

**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS  
CALIDAD DE SOFTWARE FACULTAD 4**



**Título: Procedimiento para el Aseguramiento de la Calidad en el  
Proyecto INSIGNE.**

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

**Autoras:**

Yalina Jiménez Pérez

Maidelyn Rodríguez Perdomo

**Tutora:** Ing. Liodine Miyashiro Pérez.

Ing. Dialexis Acosta Molina.

## DECLARACION DE AUDITORIA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 19 días del mes Junio del año 2008.

**Yalina Jiménez Pérez**

**Maidelyn Rodríguez Perdomo**

\_\_\_\_\_  
Firma del autor

\_\_\_\_\_  
Firma del autor

**Ing. Liodine Miyashiro Pérez**

\_\_\_\_\_  
Firma del tutor.

**Ing. Dialexis Acosta Molina**

\_\_\_\_\_  
Firma del tutor.



*En la carrera por la CALIDAD no hay línea de meta*  
*Kearns*

## AGRADECIMIENTOS

*Se que tal vez mañana sea muy tarde, por eso hoy dedico todos mis años de estudio a ustedes, para demostrarles que tan importante son para mi.*

*Primero a **Mima** que fue capaz de enfrentar la vida, decidir que yo debía existir, por hacerme una persona con mucha fe, por darme fuerzas, por hacerme capaz de lograr todo lo que me propongo y por poner a personas en mi camino que son mi razón de vivir.*

*A **Tita**, por ser la persona que más cerca he tenido en estos 5 años, en las buenas, en las malas, por apoyarme y convertirme en la persona que soy hoy, por demostrarme y quererme tanto como si fuera su hijita.*

*A **mi papa**, por estar estos 23 años a mi lado, por quererme tanto como a una verdadera hija.*

*A **Tomás** que es mi padre biológico y a mi hermana **Roxana**, a **Alejo** y a **Sabino**.*

*A **mamita**, **Noel**, **Ofelia**, **Mayi**, **Juliet**, **Pedro**, **Mirella**, **Yusniel** por demostrarme que la familia no es quien lleva nuestra sangre, por el afecto y el cariño que me han dado.*

*A **mis tutoras**, a **Anilén**, a los **vecinos** del barrio, a mis amistades de la universidad, en especial a **Yadier**.*

*A **Yeslaine** y a **Ramsés** porque sin su apoyo no hubiese realizado este trabajo de diploma.*

*A **William** por hacerme ver que la vida no se acaba, siempre hay ilusiones por vivir.*

*Y por último dedico este trabajo de diploma a mi compañera de tesis **Yalina** por el esfuerzo y su forma de enfrentar la vida.*

**MAIDELYN**

*Primero agradecer a los que no se encuentran junto a mí pero llevo en mi corazón:*

*A mi **mamita** por su cariño y su amor, por apoyarme y creer en mí, por ser mi estrella y mi razón de ser.*

*A mi **papito** por enseñarme a luchar por lo que quiero, por hacer de mí una mejor persona y ser el mejor padre del mundo.*

*A mi tío **Isidro** por quererme tanto, por ser tan especial para todos y ser un ideal a seguir.*

*A los que han llenado mi vida de alegría y amor:*

*A mi hermana **Aliena** por ser el mejor apoyo que he tenido en estos últimos años y ser mas que mi hermana mi madre.*

*A mi hermanita **Dorita** por creer en mí, por estar siempre a mi lado y ser mi yoyi.*

*A mi tía **Teresa**, mi prima **Yunieth** y mi primito **Obdalan** por ayudarme y apoyarme en estos años de la carrera.*

*A mis **tíos** y **tías** por su preocupación y apoyo en estos últimos años.*

*A **Dariela** y **Dailin** por ser mas que mis amigas mis hermanas.*

*A **Ivette**, **Lily**, **Pedro Daimí**, **Elisa**, **Carmen**, **Francis** y **Yudi**, por soportar mi mal carácter y ser un gran sostén en mis momentos de tristeza.*

*A mis **tutoras** y a todos los que han sido mis compañeros de grupo.*

*A **Yeslaine** y a **Ramsés** por su gran aporte en la realización de esta tesis.*

*A **Maidelyn** por ser una buena compañera de tesis y mostrarme a creer en mis ideas.*

*A la vida por enseñarme que todo se logra con sacrificio, por hacerme fuerte antes los momentos difíciles y mostrarme que hay que levantarse después de cada golpe.*

*Por último a **todos** los que han hecho realidad este momento.*

**YALINA**

*...A mis padres y a mis hermanas por ser las personas más especiales que la vida me ha dado.*

*YALINA*

*... A mis padres, a Williams, a todas mis amistades, en especial a Tita y a la primera persona que conocí en esta universidad, Micho.*

*MAIDY*

### RESUMEN

En la actualidad la industria del software ha evolucionado de forma extraordinaria, cada día surgen ideas, tecnologías y herramientas nuevas. Existe tendencia al crecimiento del volumen y complejidad de los productos, los proyectos se atrasan, se exige mayor calidad y productividad en menos tiempo y hay insuficiente personal calificado.

En los proyectos productivos que se desarrollan en la universidad de las Ciencias Informáticas se detectan defectos en las fases finales del proceso de desarrollo del software, en algunos casos ocasionados por problema desde las primeras etapas, originando esto retraso en la entrega y molestias al cliente que espera recibir un producto en tiempo y con la calidad adecuada.

Estas razones llevan a establecer un conjunto de actividades para asegurar la calidad del software en el proceso de desarrollo, con el propósito de que se detecten errores en el momento de haber sido creados y exista una respuesta inmediata para corregir el problema.

Se realiza un estudio de los sistemas de gestión empresarial en el mundo y la tendencia a solucionar la gestión de los procesos de las entidades cubanas, debido a que en el país solo se gestionan los procesos contables-financieros en algunas empresas.

Se identifican las principales incidencias que afectan la calidad del software en la UCI, con el objetivo de mantener bajo control cada proceso y eliminar las causas de defectos, para ello se propone un procedimiento para asegurar la Calidad durante los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis, Diseño en el proyecto INSIGNE.

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>1 CAPITULO I.....</b>	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 TENDENCIAS EN EL MUNDO.....	5
1.3 ERP EN EL MUNDO.....	6
1.3.1 ¿Qué es un sistema ERP?.....	6
1.3.2 Módulos de un sistema ERP.....	7
1.3.3 Ventajas y Desventajas de los sistemas ERP.....	8
1.3.4 ERP en la Actualidad.....	8
1.3.5 El futuro de los sistemas ERP.....	9
1.4 CALIDAD.....	9
1.4.1 Análisis de las principales definiciones de calidad.....	9
1.4.2 Calidad del Software.....	10
1.4.3 Aseguramiento de calidad del software.....	11
1.4.4 Control de la Calidad del software.....	14
1.4.5 Verificación y validación.....	15
1.4.6 Revisiones del Software.....	15
1.4.7 Auditoría.....	16
1.4.8 Prueba.....	17
1.4.9 Métricas del software.....	18
1.5 GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE.....	22
1.6 NORMA ISO (INTERNATIONAL STANDARD ORGANIZATION).....	23
1.7 MODELOS DE CALIDAD DEL SOFTWARE.....	24
1.7.1 CMMI.....	24
1.7.2 SPICE (ISO 15504).....	26
1.7.3 Comparación entre modelos.....	27
1.8 PROCESO DE DESARROLLO DE SOFTWARE.....	28
1.8.1 Proceso Unificado de Rational (Rational Unified Process- RUP).....	30
1.8.2 Flujo de Trabajo Modelamiento del Negocio.....	30
1.8.3 Flujo de Trabajo Requerimientos.....	34
1.8.4 Flujo de Trabajo Análisis y Diseño.....	36
1.9 CONCLUSIONES PARCIALES.....	40
<b>2. CAPÍTULO II.....</b>	<b>41</b>
2.1. INTRODUCCIÓN.....	41
2.2. PRINCIPALES PROBLEMAS QUE AFECTAN LA INDUSTRIA CUBANA DEL SOFTWARE.....	41
2.3. ANÁLISIS DE LOS SISTEMAS ERP EN CUBA.....	42
2.3.1. Situación en Cuba.....	42
2.3.2. Sistemas Contables-Financieros.....	43
2.3.3. ERP y Software Libre.....	47
2.3.4. Importancia del Sistema ERP en Cuba.....	47
2.4. CARACTERÍSTICAS FUNDAMENTALES DEL DESARROLLO DE SOFTWARE EN LA UCI.....	48
2.4.1. Descripción del Objeto de Estudio.....	48
2.4.2. Características de la facultad.....	49
2.4.3. Resultados de proyectos comunes en la UCI.....	51
2.5. ANÁLISIS DE LOS RESULTADOS DE LAS ENCUESTAS REALIZADAS POR LA DIRECCIÓN DE CALIDAD DE SOFTWARE.....	53
2.5.1. Análisis de las variables de control de la encuesta.....	54
2.5.2. Análisis de los resultados relacionados con la calidad de los proyectos.....	54
2.5.3. Principales problemas identificados en la encuesta.....	57
2.6. CONCLUSIONES PARCIALES.....	58
<b>3 CAPITULO III.....</b>	<b>59</b>

3.1.	INTRODUCCIÓN.....	59
3.2.	RESUMEN.....	60
3.3.	NOMBRE DEL PROCEDIMIENTO.....	60
3.4.	OBJETIVO.....	60
3.5.	ALCANCE.....	60
3.6.	REFERENCIAS.....	60
3.7.	RESPONSABLES.....	60
3.8.	TÉRMINOS Y DEFINICIONES.....	61
3.9.	NORMAS GENERALES.....	62
3.10.	DESARROLLO.....	63
3.10.1.	<i>Establecer el grupo de aseguramiento de la calidad y sus funciones.....</i>	63
3.10.2.	<i>Identificar estándares, procedimientos y lineamientos a utilizar.....</i>	66
3.10.3.	<i>Establecer y mantener el Plan de aseguramiento de la calidad.....</i>	68
3.10.4.	<i>Establecer y mantener listas de chequeos.....</i>	70
3.10.5.	<i>Desarrollar actividades de control de la calidad.....</i>	70
3.10.6.	<i>Efectuar el registro de los productos de trabajo que surgen como resultado del proceso de aseguramiento de la calidad.....</i>	76
3.10.7.	<i>Revisar los resultados de las actividades de aseguramiento de la calidad con la gerencia del proyecto y los involucrados.....</i>	76
3.10.8.	<i>Desarrollar acciones correctivas.....</i>	77
3.11.	VALIDACIÓN. MÉTODO DELPHI.....	77
3.12.	CONCLUSIONES PARCIALES.....	80
	<b>CONCLUSIONES GENERALES.....</b>	<b>81</b>
	<b>RECOMENDACIONES.....</b>	<b>83</b>
	<b>REFERENCIAS BIBLIOGRAFICAS.....</b>	<b>84</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>88</b>
	<b>ANEXOS.....</b>	<b>92</b>
	<b>ANEXO 1.</b> TABLA 1. COMPARATIVA ENTRE MODELOS CMMI Y SPICE(15504).....	92
	<b>ANEXO 2.</b> TABLA 2. FLUJO DE TRABAJO MODELAMIENTO DEL NEGOCIO.....	93
	<b>ANEXO 3.</b> TABLA 3. FLUJO DE TRABAJO REQUERIMIENTOS.....	94
	<b>ANEXO 4.</b> TABLA 4. FLUJO DE TRABAJO ANÁLISIS Y DISEÑO.....	95
	<b>ANEXO 5.</b> TABLA 5. ROLES ENCUESTADOS.....	96
	<b>ANEXO 6.</b> ESPINA DE PESCADO.....	97
	<b>ANEXO 7.</b> TABLA 6 DE TAREAS Y RESPONSABILIDADES.....	99
	<b>ANEXO 8.</b> TABLA 7 DE ESTÁNDARES Y GUÍAS.....	99
	<b>ANEXO 9.</b> TABLA 8 DE CHEQUEO.....	99
	<b>ANEXO 10.</b> FLUJO DEL PROCEDIMIENTO.....	100
	<b>ANEXO 11.</b> ENCUESTA PARA DETERMINAR EL COEFICIENTE DE COMPETENCIA DE LOS EXPERTOS.....	104
	<b>ANEXO 12.</b> CUESTIONARIO PARA EXPERTOS.....	105
	<b>GLOSARIO DE TÉRMINOS.....</b>	<b>108</b>

### INTRODUCCIÓN

Durante los primeros años de la informática, el software se consideraba como un añadido. La programación era un "arte", para el que no existían metodologías, era un proceso que se realizaba sin ninguna planificación. A partir de mitad de la década de 1960, se estableció el software como producto y aparecieron las empresas dedicadas al desarrollo y distribución masiva del mismo.

Era necesario aplicar los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los problemas de desarrollo de software. Razón para ver la ingeniería como la búsqueda de calidad, la ingeniería del software como la producción de software de calidad y el proceso de ingeniería de software como un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad.

En el mercado internacional de la actualidad, la calidad es el objetivo supremo en tanto los fabricantes se enfrentan a mayor competencia para obtener clientes. La presión para reducir los tiempos y los errores hace que resulte indispensable para las compañías establecer y mantener criterios de garantía de calidad que cumplen en su totalidad con las normatividades de la industria. Para seguir siendo competitivos, los fabricantes necesitan optimizar su ingeniería y los ciclos de vida de los productos, así como aprovechar las cadenas de suministro para introducir productos superiores en el mercado a tiempo y dentro del presupuesto.(INFOR, 2008)

Los sistemas informáticos ERP que permiten centralizar la gestión integral de las entidades empresariales y presupuestadas de un país son un factor estratégico en la promoción de la Industria Cubana del Software a nivel internacional y su desarrollo en el ámbito de las nuevas tecnologías de la, informática y las comunicaciones.

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezca la falibilidad humana son comunes. Los errores pueden empezar a darse desde el primer momento del proceso en el que los objetivos pueden estar especificados de forma errónea e imperfecta; así en los posteriores pasos del diseño y desarrollo. Debido a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad del producto final.

La Universidad de las Ciencias Informáticas persigue como objetivos centrales, el desarrollo de la Industria Cubana del SW, los servicios informáticos y la exportación de SW, razón para que el producto que se le ofrezca al cliente tenga una calidad requerida. Durante el proceso de desarrollo de software en la UCI se presenta como **situación problemática:**

- ✓ No existe una adecuada planificación del trabajo.

- ✓ No se lleva el registro de quien hizo cada actividad relacionada con el proyecto.
- ✓ No se detectan el o los problemas justo después de haber sido creados.
- ✓ No se miden los procesos, ni los productos con métricas rigurosas.
- ✓ No se lleva un control de los defectos encontrados.
- ✓ Existe falta de comunicación efectiva entre los involucrados.
- ✓ En muchos casos los proyectos se entregan fuera de fecha.
- ✓ No se ha probado un procedimiento a seguir para llevar a cabo las actividades de aseguramiento de la calidad.

Se detectó el siguiente **problema científico**: ¿Cómo asegurar la calidad del proceso de desarrollo del proyecto INSIGNE en los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño?

Para darle solución a este problema se planteó el siguiente **objeto de estudio**: Estrategias o procesos para el aseguramiento de la calidad en los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño en el proceso de desarrollo del proyecto INSIGNE.

El **campo de acción**: Aseguramiento la calidad en el proceso de desarrollo de las Líneas del proyecto INSIGNE.

Para el desarrollo del trabajo se define como **objetivo general**: Proponer un procedimiento para el Aseguramiento de la calidad en los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño del proyecto INSIGNE.

Para el cumplimiento del objetivo propuesto se plantean los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación.
2. Efectuar un análisis de la tendencia de los ERP en Cuba.
3. Realizar un análisis de los resultados de las encuestas realizadas por Dirección de Calidad de Software.
4. Elaborar la propuesta de procedimiento de aseguramiento de la calidad en el Proyecto INSIGNE.
5. Validar el procedimiento de aseguramiento de la calidad propuesto.

Llevando a cabo las siguientes **tareas de investigación**:

1. Investigar sobre los ERP en el mundo y su incidencia en la empresa actual.
2. Analizar las definiciones de calidad, calidad del software, gestión de la calidad del software, aseguramiento de la calidad, control de calidad, verificación y validación de software, las

revisiones del Software, la auditoria, las pruebas, las métricas del software y la gestión de configuración del software.

3. Analizar los estándares reconocidos para el aseguramiento de la calidad como las normas de calidad de software ISO 9000 y los modelos de evaluación de calidad del software CMMI y SPICE.
4. Determinar los procesos que gestionan la calidad de software en la UCI.
5. Obtener las mejores prácticas para garantizar el aseguramiento de la calidad en los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño.
6. Analizar de los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño el propósito, la relación con otras disciplinas, los trabajadores, las actividades que realizan y los artefactos que se obtienen.
7. Realizar una investigación de los ERP en Cuba y su influencia en el desarrollo de la empresa cubana.
8. Evaluar los principales problemas de calidad existentes en la UCI a través del análisis los resultados de las encuestas realizadas por la Dirección de Calidad de Software.
9. Determinar las causas que afectan la calidad de software en la UCI.
10. Determinar las actividades necesarias para garantizar la calidad de los entregables para los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño.
11. Adaptar las mejores prácticas de aseguramiento de la calidad en el procedimiento propuesto.
12. Seleccionar el equipo de expertos.
13. Realizar una entrevista para validar el procedimiento de aseguramiento de la calidad propuesto.
14. Analizar los resultados obtenidos en las entrevistas.

Durante el desarrollo del presente trabajo se espera como **posible resultado**: Proponer un procedimiento para asegurar la calidad en el proyecto INSIGNE con el objetivo de evaluar y controlar la calidad durante el proceso de desarrollo del software en los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño.

Cada cliente tiene necesidades particulares, por ello que el enfoque del asesoramiento tendrá en cuenta la particularidad del Proyecto INSIGNE para contribuir a agregar valor a las actividades, más allá del simple cumplimiento de los requisitos formales del sistema.

Los **Métodos Teóricos** de gran importancia en el proceso de la investigación, en la construcción y desarrollo de la teoría científica y en el enfoque general para abordar los problemas de la ciencia. El

**Analítico–Sintético**, para lograr una mejor comprensión del fenómeno, es decir, obtener los elementos más importantes relacionados con el objeto de estudio y profundizar en la esencia del mismo. El **Inductivo–Deductivo** como razonamiento para permitir dar solución al problema y así pasar del conocimiento general a otros de menor nivel de generalidad.

Los **Métodos Empíricos**, la **Entrevista** a especialistas funcionales del proyecto para conocer la necesidad que existe en el país de tener un ERP, también al Jefe de Proyecto y Jefe de Línea de Calidad con el objetivo de identificar aspectos importantes a valorar para que el proyecto se desarrolle con calidad, además a especialistas de la Dirección de Calidad de Software, con el objetivo de indagar e intercambiar criterios de lo que se hace en materia de calidad en la universidad, que permitan evaluar el estado de estos procesos. El análisis de la **Encuesta** realizada por parte de la Dirección de Calidad de Software a Directivos, Roles principales y Desarrolladores para identificar los problemas de calidad que existen específicamente en los proyectos de la Facultad 4.

La tesis consta de un Capítulo I en el que se hace un estudio sobre los temas de ERP en el mundo y su tendencia en la empresa mundial actual, la Calidad, Calidad del Software, Gestión de la calidad, Aseguramiento de la Calidad, Control de la calidad, Auditorias, Pruebas, Métricas, Verificación y validación, Revisiones, Normas ISO 9000 y modelos de evaluación de la calidad de software CMMI y SPICE, flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño definidos por RUP.

El Capítulo II, contiene un estudio de los ERP en Cuba, la necesidad del uso de estos en la gestión empresarial de las entidades cubanas, estudio de los sistemas contables existentes en el país, una caracterización de lo que se hace en materia de producción en la UCI, los problemas de la calidad en la UCI y particularmente de la Facultad 4, obtenidos del análisis de la encuesta aplicada por la Dirección de Calidad de Software a Directivos, Roles principales y Desarrolladores.

En el Capítulo III se describe la propuesta de procedimiento para el aseguramiento de la calidad del software en los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño en el proyecto INSIGNE. Contiene la validación que se le hace al procedimiento propuesto.

# CAPITULO 1

## FUNDAMENTACIÓN TEORÍA

### 1 CAPITULO I.

#### 1.1 Introducción.

En el presente capítulo se realiza un estudio de las tendencias mundiales que existen en cuanto al desarrollo de software y se observa desde el punto de vista de los ERP. Además se analizan características importantes de estos sistemas y su influencia en la empresa actual.

Se hace referencia a definiciones esenciales que se deben conocer para llevar a cabo el procedimiento, como son calidad, calidad de software, gestión de la calidad del software y sus actividades esenciales aseguramiento de la calidad y control de calidad. También se analiza de forma general la verificación y validación, las revisiones, la auditoria, las pruebas, las métricas y la gestión de configuración del software.

Se hace un análisis de los conceptos escogidos de los aspectos a tratar y de la importancia que tiene el tema de calidad en la realización de un proyecto software.

Se realiza un estudio de los estándares reconocidos a nivel internacional como las normas ISO 9000 de aseguramiento de calidad y los modelos de calidad CMMI y SPICE.

Efectuar un análisis para cada uno de los flujos de trabajo definidos por RUP es muy engorroso pues un ERP es un software de gestión empresarial muy grande, razón para considerar el estudio de los conceptos básicos de los flujos de trabajos Modelamiento del Negocio, Requerimientos, Análisis y Diseño. Orientado a los artefactos obtenidos al finalizar cada flujo en cada una de las fases de desarrollo del software. También se muestran los propósitos, relaciones de estos flujos con otros, las actividades y los trabajadores que la realizan.

#### 1.2 Tendencias en el mundo.

Las tendencias comerciales actuales y futuras obligan a las empresas a ser cada vez más competitivas; para ello es necesario que éstas tengan optimizados e integrados todos sus flujos internos de información y sus relaciones comerciales externas. Además, deben conseguirse los objetivos estratégicos como son las mejoras de la productividad, la calidad, el servicio al cliente y la reducción de costes. Las tecnologías de la información han permitido, en gran medida, la consecución de dichos objetivos. En esta área, podemos reseñar la aportación de los ERP sistemas a favor de la planificación de recursos empresariales.(PYMETICA, 2007)

Antes de la aparición de los ERP, las empresas guardaban información importante para el negocio en los sistemas informáticos de los diferentes departamentos. Cada departamento utilizaba habitualmente

diferentes sistemas y técnicas para gestionar esta información. A veces, la información se encontraba duplicada dentro de la organización, sin ser necesariamente idéntica o estar igualmente actualizada. Otras veces, mucha de esta información sólo existía en papel, lo que dificultaba mucho el acceso a esta información por parte de toda la organización.(ELECTRÓNICO, 2007)

### 1.3 ERP en el mundo.

#### 1.3.1 ¿Qué es un sistema ERP?

Un ERP<sup>1</sup> definido por Deloitte y Touche como un sistema de software de negocios que permite a las compañías:(RODRÍGUEZ, 2007)

- Automatizar e integrar la mayoría de sus procesos.
- Compartir datos comunes y prácticas a través de toda la empresa.
- Producir y acceder a la información en tiempo real.

ERP es una forma de integrar los datos y procesos de una organización en un único sistema.

Según otros autores un sistema ERP es el conjunto de actividades apoyadas por un programa multimodular, que ayuda a los fabricantes y otro tipo de negocios a gestionar las partes importantes de su empresa, incluyendo planificación de la producción, suministro de material, gestión de inventario, gestión de pedidos, etc. También incluye módulos para la gestión financiera y de los recursos humanos.(BANESPYME, 2006)

Los sistemas ERP se caracterizan por su gran capacidad de adaptación, de modularidad, de integración de la información (introducir los datos una sola vez; es decir introducción de un dato único) de universalidad, de estandarización e interfaces con otro tipo de programas. Son sistemas abiertos y multiplataforma.(PYMETICA, 2007)

Los sistemas ERP por lo general han incluido muchos de los componentes de hardware y software, con el fin de integrar toda la información de la empresa utilizada para diferentes funciones y departamentos en un mismo sistema informático lo que permite a los trabajadores de diferentes departamentos poder acceder a la misma información.

“Un sistema ERP integra toda la información de la empresa utilizada para diferentes funciones y departamentos en un mismo sistema informático. La unificación que implica un sistema ERP puede aportar beneficios significativos:

- Un acceso más completo a la información.
- Mayor rapidez y eficiencia.
- Disminución de los errores.

---

<sup>1</sup> Enterprice Resource Planning.

“Los sistemas de planeación de recursos empresariales cuentan con 3 características que los diferencian de las aplicaciones de gestión que pueden encontrarse en el mercado hoy en día y es que son sistemas Integrales, Modulares y Adaptables:

**Integrales** porque ven a las diferentes áreas o departamentos de una entidad unidos por la información que generan, por este motivo la base de datos que utilizan son centralizadas y evitan la duplicación de datos.

**Modulares** porque define por separado los procesos que se realizan en estas áreas o departamentos de manera tal que se utilizan solo los módulos que la empresa necesita para su trabajo. Por ejemplo existen módulos de contabilidad, control de inventarios, recursos humanos, costos y procesos, facturación, nómina, etc. utilizados de manera independiente por cada departamento y pueden ser adquiridos o no en dependencia de las necesidades de los clientes.

**Adaptables** porque todas las empresas no realizan los mismos procesos de negocios de igual forma incluso cuando las analizamos dentro de un mismo sector económico, por este motivo los ERP proveen a los usuarios la posibilidad de adaptarlos a sus necesidades mediante una alta parametrización de sus funciones.”(BRITO, 2006)

Cuando mejora el acceso a la información, los trabajadores y directivos tienen un mejor conocimiento de lo que está pasando en la empresa, hecho que apoya la toma de mejores decisiones comerciales.”(BANESPYME, 2006)

### 1.3.2 Módulos de un sistema ERP.

Un sistema ERP precisa estar compuesto por un conjunto básico de módulos. Como mínimo tiene que ofrecer soluciones de gestión de la producción, financieras, de compras, ventas y logística, satisfaciendo todas las necesidades de las empresas en estos aspectos.

Además de los módulos mencionados, puede contener otros, como pueden ser los de gestión de recursos humanos y de proyectos. De hecho, en este momento las soluciones de sistemas ERP que hay en el mercado incorporan cada vez más funciones para la automatización de la fuerza de ventas, inteligencia de negocios (*Business Intelligence*<sup>2</sup>), gestión de relaciones con clientes (CRM<sup>3</sup>) y optimización de la cadena de suministros (SCM<sup>4</sup>). Aunque se suele considerar al sistema CRM<sup>3</sup> y a la SCM<sup>4</sup> aplicaciones aparte, dada su potencia y alcance necesitan un sistema ERP para poder convivir y como fuente de información.(BANESPYME, 2006)

---

<sup>2</sup> Inteligencia empresarial o inteligencia de negocios.

<sup>3</sup> Customer Relationship Management.

<sup>4</sup> Supply chain management.

### 1.3.3 Ventajas y Desventajas de los sistemas ERP.

“Algunas de las ventajas de implementar un ERP son las siguientes:

- Un sistema totalmente integrado.
- La capacidad para racionalizar los diferentes procesos y flujos de trabajo.
- La posibilidad de compartir datos entre distintos departamentos de una organización.
- Mejora de la eficiencia y los niveles de productividad.
- Mejor seguimiento y previsión.
- Reducción de los costes.
- Mejora el servicio al cliente

Pero definitivamente no todo lo que rodea a un ERP es bueno, también tienen sus desventajas. Aunque por lo general las ventajas superan a las desventajas para la mayoría de las organizaciones la aplicación de un sistema de ERP, aquí están algunos de los obstáculos más comunes:

- Personalización en muchas situaciones es limitada.
- La necesidad de los procesos de negocio reengineer<sup>5</sup>.
- Sistemas ERP puede ser un costo prohibitivo para instalar y ejecutar.
- El apoyo técnico puede ser de mala calidad.
- ERP's pueden ser demasiado rígidos específicos para organizaciones que están ya sea nueva o desea mover en una nueva dirección en el futuro próximo.”(FAQ, 2007)

### 1.3.4 ERP en la Actualidad.

“En la actualidad, los sistemas ERP son la base de casi todas las empresas grandes y medianamente –grandes del mundo. Se configuran como una herramienta de gestión y otorgan una gran ventaja competitiva a las organizaciones.

Sin embargo, estos sistemas siguen evolucionando. La empresa Gartner inventó el término ERP, ha introducido recientemente un nuevo concepto: el de ERP-II. El concepto de ERP-II, se configura como un nuevo sistema que no se limita a la gestión interna de la empresa, sino que incorpora y comparte información de la cadena de suministros.

