Ciencias de la Tierra. *NORMAS SOBRE METADATOS (ISO 19115, ISO 19115-2, ISO 19139, ISO 15836).* [En línea] 2008. [Citado el: 20 de 10 de 2010.] [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1455..](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1455..)
- MINGUET MELIÁN, JESÚS M. 2003.***LA CALIDAD DEL SOFTWARE Y SU MEDIDA.* Madrid : EDITORIAL CENTRO DE ESTUDIOS RAMON ARECES, S.A., 2003.
- Miranda, Sandor Luis y Romero, Arturo Luis. 2006.***La calidad, su evolución histórica y algunos conceptos y términos asociados.* 2006.
- Monografias. *monografias.com*. [En línea] [Citado el: 22 de noviembre de 2010.] <http://www.monografias.com/trabajos56/modelo-cmmi/modelo-cmmi.shtml>.
- Monográfico Calidad del Software.* **Enero-Febrero 1999.** Enero-Febrero 1999, NOVATICA.Número 137.
- Montilva, Jonás A. 2006.***Desarrollo de Software Basado en Líneas de Productos de Software.* Mérida : s.n., 2006.
- P, Jackson y Ashton., D. 1996.***Implemente calidad de clase mundial.* Limusa : s.n., 1996.
- Piattini, M.G. y Calvo, J.A. Manzano. 1996.***Análisis y diseño detallado de aplicaciones informáticas de gestión.* 1996.
- Pressman, Roger S. 2002.***Ingeniería de Software. Un Enfoque Práctico.* 2002.
- . **2005.***ngeniería de Software. Un enfoque práctico.* s.l. : Quinta Edición, 2005.
- Profesional. [En línea] [Citado el: 16 de 2 de 2011.] <http://www.profesional.com/>.
- pruebasdesoftware. [En línea] [Citado el: 10 de 2 de 2011.] <http://www.pruebasdesoftware.com/>.
- Puglla Remache, Gabriela Noemí y León Quiñones, Lorena del Cisne. 2004.***Métricas de Proceso y Proyect.* Ecuador : s.n., 2004.

**Rivero, Lanny Canino y Rodríguez, Meydi E Martínez. 2008.***Estrategia de Aseguramiento de la calidad para el Simulador Quirúrgico.* Ciudad de la Habana, Cuba : s.n., 2008.

**Ruilova, María Esther Rojas. 2008.***Informe Ejecutivo.Métricas del Producto para el Software.* 2008.

**Santos Hernández, Vismar. 2009.***Estudio sobre la industria del Software a nivel mundial. Caracterización en América Latina y Cuba.* La Habana, Cuba : s.n., 2009.

**SCALONE, FERNANDA. 2006.***ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE.* Buenos Aires : s.n., 2006.

SEMINARIO DE FORMACION ESPECIALIZADA. [En línea] [Citado el: 4 de 3 de 2011.]

<http://www.fernandotazon.com.es/wp-content/uploads/2009/05/seminario-validacion-de-procesos1.pdf>.

Software Quality Systems. [En línea] [Citado el: 4 de 3 de 2011.] <http://www.sqs.es/es/services/validation.php>.

**Vega Lebrún, Carlos, Rivera Prieto, Laura Susana y García Santillán, Arturo. 2008.***MEJORES PRÁCTICAS PARA EL ESTABLECIMIENTO Y ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE.* Boca del Río, Ver : s.n., 2008.

**Zibert van Gricken, Carolina. 2004.** carolina.terna.net. [En línea] 2004. [Citado el: 2010 de noviembre de 9.] [http://carolina.terna.net/ingsw3/datos/Semana7\\_CMMI2.0.pdf](http://carolina.terna.net/ingsw3/datos/Semana7_CMMI2.0.pdf).

## Trabajos citados

**Bicego, A., Krzanik, L. y Kuvaja, P. June 1995.** *Tutorial: BOOTSTRAP – Assessment and Improvement Methodology*. San Francisco USA : s.n., June 1995.

Calidad del Software. [En línea] [Citado el: 9 de 2 de 2011.] <http://www.calidaddelsoftware.com/>.

**Calisoft. 2009.** Libro de Proceso para PPQ. [En línea] 2009. [Citado el: 16 de 1 de 2011.] [http://calisoft.uci.cu/index.php?option=com\\_content&view=article&id=46&Itemid=27](http://calisoft.uci.cu/index.php?option=com_content&view=article&id=46&Itemid=27).

calisoft.uci.cu. *calisoft.uci.cu*. [En línea] [Citado el: 10 de noviembre de 2010.] [http://calisoft.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com\\_content&Itemid=24](http://calisoft.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com_content&Itemid=24).

**CMMI. 2006.** *CMMI for Development, Version 1.2*. 2006.

**Cueva, Juan Manuel Lovelle. 1999.** *www.uniovi.es. www.uniovi.es*. [En línea] 21 de Octubre de 1999. [Citado el: 15 de 1 de 2011.] <http://gidis.ing.unlpam.edu.ar>.

Department of Computer Science. [En línea] [Citado el: 9 de 2 de 2011.] <http://www.ise.gmu.edu/>.

ESI. [En línea] [Citado el: 10 de 2 de 2011.] <http://www.esi.es/>.

**Garcia, Romero.** *Autores de la gestión de la calidad*.

GestionPolis. *GestionPolis.com*. [En línea] [Citado el: 25 de octubre de 2010.] <http://www.gestiopolis.com/recursos2/documentos/fulldocs/ger/caltotalmemo.htm>.

inf.educ. *www.inf.udec.cl*. [En línea] [Citado el: 26 de noviembre de 2010.] <http://www.inf.udec.cl/publicaciones/edicion1/lmonsalve.html>.