Los sistemas ERP-II también se caracterizan por el hecho de que no se centran sólo en la producción y la distribución, sino que están presentes en otros segmentos y sectores, como empresas de servicios, administraciones públicas, etc.

Técnicamente, los sistemas ERP-II han evolucionado desde sistemas monolíticos y cerrados hasta sistemas basados en Web y abiertos, que pueden integrarse e interoperar con otros sistemas. Los

---

<sup>5</sup> Reingeniería.

sistemas ERP-II se basan en sistemas modulares que permiten a los usuarios escoger los módulos con las funcionalidades que necesitan.”(BANESPYME, 2006)

### **1.3.5 El futuro de los sistemas ERP.**

El futuro de los sistemas ERP, tendrá que basarse cada vez más en la flexibilidad, que se consigue por medio de:(BANESPYME, 2006)

1. Migración incremental: Los sistemas ERP han sido tradicionalmente costosos de implantar en cuanto a tiempo y recursos. Para actuar de forma mucho más ágil, los proveedores proporcionarán productos más rápidos y fáciles de configurar, tratando de evitar proyectos largos.
2. Configuración dinámica de los sistemas ERP: Los grandes sistemas ERP se configuraban inicialmente y en ellos se introducían pocas variaciones. Actualmente, es difícil predecir los futuros cambios en el modelo de negocio, los procesos, los flujos de información, etc., lo que hará necesario sistemas de configuración dinámicos que permitan al personal no técnico ajustar cambios en las funcionalidades sin variar la lógica con que fue creado el ERP.
3. Tendencia a ofrecer componentes: Hasta ahora los ERP ofrecían módulos independientes. La tendencia actual consiste en ofrecer componentes que pueden trabajar de forma independiente y que sean capaces de integrarse eficientemente con otros (sin importar el origen).
4. Gestión de las relaciones con proveedores, socios y clientes: Anteriormente se ha referenciado a la cadena de suministros y de cómo las relaciones con todos los actores del mercado se van volviendo cada vez más estrechas. La posibilidad de interconexión con el entorno siempre tiene que estar presente.

### **1.4 Calidad.**

#### **1.4.1 Análisis de las principales definiciones de calidad.**

La Calidad tiene muchas definiciones, de forma general es la satisfacción de las necesidades y cumplimiento de las especificaciones y fabricación para cumplir con el propósito del cliente. El término "Calidad" siempre será entendido de diferente manera por cada persona, ya que para unos la Calidad reside en un producto y en otros en su servicio posventa de este producto. A continuación presentamos algunas definiciones:

Conway, consultor de Calidad, en 1988 plantea que la calidad se alcanza al “desarrollar la fabricación, administración y distribución a bajo costo de productos y servicios que el cliente quiera o necesite.” Conway en su definición hace referencia a la necesidad de observar la calidad del trabajo y desarrollar un sistema adecuado para obtenerla. (CONWAY, 1988)

Feigenbaum, presidente de la Academia internacional de Calidad en 1996 define la calidad como “Un sistema eficaz para integrar los esfuerzos de mejora de la gestión de los distintos grupos de la

organización para proporcionar productos y servicios a niveles que permitan la satisfacción del cliente.” (FEIGENBAUM, 1996)

Ishikawa, Ingeniero Químico, pionero e ideólogo indiscutible de los éxitos de la industria japonesa en materia de calidad, en 1988 manifiesta que “calidad es aquella que cumple los requisitos de los consumidores” e incluye el costo entre otros requisitos. (ISHIKAGUA, 1988)

Capacidad del software para satisfacer las necesidades de los usuarios clasificando diferentes aspectos en categorías conocidas como parámetros. (JURAN, 1970).

Conformidad o cumplimiento de los requisitos. (CROSBY, 1979)

La ISO 9000 plantea que la calidad es:” Grado en el que un conjunto de características inherentes cumple con los requisitos.” Añade dos notas:

Nota 1.El término calidad puede utilizarse acompañado de adjetivos tales como pobre, buena o excelente.

Nota 2.” Inherente” en contraposición a “asignado” significa que existe en algo, especialmente como una característica permanente. (TOIRAN, 2006)

En este caso la calidad depende de los requisitos que se planteen por los productores y si es cierto que los mismo satisfacen las necesidades de los clientes.

Después de un análisis de las diferentes definiciones se puede decir que la Calidad se determina en función de las características que hacen competitivo un producto al satisfacer las necesidades implícitas y explícitas de los usuarios finales.

### **1.4.2 Calidad del Software.**

El incremento de la necesidad por la calidad y la eficiencia en el software ha sido la causa para que los procesos de desarrollo del software actuales centren su atención en la calidad. Su trascendencia es tal que la calidad ya no es una ventaja competitiva, sino una condición necesaria que deben cumplir las empresas de desarrollo para sobrevivir en el mercado.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro.(CARRASCO, 1995). Puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño.

“Conjunto total de las características del producto de marketing, ingeniería, fabricación y mantenimiento a través del cual el producto en uso satisfará las expectativas del cliente” (JÁUREGUI, 2003)

Según Pressman 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 esperan de todo software desarrollado profesionalmente”. (PRESSMAN, 2002)

“Grado con el cual el cliente o usuario percibe que el software satisface sus expectativas” (IEEE 1028-1997)

“Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades explícitas o implícitas.” (GARCÍA, 2003)

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software. (CARRASCO, 1995)

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico. El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software. El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software. El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado. (CARRASCO *et al.*, 1995)

Son muchos los conceptos propuestos por los diferentes autores sobre calidad de software. Por la precisión de las ideas expuestas por Pressman se considera su definición como guía para la confección del procedimiento.

### **1.4.3 Aseguramiento de calidad del software.**

Gestión de la calidad: “Aspecto de la función de gestión que determina y aplica la política de la calidad, los objetivos y las responsabilidades y que lo realiza con medios tales como la planificación de la calidad, el control de la calidad, la garantía de calidad y la mejora de la calidad”. La gestión de la calidad es responsabilidad de todos los niveles ejecutivos, pero debe estar guiada por la alta dirección. Su realización involucra a todos los miembros de la organización. Para gestionar la calidad se realizan dos actividades fundamentales el control de la calidad, el aseguramiento de la calidad.(BATISTA, 2007)

El aseguramiento de la calidad o SQA por sus siglas en inglés (Software Quality Assurance) es una actividad que se aplica a nivel de proyecto durante todo el proceso de desarrollo de software y cada persona involucrada en esta actividad tiene un impacto en la calidad del software resultante.

De acuerdo a la Norma ISO 9000:2000 (GARCÍA, 2003) se define como Aseguramiento de la Calidad al “conjunto de actividades preestablecidas y sistematizadas, aplicadas al sistema de calidad, que ha

sido demostrado que son necesarias para dar confianza adecuada de que un producto o servicio satisface los requisitos para la calidad”.

CMMI v1.2 define el aseguramiento de la calidad como: Un medio planificado y sistemático para asegurar (por parte de la administración) que los estándares, prácticas, procedimientos y métodos definidos sean aplicados.(CMMI, 2006)

El aseguramiento de la calidad como actividad de protección está presente en los métodos y herramientas de análisis, diseño, programación, prueba, inspecciones técnicas formales, ajustarse a los estándares, control de la documentación del software y de los cambios realizados durante todo el proceso de desarrollo.

Todo el que este involucrado en el proceso de desarrollo del software es responsable de la calidad, desarrolladores, analistas, arquitectos, jefes de proyectos, clientes y aquellas personas que en los proyectos llamamos grupo de aseguramiento de la calidad.

El aseguramiento de la calidad, como actividad de soporte en el proceso de desarrollo, comprende procedimientos para la aplicación efectiva de métodos y herramientas, revisiones y estrategias de prueba, control de cambios, aseguramiento de ajuste a los estándares entre otras. El Instituto de Ingeniería del Software (SEI) recomienda un conjunto de actividades como: (1) establecimiento del plan de aseguramiento de la calidad para un proyecto; (2) participación en el desarrollo de la descripción del proceso de software; (3) revisión de las actividades de ingeniería del software; (4) auditorías de los procesos de software designados para verificar el ajuste con los definidos como parte del proceso de software; (5) registrar lo que no se ajuste a los requisitos e informar a los superiores; (6) coordinar el control de cambio.(PRESSMAN, 2002) CMMI v1.2 en el área de proceso de aseguramiento de la calidad propone: (1) elaborar objetivamente los procesos; (2) evaluar objetivamente los artefactos y servicios; (3) comunicar y asegurar la resolución de las no conformidades; (4) establecer registros.(CMMI, 2006)

Desde el punto de vista de la eficacia, el aseguramiento de la calidad implica generalmente, una evaluación permanente de aquellos factores que influyen en la adecuación del proyecto y de las especificaciones a las aplicaciones previstas y además, la verificación y la auditoría de las operaciones de producción, de instalación y de inspección. Para proporcionar la debida confianza, puede ser preciso que se aporten las pruebas oportunas.

### **1.4.3.1 Plan de Aseguramiento de la Calidad.**

El Plan de aseguramiento de la calidad proporciona un mapa para institucionalizar la garantía de la calidad del software. Para adaptar las directrices marcadas por los sistemas de calidad a cada

proyecto se debe generar un plan específico a cada uno de ellos: El plan de aseguramiento de la calidad.

El plan de aseguramiento de calidad generado por un grupo de aseguramiento de un proyecto sirve como guía para el desarrollo e institución de las actividades de aseguramiento de la calidad. Un plan de aseguramiento debe contener por lo general:

- Objetivos de la calidad del proyecto y su enfoque para su consecución.
- Documentación referenciada en el plan.
- Gestión del aseguramiento de la calidad.
- Documentación de desarrollo y de control o gestión.
- Estándares, normas o prácticas que hay que cumplir.
- Actividades de revisión y auditorías.
- Gestión de la configuración del software.
- Informes de problemas
- Pruebas
- Herramientas, técnicas y métodos de apoyo.
- Recogida, mantenimiento y almacenamiento de datos sobre la documentación de actividades de aseguramiento de la calidad realizadas en el proyecto.

IEEE proporciona un estándar (IEEE 730-1998) para elaborar un plan de aseguramiento. En las secciones iniciales se describen el propósito y alcance del documento. La sección de Gestión del plan describe la situación del aseguramiento de la calidad dentro de la estructura organizativa; las tareas y responsabilidades del Aseguramiento de la calidad y su emplazamiento a lo largo del proceso de software. La sección de Documentación describe cada uno de los productos de trabajo utilizados por el plan. Los Estándares y Guías muestran los estándares/prácticas que se aplican durante el proceso de software. Las sección Revisiones y Auditoría del plan identifican las revisiones y auditorías que se van a llevar a cabo por el grupo de aseguramiento de la calidad y el cliente. La sección de Pruebas hace referencia al plan de pruebas del software. El resto del plan identifica las herramientas y métodos que soportan las actividades y tareas del aseguramiento de la calidad. (PRESSMAN, 2002)

### **1.4.3.2 Equipo de Aseguramiento de la Calidad.**

El grupo de SQA sirve como representación del cliente en casa. Es decir, la gente que lleva a cabo la SQA debe mirar al software desde el punto de vista del cliente. Tiene la responsabilidad de la planificación de garantía de calidad, supervisión, mantenimiento de registros, análisis e informes.(PRESSMAN, 2002)

Por lo general, el equipo de Aseguramiento de la calidad es diferente del equipo de desarrollo, especialmente en proyectos grandes. En cuanto al tamaño de este equipo, promedio esta en una persona de Aseguramiento de la Calidad por cada 15 a 40 personas en el equipo desarrollo.

### **Papel del Equipo de Aseguramiento de la Calidad.**

Las reglas del grupo de SQA tratan de ayudar al equipo de ingeniería del software en la consecución de un producto final de alta calidad. El equipo de SQA realiza una serie de actividades que facilitan el aseguramiento de la calidad:(PRESSMAN, 2002)

1. Establecimiento de un plan de SQA para un proyecto.
2. Participación en el desarrollo de la descripción del proceso de software del proyecto.
3. Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso de software definido.
4. Auditoría de los productos del software designados para verificar el ajuste con los definidos como parte del proceso de software.
5. Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
6. Registrar lo que no se ajuste a los requisitos e informar a sus superiores.

Además de estas actividades, el equipo de SQA coordina el control y la gestión de cambios y ayuda a recopilar y analizar las métricas del software.

### **1.4.4 Control de la Calidad del software.**

Controlar la calidad durante el proceso de desarrollo de un software es de vital importancia para la obtención de un software de calidad. Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom De Marco, "usted no puede controlar lo que no se puede medir".(CARRASCO *et al.*, 1995)

En este aspecto surge el término Control de la Calidad, la cual se concibe como: "Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio" (GARCÍA, 2003), (BUADES, 2002)

El control de la 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. (PRESSMAN, 2002)

El control de la calidad se relaciona con la vigilancia permanente de todo el proceso de desarrollo y el ciclo de vida del software. Se logra mediante la observación constante del cumplimiento de cada una de las fases y actividades involucradas en el proceso de desarrollo.(MENDOZA, ; OHLSSON y OLLFORS, 2001)

Para realizar un control de calidad deben ejecutarse frecuentes inspecciones a las metodologías de trabajo y al uso de las herramientas, revisiones de prototipos y de las pruebas formales de los productos finales. El control de la calidad permite realizar las rectificaciones necesarias a cualquier falla encontrada durante el proceso de desarrollo.(MENDOZA)

### 1.4.5 Verificación y validación

La validación o verificación del proceso de desarrollo de software pueden definirse como:

- **Verificación:** se refiere al conjunto de actividades que aseguran que el software implementa correctamente una función específica.
- **Validación:** se refiere a un conjunto diferente de actividades que aseguran que el software construido se ajusta a los requisitos del cliente.

La definición de verificación y validación comprende muchas de las actividades de garantía de la calidad. Dentro de las actividades ligadas al control de la calidad se incluyen las pruebas y procesos de revisión y auditorías.

### 1.4.6 Revisiones del Software.

Las revisiones del software son un <<filtro>> para el proceso de ingeniería del software. Las revisiones se aplican en varios momentos del desarrollo del software y sirven para detectar errores y defectos que puedan así ser eliminados. Las revisiones del software sirven para <<purificar >> las actividades de ingeniería del software que sucede como resultados del análisis, el diseño y la codificación.(PRESSMAN, 2002)

Las revisiones según el estándar ANSI/IEEE Std 1028-1988(IEEE, 1989) se define como: evaluación de un elemento para determinar diferencias con los resultados planeados y recomendar mejoras.

Existen varios tipos de revisiones que pueden llevarse a cabo durante el proceso de desarrollo de software.(PRESSMAN, 2002)

- **Reuniones Informales:** Una reunión que tiene lugar alrededor de una máquina, u otro espacio en cual se discuten problemas técnicos.
- **Presentación formal:** Una presentación formal de un diseño de software, una aplicación o cualquier producto de trabajo que se realiza bajo la audiencia de clientes, ejecutivos o personal técnico.
- **Revisiones técnicas formales:** Actividad desarrollada por los ingenieros del software que tiene como objetivo (1) descubrir errores en las función, la lógica o la implementación de cualquier representación de software; (2) verificar que el software bajo revisión alcanza los requisitos; (3) garantizar que el software ha sido representado de acuerdo a varios estándares predefinidos; (4) conseguir un software desarrollado de forma uniforme.

- Las revisiones Pass Around: Revisión informal en la cual el autor distribuye el documento a varias personas para su revisión. La efectividad de este método depende del conocimiento y motivación de los revisores.
- Las revisiones Peer check: Revisión informal en la que una persona además del autor revisa el documento. El éxito de este método depende exclusivamente de la motivación y conocimiento del revisor.

Para la realización de las revisiones es recomendable definir un proceso general aplicable. A continuación se muestran un conjunto de pasos que pueden tenerse en cuenta para el desarrollo de revisiones:

1. Realizar las revisiones de acuerdo al plan de proyecto.
2. Definir el producto a revisar.
3. Definir día y hora para la revisión.
4. Determinar que es necesario y quien debe hacerlo.
5. Definir el tipo de revisión a realizar.
6. Identificar a las personas que deben participar e invitarlas indicándole su responsabilidad en la revisión.
7. Si se realiza una reunión definir quien la organizará. Desarrollar una agenda de la reunión.
8. Definir que se debe hacer durante la revisión y quien debe hacerlo.
9. Definir los criterios de éxito para la revisión (cuando se puede decir que la revisión puede finalizar).
10. Identificar y almacenar los resultados que deben conservarse de la revisión.

### **1.4.7 Auditoría.**

La Auditoría es una función de dirección cuya finalidad es analizar y apreciar, con vistas a las eventuales las acciones correctivas, el control interno de las organizaciones, para garantizar la integridad de su patrimonio, la veracidad de su información y el mantenimiento de la eficacia de sus sistemas de gestión.

Una Auditoría consiste, generalmente, en realizar una investigación para determinar el grado de cumplimiento y la adecuación de los procedimientos, instrucciones, especificaciones, códigos, estándares u otros requisitos de tipo contractual establecidos y aplicables, y permite verificar, además, la efectividad y adecuación de la implementación realizada.

Las auditorías según el estándar ISO 19011:2002(ISO, 2002) se define como: proceso sistemático, independiente y documentado para evaluar el estado actual (evidencias de la auditoría) y evaluarlas de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría.

Existen diferentes clasificaciones para las auditorías, de acuerdo a varios criterios. A continuación se listan algunas de estas clasificaciones, teniendo en cuenta el objeto de aplicación de la Auditoría, sus objetivos y características:

- Auditoría del producto: El objetivo es cuantificar el grado de conformidad del producto con las características requeridas. Las auditorías del producto software más comunes son la auditoría Funcional y la auditoría Física.
- Auditoría del proceso: El objetivo es evaluar el proceso de desarrollo o de gestión, y evaluar su completitud y efectividad, determinando dónde se puede mejorar. En el desarrollo de software se suelen realizar dos tipos de auditorías del proceso:
  - Auditorías de proyecto: cuyo objetivo es evaluar la productividad y eficacia del equipo que trabaja en un proyecto así como la efectividad de los métodos y herramientas utilizados.
  - Auditorías de gestión de proyecto: cuyo objetivo es evaluar la efectividad de las prácticas de gestión realizadas y la organización del proyecto.
- Auditoría del sistema de calidad: El objetivo es evaluar la completitud y efectividad del propio sistema de calidad establecido.

### 1.4.8 Prueba.

Las pruebas son actividades en las cuales un sistema o sus componentes se ejecutan en circunstancias previamente especificadas, los resultados se observan, registran y se realiza una evaluación de algún aspecto. Las pruebas del software son un elemento crítico para la garantía de la 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.

Algunos principios de las pruebas recogen lo siguiente:

- Las pruebas se planifican mucho antes de que comiencen.
- No son posibles las pruebas exhaustivas: El número de permutaciones de camino para incluso programas pequeños es excepcionalmente grande. Por ese motivo es imposible ejecutar todas las combinaciones de caminos durante las pruebas.(PRESSMAN, 2002)
- Para ser más eficaces, las pruebas deberían ser realizadas por un equipo independiente: El ingeniero de software que creo el sistema no es el más adecuado para realizar las pruebas del software, ya que consciente o inconscientemente tiende a probar lo que sabe que funciona.

Existen varios tipos de pruebas que pueden realizarse durante el proceso de desarrollo de software como son:

- Unitarias: Pretenden probar cada función en un archivo de programa simple (una clase en terminología de objetos).
- Integración: Pretenden comprobar la integración de los componentes, es decir, la comunicación a través de interfaces, acceso incoherente a estructuras de datos globales.  
Las pruebas de integración pueden realizarse de forma ascendente o descendente
- Validación: Pretende comprobar que se satisfacen los requisitos.
- Sistema: Se centran en comprobar la recuperación, seguridad, resistencia, rendimiento.

Asociado a los tipos de pruebas existen también técnicas de pruebas que ayudan a definir conjuntos de casos de pruebas aplicando ciertos criterios, como son:

Pruebas de caja blanca: Se centra en comprobar la interacción interna de los componentes del sistema.

Pruebas de caja negra: “Se centran en los requisitos funcionales del software. O sea, la prueba de de caja negra permite al ingeniero del software obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos”.(PRESSMAN, 2002)

### 1.4.9 Métricas del software.

IEEE (IEEE, 1990) define métrica como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.

Las métricas pueden ser utilizadas para la evaluación de las características del software. Se definen como: Una serie de medidas o pasos que ayudan a definir con mayor exactitud el desarrollo y calidad de un producto.(PRESSMAN, 2002)

Las métricas pueden agruparse según varias clasificaciones (métricas orientadas al software, proyecto, proceso y personas de forma genérica), a continuación se enuncian algunas de las clasificaciones para las métricas.(NAPAL, 2007)

Métricas técnicas: Se centran en las características de software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho (no en cómo se obtiene).

Métricas de calidad: Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

- Métricas basadas en atributos internos del producto:
  - Medidas de estructuración de un programa
  - Métricas de complejidad
  - Métricas de cobertura de pruebas
  - Métricas de calidad del diseño

- Métricas basadas en atributos externos del producto:
  - Métricas de portabilidad
  - Métricas de defectos
  - Métricas de usabilidad
  - Métricas de mantenibilidad
  - Métricas de fiabilidad
- Métricas para sistemas orientados a objetos:
  - Métricas orientadas a clases.
  - Métricas orientadas a operaciones.
  - Métricas para pruebas orientadas a objetos.

Métricas de productividad: Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

Métricas orientadas a la persona: Proporcionan medidas e información sobre la forma en que las personas desarrollan el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que se harán del personal que va a realizar el sistema.

Métricas orientadas al tamaño: Para saber en qué tiempo se va a terminar el software y cuantas personas voy a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño.

Métricas orientadas a la función: Son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcularlas, las líneas de código (LDC), las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

De forma general las métricas permiten que los desarrollados y líderes de proyectos mejoren el proceso de software, planificación y evalúen la calidad del producto. Proporcionan además una visión estratégica de la efectividad de un proceso de software.

### **1.4.9.1 Métricas y Calidad.**

El principal objetivo de los ingenieros del software es producir un sistema, aplicación o producto de alta calidad, para lo cual emplean métodos y herramientas efectivas dentro del contexto de un proceso maduro de desarrollo del software y además deben desarrollar mediciones que den como resultado sistemas de alta calidad. Para obtener esta evaluación, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad.

### **NC ISO/IEC 9126.**

La norma cubana NC-ISO/IEC 9126 describe un modelo en dos partes para calidad de los productos de software: a) calidad interna y externa y b) calidad durante el uso. La primera parte del modelo especifica seis características para la calidad interna y externa, que son además divididas en subcaracterísticas que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software. Esta parte de la NCISO/ IEC 9126 no elabora el modelo para la calidad externa e interna a un nivel inferior al de las subcaracterísticas.(INFORMACIÓN, 2005)

**Funcionalidad:** Es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas.

Esta característica está relacionada con lo que hace el software para satisfacer las necesidades, al tiempo que las otras características principalmente están relacionadas con el cuando y cómo lo hace.

- **Idoneidad:** Capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.
- **Precisión.** Capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.
- **Interoperabilidad:** Capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.
- **Seguridad (informática):** Capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.
- **Conformidad con la funcionalidad:** Capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la funcionalidad.

**Confiabilidad:** La capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas.

- **Madurez:** Capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.
- **Tolerancia ante fallos:** Capacidad del producto de software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interfase especificada.
- **Recuperabilidad:** Capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.

- **Conformidad con la confiabilidad:** capacidad del producto de software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativos a la confiabilidad.

**Usabilidad:** capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.

- **Comprensibilidad:** capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares. (Esto dependerá de la documentación y la impresión inicial que ofrezca el software.)
- **Cognoscibilidad:** capacidad del producto del software para permitirle al usuario aprender su aplicación.
- **Operabilidad:** capacidad del producto del software para permitirle al usuario operarlo y controlarlo.
- **Atracción:** capacidad del producto del software de ser atractivo o amigable para el usuario.
- **Conformidad con la usabilidad:** capacidad del producto de software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.

**Eficiencia:** capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados usados, bajo condiciones establecidas.

- **Rendimiento:** capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.
- **Utilización de recursos:** capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas.
- **Conformidad de la eficiencia:** capacidad del producto de software de adherirse a las normas o convenciones que se relacionan con la eficiencia.

**Mantenibilidad:** capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.

- **Diagnosticabilidad:** capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
- **Flexibilidad:** capacidad del producto del software para permitir la aplicación de una modificación especificada.
- **Estabilidad:** capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.

- **Contrastabilidad:** capacidad del producto del software para permitir la validación de un software modificado.
- **Conformidad de la mantenibilidad:** capacidad del producto de software para adherirse a las normas o convenciones que se relacionan con la mantenibilidad.

**Portabilidad:** capacidad de producto de software de ser transferido de un ambiente a otro.

- **Adaptabilidad:** capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines.
- **Instalabilidad:** capacidad del producto de software de ser instalado en un ambiente especificado.
- **Coexistencia:** capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.
- **Remplazabilidad:** capacidad del producto de software de ser usado en lugar de otro producto de software especificado para los mismos fines y en el mismo ambiente.
- **Conformidad con la portabilidad:** capacidad del producto de software de adherirse a las normas o convenciones relativas a la portabilidad.

### 1.5 Gestión de configuración del software.

En términos muy generales, la Gestión de Configuración del Software (GCS) se puede definir como una disciplina cuya misión es controlar la evolución de un sistema software.(ANTONIO, 2001)

Según Babich, “El arte de coordinar el desarrollo de software para minimizar la confusión, se denomina Gestión de Configuración. La Gestión de Configuración es el arte de identificar, organizar y controlar las modificaciones que sufre el software que construye un equipo de programación. El objetivo es maximizar la productividad minimizando los errores.”(ANTONIO, 2001)

Pressman (PRESSMAN, 2002)plantea: “Si no controlamos el cambio el nos controlará a nosotros y esto nunca es bueno. Es muy fácil para un flujo de cambios incontrolados llevar al caos un proyecto de software”. Por esta razón la gestión de configuración es una actividad esencial del aseguramiento de la calidad y una practica formal de la ingeniería del software.

La gestión de configuración del software es una actividad de protección que se realiza a lo largo de todo el desarrollo de software. Considerando que durante el proceso de desarrollo de software el cambio puede producirse en cualquier momento, las actividades que plantea la gestión de configuración permiten, identificar el cambio, controlar el cambio, garantizar que el cambio se realiza adecuadamente e informar del cambio a todos los interesados.

La definición estándar de de IEEE para la Gestión de Configuración del Software incluye las actividades de:(ANTONIO, 2001)

1. Identificación de la Configuración.
2. Control de Cambios en la Configuración.
3. Generación de Informes de Estado.
4. Auditoría de la Configuración.

El modelo de CMMI plantea las actividades para la gestión de configuración:(CMMI, 2006)

1. Identificar los elementos de configuración.
2. Establecer un sistema de gestión de configuración.
3. Crear líneas bases de liberaciones.
4. Dar seguimiento a las solicitudes de cambio.
5. Control sobre los elementos de configuración.
6. Establecer y mantener registros que describan la configuración.
7. Realizar auditorías a la configuración.

RUP en su disciplina de gestión de configuración establece las siguientes actividades:(JACOBSON *et al.*, 2000)

1. Identificación de los elementos de Configuración.
2. Restringir los cambios sobre los elementos.
3. Auditar los cambios.
4. Definir y gestionar la configuración.

De forma general, cuando se construye un software los elementos deben quedar identificados unívocamente. Una vez que esto se ha logrado, se procede a establecer los mecanismos para el control de cambio. Para garantizar que se mantiene la calidad durante los cambios se audita el proceso y finalmente se procede a elaborar los informes para asegurar que aquellos que necesitan conocer los cambios son informados debidamente.(PRESSMAN, 2002)

### **1.6 Norma ISO (International Standard Organization).**

La utilización de estándares como parte del proceso de aseguramiento de la calidad está presente en todo el proceso de desarrollo, y siendo más precisos, en todo el ciclo de vida. La presencia de estándares asociados directa o indirectamente al desarrollo de software es abundante.

La Serie **ISO 9000** es un conjunto de 5 normas internacionales individuales pero relacionadas entre sí, que regulan la Gestión de la Calidad y el Aseguramiento de la Calidad. Son genéricas, es decir, no se refieren a ningún tipo particular de producto, y pueden ser utilizadas igualmente para industrias o para servicios. Esas normas fueron desarrolladas para documentar en forma efectiva los elementos a ser implementados para mantener un Sistema de la Calidad eficiente en una empresa. ISO ha aportado estándares para la industria del software. Algunos de los más importantes son:

- ISO 9001. Modelo de Sistemas de Calidad para la Garantía de calidad en el Diseño, Desarrollo, Producción, Instalación y Servicios.
- ISO 9000-3. Directrices para el Uso de la ISO 9001 para el Desarrollo, Suministro y Mantenimiento de Software.
- ISO 9004-2. ISO 9004-2. Dirección de Calidad y Elementos de Sistemas de Calidad.

La aplicación de las normas ISO se basa en varias razones como son la mejora de la imagen y de la reputación de la compañía, satisfacer requisitos externos y presión del mercado, facilitar y simplificar la relación con cliente y mejora de la productividad, organización y operativa del sistema de gestión de la calidad. La finalidad principal de las normas ISO es orientar, coordinar, simplificar y unificar los usos para conseguir menores costes y efectividad.

### **1.7 Modelos de Calidad del Software.**

Existen estándares para la administración de proyectos, los cuales garantizan en gran medida obtener productos de software con calidad. 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. Estos consisten en reunir todas las actividades y funciones de forma tal que ninguna de ellas esté subordinada a las otras y que cada una se planee, controle y ejecute de un modo formal y sistemático. La correcta determinación e implementación del Modelo o Estándar de Calidad ayuda a que la empresa pueda disminuir sus costos de desarrollo, aumentar las ganancias y administrar mejor sus recursos. Existen una variedad de modelos para la gestión de la calidad del software que permiten que las Empresas de Software realicen sus tareas y funciones teniendo en cuenta la Calidad.(SCALONE, 2006)

#### **1.7.1 CMMI.**

A mediados de la década del 90, el SEI decide unificar algunos de los modelos que había desarrollado, basándose en la experiencia adquirida de las organizaciones. Específicamente integra los modelos SW-CMM, también conocido como CMM, SE-CMM y IPD-CMM; embarcándose en un esfuerzo que culmina en el año 2002 dando origen a una nueva generación llamada CMMI (Modelo de Madurez de Capacidad Integrado, en inglés Capability Maturity Model Integration).(MARTÍNEZ y NOA, 2007)

Es un modelo de mejora del proceso de madurez para el desarrollo de productos y servicios. Se compone de las mejores prácticas que se ocupan de actividades de desarrollo y mantenimiento que cubren el ciclo de vida del producto desde su concepción a través de la entrega y mantenimiento. (CMMI, 2006)

CMMI se aplica a cuatro disciplinas distintas:

- Ingeniería de Sistemas (Systems Engineering, SE): Cubre la construcción de un sistema con o sin software.
- Ingeniería de Software (Software Engineering, SW): Cubre la construcción de soluciones software.
- Desarrollo Integrado de Producto y Proceso (Integrated Product and Process Development, IPPD): Cubre la relación a largo plazo con el cliente.
- Suministradores Externos (Supplier Sourcing, SS): Cubre los procesos relacionados con la subtratación de partes del sistema.

CMMI permite enfocar la mejora de procesos y evaluaciones usando dos representaciones diferentes: continua y escalonada. La representación escalonada usa un conjunto de áreas de procesos predefinidas para definir una guía de mejora para la organización. Esta guía de mejora se caracteriza por niveles de madurez (1-5). Cada nivel de madurez proporciona un conjunto de áreas de procesos que caracterizan los diferentes comportamientos de la organización. La representación continua permite a una organización seleccionar un área de proceso (o grupo de áreas de proceso) y mejorar los procesos relacionados con ella. Esta representación utiliza niveles de capacidad (0-6) para caracterizar las mejoras relativas a un área de proceso individual. Estas representaciones permiten a la organización seguir caminos diferentes para lograr la mejora de procesos, aunque la presentación y organización de los datos sean diferentes en cada una, los contenidos son los mismos.

CMMI, incluye 22 áreas de proceso para las cuatro disciplinas que cubre actualmente y cinco niveles de madurez, el esfuerzo de alcanzar uno de ellos, dependerá fuertemente de la cultura de trabajo de cada empresa.

El Aseguramiento de la Calidad del Proceso y Producto (PPQA) es una de las Áreas de Proceso del nivel 2 de CMMI. El PPQA define los siguientes objetivos del aseguramiento de la calidad:

Evaluar objetivamente Procesos y Productos de Trabajo: Se evalúa objetivamente la adherencia de los procesos implantados y de los productos/servicios asociados con las descripciones de proceso, estándares y procedimientos aplicables.

Proporcionar Visibilidad Objetiva: Los incumplimientos son supervisados con objetividad, comunicados, y se asegura su resolución.

Esta área de proceso involucra:

- Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos.
- Identificar y documentar los elementos no conformes.

- Proporcionar información a los directores y el personal del proyecto sobre los resultados de las actividades de aseguramiento de la calidad.
- Asegurar que los elementos no conformes son arreglados.

El área de proceso de Aseguramiento de la calidad de procesos y producto apoya la entrega de productos y servicios de alta calidad al proveer al personal y a los administradores un nivel apropiado de visibilidad sobre los procesos y productos de trabajos asociados en todo el ciclo de vida del proyecto.

### **1.7.2 SPICE (ISO 15504).**

En enero de 1993, el Comité Técnico Conjunto 1 (JTC1) de la Organización Internacional para la Normalización y de la Comisión Electrotécnica Internacional (ISO/IEC) asignó al Subcomité 7 (SC7) la responsabilidad de elaborar un conjunto de normas para evaluar los procesos de software. El proyecto de elaboración de dichas normas se designa como Determinación del Mejoramiento y de la Capacidad del Proceso de Software (Software Process Improvement and Capability Determination, SPICE).

SPICE (Software Process Improvement and Capability Determination) es un modelo de madurez de procesos internacional que proporciona un marco de trabajo para la evaluación de procesos de software. Este marco lo pueden usar organizaciones interesadas por la planificación, manejo, monitorización, control y mejora de la adquisición, suministro, desarrollo, operación y soporte de software. Este modelo es una iniciativa a nivel internacional para el desarrollo de un estándar que cubre los métodos, prácticas y aplicaciones de valoración de procesos de adquisición, desarrollo, entrega, operación, evolución y servicios de productos de software.

El objetivo del modelo SPICE es proporcionar un marco de referencia para la valoración de los procesos de software. Fomentar la calidad de los productos de software y generar un proceso de valoración repetible, comparable y verificable. SPICE fomenta productos de calidad, promueve la optimización de procesos y facilita la evaluación del producto a través de los procesos de desarrollo. SPICE tiene diversos alcances, se aplica tanto a nivel directivo como a nivel de usuarios para asegurar que el proceso se encuentra alineado con las necesidades del negocio, se apoya en que los proveedores de software tengan que someterse a una sola evaluación para aspirar a nuevos negocios y busca que las organizaciones de software dispongan de una herramienta universalmente reconocida para dar soporte a su programa de mejoramiento continuo.

La evaluación de procesos tiene dos contextos principales: (1) La mejora de los procesos y (2) La determinación de la capacidad.

SPICE tiene tres características principales: (1) el marco de valor que contempla una dimensión funcional de los procesos, (2) la evidencia para la evaluación y; (3) la recurrencia dada por la selección de instancias de proyectos o productos.

Este estándar proporciona un enfoque estructurado para la evaluación de procesos de software, es decir, Organizaciones con el objetivo de: (1) comprender el estado de sus propios procesos para la mejora de los mismos, (2) determinar la idoneidad de sus propios procesos para un requerimiento particular o clases de requerimientos y (3) determinar la idoneidad de procesos de otras organizaciones para un contrato particular o clase de contratos.

El modelo establece un común denominador para una evaluación uniforme de los procesos de software, aunque la evaluación no pretende ser una nueva instancia de certificación, sino que a través de los resultados se pretende demostrar lo adecuado del mismo.

SPICE integra, al igual que CMM / CMMI, una serie de niveles por la que sus procesos deberán pasar para obtener cómo resultado final la madurez. Los niveles son: Nivel 0 Incompleto, Nivel 1- Fabricado informalmente, Nivel 2 - Planeado, Nivel 3 – Bien definido, Nivel 4 - Controlado cuantitativamente, y Nivel 5 - Mejora continua.

Existen una serie de “Prácticas genéricas” asociadas a los Niveles mencionados anteriormente. Adicionalmente hay una definición de procesos generales que abarcan a toda la organización y a través de los cuáles se identifica el cómo lograrlos: Cliente – Proveedor (CUS), Ingeniería (ENG), Administración (MAN), Apoyo o Soporte (SUP), Proyecto (PRO) y Organización (ORG). Los procesos generales son soportados por prácticas específicas que deberán cumplirse para lograr un paso de niveles, además de la estrecha relación entre los mismos. La categoría de soporte incluye un proceso de aseguramiento de la calidad.

### **1.7.3 Comparación entre modelos.**

No cabe duda de que CMM, tanto por su difusión y resultados reportados, ha sido un modelo exitoso. Por ello, a priori, dadas las similitudes, debemos esperar que CMMI ofrezca también brillantes resultados en incremento de la productividad, más rápida respuesta al mercado, reducción de defectos, disminución de costes, planificaciones fiables, etc. CMMI incluye las prácticas de institucionalización, que permiten asegurar que los procesos asociados con cada área de proceso serán efectivos, repetibles y duraderos. Guía paso a paso para la mejora, a través de niveles de madurez y capacidad (frente a ISO). Transición del ‘aprendizaje individual’ al ‘aprendizaje de la organización’ por mejora continua, lecciones aprendidas y uso de bibliotecas y bases de datos de proyectos mejorados.

En cuanto a SPICE ha sido un proyecto de lenta maduración, que ha variado mucho desde su borrador inicial en 1995, hasta los modelos que están acabando de ser publicados. Cabe destacar el considerable número de estudios de evaluación empírica realizados, pudiendo revisar sus conclusiones más significativas, en cuanto a validez predictiva, veracidad de la capacidad, demostración de su capacidad para identificación de fuerzas, debilidades y riesgos así como dirigir y priorizar el proceso de mejora.

Algunos autores pensaban que el dominio de procesos debería ser más amplio para abarcar todos los posibles ciclos de vida (algo no necesario por la adopción de modelos externos, los PRMs) y que era difícil que todos los atributos de proceso fueran universales, aplicables a todos los procesos y prácticas base.

La dimensión capacidad ha alcanzado un alto grado de dificultad y existen solapamientos con la dimensión procesos. La complejidad de las evaluaciones (y por consiguiente el costo) es significativamente más alta que en otros modelos.

En cuanto a comparativas y mapeos entre áreas clave, con la desaparición de la dimensión Proceso, esta carece de sentido con SPICE por lo que entre ISO 9001 y CMM destacaremos la sinergia entre ambos y el que cada vez más compañías consideren el uso conjunto de CMMI e ISO 9001 para aumentar la eficacia del proceso de mejora.

A modo de resumen, se presenta en el Anexo 1 una tabla comparativa con las principales características de cada modelo.

### **1.8 Proceso de desarrollo de Software.**

“Un proceso define “quién” está haciendo “qué”, “cuándo” y “cómo” para alcanzar un determinado objetivo.

Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Este conjunto de actividades, en el proceso de desarrollo de software que proponen Jacobson, Rumbaugh y Booch, tiene la misión de transformar los requerimientos del usuario en un producto de software; de manera que los integrantes del equipo y todo aquel que pueda estar interesado en el producto final, tenga la misma visión y no ocurra cuando no se aplica un proceso de desarrollo.

Por lo tanto, las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas,

proyecto, producto y proceso. Estos términos son conocidos como las cuatro “P” en el desarrollo de software.

El conocimiento y la experiencia que crea y sostiene la evolución del producto lo tienen las personas. Ellas también financian, se benefician, lo prueban y planifican. Sin un personal competente y experimentado es imposible crear productos competitivos que satisfagan las necesidades de los clientes. Además, el modo en que organiza y gestiona un producto (factibilidad del proyecto, definición del equipo que lo desarrolló, identificación y análisis de riesgos que implican su realización, planificación, nivel de entendimiento de lo que se está haciendo y sensación de que avanza en el cumplimiento) afectan a las personas involucradas en su realización.

El resultado final de un proyecto software es un producto, donde intervienen personas a través de un proceso de desarrollo de software que guía los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Típicamente, el proceso es automatizado por medio de una herramienta o un conjunto de ellas.

A continuación se definen los términos personas, proyecto, producto, proceso y herramientas.

Personas: Los principales autores de un proyecto software son los arquitectos, desarrolladores, ingenieros de prueba, y el personal de gestión que les da soporte, además de los usuarios, clientes, y otros interesados. Las personas son realmente seres humanos, a diferencia del término abstracto trabajadores, que introduciremos mas adelante.

Proyecto: Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

Producto: Artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

Proceso: Un proceso de ingeniería de software es una definición del conjunto completo de actividades necesarias para transformar los requisitos de usuario en un producto. Un proceso es una plantilla para crear proyectos.

Herramientas: Software que se utiliza para automatizar las actividades definidas en el proceso.

Un proyecto de desarrollo obtiene un versión de un producto que contiene modelos, código fuente, documentación y un ejecutable. Este producto va evolucionando durante el proceso de desarrollo desde un proyecto inicial o innovador (prototipo inicial) hasta convertirse en un reléase del proyecto.”(SOFTWARE., 2008)

### 1.8.1 Proceso Unificado de Rational (Rational Unified Process- RUP).

“RUP es un proceso de desarrollo de software que constituye una forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo. Tiene como objetivo la producción de software de calidad dentro de plazos y costos predecibles.

También se considera un producto desarrollado y mantenido por Rational y que es actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo a la experiencia obtenida. RUP es un proceso y en su modelación define como sus principales elementos:

**Trabajadores (“quién”):** Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de los elementos.

**Actividades (“cómo”):** Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula los elementos.

**Artefactos (“qué”):** Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

**Flujo de actividades (“Cuándo”):** Secuencia de actividades realizadas por los trabajadores y que producen un resultado de valor observable.

RUP agrupa las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son flujos de ingeniería del desarrollo del software y los tres últimos de apoyo. Se especificara en los siguientes flujos de trabajo:

**Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

**Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

**Análisis y Diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.”(SOFTWARE., 2008)

### 1.8.2 Flujo de Trabajo Modelamiento del Negocio.

#### Desarrollo del flujo de trabajo Modelamiento del Negocio.

“El Modelamiento de Negocios se efectúa para valorar el negocio para el cual el sistema de información se está construyendo y para determinar mejor las necesidades y problemas a ser resueltos por los sistemas de información. Los modelos del negocio proveen una base para la comunicación entre los analistas de sistemas y los desarrolladores para incrementar su entendimiento del negocio y

para identificar oportunidades de mejorar el negocio. También, los gerentes de proyecto usan los modelos del negocio para ayudarse a estimar los costos del proyecto.

El Modelamiento del Negocio debería hacerse antes del desarrollo de software para obtener un buen entendimiento de sus procesos del negocio. Sin embargo, el Modelamiento del Negocio sólo debe ser efectuado si se está cambiando la manera en que se hace negocio. Si sólo se está añadiendo una nueva característica a un sistema existente, entonces RUP no recomienda que usted empiece con un modelamiento del negocio. En ese caso, RUP recomienda que usted empiece con Requerimientos.”(RUP/EASY, 2004)

### **Propósito:**

- Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
- Derivar los requerimientos del sistema que va a soportar la organización.
- Evaluar el impacto de cambio en la organización.

### **Relación con otras disciplinas o flujos de trabajo.**

Requerimientos utiliza el Modelamiento de Negocio como una importante contribución para entender los requerimientos del sistema.

Análisis y Diseño disciplina que usa los modelos de negocio para definir los sistemas de software que se ajusta dentro de la organización.

Despliegue disciplina que usa el Modelamiento de Negocio como ayuda para planificar el despliegue en un sistema de software.

Ambiente disciplina que desarrolla y mantiene artefactos de soporte, tal como las pautas del Modelamiento de Negocio.

### **Trabajadores**

Analista del Proceso de Negocio: Responsable de la arquitectura del negocio por lo que dirige y coordina el proceso de Modelo del Negocio. Decide cuáles son actores y los procesos del negocio y las relaciones entre ellos y cuáles son las reglas de negocio a tener en cuenta.

Diseñador del Negocio: Describe los procesos de negocio y como parte de la realización de estos procesos identifica a las entidades y trabajadores del negocio y sus relaciones. Define cuáles son los requerimientos en la automatización.

Revisor Técnico: Revisa formalmente el modelo de casos de uso del negocio y de objetos de negocio obtenido.

### **Artefactos**

Documento de Arquitectura del Negocio: Proporciona una visión general integral de los aspectos arquitectónicamente significativos del negocio de un número diferentes de perspectivas.

Especificaciones Suplementarias del Negocio: Otras descripciones contenidas en documentos u obtenidas por otras vías; que permitan un mayor entendimiento del negocio y que contribuyan a su modelamiento.

Glosario de Términos: Lista de concepto asociados al negocio que son comúnmente usados y que deben ser del dominio del equipo de desarrollo para poder modelar el negocio y dar una solución a la problemática encontrada.

Metas del Negocio: Descripción de la meta, usada para proveer más información. Es un requisito que debe quedar satisfecho para el negocio. Describen el valor que se desea alcanzar en el futuro para una medida en particular y por consiguiente puede ser usado para planificar y gestionar las actividades del negocio.

Modelo de Análisis del Negocio (Diagrama de Objetos o Clases): Es un modelo de objetos que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

Modelo de CUN: Describe los procesos de negocio de una empresa en términos de casos de uso y actores del negocio, que se corresponden con los procesos del negocio y los clientes, respectivamente.

Reglas del Negocio: Especifica las reglas de negocio que describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

El proceso de especificación implica que hay que “identificarlas” dentro del negocio, “evaluar” si son relevantes dentro del campo de acción que se está modelando e “implementarlas” en la propuesta de solución.

Valoración del Objetivo de la Organización: Describe el estado actual de la organización en la cual el sistema será desplegado. La descripción será en términos de procesos actuales, herramientas, personas competentes, personas con actitudes, clientes, competidores, tendencias técnicas, problemas, y áreas de mejora.

Visión del Negocio: Define las metas establecidas y los objetivos en los cuales el Modelo del Negocio basa sus esfuerzos para alcanzarlos.

Actores del Negocio: Para cada actor del negocio que se identifica se debe escribir una breve descripción que incluya sus responsabilidades y por qué interactúa con el negocio. Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información

externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Casos de Uso del Negocio (CUN): Breve descripción del rol y el propósito del Caso de uso. Un proceso de negocio es un grupo de tareas relacionadas lógicamente que se llevan a cabo en una determinada secuencia y manera y que emplean los recursos de la organización para dar resultados en apoyo a sus objetivos.

Un caso de uso del negocio representa a un proceso de negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del negocio. Desde la perspectiva de un actor individual, define un flujo de trabajo completo que produce resultados deseables.

Para identificar los procesos de negocio es muy importante tener en cuenta que deben generar un valor para el negocio o mitigar los costos del negocio.

Entidades del Negocio: Breve descripción del rol y el propósito de la Entidad del Negocio. Las entidades de negocio representan a los objetos que los trabajadores del negocio toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio. Comúnmente representan un documento o una parte esencial de un producto. Algunas veces representa cosas no tangibles como el conocimiento acerca de un mercado o cliente.

Eventos del Negocio: Breve descripción de los Eventos del Negocio. Representa un evento significativo en las actividades del negocio que requiere acción inmediata.

Realización de los CUN: La realización de un caso de uso de negocio muestra cómo colaboran los trabajadores y entidades de negocio para ejecutar el proceso. Es una descripción en detalle del caso de uso que puede ser especificado lo mismo textualmente o gráficamente mediante un diagrama en UML.

Sistema del Negocio: Breve descripción del rol y el propósito del Paquete. Encapsula un los roles determinados y los recursos que complementan un propósito específico y define un conjunto de responsabilidades con las cuales ese propósito puede ser logrado.

Trabajadores del Negocio: Breve descripción del rol y el propósito del Paquete. Encapsula un los roles determinados y los recursos que complementan un propósito específico y define un conjunto de responsabilidades con las cuales ese propósito puede ser logrado.

Registro de Revisión: Creado para capturar los resultados de la revisión de actividades en las que son revisadas uno o más artefactos.

Los trabajadores con las actividades que realiza y los artefactos que se obtienen. Ver Anexo 2.

### 1.8.3 Flujo de Trabajo Requerimientos.

#### Desarrollo del flujo de trabajo Requerimientos.

“Se debe manejar las generaciones (versiones) de requerimientos y su documentación. La Administración de Requerimientos incorpora la identificación, organización y documentación de los cambios a los requerimientos de un proyecto. Es una parte integral de la actividad de desarrollo de software. La Administración de Requerimientos establece un entendimiento común y acuerdo entre el cliente y el equipo del proyecto acerca de los requerimientos del cliente. Una Administración de Requerimientos efectiva incluye el mantener requerimientos claros. Mantener atributos acerca de los requerimientos (tales como estado, prioridad), proveer seguimiento a otros requerimientos y componentes y, proveer de los recursos adecuados y fondos para administrar los requerimientos.”(RUP/EASY, 2004)

#### Propósito

- Establecer y mantener un acuerdo con los clientes y los otros interesados acerca de que debe hacer el sistema.
- Proveer a los desarrolladores del sistema de un mejor entendimiento de los requerimientos del sistema.
- Definir los límites (o delimitar) del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
- Proveer una base para la estimación de costo y tiempo necesarios para desarrollar el sistema.
- Definir una interfaz de usuario para el sistema, enfocada en las necesidades y objetivos del usuario.

#### Relación con otras disciplinas o flujos de trabajo.

Modelamiento del Negocio disciplina que provee reglas de negocio, un modelo de Casos de Usos del Negocio y un Modelo de Análisis del Negocio, incluyendo un modelo de Dominio y un contexto organizacional para el sistema.

Análisis y Diseño disciplina que toma como entrada principal (el Modelo de Casos de Usos y el Glosario) de los Requerimientos. Durante el flujo de Análisis y Diseño pueden ser encontrados defectos en el modelo de casos de usos, se generan peticiones de cambio y se aplican al modelo de caso de uso.

Prueba disciplina que valida el sistema contra (entre otras cosas) el modelo de Casos de Usos. Los casos de Usos y las Especificaciones Suplementarias proveen una contribución en los requerimientos usados en la definición de la evaluación, y en la prueba subsiguiente y actividades de evaluación.

Administración y Configuración de Cambios disciplina que provee el mecanismo de Control de cambios.

Gestión de Proyecto disciplina que planifica el proyecto y cada iteración (descrita en un Plan de Iteración). El modelo de Casos de Usos y el Plan de Administración Requerimientos son contribuciones importantes a las actividades de planificación de iteraciones.

Ambiente disciplina que desarrolla y mantiene los artefactos de Soportes que son usados durante la Administración de Requerimientos y el Modelo de Casos de Usos, tal como las Pautas del Modelo de Casos de Usos y las Pautas de Interfaz de Usuario capturada en la Guía Específica del Proyecto.

### **Trabajadores**

Analista del Sistema: Define los alcances del sistema e identifica a los actores y casos de uso que permiten modelar completa y consistentemente el sistema.

Arquitecto de Software: Describe la vista de la arquitectura del modelo de casos de uso, definiendo la prioridad de cada caso de uso para decidir en qué iteración será desarrollado cada uno. Es el responsable de la arquitectura del software la cual incluye las decisiones técnicas que contiene el conjunto de diseño e implementación del proyecto.

Especificador de Requerimientos: Describe detalladamente cada caso de uso de acuerdo a las funcionalidades que engloba.

Revisor Técnico: Es el responsable de contribuir a la retroalimentación de los procesos revisados. Esta involucrado en la revisión técnica de los artefactos del proyecto. Responsable de proveer en forma una apropiada retroalimentación de los artefactos a revisar.

### **Artefactos**

Atributos de los Requerimientos: Un repositorio de los requerimientos del proyecto, atributos y dependencias que ayudan en la administración de cambio de las perspectivas de los requerimientos.

Documento Visión (Sistema): Define la vista de los stakeholders del producto a desarrollar, especificando en términos de los stakeholders las necesidades y características claves. Contiene una visión de los requerimientos fundamentales, esta provee una base contractual más detallada de los requerimientos técnicos.

Especificaciones suplementarias: Captura los requerimientos del sistema que no son fácil de capturar en el comportamiento de los artefactos del requerimiento tal como la especificación de caso de uso.

Glosario de Términos: Términos comunes que se utilizan para describir el sistema.

Modelo de Casos de Uso (Sistema): Es un modelo del sistema que contiene actores, casos de uso y sus relaciones. Es usado como una entrada esencial para las actividades de análisis, diseño y prueba.

Plan de Administración de Requerimientos: Describe el artefacto requerimiento, tipos de requerimientos y sus respectivos atributos, especificando la información a coleccionar y mecanismos de control a ser usados para medir, reportar y controlar cambios a los requerimientos del producto.

Requisitos de los Stakeholders: Contiene algún tipo de demanda de los stakeholder (clientes, usuarios finales, etc.) a tener el sistema a desarrollar. También contiene referencias de algún tipo de fuente externa que el sistema debe cumplir.

Documento de Arquitectura: Provee una vista global de la arquitectura del sistema, usando las diferentes vistas arquitectónicas para describir los diferentes aspectos del sistema.

Requerimientos del Software: Especifica las condiciones o capacidades que le sistema debe cumplir.

Casos de Uso (Sistema): Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

Especificaciones de los Requerimientos del Software: Captura los requerimientos del sistema completo o de una parte de este.

Paquete de Casos de Uso: Colección de casos de uso, actores, relaciones, diagramas y otros paquetes que son usados para estructura del modelo de caso de uso en partes más pequeñas.

Requerimientos del Software:

Requerimientos funcionales: Son capacidades o condiciones que el sistema debe cumplir. Los requerimientos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen.

Requerimientos no funcionales: Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable, por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado.

Registro de Revisión: Creado para capturar los resultados de la revisión de actividades en las que son revisadas uno o más artefactos.

Los trabajadores con las actividades que realiza y los artefactos que se obtienen. Ver Anexo 3.

### **1.8.4 Flujo de Trabajo Análisis y Diseño.**

#### **Desarrollo del flujo de trabajo Análisis y Diseño.**

“El propósito del Flujo de Trabajo de Análisis y Diseño es empezar a realizar los casos de uso desarrollados durante el Flujo de Trabajo de Requerimientos. Es decir, tomar el Modelo de Casos de Uso, el Glosario y las Especificaciones Suplementarias creadas en el Flujo de Trabajo de Requerimientos y generar un modelo de diseño que pueda ser usado por los desarrolladores durante

el Flujo de Trabajo de Implementación. El Análisis se enfoca en trasladar los requerimientos funcionales a conceptos de software.

El Flujo de Trabajo de Análisis y Diseño toma los casos de uso documentados del Flujo de Trabajo de Requerimientos y del Flujo de Trabajo Modelamiento de Negocio y los traslada a elementos de diseño que serán usados para construir el sistema. Por medio de usar varias actividades y modelos el Flujo de Trabajo de Análisis y Diseño busca destilar la información recogida de los stakeholders en información que los programadores podrán usar. Al final, un Modelo de Diseño, el documento de Arquitectura del Software, el Modelo de Despliegue y una Realización de Casos de Uso por cada Caso de Uso describirán el sistema.”(RUP/EASY, 2004)

### **Propósito**

- Transformar los requerimientos a diseños del sistema.
- Desarrollar una arquitectura robusta para el sistema.
- Adaptar el diseño para hacerlo corresponder con el ambiente de implementación y ajustarla para un desempeño esperado.

### **Relación con otras disciplinas o flujos de trabajo.**

Modelamiento de Negocio disciplina que provee un contexto organizacional para el sistema.

Requerimientos disciplina que provee una entrada principal para el Análisis y Diseño.

Implementación disciplina que implementa el diseño.

Prueba disciplina que prueba el sistema diseñado durante el Análisis y Diseño.

Ambiente disciplina que desarrolla y mantiene los artefactos de soporte que son usados durante el Análisis y Diseño.

Gestión de Proyecto disciplina que planifica el proyecto y cada iteración (en el Plan de Iteración).

### **Trabajadores**

Arquitecto de Software: Es el responsable de la arquitectura del software la cual incluye las decisiones técnicas que contiene el conjunto de diseño e implementación del proyecto.

Diseñador: Responsable de diseñar parte del sistema, dentro de los requerimientos especificados, la arquitectura y desarrollo de los procesos del proyecto.