**Informática, Departamento de Control de Calidad y Auditoría. 2000.** *Control de Calidad en los Sistemas*. 2000.

inteco. *inteco.es*. [En línea] [Citado el: 28 de noviembre de 2010.] [http://www.inteco.es/calidad\\_TIC/calidad\\_TIC/estudio\\_indicadores.pdf](http://www.inteco.es/calidad_TIC/calidad_TIC/estudio_indicadores.pdf).

**ISO. 2001.** *ISO 9000: "Sistema de Gestión de la Calidad"*. 2001.

—. **2004.** *ISO/IEC 90003: "Guía para la Aplicación de la ISO 9000 al Software de Computación"*. 2004.

**ISO/IEC9126-3.** ISO/IEC9126-3. *Métricas Internas de la Calidad del Producto de Software. Estándares de Calidad*. [En línea] [Citado el: 14 de 04 de 2011.] [http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso\\_9126\\_3](http://www.mena.com.mx/gonzalo/maestria/calidad/presenta/iso_9126_3).



ITIL. *itil.co.uk*. [En línea] [Citado el: 10 de noviembre de 2010.] <http://www.itil.co.uk/>.

*Los sistemas de información geográficas SIG: Definición, características, estado actual y tendencias de desarrollo*. **Díaz Cisneros, Luís R y Candeaux, Rafael Duffatt. Julio de 1994**. Revista Internacional de Ciencias de la Tierra, págs. 100 -131.

**Malevski, Yoram. 1995.** *Manual de gestión de la calidad total a la medida*. Guatemala : Piedra, 1995.

**2008.** Mapping Interactivo. Revista Internacional de Ciencias de la Tierra. *NORMAS SOBRE METADATOS (ISO 19115, ISO 19115-2, ISO 19139, ISO 15836)*. [En línea] 2008. [Citado el: 20 de 10 de 2010.] [http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=1455..](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1455..)

**MINGUET MELIÁN, JESÚS M. 2003.** *LA CALIDAD DEL SOFTWARE Y SU MEDIDA*. Madrid : EDITORIAL CENTRO DE ESTUDIOS RAMON ARECES, S.A., 2003.

**Miranda, Sandor Luis y Romero, Arturo Luis. 2006.** *La calidad, su evolución histórica y algunos conceptos y términos asociados*. 2006.

Monografías. *monografias.com*. [En línea] [Citado el: 22 de noviembre de 2010.] <http://www.monografias.com/trabajos56/modelo-cmmi/modelo-cmmi.shtml>.

*Monográfico Calidad del Software*. **Enero-Febrero 1999**. Enero-Febrero 1999, NOVATICA. Número 137.

**Montilva, Jonás A. 2006.** *Desarrollo de Software Basado en Líneas de Productos de Software*. Mérida : s.n., 2006.

**P, Jackson y Ashton., D. 1996.** *Implemente calidad de clase mundial*. Limusa : s.n., 1996.

**Piattini, M.G. y Calvo, J.A. Manzano. 1996.** *Análisis y diseño detallado de aplicaciones informáticas de gestión*. 1996.

**Pressman, Roger S. 2002.** *Ingeniería de Software. Un Enfoque Práctico*. 2002.

—. **2005.** *ingeniería de Software. Un enfoque práctico*. s.l. : Quinta Edición, 2005.

Profesional. [En línea] [Citado el: 16 de 2 de 2011.] <http://www.profesional.com/>.

pruebasdesoftware. [En línea] [Citado el: 10 de 2 de 2011.] <http://www.pruebasdesoftware.com/>.

**Puglla Remache, Gabriela Noemí y León Quiñones, Lorena del Cisne. 2004.** *Métricas de Proceso y Proyect*. Ecuador : s.n., 2004.

**Rivero, Lanny Canino y Rodríguez, Meydi E Martínez. 2008.** *Estrategia de Aseguramiento de la calidad para el Simulador Quirúrgico*. Ciudad de la Habana, Cuba : s.n., 2008.

**Ruilova, María Esther Rojas. 2008.***Informe Ejecutivo.Métricas del Producto para el Software.* 2008.

**Santos Hernández, Vismar. 2009.***Estudio sobre la industria del Software a nivel mundial. Caracterización en América Latina y Cuba.* La Habana, Cuba : s.n., 2009.

**SCALONE, FERNANDA. 2006.***ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE.* Buenos Aires : s.n., 2006.

SEMINARIO DE FORMACION ESPECIALIZADA. [En línea] [Citado el: 4 de 3 de 2011.]

<http://www.fernandotazon.com.es/wp-content/uploads/2009/05/seminario-validacion-de-procesos1.pdf>.

Software Quality Systems. [En línea] [Citado el: 4 de 3 de 2011.] <http://www.sqs.es/es/services/validation.php>.

**Vega Lebrún, Carlos, Rivera Prieto, Laura Susana y García Santillán, Arturo. 2008.***MEJORES PRÁCTICAS PARA EL ESTABLECIMIENTO Y ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE.* Boca del Río, Ver : s.n., 2008.

**Zibert van Gricken, Carolina. 2004.** carolina.terna.net. [En línea] 2004. [Citado el: 2010 de noviembre de 9.] [http://carolina.terna.net/ingsw3/datos/Semana7\\_CMMI2.0.pdf](http://carolina.terna.net/ingsw3/datos/Semana7_CMMI2.0.pdf).