Diseñador de Cápsula: Responsable de diseñar el artefacto Cápsula, garantizando que el sistema responda en forma a los eventos de acuerdo a los requerimientos especificados.

Diseñador de Base de Datos: Responsable del diseño de los datos persistentes almacenados a usar en el sistema. Define un diseño detallado de la base de datos que incluye tablas, índices, vistas, constraints, triggers, procedimientos almacenados y otras construcciones específicas de la base datos necesarias para el almacenamiento , recuperar y eliminar objetos persistentes.

Diseñador de Interfaz de Usuario: Coordina el diseño de la interfaz de usuario. El diseñador de la interfaz de usuario también está involucrado en juntar requisitos de usabilidad y diseños candidatos de la interfaz de usuario para los requisitos especificados.

Diseñador de Prueba: Responsable de definir los métodos de prueba y garantizar su implementación. Identifica las técnicas apropiadas, herramientas y pautas requeridas para la implementación de pruebas, y brinda una guía de los requerimientos a probar.

Revisor Técnico: Es el responsable de contribuir a la retroalimentación de los procesos revisados. Está involucrado en la revisión técnica de los artefactos del proyecto. Responsable de proveer en forma una apropiada retroalimentación de los artefactos a revisar.

### **Artefactos**

Modelo de Análisis: Un modelo de objetos describe la realización de los CU, y cuales sirven como abstracción del artefacto: modelo de diseño. El modelo de análisis contiene los resultados del análisis de los CU, instancias del artefacto: clases del análisis. Contiene: clases del análisis.

Modelo de Despliegue: El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos.

Modelo de Diseño: El modelo de diseño es un modelo de objetos que describe la realización de los CU, y sirve como una abstracción del modelo de implementación y el código fuente. Es usado como una entrada inicial en las actividades de implementación y prueba. Contiene: protocolo, capsula, realización de CU, señales, eventos, subsistema de diseño, paquetes de diseño, interfaces, clases del diseño, clases de prueba, diseño de pruebas.

Documento de Arquitectura de Software: Provee una vista abarcadora de la arquitectura del sistema, usando un número de vistas de la arquitectura diferentes para describir diferentes aspectos del sistema.

Arquitectura de Referencia: Una arquitectura de referencia es en esencia un patrón de arquitectura predefinido, o una entrada de patrones, posiblemente parcial o completamente ejemplificada, diseñada y acreditada para el uso en negocios particulares y en contextos técnicos, junto con artefactos de soporte que hacen posible su uso. Con frecuencia, estos artefactos son recogidos de proyectos anteriores.

Interfaces: Un elemento del modelo que define una entrada de comportamientos (entrada de operaciones) ofrecidas por un elemento clasificador del modelo (específicamente clases, subsistemas o componentes). Un clasificador puede usar una o más interfaces.

Señal: Una señal es un estímulo asíncrono desde un objeto o una instancia a otro objeto o instancia.

Eventos: Una especificación de un acontecimiento en espacio y tiempo, menos formal, un acontecimiento de algo a lo que el sistema debe responder.

Protocolos: Una especificación común de una entrada a los puertos del artefacto cápsula.

Arquitectura (Esbozo): Es una solución la cual puede ser conceptual para los requisitos significativos de la arquitectura que son identificados en etapas tempranas del inicio.

Clases de Análisis: Las clases de análisis representan un modelo conceptual temprano para “cosas en el sistema que tienen responsabilidades y comportamientos”.

Clases de Diseño: Describe un conjunto de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos, y la semántica.

Describe una colección de objetos que comparten las mismas responsabilidades, relaciones, operaciones, atributos y semántica.

Clases de Pruebas: Una clase especializada del modelo de diseño que representa un comportamiento específico de prueba que va a ser soportado por el SW.

Paquete de Diseño: Un paquete de diseño es una colección de clases relaciones, realizaciones de CU, diagramas y otros paquetes. Es usado para estructurar el modelo de diseño mediante su división en partes más pequeñas.

Realización de Casos de Uso: Una realización de CU describe como un CU particular es realizado dentro del modelo de diseño, en términos de colaboración de objetos.

Subsistema de Diseño: Una parte del sistema que encapsula comportamientos, expone un conjunto de interfaces, y empaqueta otros elementos del modelo. Desde el exterior, una subsistema es un único elemento del diseño que colabora con otros elementos del modelo para cumplir sus responsabilidades. Las interfaces visibles externamente y sus comportamiento son referidos como especificaciones del subsistema. Internamente, un subsistema es una colección de elementos del modelo (clases del diseño y otros subsistemas) que entienden la interfaz y el comportamiento de la especificación del subsistema. Esto es referido como realización del subsistema.

Cápsula: Es un patrón de diseño específico el cual representa un hilo de control encapsulado en el sistema.

Una capsula es representada como una clase, estereotipada «capsule».

Modelo de Datos: Describe la representación lógica y física de los datos persistentes usados por la aplicación. En los casos donde la aplicación va a utilizar una sistema de administración de base de datos relacionales (RDBMS), el modelo de datos puede incluir también elementos del modelo para procedimientos almacenados, triggers, constraints, etc. Que definen la interacción de los componentes de la aplicación y el RDBMS.

Mapa de Navegación: El mapa de navegación expresa la estructura de los elementos de la UI en el sistema, junto con sus caminos de navegación potenciales.

Prototipo de Interfaz de Usuario: Es un ejemplo de la interfaz de usuario que es construida con el fin de explorar y/o validar el diseño de la interfaz de usuario.

Diseño de prueba: Descripción de los elementos estructurales de prueba y la realización de los casos de prueba.

Registro de Revisión: Creado para capturar los resultados de la revisión de actividades en las que son revisadas uno o más artefactos.

Los trabajadores con las actividades que realiza y los artefactos que se obtienen. Ver Anexo 4.

### **1.9 Conclusiones Parciales.**

El estudio que se realiza en este capítulo demuestra la importancia de asegurar la calidad durante el proceso de desarrollo del software, para obtener al final del ciclo de vida un producto que satisfaga las necesidades del cliente. Facilitó especificar la función y el comportamiento de algunos de los modelos de calidad más reconocidos, aplicados mundialmente y seleccionar el Modelo de Calidad de Software CMMI para el desarrollo del procedimiento, indicar la interacción con otros elementos del sistema y la necesidad de establecer normas de calidad. Conocer los trabajadores y las actividades que se realizan para obtener los diferentes artefactos en cada uno de los flujos de trabajos especificados. Brinda una visión de cada estándar estudiado para la confección del procedimiento que se va a proponer más adelante.

## CAPITULO 2

### PERSPECTIVAS DEL ERP EN CUBA.

### SITUACIÓN ACTUAL DE LA CALIDAD EN LA UCI.

#### 2. Capítulo II.

##### 2.1. Introducción.

En el presente capítulo se realiza una investigación de los principales problemas que influyen en la calidad del software en la universidad. Se efectúa una caracterización de forma general de la UCI y específicamente de la Facultad 4 con el objetivo de enmarcar la investigación que se efectúa. Se analizan estadísticamente los resultados obtenidos a partir de una encuesta realizada a los Directivos, Roles Principales y Desarrolladores de la universidad, filtrando estos datos para la facultad, con el objetivo de definir los principales problemas que afectan la calidad del software en la universidad, utilizando como herramienta el STATGRAPHICS. Se realiza un análisis de los ERP en Cuba abordando la situación que existe, la vinculación de estos sistemas con el software libre, los sistemas contables financieros certificados, la importancia del desarrollo de un software de este tipo para las empresas cubanas, además de los resultados obtenidos al realizar un estudio de los proyectos productivos en la universidad.

##### 2.2. Principales problemas que afectan la industria cubana del software.

“El desarrollo de una Industria Nacional de Software es una tarea de gran prioridad para el Estado cubano debido a la alta perspectiva económica que posee, así como para el aseguramiento de un grupo importante de actividades del país. A pesar de ello, los resultados alcanzados no cubren las expectativas, ya que la productividad es baja, la cantidad real de recursos a consumir -en tiempo principalmente- es casi impredecible y el trabajo realizado casi nunca tiene la calidad y profesionalidad requerida. Los proyectos están excesivamente tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas no se detectan en este medio indisciplinados y caóticos de desarrollo.

En estudios realizados por el Centro de Estudios de Ingeniería de Sistemas (CEIS) en empresas nacionales se detectaron problemas relacionados con:”(DAPENA, 2003)

- El personal disponible en estas empresas es aún escaso aunque tiene un alto nivel de preparación.
- Existe una gran desorganización en las empresas lo que no permite aplicar técnicas, modelos o estándares que ayudarían al desarrollo de esta.
- Existen pocos clientes y no se ha creado una disciplina para el intercambio con ellos.

- Resulta muy pobre el mercado externo.
- No existe una cultura de producción de software bajo parámetros de terminación y calidad, donde se actúe bajo conceptos y estudios técnicamente fundamentados por equipos multidisciplinarios y competentes dirigidos a la creación de un producto orientado a determinado mercado.
- Hay mala calidad en gran parte del software que se produce en el país y es indispensable la solución de este problema lo más breve posible, pues la calidad del producto que desarrollan las empresas nacionales es clave para mejorar su competitividad, y teniendo la calidad del producto como elemento distintivo, estas pueden encontrar nuevos mercados.

Los resultados de este estudio arrojan la necesidad de establecer políticas de calidad en las organizaciones del país con el fin de mejorar la productividad de las empresas y lograr una adecuada satisfacción de los clientes tanto en el mercado nacional como en el internacional.

### **2.3. Análisis de los sistemas ERP en Cuba.**

#### **2.3.1. Situación en Cuba.**

La Oficina Nacional de Informatización, ha reconocido que la situación general en Cuba de las entidades de desarrollo de aplicaciones informáticas y de las propias aplicaciones en explotación que abarcan la actividad presupuestada, empresarial productiva o de servicios, esta caracterizada por:

- Una considerable presencia de sistemas informáticos que están desarrollados sobre plataformas envejecidas y con poco o ningún criterio de seguridad y auditoria (en el orden técnico y funcional).
- Son productos que se caracterizan por abordar solamente partes del problema de la gestión de la empresa o la unidad presupuestada y no soportan mecanismos estándares de integración con otras aplicaciones.
- La mayoría fueron desarrollados para un ambiente multiusuario, casi ninguno bajo conceptos de informática multicapa y distribuida en la red. Lo más general es, el desarrollo sobre arquitectura Cliente-Servidor de base de datos.
- Los sistemas más potentes que actualmente están en explotación en Cuba son extranjeros y no abarcan todas las operaciones de gestión por una incompleta implementación en la entidad o porque no lo soportan.
- Pocas entidades desarrollan utilizando software libre y con estándares abiertos, por lo cual hay casi un desconocimiento generalizado sobre estas actuales plataformas de desarrollo.

La dirección del país, como parte del fortalecimiento de la gestión de las entidades y la informatización de la sociedad cubana ha planteado la necesidad de informatizar los procesos de gestión de las

entidades presupuestadas y empresariales a escala nacional utilizando plataformas confiables y eficientes.

En la actualidad no existe en Cuba un sistema informático integral de gestión que cumpla con la totalidad de los requerimientos de funcionalidad, interoperabilidad y seguridad que espera el gobierno cubano de una solución de este tipo, de manera que pueda ser utilizada como herramienta para potenciar el cumplimiento de las funciones de las entidades a todos los niveles con un máximo de racionalidad y control de los recursos financieros, materiales y humanos.

### **2.3.2. Sistemas Contables-Financieros.**

Un sistema contable financiero comprende los métodos, procedimientos y recursos utilizados por una entidad para llevar un control de las actividades financieras y resumirlas en forma útil para la toma de decisiones. (JOSAR)

Las alternativas que se ejemplifican corresponden a sistemas nacionales certificados, con características de desarrollo en plataformas totalmente propietarias y aplicaciones de escritorio. Los motores de bases de datos utilizados son MS ACCESS, MS SQL SERVER y ORACLE en diferentes versiones. Los sistemas Contables-Financieros certificados en Cuba hasta marzo del 2008:

VERSAT SARASOLA, ASSETS NS, EXACT GLOBE PARA WINDOWS, RODAS XXI, KEWAN, SAGE LINEA 100, SISCOMIP, BK-MIS, ABACO, SAGE MAS 500, STOCK EMPRESARIAL, TPV SOFT, MISTRAL POS, Sistema e-TES, Mod almacén del Sistema Hotel, CONDOR, NOMINA, SUITE ATENAS, FACSI+, ZUN, 1C: EMPRESA, Aqua Ebs-Disaic, SICEMA PLUS, CONTACUB de ADDIS, ACON, SENTAÍ, SISCONT5. Se describe a continuación, una síntesis de funcionalidades generales:

#### **2.3.2.1. VERSAT SARASOLA.**

El programa Versat-Sarasola, sistema cubano de contabilidad confiable, permite enviar información eficaz, de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión.

Fue este el primer sistema de contabilidad cubano certificado, en cuya evaluación participaron el Ministerio de Finanzas y Precios, consultorías internacionales y el organismo encargado de la seguridad informática. Es un sistema económico integrado. Constituido por 12 módulos:

- Configuración y seguridad.
- Contabilidad general y de gastos.
- Costos y procesos.
- Análisis económico empresarial.
- Control de activos fijos.
- Finanza y caja.

- Planificación y presupuestos.
- Control de inventarios.
- Pago de salario.
- Paquete de gestión.
- Contratación.
- Facturación.

Actualmente lo utilizan alrededor de 200 entidades de varias provincias y en lo adelante lo introducirán más de dos mil 500 unidades presupuestadas del país, entre las que figuran organismos de la Administración Central del Estado, las direcciones municipales de finanzas, tesorerías, la ONAT y otros.

#### **2.3.2.2. RODAS XXI Versión 3.0.**

Sistema multiempresa y multiusuario creado por CITMATEL para la automatización de la gestión empresarial. Contiene diferentes módulos que pueden usarse integrados o independientes:

- Contabilidad.
- Efectivo Caja y Banco.
- Nóminas.
- Activos Fijos Tangibles.
- Inventarios.
- Cobros y Pagos.
- Facturación.
- Finanzas.
- Tele-cobranza.

Además, cuenta con el módulo Administrador, que brinda mayor integralidad al sistema y garantiza facilidades adicionales durante su instalación y explotación.

#### **2.3.2.3. SISCONT5.**

El sistema se aviene a las definiciones y conceptos del Ministerio de la Industria Básica aunque por las acciones contables financieras que permite puede ser utilizado en otras entidades nacionales.

Está formado por varios Módulos:

- Efectivo en Caja y Bancos.
- Inventarios.
- Cobros y Pagos.
- Facturación.

- Activos Fijos Tangibles.
- Nóminas.
- Contabilidad.

Puede ser explotado en régimen monousuario y multiusuario. Se define para monoentidad y multientidad, para esta última existe el control de su acceso para las entidades en un mismo equipo de cómputo como servidor.

#### **2.3.2.4. CONDOR.**

Sistema automatizado de alta complejidad y seguridad que abarca todos los aspectos del proceso contable de una entidad, tales como la dualidad de moneda y el pago por resultados. Está formado por varios Módulos:

- Activos Fijos.
- Contabilidad General.
- Nóminas.
- Control de Inventarios.
- Condexce.
- Recursos Humanos.

Este brinda mayor autonomía al cliente para efectuar cambios de estructura sin necesidad de la intervención de especialistas, quedando registrados de forma que puedan ser auditables. Incluye la contabilidad multimonedas.

#### **2.3.2.5. SISCOMIP.**

Es un Sistema Contable Integral diseñado para las empresas del Ministerio de la Industria Pesquera. Su arquitectura no es modular. Está concebido para llevar a cabo el registro contable de la actividad económica, a nivel de documento primario, en una entidad cualquiera compuesta por unidades, de forma compartimentada o no, y está en explotación actualmente en entidades pertenecientes al Ministerio de la Industria Pesquera. Está formado por varios subsistemas:

- General.
- Inventarios.
- Finanzas.
- Comercial.
- Nóminas.
- Activos Fijos.

Permite intercambio de información entre los subsistemas de forma automática y por opciones. Puede ser implementado en entornos de redes o en máquinas aisladas.

#### **2.3.2.6. SICEMA PLUS.**

Sistema que permite la generación automática de la contabilidad a partir de la información primaria de las entidades. Cuenta con el proceso de Re-indización opcional y automática. Permite trabajar en distintas monedas. Está formado por varios Módulos:

- Nóminas.
- Contabilidad.
- Efectivo en Caja y Banco.
- Inventarios.
- Facturación.
- Cobros y Pagos.
- Activos Fijos Tangibles.

Puede trabajar en régimen monousuario aunque esta diseñado para trabajar en régimen multiusuario con explotación en redes locales. Cuenta con las opciones que le permiten importar y exportar Base Datos.

#### **2.3.2.7. SISTEMA eTES (Total Enterprise Solution) Versión 1.3.4.0.**

El Sistema eTES ha sido concebido como un sistema integrado de gestión empresarial por lo que lleva implícito en su diseño características de integración, acceso directo y remoto, conectividad, uso y protección de la base informativa única, reglas de seguridad y de autenticación, concepción modular y compartimentación, entre otras. Tiene una arquitectura de tres capas (Almacenamiento de Datos, Reglas del Negocio, Presentación). Está formado por varios Módulos:

- Data Manager (Administrador).
- Account (Contabilidad, Cobros y Pagos, Efectivo en Caja y Bancos).
- Assets (Activos Fijos).
- PayRool (Nóminas).
- Stocks (Inventarios).
- Sales (Facturación).

El módulo Account es multimoneda y los módulos, Assets, PayRool, Stocks son doble moneda.

Estos Sistemas Contables-Financieros forman parte de una diversidad de soluciones de gestión con diferentes grados de seguridad, niveles de funcionalidades y deficiente proceso de soporte técnico que

afectan la oportunidad, la fiabilidad y el intercambio ordenado de la información para el proceso de toma de decisiones, apreciándose una tendencia al uso de soluciones extranjeras.

### **2.3.3. ERP y Software Libre.**

La idea de crear un ERP nacional en Software Libre no es solamente cubana sino una necesidad de otros países. Los ERP no han estado ajenos a la corriente de Software Libre, existen muchos ejemplos que van desde sistemas propietarios de grandes compañías que permiten instalarlos sobre servidores Linux hasta esfuerzos más modestos pero apoyados incluso por empresas medianas, ejemplos de ERP libres son TinyERP, OpenBravo, FacturaLUX, openXpertya, Compiere, ERP5.

Las dos desventajas principales que tienen en el caso de Cuba es que algunos usan tecnologías que pueden ser riesgosas en la situación actual de bloqueo que sufre el país, por ejemplo OpenBravo y OpenXpertya están basados en la plataforma J2EE cuya máquina virtual es propiedad de SUN que es una empresa norteamericana y aunque haya comenzado a liberar el código de esta máquina virtual sigue estando bajo las leyes de su gobierno que bloquea por todos los medios el acceso a la tecnología informática, además J2EE requiere un consumo de memoria elevado en comparación con otras plataformas como pudiera ser Python/Zope o PHP/Apache. Como segunda desventaja se puede señalar que estos sistemas han sido diseñados para empresas capitalistas que tienen un modelo de gestión y de procesos bastante diferente a las entidades cubanas donde la economía es centralizada y operan otros mecanismos. No quiere decir que estos ERP libres sean totalmente inadecuados, pero hacerle las adaptaciones necesarias sería tan engorroso que es preferible diseñar uno específico para satisfacer las necesidades nacionales.(BRITO, 2006)

No se puede ver separada la definición de la plataforma con las características del parque de computadoras presente o futuro en el país. Si el país se ve obligado a invertir millones de dólares para comprar equipamientos de avanzadas porque el sistema requiere de grandes prestaciones de hardware entonces no se habría hecho mucho pues sólo algunas empresas podrían darse ese lujo y las demás entidades del país no tendrían los recursos necesarios para disfrutar de las ventajas del nuevo sistema.

### **2.3.4. Importancia del Sistema ERP en Cuba.**

Contar con un ERP cubano que permita estandarizar la comunicación entre las empresas nacionales, que pueda servir como herramienta para la gestión económica de los recursos, no sólo a nivel gubernamental sino también a nivel de provincia o municipios, es una meta alcanzable pero trabajosa. Con el desarrollo de un sistema ERP se pretende lograr un producto altamente configurable y modular de alcance nacional para la gestión integral de los procesos en las entidades cubanas, brindando de este modo un impulso decisivo a la informatización de la sociedad cubana.

Permite la gestión de la información referida a los Sistemas Contables y Financieros con la totalidad de las funcionalidades de los sistemas nacionales certificados.

Brinda una solución nacional que permite, a diferencia de otros productos similares, la gestión integral de las entidades presupuestadas y empresariales, basada en los principios de independencia tecnológica y con funcionalidades generales de los procesos y las particularidades de la economía cubana.

Una solución satisfactoria implica:

1. Producto nacional que respalda los lineamientos nacionales de desarrollo de soluciones bajo el principio de independencia tecnológica.
2. Seguridad, oportunidad y fiabilidad en la gestión de la información de las entidades nacionales y de los diferentes niveles de gobierno.
3. Ordenamiento en la producción nacional de Software de gestión.
4. Disponer de una plataforma de intercambio de información controlado, flexible y ágil.

Este sistema logra vencer las barreras del software que existen en las entidades presupuestadas o empresariales para migrar hacia las nuevas posibilidades que brinda el Software Libre, ya que ofrece las funcionalidades necesarias para que las entidades puedan continuar desarrollando su función social sin problema alguno y mejorar el control de los procesos y la eficiencia económica.

La solución posibilita cubrir las necesidades de informatización del país. No obstante por la complejidad y tamaño de estas soluciones se requiere la disponibilidad de un grupo importante de recursos materiales, infraestructura, capital humano y organización para su desarrollo.

Se espera que dentro de un futuro muy cercano el país pueda contar con una infraestructura de software propia que le permita afrontar los desafíos del nuevo milenio y lo sitúe a la vanguardia de la Informática como lo es en otros campos del quehacer humano.

## **2.4. Características fundamentales del desarrollo de software en la UCI.**

### **2.4.1. Descripción del Objeto de Estudio.**

La UCI es una universidad productiva, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. La Producción de Software y Servicios Informáticos se basa en la integración de los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva. Entre los servicios más importantes se brindan los de calidad de software, arquitectura y tecnología, servicios legales, diseño de comunicación visual.

La producción se concentra en el desarrollo de proyectos en más de 30 Polos Productivos y se destacan resultados en las esferas de salud, educación, software libre, teleformación, sistemas

legales, realidad virtual, automatización, bioinformática, procesamiento de imágenes y señales, entre otras.

Promueve el desarrollo de productos y servicios informáticos en aquellas ramas donde Cuba tiene un reconocido prestigio en el mundo a través del concurso de los mejores especialistas del país para lograr una solución de calidad y de impacto internacional.

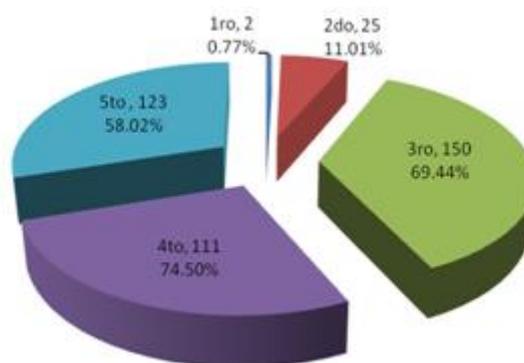
Desarrolla programas de informatización de la sociedad cubana a través de la relación con entidades nacionales, los resultados alcanzados se extienden por todo el país.

En el 2005 fue creada la empresa comercializadora de todos los productos y servicios informáticos que se desarrollen en la Infraestructura Productiva de la UCI formada por:

- 30 Polos Productivos que desarrollan proyectos temáticos ubicados en las 10 Facultades, estas cuentan con un Vicedecanato de Producción e Investigación con asesores de Arquitectura, Calidad y Estrategia.
- 12 direcciones especializadas para coordinar y dar servicio al ciclo completo de las producciones de software que cuenta con personal capacitado que trabaja en la documentación del proceso y lineamientos que rigen la producción de la universidad.
- Un capital humano capaz de generar resultados equivalentes a 6 millones de horas hombres al año lo que significa el desarrollo de 100 grandes sistemas al año si se crean las condiciones técnico – organizativas necesarias.

**2.4.2. Características de la facultad.**

La Facultad 4 cuenta con un total de 1065 estudiantes de estos están vinculados a proyectos 411 que representan un 38.59 %. La asignación de estudiantes por año es:



**Figura1.** Asignacion de estudiantes a proyectos.

Se reorganiza por Polos Productivos de 1ero a 5to. Estos son:

1. Sistemas Tributarios y de Aduanas.
2. Gestión de Comunidades.

3. Sistemas Contables, Financieros y Bancarios.

4. Sistemas para la Defensa.

Los Polos son agrupaciones de temáticas de los proyectos que existen. Se trabaja para alcanzar los primeros niveles de certificación, el área más crítica en este sentido es la investigación asociada que sustente el desarrollo de productos. Está en análisis la composición y el objeto de cada Polo en la facultad, existen desarrollo de soluciones similares en diferentes Polos como es el caso de los sistemas de Recursos Humanos desarrollados para el MINFAR y la Aduana y SIGEP. Además se potencia la investigación de los grupos de inteligencia empresarial y sistemas de Bases de Datos (Minería de Datos para el análisis de información empresarial), sistemas de apoyo a las organizaciones para la toma de decisiones.

Los proyectos productivos que se desarrollan actualmente son:

- Sistema Único de Aduana
- Información adelantada de pasajeros
- Sistema de Gestión de Recursos Humanos para la Aduana
- Planificación de Recursos para el MINFAR
- Recursos Humanos para el MINFAR (comités Militares)
- Sistema de Contabilidad Material para el MINFAR
- Sistema de Contabilidad Financiera para el MINFAR
- Informatización de la Residencia
- Información y Gestión de Establecimientos Penitenciarios (SIGEP)
- Migración del Sistema de la Banca Nacional
- INSIGNE<sup>6</sup>
- Metadatos

Entre los principales clientes está a Aduana General de la República, MPPRIJ, Residencia UCI, MINFAR, MFP y Banco Nacional.

Se han alcanzado resultados como:

Relaciones de colaboración satisfactorias con la Aduana General de la República.

1. SUA<sup>7</sup> en explotación.
2. Reconocimiento de la Organización Mundial de Aduanas.

---

<sup>6</sup> Integración de Sistemas para la Gestión de Entidades.

<sup>7</sup> Sistema Único de Aduana.

Colaboración con el Ministerio del Interior y Justicia de Venezuela en el Marco de los convenios del Alba.

1. Piloto en la Instalación Penitenciaria de Carabobo, Venezuela.

Colaboración con el MINFAR en el desarrollo de soluciones y despliegue de las mismas: Comités Militares.

### **2.4.3. Resultados de proyectos comunes en la UCI.**

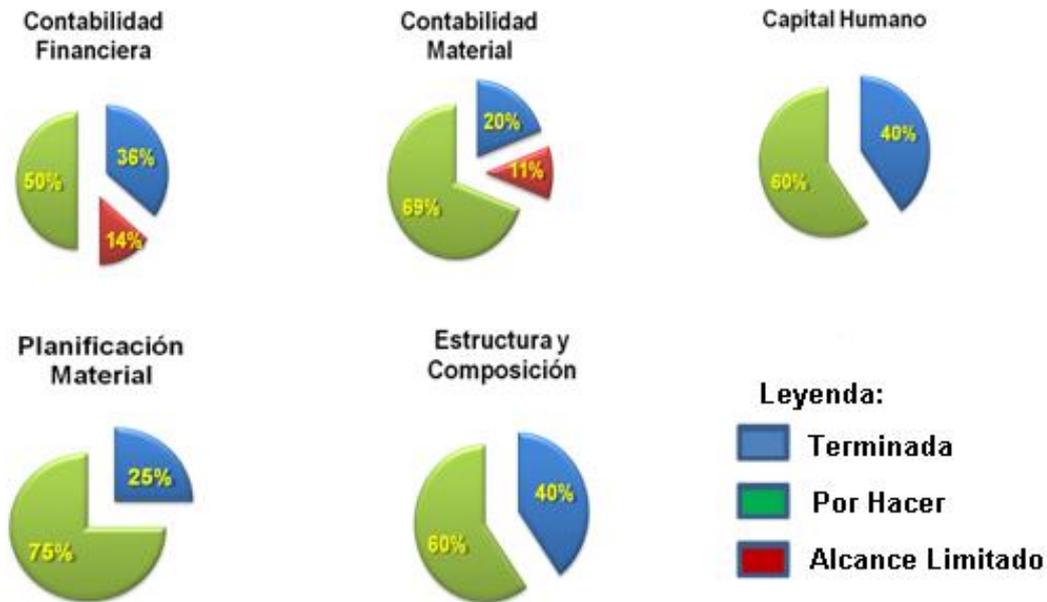
En el proyecto el primer resultado forma parte del Sistema “Soluciones para la Gestión de Entidades”, y comprende en su primera fase elementos de los Sistemas Contable, Capital Humano, Logística y de Estructura y Composición. De forma general abarca los módulos de:

- Configuración.
- Estructura y composición.
- Contabilidad (Incluye Estados Financieros).
- Caja, Banco, Cobros y Pagos.
- Inventario.
- Facturación.
- Activos Fijos Tangibles e Intangibles.
- Costo y procesos.
- Capital Humano (Incluye Nóminas).
- Planificación Empresarial y Presupuestaria.
- Generador de reportes.

Se realiza un análisis para identificar los resultados y proyectos que pueden contribuir con la solución de un ERP en la universidad. Se obtiene la siguiente información:

- Proyecto UCID (ERP FAR).
  - Contabilidad Financiera.
  - Contabilidad Material.
  - Planificación.
  - Estructura y Composición.
  - Cuadros.
  - Configuración.
- Facultades.
  - Capital Humano, en desarrollo para la AGR.
  - SIGE, en desarrollo para la ONE.

Al efectuarse un diagnóstico a los módulos del proyecto UCID, en la universidad para comprobar el grado de aprovechamiento, con la solución, desde el punto de vista funcional se identifican los siguientes datos:



**Figura 2.** Resultados por módulos del proyecto UCID.

Se examinan todos estos resultados y se obtiene de forma general que solo una parte de las funcionalidades terminadas del proyecto UCID pueden aportar al desarrollo del proyecto INSIGNE.



**Figura 3.** Resultado general del proyecto UCID.

Se observa que esto es solo una parte de lo que se aspira que sea el ERP Cubano, pero es una solución que no esta muy lejos de la realidad pues es un software que funciona actualmente en las FAR y que se esta en proceso de desarrollo (otros módulos).

## **2.5. Análisis de los resultados de las encuestas realizadas por la Dirección de Calidad de Software.**

Para el análisis del aseguramiento de la calidad en la universidad se ha tomado como base un estudio realizado en la UCI en el primer semestre del presente curso 2007-2008 por parte de la Dirección de Calidad de Software, con el objetivo de realizar un diagnóstico general, con preguntas enfocadas a temas de planificación, calidad mediciones, organizativas. A partir del cuestionario realizado por parte de la dirección de calidad se han seleccionado las preguntas relacionadas con el tema de aseguramiento de la calidad y sólo de los proyectos vinculados con la facultad 4.

### **Planificación**

- ¿En su proyecto se realizan reuniones para planificar y organizar el trabajo?
- ¿Tiene plan de trabajo individual para sus actividades en el proyecto?
- ¿En su proyecto se gestionan los costos, plazos y recursos?
- ¿Se establece una fecha fija para la revisión y control de las entregas del proyecto?
- ¿Se firma en un acta de aceptación la entrega de la documentación al cliente?
- ¿Se firma con el cliente el documento donde se acuerde la responsabilidad de las partes?

### **Calidad**

- ¿En su proyecto existe el expediente de las pruebas realizadas?
- ¿Se le entrega con defectos el producto al cliente?
- ¿Considera que debe recibir capacitación para desempeñar su rol?
- ¿Su proyecto cuenta con equipo de calidad?
- ¿Se ejecutan las pruebas de aceptación con el cliente?
- ¿Tiene conocimiento de los indicadores que permiten notar que alguno de los Proyectos está en una situación alarmante y los mantiene actualizados?

### **Mediciones**

- ¿Utiliza herramientas para el control de versiones?
- ¿Se definen estándares a utilizar en el proyecto?
- ¿Se realizan pruebas internas en el proyecto?
- ¿En su proyecto se utiliza alguna herramienta para la automatización del proceso de pruebas?
- ¿Aplica métricas en su Proyecto?
- ¿Se posee registros de mediciones relevantes de los proyectos de su facultad, que permitan realizar estimaciones para planificaciones futuras, análisis de avances o cualquier otro estudio a favor de mejorar los procesos de desarrollo o los proyectos?

### **Organizativas**

Seleccione la facultad a la que usted pertenece.

Determine su nivel de participación en las acciones para el análisis de los problemas y la toma de decisiones en los proyectos productivos:

Determine su nivel de participación en las acciones del proceso productivo que se proponen.

¿Están definidos los roles de los miembros del proyecto?

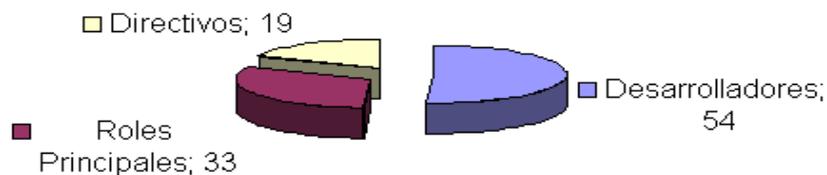
¿Están definidas las responsabilidades de los miembros del proyecto?

Los roles y responsabilidades de su proyecto se definen:

¿Le interesa realmente el rol que desempeña en su proyecto?

### 2.5.1. Análisis de las variables de control de la encuesta.

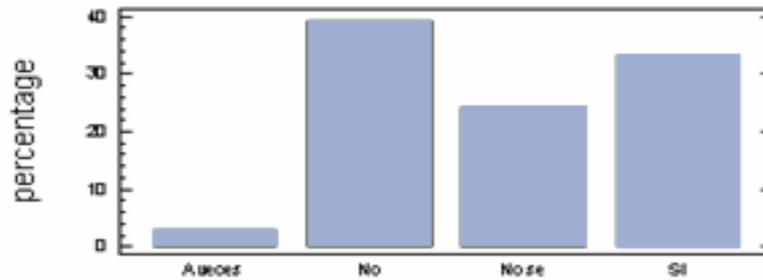
El diagnóstico fue realizado a los representantes de los proyectos que se realizan en la facultad (Arquitectura, Calidad, ERP, ERP- MINFAR, SIGEP, Sistema Único de Aduana, INSIGNE ) tanto a Directivos, Roles principales como a desarrolladores, se diagnosticaron 19 directivos, 54 desarrolladores y 33 roles principales. Ver **Anexo 5**.



**Figura 5.** Total de personas diagnosticadas.

### 2.5.2. Análisis de los resultados relacionados con la calidad de los proyectos.

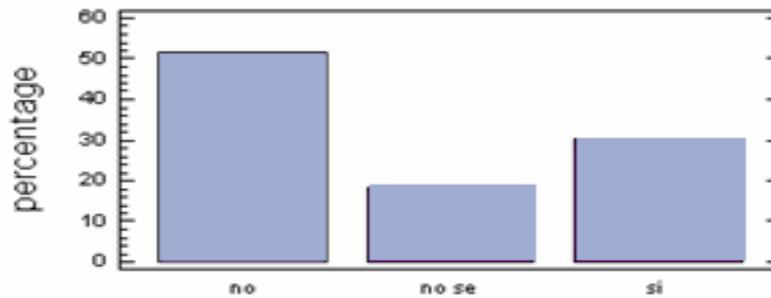
En el diagnóstico realizado se detectó que existen problemas graves de planificación pues a pesar de la existencia de dirigentes y un grupo de calidad en cada proyecto, no se planifican y organizan las reuniones para controlar el trabajo en el proyecto de la forma requerida, en ocasiones no se establecen fechas fijas para revisar y controlar los entregables del proyecto, además algunos de los involucrados en el proyecto no conocen de la firma de un acta de aceptación de entrega de la documentación al cliente.



Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	1	1	0,0303	1	0,0303
2	2	13	0,3939	14	0,4242
3	3	8	0,2424	22	0,6667
4	4	11	0,3333	33	1,0000

**Figura 5.** ¿Se firma un acta de aceptación de la entrega de la documentación del proyecto?

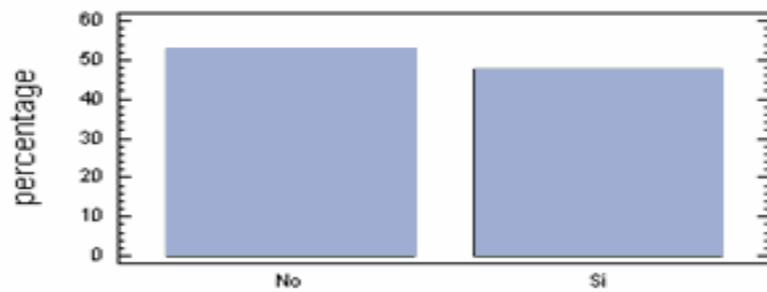
En cuanto a la calidad podemos ver que en los proyectos de la facultad un 52 % de los encuestados no conoce que existe un expediente que controle las pruebas realizadas durante el ciclo de vida del proyecto.



Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	1	17	0,5152	17	0,5152
2	2	6	0,1818	23	0,6970
3	3	10	0,3030	33	1,0000

**Figura 6** ¿En su proyecto existe el expediente de las pruebas realizadas?

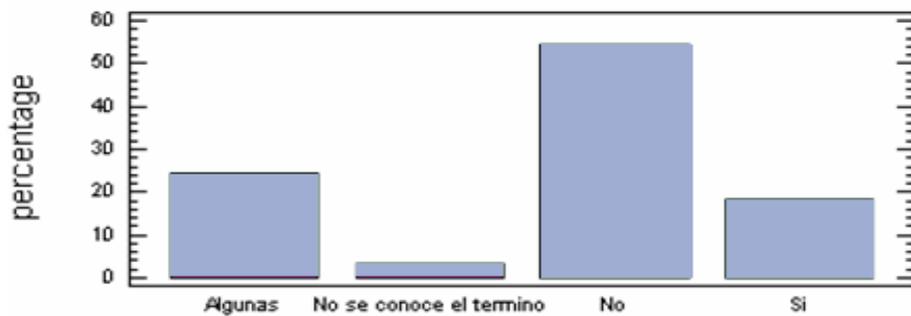
Existen integrantes del grupo de desarrollo que no conocen de los indicadores que monitoreen el proceso de desarrollo de software.



Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	1	10	0,5263	10	0,5263
2	2	9	0,4737	19	1,0000

**Figura 7** ¿Se tiene conocimiento de los indicadores que permiten notar que alguno de los proyectos esta en una situación alarmante?

En la mayoría de los proyectos no se utilizan métricas para evaluar las características del software y definir con mayor exactitud el desarrollo y calidad del producto que se esta desarrollando.



Class	Value	Frequency	Relative Frequency	Cumulative Frequency	Cum. Rel. Frequency
1	1	8	0,2424	8	0,2424
2	2	1	0,0303	9	0,2727
3	3	18	0,5455	27	0,8182
4	4	6	0,1818	33	1,0000

**Fig. 1.4** ¿Se Aplican métricas en su proyecto?

Es también importante conocer que la preparación requerida para asumir el rol que desempeñan los desarrolladores la adquirieron sobre la marcha y esto muchas veces afecta la calidad de los productos finales.

### **2.5.3. Principales problemas identificados en la encuesta.**

De los resultados obtenidos a partir del análisis estadístico del diagnóstico y de los problemas identificados por Febles (FEBLES,2004) se confecciona una espina de pescado para representar las dificultades que existen en la facultad que afectan la calidad del software y causan insatisfacción de las necesidades del cliente. Ver anexo 6. Se identifican los principales problemas clasificados en cuanto:

#### **Organizativas:**

- P1: Existe falta de comunicación efectiva entre los involucrados (usuarios, desarrolladores, administradores, clientes e investigadores).
- P8: No existe una preparación requerida para desempeñar los roles que se asignan, la mayoría aprenden sobre la marcha.
- P9: No todas las personas que tienen un rol definido están interesados por el mismo.

#### **Planificación y Seguimiento:**

- P4: No existe un procedimiento para llevar a cabo las actividades relacionadas con el desarrollo del software.
- P13: Casi nunca se firma un acta de aceptación de la entrega de la documentación al cliente.
- P14: Los proyectos están excesivamente tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas no se detectan en este medio indisciplinado y caótico de desarrollo.

#### **Mediciones:**

- P15: La evaluación se limita a pruebas de caja negra en la fase final de los proyectos.
- P17: No se exige responsabilidades, ni se puede medir productividad, cantidad de errores cometidos a nivel individual.
- P21: No se miden los procesos, ni los productos con métricas rigurosas lo que implica que no se pueda controlar ni mejorar el proceso a partir de los resultados en su ejecución.

#### **Calidad:**

- P16: No existe el expediente de las pruebas realizadas.
- P20: No se pueden detectar con antelación los problemas en el software afectando esto finalmente la calidad en los productos.
- P22: No se tiene conocimiento de los indicadores que permiten notar que alguno de los Proyectos está en una situación alarmante.

**2.6. Conclusiones parciales.**

El análisis realizado en el presente capítulo, de la actividad productiva en la Universidad de las Ciencias Informáticas muestra las dificultades que existen en los proyectos productivos, más específicamente de la Facultad 4 que afectan la calidad del producto, por ello la necesidad de establecer la manera de asegurar la calidad durante el ciclo de vida del software para mejorar la productividad y que se satisfagan las necesidades del cliente.

También se realiza un estudio sobre la importancia que tiene para las empresas cubanas contar con un sistema ERP que permita estandarizar la comunicación entre las empresas nacionales, y la gestión económica de los recursos a nivel nacional, provincial y municipal.

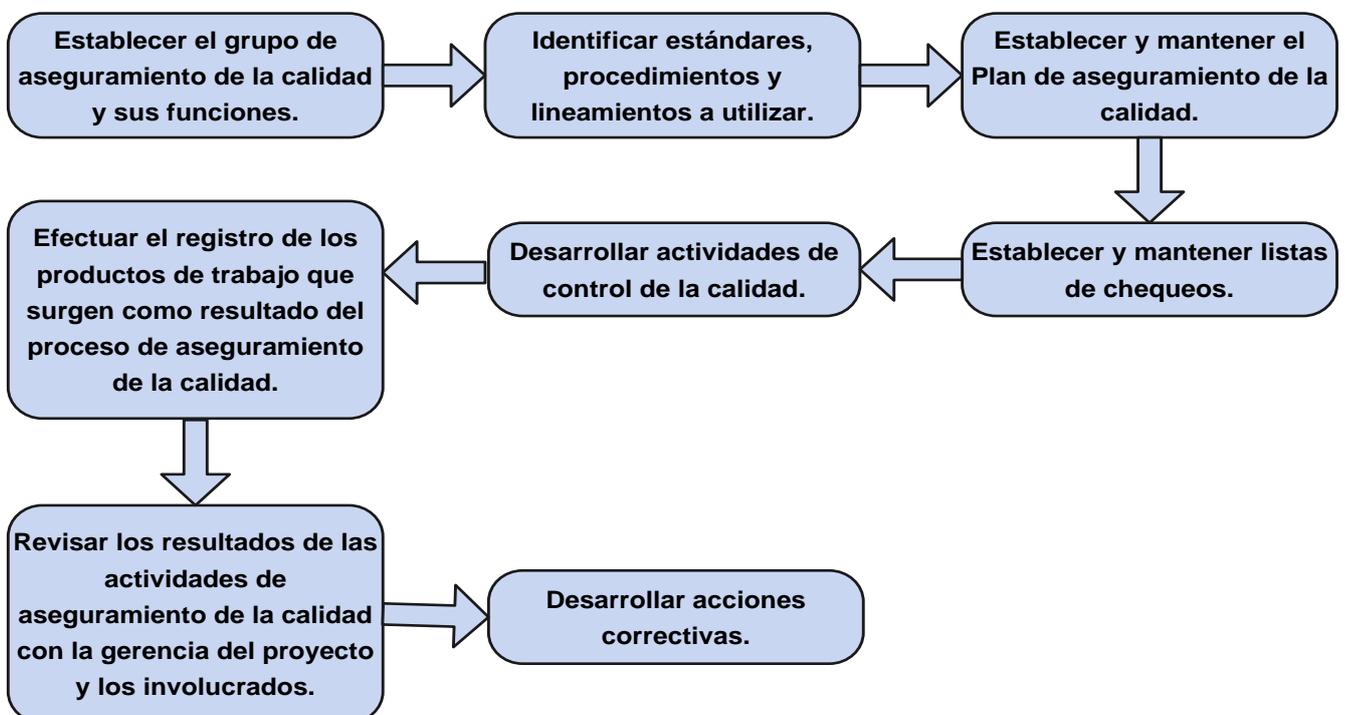
CAPITULO 3

PROPUESTA DEL PROCEDIMIENTO.

3 CAPITULO III.

3.1. Introducción.

En el presente capítulo se explica la propuesta de procedimiento para el Aseguramiento de la Calidad del Software en el proyecto INSIGNE, en los flujos de trabajo Modelamiento del Negocio, Requerimiento, Análisis y Diseño. Se describen los pasos a seguir para el correcto desarrollo del procedimiento y se referencia la documentación para la realización de dicha propuesta. Se propone una serie de etapas que dentro contienen actividades y estas tareas a realizar para culminar exitosamente y con calidad los flujos de trabajo especificados. De forma general las etapas especificadas son Establecer el grupo de aseguramiento de la calidad y sus funciones, Identificar estándares, procedimientos y lineamientos a utilizar, Establecer y mantener el Plan de aseguramiento de la calidad, Establecer y mantener listas de chequeos, Desarrollar las actividades de control de la calidad, Efectuar el registro de los productos de trabajo que surgen como resultado del proceso de aseguramiento de la calidad, Revisar los resultados de las actividades de aseguramiento de la calidad con la gerencia del proyecto y los involucrados y por último en caso de existir no conformidades Desarrollar acciones correctivas.



### 3.2. Resumen.

En la actualidad, el desarrollo de software ha tomado gran importancia en el mundo, siendo ésta cada vez más creciente por lo que obtener productos finales con la mayor calidad y, por consiguiente, con un mayor grado de satisfacción de sus clientes es el objetivo principal.

El procedimiento propuesto para el Aseguramiento de la Calidad del Software en el proyecto INSIGNE, especifica 8 etapas que contienen actividades y estas tareas a seguir por el grupo de aseguramiento de la calidad para asegurar la calidad en los flujos de trabajo Modelamiento del Negocio, Requerimiento, Análisis y Diseño. Ver Flujo del procedimiento, Anexo 10. Para la realización de dicho procedimiento se utilizo estándares reconocidos y aplicados mundialmente, es el caso CMMI v1.2 que provee una guía para mejorar los procesos de una organización y su capacidad de gestionar el desarrollo, la adquisición y el mantenimiento de los productos software.

El Aseguramiento de la Calidad de Software es un conjunto de actividades sistemáticas y planificadas que apoyan a que los productos y procesos de software cumplan con los requisitos, estándares y procesos definidos.

### 3.3. Nombre del procedimiento.

Aseguramiento de la Calidad para el proyecto INSIGNE.

### 3.4. Objetivo.

Establecer y detallar actividades a seguir por el grupo de aseguramiento de la calidad del proyecto INSIGNE, orientar sobre los principios, disposiciones generales, reglas y normativas a seguir para asegurar la calidad durante el desarrollo de los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño.

### 3.5. Alcance.

Es aplicable para asegurar la calidad en el Proyecto INSIGNE para los flujos de trabajo Modelamiento del Negocio, Requerimientos, Análisis y Diseño.

### 3.6. Referencias.

Manual de procedimientos IPP-1000:2008.

Área de proceso PPQA en CMMI v1.2.

### 3.7. Responsables.

**Ejecuta:** Grupo de Aseguramiento de la Calidad del Proyecto INSIGNE.

**Responsable de su ejecución:** Jefe de la Línea de Calidad y Grupo de Aseguramiento de la Calidad del Proyecto INSIGNE.

**Revisa y actualiza este procedimiento:** Jefe de Proyecto INSIGNE, Jefe y especialistas de la Línea de calidad del proyecto.

**Fiscaliza su cumplimiento:** Jefe de Proyecto INSIGNE.

### 3.8. Términos y definiciones.

**( a ):** Artefatos.

**Acción correctiva:** Acción tomada para eliminar la causa de una no conformidad detectada u otra situación indeseable. (PEREIRO, 2005)

**Actividad:** Conjunto de operaciones o tareas propias de una persona o entidad. (REAL ACADEMIA ESPAÑOLA)

**Artefacto de software:** Cualquier cosa que resulte del proceso de desarrollo de software; por ejemplo: documentos de requisitos, especificaciones, diseños, software, etc. (SQUAC, 2006)

**Auditoría interna:** Actividad independiente y objetiva de aseguramiento y consulta, concebida para agregar valor y mejorar las operaciones de una organización. Ayuda a una organización a cumplir sus objetivos aportando un enfoque sistemático y disciplinado para evaluar y mejorar la eficacia de los procesos de gestión de riesgos, control y gobierno. (INTERNA, 2002)

**Competencias:** Atributos personales y aptitudes demostradas para aplicar conocimientos y habilidades.

**CP:** Caso de Prueba. Conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Siempre es ejecutada como una unidad, desde el comienzo hasta el final.(AGUILAR, 2008)

**DCS:** Dirección de Calidad de Software de la UCI, es un centro de referencia de calidad y órgano de certificación de software, conocido y acreditado a nivel nacional.

**Especialista funcional:** Encargado de aclarar todas las dudas que surjan del negocio a automatizar durante la captura de requisitos y el resto del desarrollo, servir de apoyo al equipo de desarrollo para lograr el entendimiento correcto del negocio y participar en las pruebas de calidad que se realicen.(TERUEL, 2007)

**Etapa:** Fase en el desarrollo de una acción u obra. (REAL ACADEMIA ESPAÑOLA)

**Grupo de aseguramiento de la calidad:** grupo de personas que tienen la responsabilidad de la planificación de aseguramiento de la calidad, supervisión, mantenimiento de registros, análisis e informes. Además coordina el control y la gestión de cambios; ayuda a recopilar y analizar las métricas del software. (SCALONE, 2006)

**Jefe de Línea:** Administrador de la Calidad. Encargado de asegurar la calidad en el proceso de desarrollo de software y que la aplicación producida se ajusta a las especificaciones y está

razonablemente libre de errores, proporciona una metodología para realizar las pruebas, evalúa los resultados que se obtienen en las pruebas de calidad.(TERUEL, 2007)

**Jefe de Proyecto:** Encargado de la definición del proyecto, toma las decisiones en el proyecto, aprueba las tecnologías a usar en el desarrollo del proyecto, coordina y organiza las tareas que se asignan a los miembros del equipo de desarrollo, gestiona los recursos y materiales necesarios para el proyecto y para el equipo de desarrollo y lleva a cabo todo el proceso de gestión de proyecto.(TERUEL, 2007)

**Plan de Gestión de Configuración:** Describe todas las actividades de gestión de Configuración y control de cambios (CCM) que se llevan a cabo durante el proceso de desarrollo del software. Detalla el cronograma de actividades, la asignación de responsabilidades, y los recursos requeridos para llevar a cabo la CM. Establece las políticas para la gestión de configuración y el control de los cambios.(SOFTWARE, 2007)

**Plan de Pruebas:** En el Plan de pruebas se identifican los objetivos y la estrategia de las pruebas. Se define además cómo monitorear y evaluar el progreso de las pruebas.(AGUILAR, 2008)

**Procedimiento:** Sucesión cronológica de operaciones concatenadas entre sí, que se constituyen en una unidad de función para la realización de una actividad o tarea específica dentro de un ámbito predeterminado de aplicación. Todo procedimiento involucra actividades y tareas del personal, determinación de tiempos de métodos de trabajo y de control para lograr el cabal, oportuno y eficiente desarrollo de las operaciones. (DEFINICION.ORG, 2008)

**Proceso:** Conjunto de recursos y actividades interrelacionados que transforman elementos de entradas en elementos de salida. (CONTENIDOS, 2006)

**( r ):** Responsable.

**Responsable:** Obligado a responder de algo o por alguien. (REAL ACADEMIA ESPAÑOLA)

**RTF:** Revisiones técnicas formales. Es una actividad de garantía de la calidad del software llevada a cabo por los ingenieros del software (y otros). (PRESSMAN, 2002)

**SQA:** Aseguramiento de la Calidad del Software.

**Tarea:** Trabajo que debe hacerse en tiempo limitado. (REAL ACADEMIA ESPAÑOLA)

### 3.9. Normas generales.

3.9.1. El aseguramiento de la calidad es una actividad de soporte que se realiza durante todo el ciclo de vida del proyecto. Por lo tanto se determina concluido el proceso una vez que el proyecto ha concluido.

3.9.2. Las competencias a considerar en la evaluación del personal que integre el grupo de aseguramiento las define el Jefe de la Línea de calidad.

- 3.9.3. El grupo de aseguramiento de la calidad es diferente del equipo de desarrollo del software.
- 3.9.4. El responsable de asignar los roles y responsabilidades del grupo de aseguramiento de la calidad es el Jefe de la Línea de calidad.
- 3.9.5. Para conformar el Plan de aseguramiento de la calidad, las Listas de chequeo y el documento de las No Conformidades se utilizan las plantillas que propone la Dirección de Calidad de Software.
- 3.9.6. El Jefe del equipo de revisiones y pruebas es el Responsable de la calidad en la Línea.
- 3.9.7. Para hacer una revisión el Responsable de la Línea contacta con el Jefe de Línea.
- 3.9.8. Las revisiones y pruebas a realizarse serán aprobadas por el Jefe de Proyecto y el Jefe de la Línea de calidad.
- 3.9.9. Los desarrolladores tienen que entregar la información solicitada por el equipo revisor, el día planificado para la fecha de inicio de la revisión.
- 3.9.10. Los desarrolladores forman parte del equipo de revisión.
- 3.9.11. Los especialistas funcionales tienen que formar parte del equipo de revisión y prueba que van a revisar y probar la calidad de los Productos Software existentes.
- 3.9.12. Las pruebas de software realizadas a los productos existentes quedaran documentadas en el Documento de Sistemas Software evaluados.
- 3.9.13. La auditoria interna del proceso la realizan el Jefe de Proyecto y el Jefe de la Línea de calidad.
- 3.9.14. Las revisiones y las auditorías deben quedar documentadas en los informes finales de Revisión y Auditoría.

### **3.10. Desarrollo.**

#### **3.10.1. Establecer el grupo de aseguramiento de la calidad y sus funciones.**

##### **3.10.1.1. Conformar grupo de aseguramiento de la calidad.**

Se crea un grupo de aseguramiento de la calidad agrupado en la Línea de calidad definida por el proyecto, ese equipo es diferente del equipo de desarrollo del software, lo cual permite que se detecten con facilidad los errores que atenten contra la calidad del software. Además por cada línea definida en el proyecto existen personas encargadas de la calidad.

Para seleccionar el personal que va a formar parte del grupo de aseguramiento de la calidad se tiene en cuenta las competencias teniendo en cuenta los objetivos al que se aspire, esto garantiza que el personal seleccionado sea el indicado para cumplir con la tarea que se debe desarrollar, pues permite determinar los conocimientos, las habilidades y valores de los implicados en cuanto al tema que es objeto del diagnóstico.

( r ): Jefe de la Línea de Calidad y Jefe de Proyecto.

( a ): Estructura Proyecto ERP.

### 3.10.1.2. Asignar roles y responsabilidades del grupo de aseguramiento de la calidad.

Se establece dentro del grupo de aseguramiento de la calidad diferentes roles que tendrán las responsabilidad de garantizar las actividades definidas para asegurar la calidad. Se realiza teniendo en cuenta las competencias de cada uno de los que formen parte del grupo aseguramiento de la calidad.

Se define el número de personas por roles en el equipo de aseguramiento de la calidad:

**Administrador de la calidad:** responsable de asegurar la calidad del producto y de los procesos involucrados en el desarrollo del software, además es defensor de las pruebas, planificación y administración de recursos, resolución de problemas que impidan las pruebas y participa en la evaluación de calidad del producto final. Además es responsable de:

- Hacer entrevistas y visitas a los clientes durante todo el ciclo de desarrollo del software; al culminar esta actividad debe realizar un informe con las necesidades del cliente.
- Identificar las propiedades de calidad que debe cumplir el software, para esto hace un listado de las propiedades según las necesidades del cliente y los requisitos funcionales y no funcionales que debe cumplir cada software; debe insertarlas en el plan de aseguramiento de la calidad.
- Definir las métricas que se utilizarán para medir las propiedades de calidad que se identifiquen; las mismas se insertan en el plan de aseguramiento de la calidad.
- Revisar las entregas semanales que hace el equipo de desarrollo de software, debe asegurar que los productos se realicen en la fecha señala en el plan del proyecto y que además se realicen todos los artefactos vinculados de forma directa con la calidad del producto; esta actividad culmina con un informe de revisión.
- Revisar la calidad del producto y hacer un resumen de la situación actual del producto que se esta revisando; este resumen se introduce en el informe de revisión.
- Elaborar el Plan de aseguramiento de la calidad y el Plan de pruebas en la fase inicial del desarrollo del proyecto el administrador de la calidad elabora el plan de aseguramiento de la calidad y le da un seguimiento sistemático en las restantes fases, sin embargo debe elaborar un plan de prueba para cada fase de desarrollo.

**Analista de prueba:** responsable de identificar y definir las pruebas requeridas, monitorear el progreso de la prueba y el resultado en cada ciclo de prueba y evaluar la calidad total

experimentada como un resultado de las actividades de prueba. Representa las necesidades de los stakeholder que no tienen representación directa con el proyecto. Además es responsable de:

- Hacer entrevistas y visitas a los clientes durante todo el ciclo de desarrollo del software.
- Identificar los métodos de pruebas y los procedimientos de prueba que se realizarán en las diferentes fases de desarrollo; es importante aclarar que este plan de prueba es totalmente diferente al plan de prueba que debe realizar el equipo de desarrollo de software en el flujo de prueba que propone RUP.

**Diseñador de prueba:** responsable de definir los casos de prueba y asegurar su implementación exitosa. Identifica las técnicas apropiadas, herramientas e instrucciones para implementar las pruebas necesarias. Además es responsable de:

- Diseñar los casos de pruebas que se realizarán en todo el ciclo de vida del software; para ello construye el documento diseño de casos de pruebas.
- Evaluar y documentar el resultado de las pruebas realizadas al software.
- Elaborar las listas de chequeo que se realizarán en las diferentes fases.

**Probador:** responsable de ejecutar las pruebas en las diferentes fases de desarrollo del software y el encargado de registrar los resultados de las pruebas. Además es responsable de:

- Ejecutar los casos de pruebas y anotar las no conformidades encontradas durante la realización de las pruebas.
- Realizar las listas de chequeos definidas por el diseñador de pruebas.

( r ): Jefe de la Línea de Calidad.

( a ): Documento de Roles y Responsabilidades.

### 3.10.1.3. Asignar recursos al grupo de aseguramiento de la calidad.

El objetivo fundamental de esta actividad es asignar recursos o herramientas de trabajo definidas por la dirección del proyecto al grupo de aseguramiento de la calidad para el desarrollo de las actividades de aseguramiento de la calidad. Las 10 computadoras ubicadas en el laboratorio 808 con que cuenta la Línea de Calidad son asignadas a cada integrante del grupo teniendo en cuenta las tareas que realiza dentro del proyecto.

( r ): Jefe de Proyecto y Jefe de la Línea de Calidad.

( a ): Capital Humano ERP.

### 3.10.1.4. Capacitar al grupo de aseguramiento de la calidad.

Se llevan a cabo diferentes acciones para capacitar a los integrantes del grupo de aseguramiento de la calidad. Es necesario que cada persona que conforme de este grupo tenga amplio conocimiento en temas de calidad, por lo que se proponen algunos temas en los que pueden ser entrenados:

- Cómo llevar a cabo los procesos de Revisiones, Revisiones Técnicas Formales y Auditorias.
- Cómo identificar las Desviaciones resultantes de los procesos de Revisiones, Revisiones Técnicas Formales y Auditorias.
- Cómo medir los procesos a través de métricas.
- Cómo evaluar los procesos a través de métodos de evaluación.
- Cómo desarrollar las pruebas al software.
- Las responsabilidades que tiene una persona a la hora de ocupar un rol determinado, como parte del grupo aseguramiento de la calidad del proyecto.
- Actividades específicas del grupo aseguramiento de la calidad.

Además se debe conocer las principales ideas del proyecto que se esta desarrollando, no solo debe estar entrenado en temas de calidad sino en todo lo referente al entorno de procesos que se definan a desarrollar, porque tienen que velar por obtener un producto final con excelente calidad. Se imparten cursos de:

- Caja blanca.
- Diseño de casos de prueba.

( r ): Jefe de Proyecto y Jefe de la Línea de Calidad.

( a ): Plan de capacitación.

### **3.10.2. Identificar estándares, procedimientos y lineamientos a utilizar.**

#### **3.10.2.1. Determinar estándares de calidad.**

Seleccionar los estándares a seguir para asegurar la calidad en el proyecto. Basándose en los modelos y normas más utilizados en IP entre ellos:

- CMMI Integración del Modelo de Capacidad y Madurez.
- ISO 9001:2000 Sistema de Gestión de la Calidad. Requisitos.
- ISO 9126 Métricas de Software.
- ISO 9127 La documentación del software.
- ISO 12119 Evaluación y test de producción de software.
- ISO 12207 Procesos del ciclo de vida del software.

- ISO 15486 Gestión de la configuración de software.
- ISO 19011:2002 Directrices para la auditoría de los Sistemas de Gestión de la Calidad.
- IEEE 829 Estándar para la documentación de las pruebas del software.
- IEEE 1008 Estándar para las pruebas de unidad de software.
- IEEE 1028 Estándar para revisiones de Software.
- IEEE 1045 Métricas de productividad.
- IEEE 1061 Métrica de calidad del software.
- IEEE 1233 Guía para el desarrollo de especificaciones de requerimiento del sistemas.
- IEEE 1540 Standard for Software Life Cycle Processes—Risk Management.

( r ):Grupo de aseguramiento de la calidad.

( a ): Plan de Aseguramiento de la Calidad.

### **3.10.2.1.1. Definir arquitectura de la información.**

Se define cuál va a ser el contenido de la aplicación, sus objetivos, herramientas, cómo se llamarán los botones y en qué posición éstos van a estar, cómo se va a relacionar el usuario con el sitio, grupos de contenidos, tabla de contenidos, mapa del sitio y sistema de interacción entre otras. Se realizara a través de tres fases:

- La combinación de esquemas de organización, etiquetado y navegación, dentro de un sistema de información.
- El diseño estructural de un espacio de información para facilitar la terminación de tareas y el acceso intuitivo al contenido.
- El arte y la ciencia de estructurar y clasificar sitios Web e intranets, para ayudar a las personas a encontrar y administrar información.

( r ): Grupo de aseguramiento de la calidad.

( a ): Plantilla DCS - Arquitectura de Información v1.0.

### **3.10.2.2. Seleccionar lineamientos de calidad.**

Seleccionar los lineamientos, de los establecidos por la Dirección de Calidad de Software, a utilizar durante el desarrollo de los flujos de trabajo Modelamiento del Negocio, Requerimientos y Análisis y Diseño. Estos son:

- Proyecto
- Trabajo en equipo
- Planificación
- Gestión de Configuración:
- Trabajo personal.

- Modelos de calidad.
- Vocabulario.
- Requisitos.
- Arquitectura.
- Investigación.

( r ): Grupo de aseguramiento de la calidad.

( a ): Plan de Aseguramiento de la Calidad.

### **3.10.3. Establecer y mantener el Plan de aseguramiento de la calidad.**

#### **3.10.3.1. Conformar el Plan de aseguramiento de la calidad.**

Se conforma el Plan de aseguramiento de la calidad para asegurar la evaluación objetiva de los productos y procesos identificados. Para ello deben planificarse tipos de actividades evaluadoras como revisiones, auditorías y pruebas sobre los procesos y productos según sea necesario. La evaluación de los procesos debe realizarse a través de la adherencia de los mismos a los procedimientos, estándares y lineamientos identificados. Las actividades de evaluación de los productos de trabajo deben realizarse antes de la entrega de los mismos al cliente y en los hitos de su desarrollo establecido en el plan de proyecto. Para documentar el Plan de Aseguramiento de la calidad se utiliza la Plantilla DCS - Plan aseguramiento de la calidad v1.0 definida en el expediente de proyecto.

##### **3.10.3.1.1. Definir propósito.**

Se expone el propósito del Plan aseguramiento de la calidad donde se reflejan las actividades de auditoría, revisión y evaluación de los procesos en el desarrollo de estos flujos.

##### **3.10.3.1.2. Definir alcance.**

Se define el alcance del Plan aseguramiento de la calidad teniendo en cuenta el alcance del Plan de Desarrollo de Software del Proyecto aprobado para estos flujos, ya que estos dos planes se llevan en conjunto.

##### **3.10.3.1.3. Definir objetivos de calidad.**

Se incluyen los requerimientos del proyecto que están alineados con los requerimientos de calidad definidos por la dirección.

##### **3.10.3.1.4. Definir tareas y responsabilidades.**

Se muestra una tabla que registra la asignación de responsables y participantes por tareas. Ver Anexo 7

**3.10.3.1.5. Definir mediciones y métricas.**

Se describe todo el proceso de métricas que se realizará como producto del monitoreo del trabajo.

**3.10.3.1.6. Definir estándares y guías.**

Lista de los estándares y guías utilizadas por el Plan aseguramiento de la calidad, acorde a los que plantea el Plan de Desarrollo de Software del proyecto INSIGNE. Ver Anexo 8.

**3.10.3.1.7. Definir Plan de revisiones y auditorias que incluye:**

- **Tareas generales de Revisiones y Auditorias** [Se describa brevemente cada tipo de revisión y auditoria que se llevará a cabo en el proyecto.]
- **Cronograma** [Detallar el cronograma para las revisiones y auditorias. Este debe incluir las revisiones y auditorias programadas en las fechas principales del proyecto, así como revisiones que son provocadas por la entrega de artefactos del proyecto.]
- **Organización y responsabilidades** [Se listan los involucrados en cada una de las actividades de revisión y de auditoria identificadas.]
- **Resolución de problemas y actividades de corrección** [Se describe los procedimientos para informar y manejar problemas identificados durante las revisiones y auditorias del proyecto.]
- **Herramientas, técnicas y metodologías** [Se describe las herramientas, técnicas o metodologías específicas que serán usadas para llevar a cabo las actividades de revisión y de auditoria identificadas en este plan.]

**3.10.3.1.8. Definir pruebas y evaluaciones.**

Se definen las pruebas a realizar durante el proceso de desarrollo del proyecto. Se hace referencia al Plan de pruebas.

**3.10.3.1.9. Definir herramientas, técnicas y metodologías.**

Listar todas las herramientas, técnicas y metodologías utilizadas en las actividades del Plan de Calidad.

**3.10.3.1.10. T: Definir la gestión de configuración.**

Se hace referencia al Plan de Gestión de Configuración.

**3.10.3.1.11. Establecer registros de calidad.**

Descripciones de varios registros de calidad que se mantendrán durante el proyecto, incluyendo cómo y dónde cada tipo de registro se guardará y por cuánto tiempo.

( r ): Grupo de aseguramiento de la calidad.

( a ): Plan de Aseguramiento de la Calidad.

### 3.10.4. Establecer y mantener listas de chequeos.

#### 3.10.4.1. Conformar las listas de chequeos.

Conformar las listas de chequeos para la realización de las revisiones en el proyecto. Para la confección de estas se utiliza la Plantilla DCS - Listas de chequeo definido en el expediente de proyecto, contiene:

**Fecha** [Se escribe la fecha en la que se realizo.]

**Responsable que la aplicó** [Se escribe el nombre de la persona que trabajo con ella.

Tiene una tabla que resume los siguientes aspectos. Ver Anexo 9]

Para la confección de la tabla tener en cuenta:

- **Nivel** solo se define cuando existen aspecto de mayor importancia que otros, para ello se usarán signos de exclamación (!), entre más tenga más importante es.
- **Criterio de evaluación** se ubicará el aspecto a evaluar, siempre con una redacción nítida
- **Evaluación** es para el caso de la aplicación de la Lista de Chequeo. Dicha evaluación se encontrará en el rango de 0-5
- **N.P.** Significa No Procede y es para el caso de la aplicación que ese aspecto no sea factible su valoración
- **Observaciones** es para cualquier cosa que quiera incluir la persona que aplica dicha lista
- **Métricas, Submétricas y % cumplimiento** es para la aplicación de métricas a las iteraciones realizadas.

( r ): Grupo de aseguramiento de la calidad.

( a ): Listas de chequeo.

### 3.10.5. Desarrollar actividades de control de la calidad.

Se realizan las actividades descritas en el plan de aseguramiento de la calidad del proyecto y se documentan e informan los resultados, problemas y no conformidades detectados en cada una de las actividades. Las no conformidades serán documentadas utilizando la Plantilla DCS-No Conformidades (ampliada) v1.0 del expediente de proyecto.

Se realizan RTF como método formal para asegurar la calidad, teniendo como objetivos descubrir errores antes que el producto esté en manos del cliente, encontrar desviaciones, documentarlas e informarlas al Jefe de Proyecto, de acuerdo a los requerimientos y estándares establecidos.

Se definen aspectos de revisiones a tener en cuenta:

- Verificar si cada uno de los documentos, artefactos o productos fueron creados en la fecha establecida y por los roles identificados en el proyecto.
- Verificar si cada uno de los documentos, artefactos o productos tiene buena calidad teniendo en cuenta las métricas.
- Comparar los resultados obtenidos con los deseados.

### 1.10.5.1. Conformar el equipo de revisiones y pruebas.

Se conforma un **equipo de revisiones** por cada Línea de desarrollo que lleva a cabo todas las revisiones establecidas en el plan de aseguramiento de la calidad con el propósito de encontrar errores. El equipo debe estar integrado de 3 a 5 personas con los siguientes roles:

**Jefe de revisión:** encargado de llevar a cabo la preparación del proceso de revisión para lograr el éxito en la misma.

**Desarrollador:** persona que desarrolló el producto que será revisado.

**Revisores:** quienes llevarán a cabo la revisión.

Una vez definido el cronograma de las revisiones se ejecutan las siguientes acciones:

- El Jefe de revisión contacta al Jefe de Línea, quien evalúa la disponibilidad de los procesos, artefacto y productos.
- Los revisores se preparan por adelantado, revisando los procesos, artefacto o productos y toman las notas pertinentes.
- El Jefe de Línea establece un horario para la revisión planificada.
- Se realiza la revisión en la que participa el Jefe de revisión, el desarrollador y los revisores.
- El desarrollador explica el proceso, artefacto o productos y los revisores emiten criterios según la preparación previamente realizada.
- El Jefe de revisión toma nota de los problemas y errores válidos que aparezcan.
- Al terminar la revisión, el Jefe de revisión documenta cada proceso, artefacto o productos en el Informe Final de Revisión.

Se conforma el **equipo de pruebas** por cada Línea de desarrollo, los que además de diseñar las pruebas ejecutan las mismas. El equipo debe estar conformado por los siguientes roles:

**Jefe de Pruebas:** Es el responsable del éxito de la prueba, Este rol es el defensor de prueba y de la calidad, planifica y administra los recursos, resuelve los problemas que impidan las pruebas.

**Analista de pruebas:** Es el responsable de identificar y definir las pruebas requeridas, monitorear el progreso de las mismas y el resultado en cada ciclo de pruebas, evaluando la

calidad total experimentada como un resultado de las actividades de prueba. Este rol lleva la responsabilidad para representar apropiadamente las necesidades de los stakeholder que no tienen representación regular y directa en el proyecto.

**Diseñador de prueba:** Es el responsable de definir el método de prueba y asegurar su implementación exitosa. El rol incluye identificar técnicas apropiadas, herramientas e instrucciones para implementar las pruebas necesarias y encauzar los recursos correspondientes para las pruebas.

**Probadores:** Conduce las pruebas necesarias y el registro del resultado de la mismas.

Una vez definido el cronograma de pruebas se ejecutan las siguientes acciones:

- Presentación del proceso de pruebas por parte del Jefe de Pruebas al Jefe de Línea, enfatizando en aspectos claves como la metodología a aplicar y las herramientas que se utilizarán.
- Entrenamiento del equipo de pruebas mediante los cursos de capacitación.
- Garantizar recursos físicos, herramientas y espacio al equipo de prueba.
- Asignación de puestos de trabajo compartiendo espacio con el equipo de desarrollo con facilidades de comunicación y posibilidades de reuniones periódicas.

**(r):** Jefe de la Línea de Calidad.

**(a):** Documento de Roles y Responsabilidades.

### **1.10.5.2.Revisar y probar la calidad de los productos software existentes.**

Se requiere de una revisión de los productos iniciales con que cuenta el proyecto para evaluar la calidad de los mismos y conocer que aportaría a la solución teniendo en cuenta la norma ISO 9126.(INFORMACIÓN, 2005)

- Revisar el diseño de lo que ya está implementado por las FAR para obtener resultado y definir las pautas de diseño, teniendo en cuenta las métricas internas de precisión, mantenibilidad, diagnosticabilidad, flexibilidad, como características de la funcionalidad y eficiencia.
- Revisar el estado del código de lo que ya está implementado por la FAR, para comparar los resultados con lo que se aspira por el proyecto. Se utilizan la métrica interna relacionada con la precisión para comprobar si cumple con las funcionalidades deseadas y flexibilidad para verificar si el producto permite la aplicación de una modificación.
- Revisar los Sistemas Contables Financieros certificados en el país por parte los especialistas funcionales. Se verifican las funcionalidades de los productos con respecto a

las del proyecto. Se revisa el código de los procesos definidos en estos, siguiendo las métricas internas relacionadas con las propiedades de funcionalidad, confiabilidad, usabilidad, eficiencia, portabilidad de los productos software. Se tiene en cuenta si los productos tienen seguridad, pueden ser modificados, son flexibles a cambios en el código y si son adaptables a cualquier ambiente especificado. Estos productos son:

- VERSAT SARASOLA.
  - RODAS XXI Versión 3.0.
  - SISCONT5.
  - CONDOR.
  - SISCOMIP.
  - SICEMA PLUS.
  - SISTEMA eTES.
- Realizar pruebas de caja Blanca a los Sistemas Contables Financieros seleccionados por la dirección del proyecto, para analizar el código de los mismos con el objetivo de reutilizarlos. Se recomienda para la confección de la documentación de las pruebas realizadas utilizar el estándar IEEE 829.

**(r):** Equipo de revisiones y pruebas.

**(a):** Informe Final de Revisión y Documento de Sistemas Software evaluados.

### **1.10.5.3. Realizar revisión administrativa.**

#### **1.10.5.3.1. Revisar planificación del proyecto.**

Se establecen revisiones a los documentos generados por la planificación del proyecto siguiendo las métricas internas aplicadas a los productos software. Se mide la seguridad, comprensibilidad, mantenibilidad, además se revisa otras características particulares.

Documento de Roles y Responsabilidades del proyecto: se revisa cada uno de los roles y responsabilidades identificadas según las necesidades del proyecto, se verifica que este documento haya sido creado en la fecha establecida y por el Jefe de Proyecto.

Plan de Capacitación del proyecto: se revisa que este plan tenga especificado los cursos de capacitación, los responsables de impartirlos, las personas que reciben los mismos y una fecha a realizar. También se verifica que haya sido creado por Jefe de Proyecto y Jefes de Línea.

Documento Estructura del proyecto ERP: verificar que haya sido creado por el planificador y que se cumpla en el proyecto con lo establecido en el mismo.

Cronograma de integración ERP: se revisan las actividades escritas y que haya sido creado por Jefe de proyecto.

Una vez probado la existencia de los documentos es evidente verificar que coincidan los datos de los documentos con los datos reales del proyecto:

- Se verifica en una reunión con los jefes de Línea que coincidan los roles y responsabilidades asignados en el proyecto con lo establecido en el Documento de Roles y Responsabilidades,
- Se verifica que se estén impartiendo los cursos de capacitación y que las personas que lo estén recibiendo sean las propuestas en el Plan de Capacitación del proyecto, haciendo una inspección dado el cronograma especificado en este plan.
- Se miden los conocimientos de los cursos impartidos al grupo de desarrollo, realizando un diagnóstico y otras evaluaciones definidas por los Jefes de Línea.
- Verificar el cumplimiento de las actividades definidas en el cronograma integración ERP, chequeando hasta la fecha todas las actividades propuestas.

### **1.10.5.3.2. Revisar Plan de iteración.**

Se realiza la revisión del documento Plan de iteración verificando que tenga bien definida la Introducción (Propósito, Alcance, Definiciones y abreviaciones, Referencias, Contenido), detalle de las actividades, su duración y fechas, y metas intermedias en el Plan (Actividades, Calendario, Entregas de la iteración), los Recursos que se utilizan para la iteración, la lista de los casos de uso que se desarrollan en esta iteración y los Criterios de Evaluación.

Se analizar que riesgo aportan cada una de las iteraciones definidas en el Plan de Iteración, además se verifica que este plan haya sido creado por el Jefe de Línea, y después de haber verificado la planificación y estado actual del proyecto.

**(r):** Equipo de revisiones.

**(a):** Informe Final de Revisión y Plantilla DCS - No Conformidades.

### **1.10.5.4. Revisar los artefactos generados.**

Se establece una revisión detallada para cada uno de los artefactos generados en los flujos de trabajo Modelamiento del Negocio, Requerimiento y Análisis y Diseño, siguiendo las métricas internas, verificando el cumplimiento de los documentos establecidos, y se proponen utilizar las listas de chequeo correspondientes a cada uno de estos o mediante la descripción de los mismos.

Los artefactos deben revisarse en pequeñas porciones para encontrar el mayor número de

errores y lograr un buen desarrollo de estos.

Se utiliza en el proyecto la notación de Modelado de Procesos de Negocio BPNM, por lo que se reajustan los artefactos, definidos en el Modelo de Desarrollo del proyecto, según las necesidades principales:

- Glosario de Términos.
- Modelo Conceptual.
- Descripción de Procesos.
- Seguimiento a los Requisitos.
- Plan de Mitigación de Riesgos.
- Pautas de Diseño.
- Documentación Arquitectónica.
- Modelo de Diseño.
- Modelo de Datos.

**(r):** Equipo de revisiones.

**(a):** Informe Final de Revisión y Plantilla DCS - No Conformidades.

### **1.10.5.5. Revisar procesos de negocio.**

Se establece realizar una revisión de los procesos de negocio definidos, verificar el cumplimiento tanto de las características funcionales, como técnicas establecidas para el proyecto hasta Análisis y Diseño, siguiendo el Cronograma de Integración del ERP.

**(r):** Equipo de revisiones.

**(a):** Informe Final de Revisión y Plantilla DCS - No Conformidades.

### **1.10.5.6. Realizar auditoría interna del proceso.**

Se define el programa de la auditoría interna a realizarse, fecha de la auditoría, los auditados, los objetivos, alcance y recursos necesarios para llevarla a cabo. Se efectúa una auditoría de los procesos con el objetivo de evaluar el proceso de desarrollo o de gestión, y evaluar su completitud y efectividad, determinando dónde se puede mejorar. Se hacen auditorías al proyecto para evaluar la productividad y eficacia del equipo que trabaja, así como la efectividad de los métodos y herramientas utilizados. Se realizan auditorías de gestión de proyecto cuyo objetivo es evaluar la efectividad de las prácticas de gestión realizadas y la organización del proyecto.

Se verifica que en las auditorías de procesos estén involucrados el Jefe de Proyecto, Planificador y los probadores del proyecto quienes son los que ejecutan las listas de chequeos previamente

elaboradas y detallan si el proceso es aprobado o no, en caso de que no sea aprobado debe ser informado y registrado y además debe indicarse la fecha de realización de la próxima auditoria, emiten un informe evaluativo de las desviaciones encontradas y establecen un plan de medidas correctivas. Utilizar el Procedimiento IPP-3201:2008-Auditorías a la actividad Productiva definido por la Dirección de Calidad de Software.

**(r):** Jefe de Proyecto y Jefe de Línea de Calidad.

**(a):** Informe Final de Auditoría y Plantilla DCS - No Conformidades.

### **3.10.6. Efectuar el registro de los productos de trabajo que surgen como resultado del proceso de aseguramiento de la calidad.**

Efectuar el registro de los productos de trabajo que surgen como resultado del proceso de aseguramiento de la calidad.

Plantilla DCS - Listas de chequeo.

Plantilla DCS - Plan aseguramiento de la calidad v1.0

Plantilla DCS - Plan de pruebas v1.0

Plantilla DCS - No conformidades (reducida) v1.0

Plantilla DCS - Diseño casos de prueba v2.0

Plantilla DCS - Respuestas a No Conformidades v1.0.

Documento de Sistemas Software evaluados.

Plantilla DCS - Arquitectura de Información v1.0.

Informe Final de Revisión.

Informe Final de Auditorías.

**(r):** Grupo de aseguramiento de la calidad.

### **3.10.7. Revisar los resultados de las actividades de aseguramiento de la calidad con la gerencia del proyecto y los involucrados.**

Se realiza con anterioridad una revisión a todas las actividades planificadas y ejecutadas por el grupo aseguramiento de la calidad del proyecto, así como fechas y duración de las mismas, también se analizan los documentos y demás artefactos referentes al proceso, para tener un control total de la calidad en el proyecto.

Se revisa por parte del Jefe de Línea de Calidad el estado y resultado de las actividades con los responsables de calidad de las restantes líneas y el Jefe de Proyecto en el período definido en el plan de aseguramiento de la calidad. Durante esta actividad además el Jefe de Línea de Calidad informará al Jefe de Proyecto las no conformidades no resueltas si fuera necesario y de conjunto con él se toman

acciones correctivas cuando los requerimientos y objetivos del aseguramiento de la calidad no son satisfechos.

Se analiza y valoran el estado en que se encuentran las actividades de aseguramiento de la calidad, después se evalúa teniendo en cuenta las tareas definidas en el área de proceso referida al aseguramiento de la calidad en CMMI para así saber lo que se debe fortalecer o mejorar.

**(r):** Jefe de Línea de Calidad y Jefe de Proyecto.

### **3.10.8. Desarrollar acciones correctivas.**

Se realizan acciones correctivas cuando los requerimientos y objetivos del aseguramiento de la calidad no son satisfechos o cuando el progreso de las actividades difiere significativamente de lo previsto en el plan de aseguramiento. Para ello se sigue el plan para resolución de No Conformidades detectadas. Las acciones correctivas conllevan a la redefinición de las actividades definidas en el plan de aseguramiento.

**(r):** Jefe de Línea de Calidad y Jefe de Proyecto.

### **3.11. Validación. Método Delphi.**

Delphi consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos futuros. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. Por lo tanto, la capacidad de predicción de la Delphi se basa en la utilización sistemática de un juicio intuitivo emitido por un grupo de expertos.

El Delphi es uno de los métodos de pronosticación más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de un grupo de expertos en el tema tratado. Permite rebasar el marco de las condiciones actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos que pueden estar basadas en un análisis lógico, como en su experiencia intuitiva.

Para aplicar el método y garantizar la calidad de los resultados, para lanzar y analizar la Delphi se siguieron tres etapas fundamentales:

- Elección de expertos.
- Elaboración del cuestionario, para validación de la propuesta.
- Desarrollo práctico y explotación de resultados.

#### Elección de expertos.

Se entiende por experto, tanto al individuo en sí como a un grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia.

La autenticidad de la valoración de los criterios de expertos puede ser determinada solamente, sobre la base de la solución práctica del problema y el análisis de los resultados.

Por experto se considera la persona capaz de ofrecer valoraciones concluyentes sobre la calidad de software y específicamente del conocimiento que posea del aseguramiento de la calidad del mismo.

La selección del posible equipo de experto se realizó bajo los siguientes criterios:

- Graduado de nivel superior.
- Conocimientos sobre la calidad y el aseguramiento de la calidad de software.
- Conocimientos y habilidades en actividades de desarrollo de software.

La selección de los posibles candidatos se realizó en la UCI, luego se les aplicó una encuesta para determinar el coeficiente de competencia en la temática “aseguramiento de la calidad”. Ver Anexo 10.

### Elaboración de los cuestionarios para validación de la propuesta.

Luego de haber seleccionado los expertos se elabora un cuestionario que se adapte a las características de los expertos, para validar la propuesta de procedimiento. Ver Anexo 11.

Las preguntas realizadas a los expertos son de tipo cualitativas, están medidas en una escala nominal, en su mayoría brindan valores de \_\_\_Si \_\_\_No \_\_\_No se.

### Desarrollo práctico y explotación de resultados.

Después de tener todas las encuestas realizadas:

Se determina el coeficiente de competencias de cada experto.

La competencia de los expertos se determina por el coeficiente K, el cual se calcula de acuerdo con la opinión del candidato sobre su nivel de conocimiento acerca del problema que se está resolviendo, y respecto a las fuentes que le permiten argumentar sus criterios.

El coeficiente K se calcula para cada experto mediante la siguiente expresión:

$$K = 1/2 (Kc + Ka)$$

Donde:

Kc = Coeficiente de conocimiento que tiene el experto sobre el tema en cuestión.

Ka = Coeficiente de argumentación o fundamentación de los criterios del experto.

El coeficiente de conocimiento (**Kc**) está en dependencia de la autovaloración del experto en relación a su nivel de conocimientos, que marca en una tabla. El número resultante se multiplica por 0,1. Se muestra el resultado de los cálculos en la tabla.

### CAPITULO 3: PROPUESTA DEL PROCEDIMIENTO

<b>Expertos</b>	1	2	3	4	5	6	7
<b>Kc</b>	0.8	1	0.9	0.7	0.5	0.7	0.7

El **coeficiente** de argumentación (**Ka**) se obtiene como resultado de la suma de los puntos alcanzados, a partir de la siguiente tabla patrón:

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

Esta tabla patrón se le presenta en la encuesta al experto sin cifras para que marque con una (x) el grado de influencia de las fuentes, de acuerdo a los niveles ALTO, MEDIO y BAJO.

Se muestran los resultados en la tabla:

<b>Expertos</b>	1	2	3	4	5	6	7
<b>Ka</b>	0.7	1	0.9	0.9	0.7	0.9	0.9

Finalmente el coeficiente de competencias para cada experto es:

<b>Expertos</b>	1	2	3	4	5	6	7
<b>K</b>	0.75	1	0.9	0.8	0.7	0.9	0.8
	<b>Media</b>	<b>Alta</b>	<b>Alta</b>	<b>Alta</b>	<b>Media</b>	<b>Alta</b>	<b>Alta</b>

Una vez calculado K, si:

$0.8 \leq K \leq 1$  K es alto, el Experto (i) tiene competencia alta.

$0.5 \leq K < 0.8$  K medio, el Experto (i) tiene competencia media.

$0 \leq K < 0.5$  K es bajo, el Experto (i) tiene competencia baja.

Se seleccionaron 7 personas como expertos para realizar la validación del procedimiento propuesto. La encuesta se realizó mediante una entrevista personal con cada uno de ellos.

Los expertos le confieren gran relevancia el aseguramiento de la calidad del software y consideran importante la utilización de un procedimiento para realizar esta actividad en el proyecto INSIGNE. Opinan que el uso del procedimiento propuesto aumentaría la calidad del proceso de desarrollo del software y con ello la de los productos que se obtienen en los flujos de trabajo Modelamiento del Negocio, Requerimiento y Análisis y Diseño.

Ellos coinciden con las etapas propuestas en el procedimiento y con la mayoría de las actividades a realizar en cada una de estas. Se detectó que no se realizan pruebas en los flujos de trabajo Modelamiento del Negocio, Requerimiento y Análisis y Diseño por lo que se considera no efectuar esta actividad en la etapa Desarrollar actividades de control de la calidad. Sugieren que pueden realizarse otras actividades de acuerdo con las características del proyecto INSIGNE.

Estos creen que pueden presentarse dificultades durante el uso del procedimiento propuesto debido a que no se siga como está definido, que no exista una asimilación de un trabajo con calidad por los miembros del proyecto, dificultades en la comprensión, que existan atrasos en el cronograma que impidan que las actividades propuestas puedan realizarse.

Como resultado final evalúan de muy bueno y bueno el procedimiento propuesto.

### **3.12. Conclusiones parciales.**

La propuesta de procedimiento para asegurar la calidad en los flujos de trabajo el Modelamiento de Negocio, Requerimiento, Análisis y Diseño evalúa los productos y procesos mediante estándares de calidad previamente establecidos. Se identifican y documentan los defectos encontrados durante la revisión y la auditoría. Además se informan a la dirección del proyecto los resultados de las actividades de aseguramiento de la calidad realizadas. Se efectúan acciones para resolver los problemas identificados.

Se realiza una validación por expertos al procedimiento propuesto para comprobar se puede aplicar al proyecto INSIGNE.

### CONCLUSIONES GENERALES

Al término de la investigación se plantean las siguientes conclusiones:

1. Se efectúa una extensa búsqueda bibliográfica y en Internet relacionada con el tema, lo que proporciona una serie de elementos teóricos importantes en el desarrollo del procedimiento para el aseguramiento de la calidad.
2. Es necesario preestablecer un conjunto de actividades a realizar por los involucrados en el proyecto que aseguren el cumplimiento de los requisitos establecidos por los clientes.
3. Se selecciona el modelo CMMI como referencia debido a que el mismo contiene un área de proceso de Aseguramiento de la calidad de procesos y producto, que apoya la entrega de productos y servicios de alta calidad, al proveer al personal y a los administradores un nivel apropiado de visibilidad sobre los procesos y productos de trabajos asociados en todo el ciclo de vida del proyecto.
4. El análisis realizado de la tendencia de los ERP en Cuba muestra la necesidad de implementar un sistema que gestione los procesos en las entidades presupuestadas y empresariales a escala nacional utilizando plataformas confiables y eficientes.
5. El análisis de la encuesta realizada por la Dirección de Calidad de Software en la UCI muestra la necesidad de establecer un procedimiento para asegurar la calidad del proyecto INSIGNE en los flujos de trabajo Modelamiento del Negocio, Requerimientos y Análisis y Diseño.
6. Se diseña un procedimiento para el aseguramiento de la calidad del proyecto INSIGNE a partir de la experiencia nacional e internacional, el cual se compone de 8 etapas con diversas actividades. En cada etapa se utilizan una serie de herramientas y conceptos que le confieren una alta flexibilidad al procedimiento.
7. El procedimiento propuesto evalúa los productos y procesos. Se identifican, documentan y efectúan acciones correctivas de los defectos encontrados durante la realización de las diferentes actividades de aseguramiento de calidad, además de guiar el trabajo a realizar por el grupo de aseguramiento de la calidad.
8. Se seleccionan 7 expertos que evalúan la propuesta del procedimiento a través de su revisión y la aplicación de un cuestionario. Cinco de los expertos consultados poseen competencia alta y dos competencias media.
9. Los expertos consultados coinciden en:
  - El aseguramiento de la calidad del software tiene gran relevancia.
  - Es importante la utilización de un procedimiento para asegurar la calidad del software en el proyecto INSIGNE.

- El uso del procedimiento propuesto elevará la calidad del proceso de desarrollo del software y con ello la de los productos que se obtienen en los flujos de trabajo Modelamiento del Negocio, Requerimiento y Análisis y Diseño.
- Coinciden con las etapas propuestas en el procedimiento.
- Evalúan de muy bueno y bueno el procedimiento propuesto para el aseguramiento de la calidad.

## RECOMENDACIONES

Después del desarrollo de este trabajo de diploma se recomienda:

- Llevar a cabo un procedimiento que incluya los restantes flujos de trabajo.
- Dar seguimiento a las actividades de aseguramiento de la calidad propuestas.
- Analizar y desarrollar las métricas definidas, así como también analizar y estudiar otras métricas que sean de mejor entendimiento.
- Realizar las revisiones al proceso de solicitud de cambios con el fin de controlar y coordinar lo que ocurre el proyecto.
- Probar el procedimiento propuesto en el proyecto INSIGNE y evaluar los resultados obtenidos en el proceso de ejecución.
- Investigar sobre herramientas que faciliten el proceso de aseguramiento de la calidad propuesto.

REFERENCIAS BIBLIOGRAFICAS

- AGUILAR, V. H. *Conf. # 5 "Pruebas". Ingeniería de Software II* Ciudad de la Habana: [Consultado el: 15 mayo de 2008]. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=22393>.
- ANTONIO ANGÉLICA, *Gestión de Configuración*, Chile, SPIN, 2001
- BANESPYME, C. Y. L. E. *Guía de Sistemas de Planificación de Recursos Empresariales (ERP)* [Consultado el: 10 enero de 2008]. Disponible en: <http://www.banespyme.org/imagesWeb/Publicaciones/10/guiasbanesto-3.pdf>.
- BATISTA, M. L. R. *Propuesta de Procedimiento para el Aseguramiento de la Calidad del Software en los proyectos productivos de la Facultad 7*. . Tutor: González, I. Y. T. Tesis de Grado, Universidad de las Ciencias Informáticas 2007.
- BUADES, G. "Calidad en Ingeniería del Software" [Presentación Power Point]. [Consultado el: febrero de 2007]. Disponible en: <http://dmi.uib.es/~bbuades/calidad/index.htm>.
- CARRASCO, O. M. F.; LEÓN, D. G., et al. Un enfoque actual sobre la calidad del software. . *Revistas de la Editorial Ciencias Médicas*, 1995, vol. 3, nº [Consultado el: 9 enero 2008]. Disponible en: [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm). ISSN 1024-9435.
- CMMI, P. T. *CMMI-DEV, V1.2 Improving processes for better products* United States of America: [Consultado el: 20 octubre de 2007]. Disponible en: [http://calidadsoft.prod.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com\\_content&Itemid=24](http://calidadsoft.prod.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com_content&Itemid=24).
- CONTENIDOS, G. D. *Vocabulario a conocer: UNE 166000*. publicado el: 2008 de 2006, última actualización: 2008. 2 p. Disponible en: [http://www.innovacion.com.es/index2.php?option=com\\_content&do\\_pdf=1&id=115](http://www.innovacion.com.es/index2.php?option=com_content&do_pdf=1&id=115).
- CONWAY. *Gestión de la Calidad. Conceptos y Filosofía* Disponible en: <http://www.scribd.com/doc/2628724/GESTION-DE-LA-CALIDAD-CONCEPTOS-Y-FILOSOFIAS>.
- CROSBY, P. B. *Quality Is Free: The Art of Making Quality Certain: How to Manage Quality - So That It Becomes A Source of Profit for Your Business (Hardcover)*. Primera Edición ed. New York: McGraw-Hill, 1979. 309 p. ISBN 978-0070145122
- DAPENA, M. D. D. *Calidad de los proyectos de software:Revisiones utilizando razonamiento basado en casos*. Ciudad de La Habana: publicado el: 2007 de 2003, última actualización: 2007. [Consultado el: 17 diciembre]. vol. 024, 7 p. Disponible en: <http://bdigital.eafit.edu.co:8080/bdng/query/single.xsp?id1=EAFITHIP0070240002200302>.

- DEFINICION.ORG. *Definición de procedimiento* [Consultado el: 20 mayo de 2008]. Disponible en:  
<http://www.definicion.org/procedimiento>.
- ELECTRÓNICO, P. D. D. D. N. *Sistemas de gestión integral (ERP)* [Consultado el: 13 marzo de 2008].  
Disponible en:  
[http://enegocio.equalmultiplica.net/eNegocio/opencms/enegocio/repositorio/topicos/topico\\_1166089427810.html?url=/enegocio/informacion-tematica/tematica\\_0002.html](http://enegocio.equalmultiplica.net/eNegocio/opencms/enegocio/repositorio/topicos/topico_1166089427810.html?url=/enegocio/informacion-tematica/tematica_0002.html).
- FAQ, E. D. T. *¿Qué es el ERP?* [Consultado el: 10 enero de 2008]. Disponible en: <http://www.tech-faq.com/lang/es/erp.shtml>.
- FEBLES, A.: *“Un modelo de referencia para la gestión de configuración en la pequeña y mediana empresa de software”*. Tesis presentada en opción del grado de Doctor en Ciencias Técnicas. Instituto Superior José Antonio Echeverría, 2004.
- FEIGENBAUM. *Gestión de la Calidad. Conceptos y Filosofía* Disponible en:  
<http://www.scribd.com/doc/2628724/GESTION-DE-LA-CALIDAD-CONCEPTOS-Y-FILOSOFIAS>.
- GARCÍA, M. *“Introducción a los conceptos de calidad”* [Página Web]. [Consultado el: octubre de 2007]. Artículo basado en las normas ISO. Disponible en:  
<http://www.mgar.net/soc/isointro.htm>.
- IEEE, *“ANSI/IEEE Std 10281988 IEEE standard for software reviews and audits,”* Jun.10 1989; <http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel1/2366/1231/00029123.pdf>.
- IEEE, *“IEEE Standard Glossary of Software Engineering Terminology,”* Sep. 1990;  
[http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsptp=&isnumber=4148&arnumber=159342&punumber=2238](http://ieeexplore.ieee.org/xpl/freeabs_all.jsptp=&isnumber=4148&arnumber=159342&punumber=2238)
- INFOR. *Infor ERP (Enterprise Resource Planning) > Gestión de Calidad. Mejor calidad para lograr mayor productividad.* [Consultado el: 12 febrero de 2008]. Disponible en:  
<http://latinamerica.infor.com/soluciones/erp/qualitymanagement/>.
- INFORMACIÓN, C. T. Y. D. N. N. C. D. T. D. L. *NC ISO/IEC 9126. INGENIERÍA DE SOFTWARE— CALIDAD DEL PRODUCTO— PARTE 1: MODELO DE LA CALIDAD* Ciudad de La Habana.: Cuban National Bureau of Standards, [Consultado el: 5 marzo de 2008]. Disponible en:  
<http://calidadsoft.prod.uci.cu/tmp/documentos/normas/iso/NC-ISO-IEC%209126-1.pdf>.
- INTERNA, I. D. A. *Auditoría interna* [Consultado el: 1 junio de 2008]. Disponible en:  
[http://www.fao.org/aud/es/internal\\_es.html](http://www.fao.org/aud/es/internal_es.html).
- Institute of Electrical and Electronics Engineers: *“Standard for Software Reviews”* IEEE Std. 1028-1997.

ISHIKAGUA. *Gestión de la Calidad. Conceptos y Filosofía* Disponible en:

<http://www.scribd.com/doc/2628724/GESTION-DE-LA-CALIDAD-CONCEPTOS-Y-FILOSOFIAS>.

ISO, "ISO 19011:2002 Auditorias de gestión de calidad y/o ambiental," Oct. 2002; [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=31169](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=31169).

ISO. *ISO 19011:2002 Directrices para la auditoría de los sistemas gestión de la calidad y/o ambiental* Disponible en: <http://calidadsoft.prod.uci.cu/tmp/documentos/normas/iso/NORMA%2019011%28Trad.Certif.%29.pdf>.

JACOBSON, I.; BOOCH, B., et al. *El Proceso Unificado de Desarrollo de Software*. Editado por: Educación.S.A., P. Madrid: 2000. ISBN 84-7829-036-2.

JÁUREGUI, A. J. *Administración de la Calidad. Bases teóricas y metodológicas del Modelo de Calidad Total*. de 2008]. Disponible en: <http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/acbtmmct.htm>.

JURAN, J.; GRZYNA, F.; *Quality planning and analysis: From Product development through use*, McGraw-Hill, 1970

MARTÍNEZ, A. G. y NOA, M. R. *Procedimiento para el Control y el Aseguramiento de la Calidad en el Flujo de Trabajo Análisis y Diseño de software en la UCI*. Tutor: Calvo., L. B. Tesis de Grado, Universidad de las Ciencias Informáticas, 2007.

MENDOZA, L. E. *Sistemas de información III. Teoría* Venezuela: [Consultado el: 10 diciembre de 2007]. Disponible en: [http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDa\)/PS6117%20Calidad%20del%20Software.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDa)/PS6117%20Calidad%20del%20Software.pdf).

M., L. E. M. *Sistemas de información III. Teoría* [Consultado el: 5 enero de 2008]. Disponible en: [http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDa\)/PS6117%20Calidad%20del%20Software.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDa)/PS6117%20Calidad%20del%20Software.pdf).

NAPAL. IRINA. "Las métricas de software en la actualidad. Su necesidad de aplicación.," Oct. 2007; III Conferencia Científica de la Universidad de las Ciencias Informáticas.

OHLSSON, D. y OLLFORS, M. *ERP – More than just Ones and Zeros- Investigating the Costs and Benefits of Enterprise Resource Planning Systems* -. Masters Thesis, Graduate Business School. Göteborg University, 2001.

PEREIRO, J. *Correcciones y acciones correctivas, acciones diferentes que se confunden*. 2005, 1 p. Disponible en: <http://www.portalcalidad.com/article.php?storyid=51>

- PRESSMAN, R. S. *Ingeniería del Software. Un enfoque práctico.* . Quinta Edición. ed. Madrid: McGraw-Hill, 2002. vol. 1, 601 p. ISBN 84-481-3214-9.
- PYMETICA. *DEFINICIÓN DE LOS SISTEMAS ERP (Enterprise Resource Planning o Sistemas de Planificación de Recursos Empresariales).* [Consultado el: 25 enero de 2008]. Disponible en: <http://www.tecnologiaparapymes.com/contenido/sysdow/descargas/DEFINICI%C3%93N%20DE%20LOS%20SISTEMAS%20ERP%202007.pdf>.
- RODRÍGUEZ, V. G. *El impacto de un ERP en la empresa* Guadalajara: [Consultado el: 20 enero de 2008]. Disponible en: <http://secretosenred.com/articulos/2982/1/EL-IMPACTO-DE-UN-ERP-EN-LA-EMPRESA/Paacutegina1.html>.
- RUP/EASY. *Guía Metodológica de Desarrollo de Distemas* [Consultado el: 16 diciembre de 2007]. Disponible en: [http://www.solucionesracionales.com/archivos/rupe\\_guia.pdf](http://www.solucionesracionales.com/archivos/rupe_guia.pdf).
- SCALONE, L. F. *Estudio comparativo de los modelos y estandares de calidad del software.* Tutor: Martinez, D. R. G. Tesis de Maestria, UNIVERSIDAD TECNOLOGICA NACIONAL, 2006.
- SOFTWARE., D. D. I. D. *Introducción a la Ingeniería de Software.* La Habana: [Consultado el: 10 diciembre de 2007]. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
- SOFTWARE., D. D. I. D. *Introducción a la Ingeniería de Software.* La Habana: [Consultado el: 10 diciembre de 2007]. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
- SQUAC. *Glosario de calidad del software* [Consultado el: 15 mayo de 2008]. Disponible en: [http://squac.iti.upv.es/glosario\\_calidad/](http://squac.iti.upv.es/glosario_calidad/).
- TERUEL, K. P. *Principales Roles para Equipos de Desarrollo de Software v 0.3.* 2007, 13 p. Disponible en: [http://calidadsoft.prod.uci.cu/index.php?view=article&catid=10%3Aart&id=19%3Aroles-equipo-proyecto&option=com\\_content&Itemid=24](http://calidadsoft.prod.uci.cu/index.php?view=article&catid=10%3Aart&id=19%3Aroles-equipo-proyecto&option=com_content&Itemid=24)
- TOIRAN, Y., *Organización en Procesos del Departamento de Detención a Estudiantes Extranjeros.*2006, Universidad Central de las Villas: Villa Clara.

---

---

## Bibliografía

- AGUILAR, V. H. *Conf. # 5 "Pruebas". Ingeniería de Software II* Ciudad de la Habana: [Consultado el: 15 mayo de 2008]. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=22393>.
- BANESPYME, C. Y. L. E. *Guía de Sistemas de Planificación de Recursos Empresariales(ERP)* [Consultado el: 10 enero de 2008]. Disponible en:  
<http://www.banespyme.org/imagesWeb/Publicaciones/10/guiasbanesto-3.pdf>.
- BATISTA, M. L. R. *Propuesta de Procedimiento para el Aseguramiento de la Calidad del Software en los proyectos productivos de la Facultad 7.* . Tutor: González, I. Y. T. Tesis de Grado, Universidad de las Ciencias Informaticas 2007.
- BRITO, H. R. G. *ERP cubano, un paso estratégico para la consolidación del Software Libre en Cuba.* Ciudad de La Habana: [Consultado el: 15 octubre de 2007]. Disponible en:  
[http://www.informaticahabana.com/evento\\_virtual/files/SWL11.pdf](http://www.informaticahabana.com/evento_virtual/files/SWL11.pdf).
- CARRASCO, O. M. F.; LEÓN, D. G., *et al.* Un enfoque actual sobre la calidad del software. . *Revistas de la Editorial Ciencias Médicas*, 1995, vol. 3, nº [Consultado el: 9 enero 2008]. Disponible en:  
[http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm). ISSN 1024-9435.
- CMMI, P. T. *CMMI-DEV, V1.2 Improving processes for better products* United States of America: [Consultado el: 20 octubre de 2007]. Disponible en:  
[http://calidadsoft.prod.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com\\_content&Itemid=24](http://calidadsoft.prod.uci.cu/index.php?view=article&catid=12%3Anormas-y-estandares-internacionales&id=26%3Acmmi&option=com_content&Itemid=24).
- CONTENIDOS, G. D. *Vocabulario a conocer: UNE 166000.* publicado el: 2008 de 2006, última actualización: 2008. 2 p. Disponible en:  
[http://www.innovacion.com.es/index2.php?option=com\\_content&do\\_pdf=1&id=115](http://www.innovacion.com.es/index2.php?option=com_content&do_pdf=1&id=115).
- CONWAY. *Gestión de la Calidad. Conceptos y Filosofía* Disponible en:  
<http://www.scribd.com/doc/2628724/GESTION-DE-LA-CALIDAD-CONCEPTOS-Y-FILOSOFIAS>.
- CROSBY, P. B. *Quality Is Free: The Art of Making Quality Certain: How to Manage Quality - So That It Becomes A Source of Profit for Your Business (Hardcover).* Primera Edicion ed. New York: McGraw-Hill, 1979. 309 p. ISBN 978-0070145122

- DAPENA, M. D. D. *Calidad de los proyectos de software: Revisiones utilizando razonamiento basado en casos*. Ciudad de La Habana: publicado el: 2007 de 2003, última actualización: 2007. [Consultado el: 17 diciembre]. vol. 024, 7 p. Disponible en: <http://bdigital.eafit.edu.co:8080/bdng/query/single.xsp?id1=EAFITHIP0070240002200302>.
- DEFINICION.ORG. *Definición de procedimiento* [Consultado el: 20 mayo de 2008]. Disponible en: <http://www.definicion.org/procedimiento>.
- ELECTRÓNICO, P. D. D. N. *Sistemas de gestión integral (ERP)* [Consultado el: 13 marzo de 2008]. Disponible en: [http://enegocio.equalmultiplica.net/eNegocio/opencms/enegocio/repositorio/topicos/topico\\_1166089427810.html?url=/enegocio/informacion-tematica/tematica\\_0002.html](http://enegocio.equalmultiplica.net/eNegocio/opencms/enegocio/repositorio/topicos/topico_1166089427810.html?url=/enegocio/informacion-tematica/tematica_0002.html).
- FAQ, E. D. T. *¿Qué es el ERP?* [Consultado el: 10 enero de 2008]. Disponible en: <http://www.tech-faq.com/lang/es/erp.shtml>.
- FEIGENBAUM. *Gestión de la Calidad. Conceptos y Filosofía* Disponible en: <http://www.scribd.com/doc/2628724/GESTION-DE-LA-CALIDAD-CONCEPTOS-Y-FILOSOFIAS>.
- INFOR. *Infor ERP (Enterprise Resource Planning) > Gestión de Calidad. Mejor calidad para lograr mayor productividad*. [Consultado el: 12 febrero de 2008]. Disponible en: <http://latinamerica.infor.com/soluciones/erp/qualitymanagement/>.
- INFORMACIÓN, C. T. Y. D. N. N. C. D. T. D. L. *NC ISO/IEC 9126. INGENIERÍA DE SOFTWARE— CALIDAD DEL PRODUCTO— PARTE 1: MODELO DE LA CALIDAD* Ciudad de La Habana.: Cuban National Bureau of Standards, [Consultado el: 5 marzo de 2008]. Disponible en: <http://calidadsoft.prod.uci.cu/tmp/documentos/normas/iso/NC-ISO-IEC%209126-1.pdf>.
- INTERNA, I. D. A. *Auditoría interna* [Consultado el: 1 junio de 2008]. Disponible en: [http://www.fao.org/aud/es/internal\\_es.html](http://www.fao.org/aud/es/internal_es.html).
- ISHIKAGUA. *Gestión de la Calidad. Conceptos y Filosofía* Disponible en: <http://www.scribd.com/doc/2628724/GESTION-DE-LA-CALIDAD-CONCEPTOS-Y-FILOSOFIAS>.
- ISO. *ISO 19011:2002 Directrices para la auditoría de los sistemas gestión de la calidad y/o ambiental* Disponible en: <http://calidadsoft.prod.uci.cu/tmp/documentos/normas/iso/NORMA%2019011%28Trad.Certif.%29.pdf>.

- JACOBSON, I.; BOOCH, B., *et al.* *El Proceso Unificado de Desarrollo de Software*. Editado por: Educación.S.A., P. Madrid: 2000. ISBN 84-7829-036-2.
- JÁUREGUI, A. J. *Administración de la Calidad. Bases teóricas y metodológicas del Modelo de Calidad Total*. de 2008]. Disponible en:  
<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/acbtmmct.htm>.
- JOSAR, C. *La contabilidad y el sistema contable* Maracaibo: [Consultado el: 15 de febrero de 2008].
- MARTÍNEZ, A. G. y NOA, M. R. *Procedimiento para el Control y el Aseguramiento de la Calidad en el Flujo de Trabajo Análisis y Diseño de software en la UCI*. . Tutor: Calvo., L. B. Tesis de Grado, Universidad de las Ciencias Informaticas, 2007.
- MENDOZA, L. E. *Sistemas de información III. Teoría* [Consultado el: 5 enero de 2008]. Disponible en:  
[http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDA\)/PS6117%20Calidad%20del%20Software.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDA)/PS6117%20Calidad%20del%20Software.pdf).
- OHLSSON, D. y OLLFORS, M. *ERP – More than just Ones and Zeros- Investigating the Costs and Benefits of Enterprise Resource Planning Systems* -. Masters Thesis, Graduate Business School. Göteborg University, 2001.
- PEREIRO, J. *Correcciones y acciones correctivas, acciones diferentes que se confunden*. 2005, 1 p. Disponible en: <http://www.portalcalidad.com/article.php?storyid=51>
- PRESSMAN, R. S. *Ingeniería del Softwar.Un enfoque práctico*. . Quinta Edición. ed. Madrid: McGraw-Hill, 2002. vol. 1, 601 p. ISBN 84-481-3214-9.
- PYMETICA. *DEFINICIÓN DE LOS SISTEMAS ERP (Enterprise Resource Planning o Sistemas de Planificación de Recursos Empresariales)*. [Consultado el: 25 enero de 2008]. Disponible en:  
<http://www.tecnologiaparapymes.com/contenido/sysdow/descargas/DEFINICI%C3%93N%20DE%20LOS%20SISTEMAS%20ERP%202007.pdf>.
- RODRÍGUEZ, V. G. *El impacto de un ERP en la empresa* Guadalajara: [Consultado el: 20 enero de 2008]. Disponible en: <http://secretosenred.com/articulos/2982/1/EL-IMPACTO-DE-UN-ERP-EN-LA-EMPRESA/Paacuteginal.html>.
- RUP/EASY. *Guía Metodológica de Desarrollo de Distemas* [Consultado el: 16 diciembre de 2007]. Disponible en: [http://www.solucionesracionales.com/archivos/rupe\\_guia.pdf](http://www.solucionesracionales.com/archivos/rupe_guia.pdf).

- SCALONE, L. F. *Estudio comparativo de los modelos y estandares de calidad del software*. Tutor: Martínez, D. R. G. Tesis de Maestria, UNIVERSIDAD TECNOLÓGICA NACIONAL, 2006.
- SOFTWARE, D. D. I. D. *Conf. # 6 “ Gestión de Configuración ”. Ingeniería de Software II* Ciudad de la Habana: [Consultado el: 15 mayo de 2008]. Disponible en: <http://teleformacion.uci.cu/course/view.php?id=43>.
- SOFTWARE., D. D. I. D. *Introducción a la Ingeniería de Software*. La Habana: [Consultado el: 10 diciembre de 2007]. Disponible en: <http://teleformacion.uci.cu/mod/resource/view.php?id=6655>.
- SQUAC. *Glosario de calidad del software* [Consultado el: 15 mayo de 2008]. Disponible en: [http://squac.iti.upv.es/glosario\\_calidad/](http://squac.iti.upv.es/glosario_calidad/).
- TERUEL, K. P. *Principales Roles para Equipos de Desarrollo de Software v 0.3*. 2007, 13 p. Disponible en: [http://calidadsoft.prod.uci.cu/index.php?view=article&catid=10%3Aart&id=19%3Aroles-equipo-proyecto&option=com\\_content&Itemid=24](http://calidadsoft.prod.uci.cu/index.php?view=article&catid=10%3Aart&id=19%3Aroles-equipo-proyecto&option=com_content&Itemid=24).

## ANEXOS

**Anexo 1.**Tabla1. Comparativa entre modelos CMMI y SPICE(15504).

	CMMI	ISO 15504
Ámbito de aplicación	Software y Sistemas	Software y Sistemas
En su favor	El de mayor prestigio	Más consensuado y probado
En su contra	Difícil de entender, mayor inversión, prescriptivo	Difícil en capacidad, complejo para evaluar
Procesos	Estructura propia	Delega en ISO 12207, por mayor aplicabilidad
Validación	Encuestas satisfacción y casos de estudio	'Trials' y esfuerzo empírico
Objetivo	Mejora del proceso, determinación capacidad contratista	Valoración del proceso y guía para la mejora.
Representación	Continua y por etapas	Continua (por etapas a nivel de proceso)
Técnicas análisis	Cuestionarios de evaluación	Varios
Método para mejora de procesos	IDEAL, mapa guiado	SPICE 4ª Parte

**Anexo2.** Tabla 2. Flujo de Trabajo Modelamiento del Negocio.

Trabajador	Actividades	Artefactos
Analista del Proceso de Negocio.	<p>Ajustar los Objetivos.</p> <p>Buscar Actores del Negocio y Casos de Uso.</p> <p>Capturar un Vocabulario Común del Negocio.</p> <p>Definir la Arquitectura del Negocio</p> <p>Evaluar los Objetivos de la Organización.</p> <p>Identificar las Metas del Negocio.</p> <p>Mantener las Reglas del Negocio.</p> <p>Modelar la Estructura de CUN.</p>	<p>Documento de Arquitectura del Negocio</p> <p>Especificaciones Suplementarias del Negocio.</p> <p>Glosario de Términos</p> <p>Metas del Negocio</p> <p>Modelo de Análisis del Negocio (Diagrama de Objetos o Clases)</p> <p>Modelo de CUN</p> <p>Reglas del Negocio</p> <p>Valoración del Objetivo de la Organización</p> <p>Visión del Negocio</p>
Diseñador del Negocio	<p>Definir Requerimientos Automatizables.</p> <p>Detallar las Entidades del Negocio.</p> <p>Detallar los CUN.</p> <p>Detallar los Trabajadores del Negocio.</p> <p>Encontrar Trabajadores y Entidades del Negocio.</p>	<p>Actores del Negocio.</p> <p>Entidades del Negocio.</p> <p>Eventos del Negocio.</p> <p>Realización de los CUN.</p> <p>Sistema del Negocio.</p> <p>Trabajadores del Negocio.</p>
Revisor Técnico	<p>Revisar el Modelo de Análisis</p> <p>Revisar el Modelo de CUN</p>	<p>Registro de Revisión</p>

**Anexo3.** Tabla 3. Flujo de Trabajo Requerimientos.

Trabajador	Actividades	Artefactos
Analista del Sistema	Administrar Dependencias. Capturar el Vocabulario Común. Desarrollador el Plan de Administración de Requerimientos. Encontrar Actores y CU. Estructurar el Modelo de CU. Respuesta de los Requisitos de los Stakeholder (Obtener Requisitos). Visión de Desarrollo (Visión).	Atributos de los Requerimientos. Documento Visión (Sistema). Especificaciones suplementarias. Glosario de Términos. Modelo de Casos de Uso (Sistema). Plan de Administración de Requerimientos. Requisitos de los Stakeholders.
Arquitecto de Software	Priorizar los CU	Documento de Arquitectura. Requerimientos del Software.
Especificador de Requerimientos	Detallar los Casos de Uso. Detallar los Requerimientos del Software.	Casos de Uso (Sistema) Especificaciones de los Requerimientos del Software Paquete de Casos de Uso Requerimientos del Software
Revisor Técnico	Revisar los Requerimientos.	Registro de Revisión

**Anexo 4.** Tabla 4. Flujo de Trabajo Análisis y Diseño.

Trabajador	Actividades	Artefactos
Arquitecto de Software	<p>Análisis de la Arquitectura</p> <p>Construir un esbozo de la Arquitectura.</p> <p>Describir la Arquitectura en tiempo de ejecución.</p> <p>Describir la Distribución.</p> <p>Evaluar la Vialidad de la Arquitectura (esbozo)</p> <p>Identificar Elementos de Diseño</p> <p>Identificar Mecanismos de Diseño</p> <p>Incorporar Elementos de Diseño Existentes.</p>	<p>Modelo de Análisis</p> <p>Modelo de Despliegue</p> <p>Modelo de Diseño</p> <p>Documento de Arquitectura de Software</p> <p>Arquitectura de Referencia</p> <p>Interfaces</p> <p>Señal</p> <p>Eventos</p> <p>Protocolos</p> <p>Arquitectura (Esbozo)</p>
Diseñador	<p>Caso de Uso de Análisis</p> <p>Caso de Uso de Diseño</p> <p>Clases de Diseño</p> <p>Diseño de Subsistema</p> <p>Diseño de elementos de Pruebas</p>	<p>Clases de Análisis</p> <p>Clases de Diseño</p> <p>Clases de Pruebas</p> <p>Paquete de Diseño</p> <p>Realización de Casos de Uso Subsistema de Diseño</p>
Diseñador de Cápsula	Diseñar Cápsula.	Cápsula.
Diseñador de Base de Datos	Diseño de la BD	Modelo de Datos
Diseñador de Interfaz de Usuario	<p>Diseño de Interfaz de Usuario</p> <p>Prototipo de Interfaz de Usuario</p>	<p>Mapa de Navegación</p> <p>Prototipo de Interfaz de Usuario</p>
Diseñador de Prueba	<p>Define los métodos de prueba.</p> <p>Identifica los mecanismos de prueba.</p> <p>Define los elementos de prueba.</p> <p>Define el ambiente de configuración de prueba.</p> <p>Estructura la implementación de</p>	Diseño de prueba

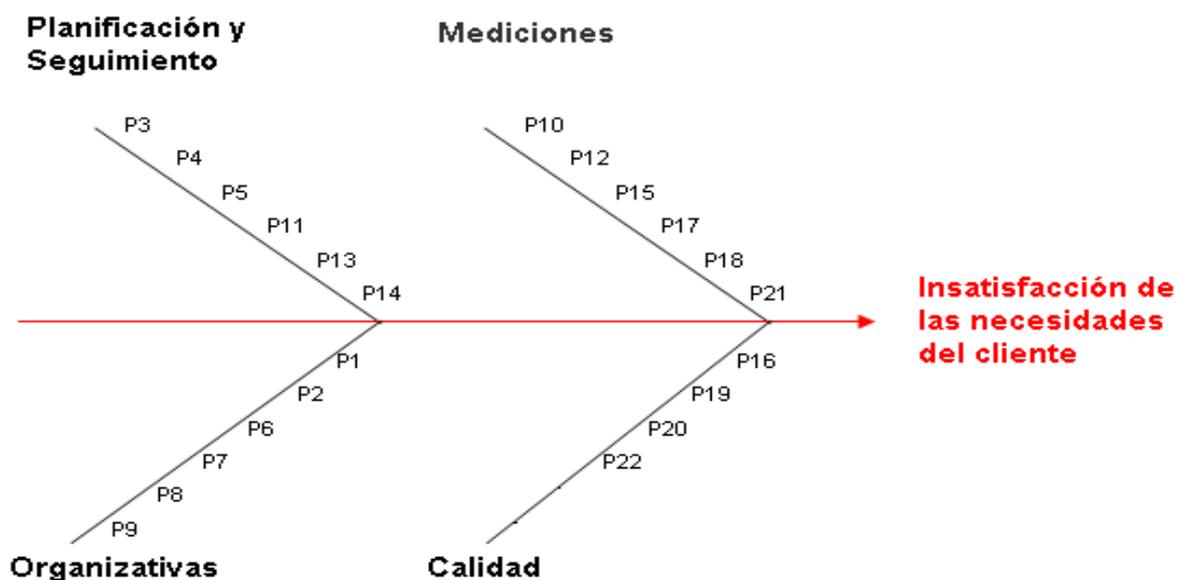
	prueba.	
Revisor Técnico	Revisión de la Arquitectura. Revisión del Diseño.	Registro de Revisión

**Anexo 5.** Tabla 5. Roles Encuestados.

Directivos	Roles Principales	Desarrolladores
Presidente de la FEU	Analista principal	Analistas
Jefe del departamento de ingeniería de software	Arquitecto principal	Planificador
Vicedecano de residencia	Jefe del grupo de desarrollo	Investigador
Jefe del departamento de Ciencias sociales.	Líder del proyecto	Implementador
Decano	Planificador	Arquitecto
Jefe del Departamento Ciencias Básicas		Probador
Jefe del Departamento Técnicas de Programación		Documentador
Secretario General Sindicato Vicedecano de Producción		
Asesor de Calidad		
Jefe del Departamento Sistemas Digitales		
Vicedecano de Formación		
Secretario Comité Primario de la UJC		
Secretario Docente		
Asesor de Calidad		

3 Jefes del polo

### Anexo 6. Espina de Pescado.



#### Leyenda:

- P1: Existe falta de comunicación efectiva entre los involucrados (usuarios, desarrolladores, administradores, clientes e investigadores).
- P2: No se puede comparar con proyectos anteriores y aprender de estos
- P3: No se realizan las reuniones necesarias para hacer una buena planificación del trabajo.
- P4: No existe un procedimiento para llevar a cabo las actividades relacionadas con el desarrollo del software.
- P5: No existe una adecuada planificación del trabajo.
- P6: No se brinda eficientemente el Soporte de Software.
- P7: No hay eventos predefinidos para hacer pedidos de cambios.
- P8: No existe una preparación requerida para desempeñar los roles que se asignan, la mayoría aprenden sobre la marcha.
- P9: No todas las personas que tienen un rol definido están interesados por el mismo.
- P10: No se conoce sobre las herramientas para el control de versiones.

- P11: No se puede dar seguimiento al progreso del proyecto (no hay definición de tareas, relaciones entre estas, cronogramas, puntos de chequeo, etc.)
- P12: No se lleva el registro de quien hizo cada actividad relacionada con el proyecto.
- P13: Casi nunca firma un acta de aceptación de la entrega de la documentación al cliente.
- P14: Los proyectos están excesivamente tarde y los beneficios que pudieran obtenerse al utilizar los mejores métodos e instrumentos en las distintas etapas no se detectan en este medio indisciplinado y caótico de desarrollo
- P15: La evaluación se limita a pruebas de caja negra en la fase final de los proyectos.
- P16: No existe el expediente de las pruebas realizadas.
- P17: No se exige responsabilidades, ni se puede medir productividad, cantidad de errores cometidos a nivel individual.
- P18: No se posee registros de mediciones relevantes de los proyectos de su facultad, que permitan realizar estimaciones para planificaciones futuras, análisis de avances o cualquier otro estudio a favor de mejorar los procesos de desarrollo.
- P19: No se pueden ver los problemas de calidad (cantidad de errores) en distintas áreas de trabajo.
- P20: No se pueden detectar con antelación los problemas en el software afectando esto finalmente la calidad en los productos.
- P21: No se miden los procesos, ni los productos con métricas rigurosas lo que implica que no se pueda controlar ni mejorar el proceso a partir de los resultados en su ejecución.
- P22: No se tiene conocimiento de los indicadores que permiten notar que alguno de los Proyectos está en una situación alarmante.

**Anexo 7.** Tabla 6 de Tareas y Responsabilidades.

<b>Tarea de Aseguramiento de calidad</b>	<b>Precondición Al finalizar la fase:</b>	<b>Poscondición Antes de la fase:</b>	<b>Responsable</b>	<b>Comentarios</b>
Se exponen las tareas que serán llevadas a cabo para asegurar la calidad en el proyecto en los flujos de trabajos especificados.			Nombre de responsable del cumplimiento de la tareas	Breve descripción de la tarea a desarrollar.

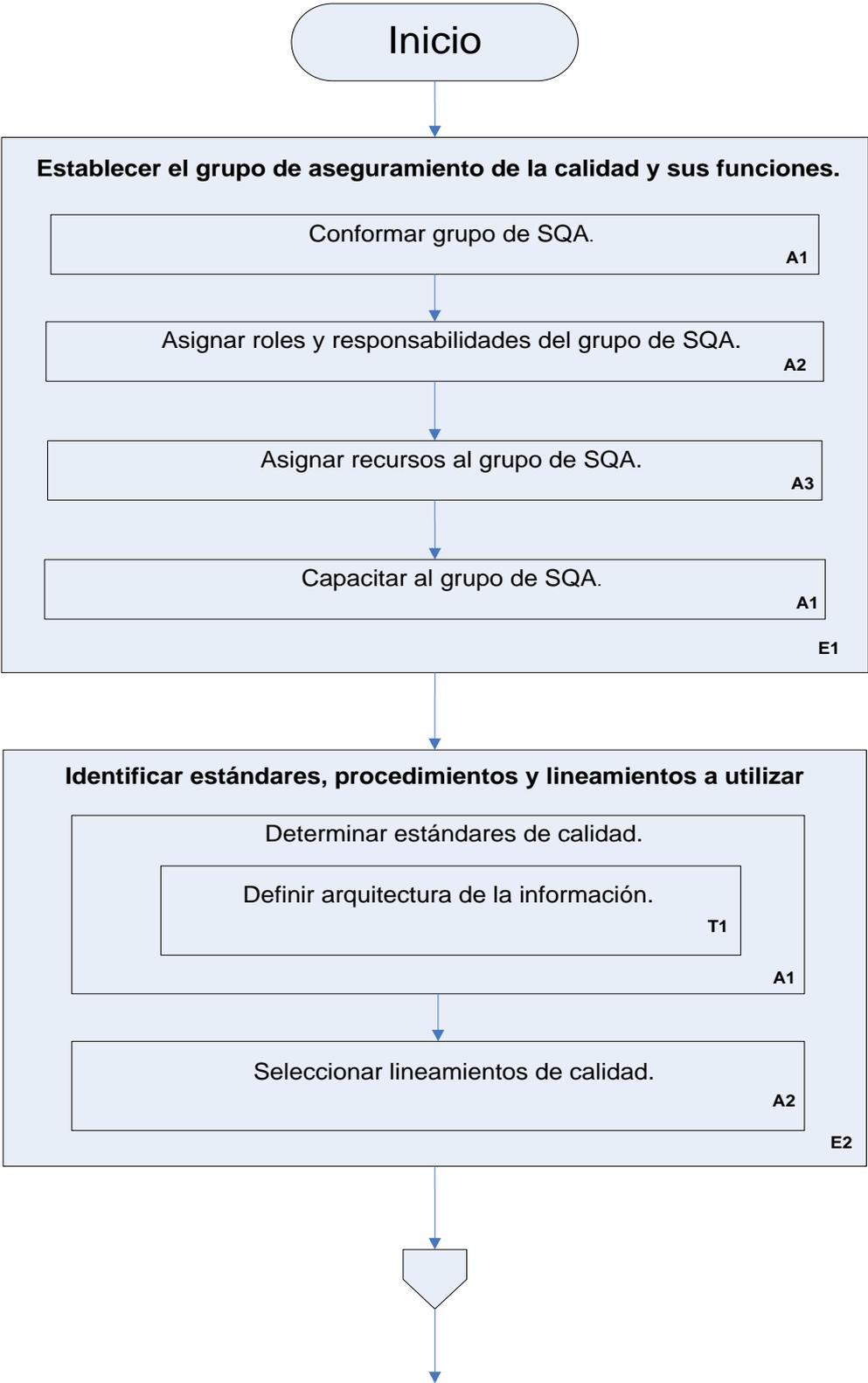
**Anexo 8.** Tabla 7 de Estándares y Guías.

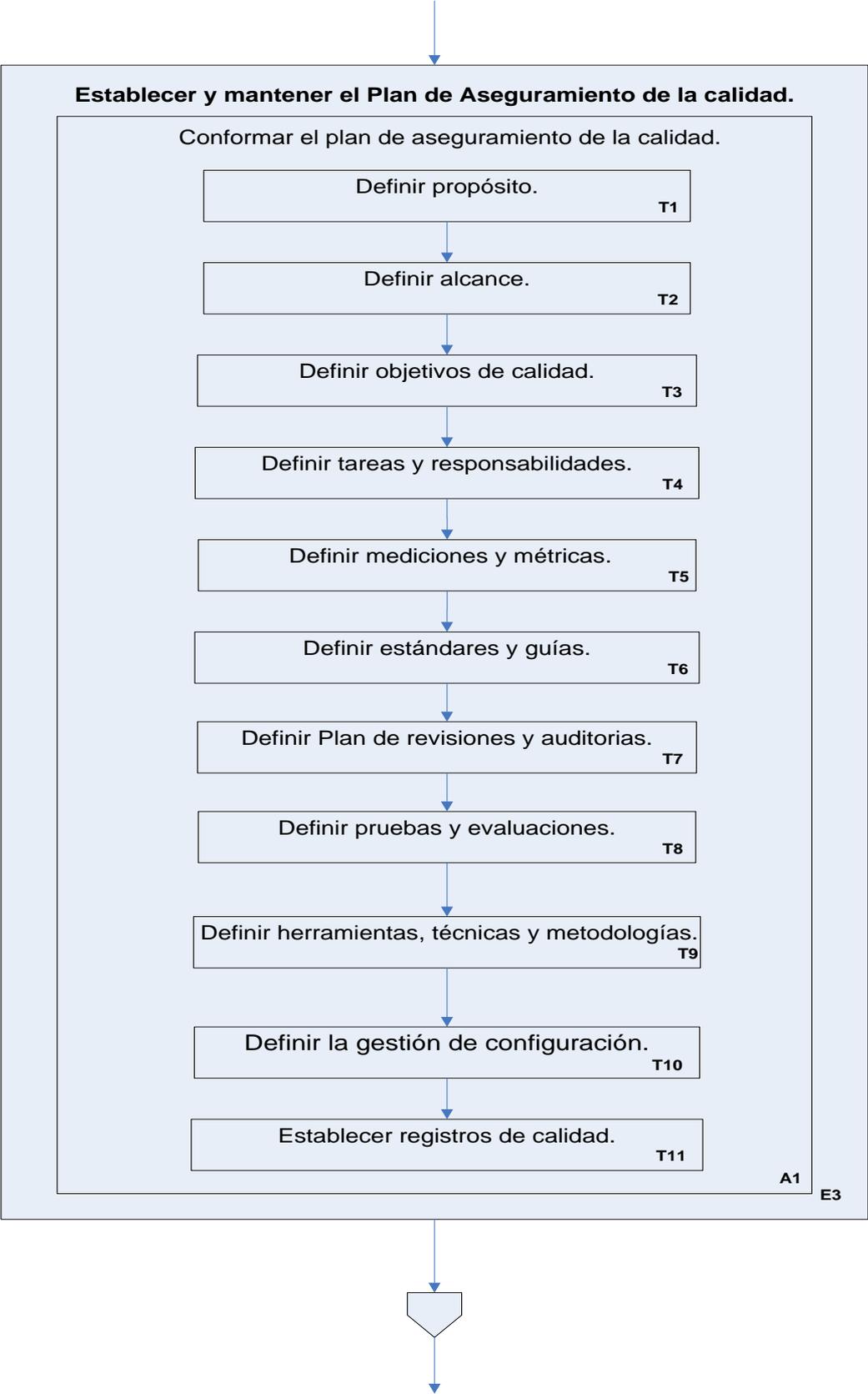
<b>Estándar y Guía</b>	<b>Ubicación</b>	<b>Comentarios</b>
Estándares por el cual se van a regir e grupo de aseguramiento de la calidad.	Lugar donde se almacena.	Comentarios acerca de los estándares a utilizar en el proyecto.

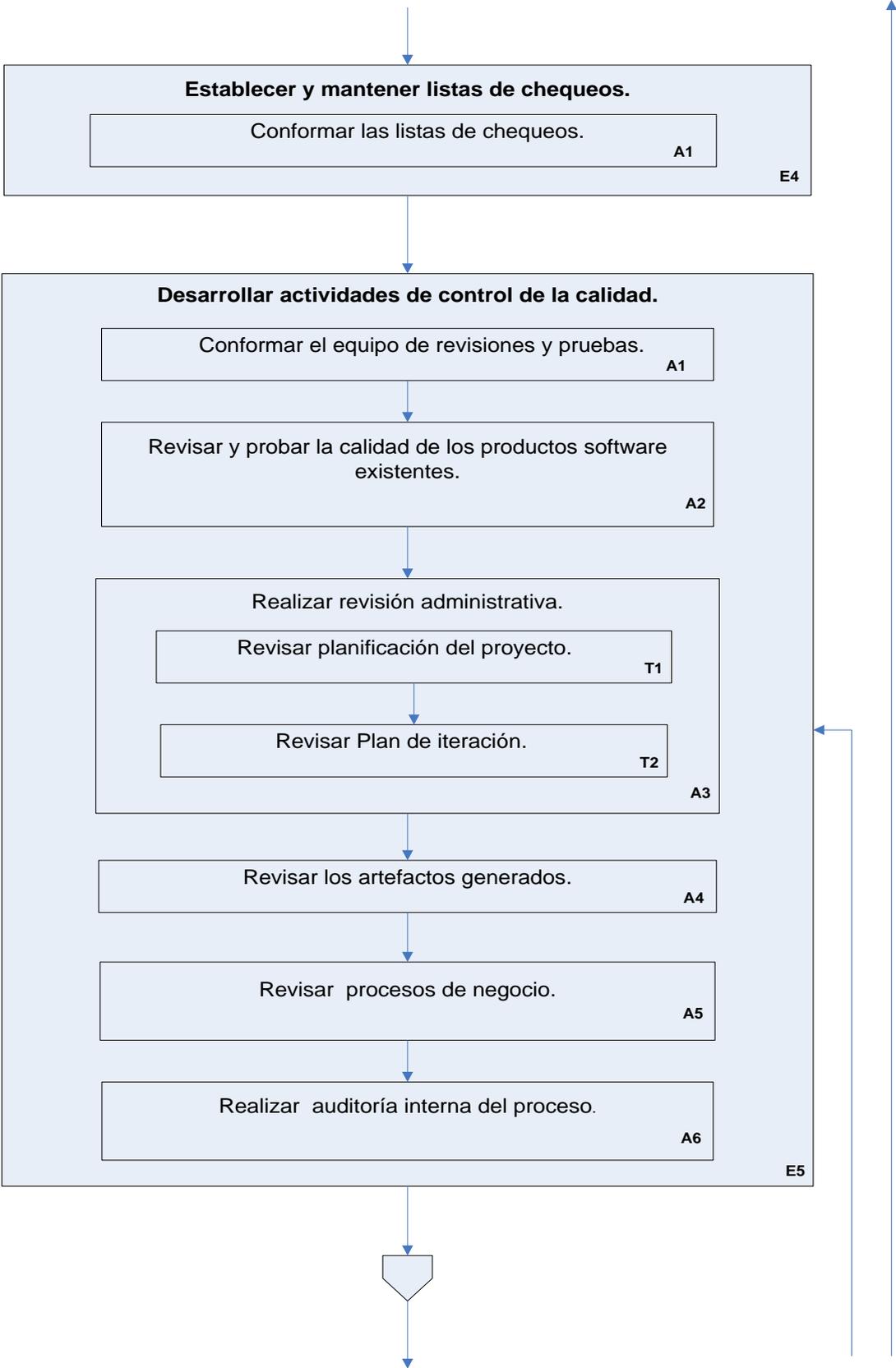
**Anexo 9.** Tabla 8 de Chequeo.

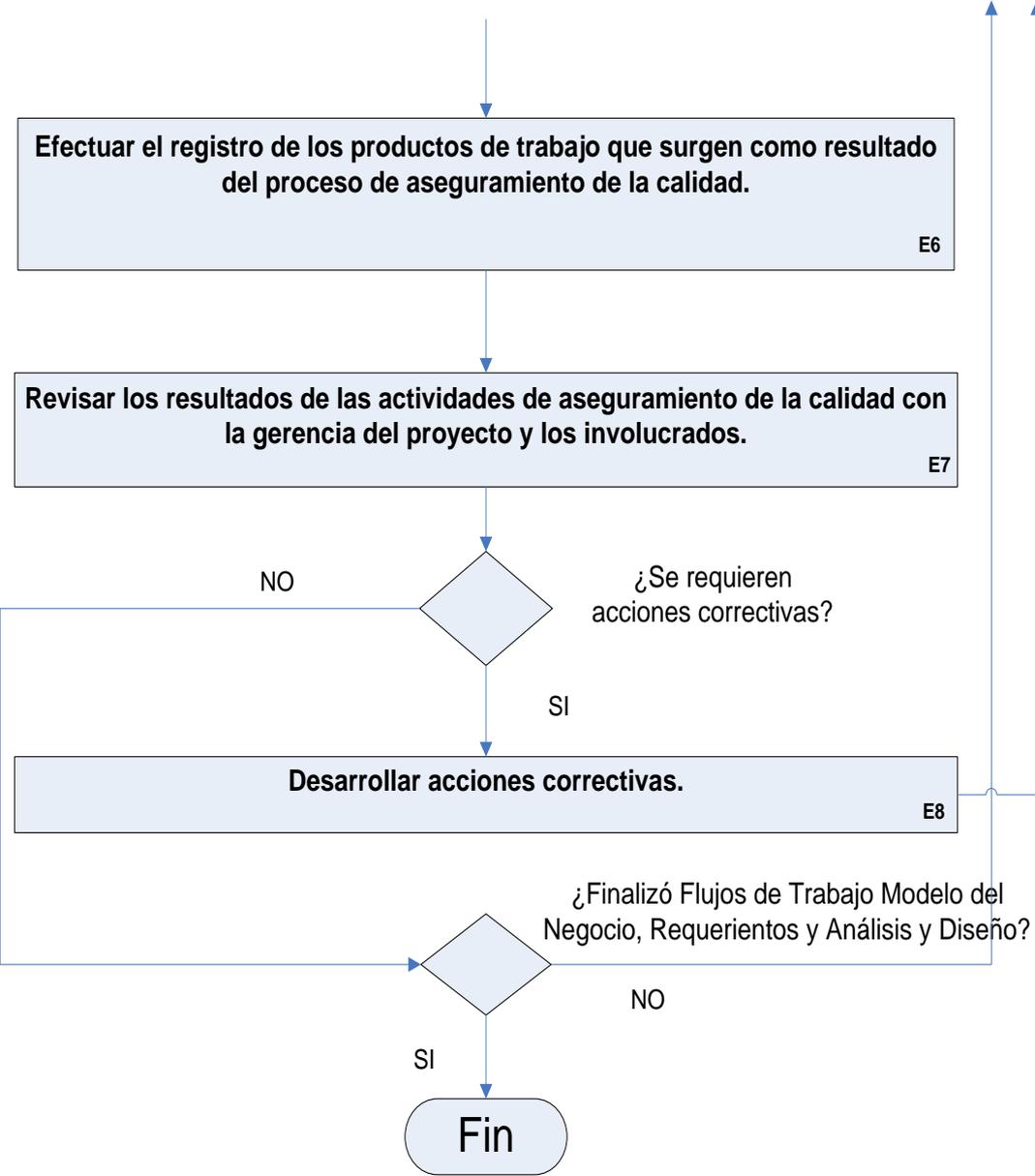
<b>Nivel</b>	<b>Criterio de evaluación</b>	<b>Evaluación</b>	<b>N.P</b>	<b>Observaciones</b>	<b>Métrica</b>	<b>Submétrica</b>	<b>%cumplimiento</b>

Anexo 10. Flujo del procedimiento.









**Anexo 11.** Encuesta para determinar el coeficiente de competencia de los expertos.

Compañero (a):

En la ejecución de la presente tesis, deseamos someter a la valoración de un grupo de expertos, la propuesta del procedimiento para el Aseguramiento de la Calidad en el Proyecto Solución para la Gestión de Entidades. Para ello necesitamos conocer el grado de dominio que UD. posee sobre el aseguramiento de la calidad durante el proceso de desarrollo del software; y con ese fin deseamos que responda lo que se le pide a continuación.

Nombre y apellidos: \_\_\_\_\_

Labor que realiza: \_\_\_\_\_

Años de experiencia: \_\_\_\_\_ Especialidad: \_\_\_\_\_

Categoría docente: \_\_\_\_\_ Categoría científica: \_\_\_\_\_

1.- Marque con una cruz (X) el grado de conocimiento que UD. tiene sobre la temática que se investiga:

0	1	2	3	4	5	6	7	8	9	10

2.- Marque con una cruz (X) las fuentes que le han servido para argumentar el conocimiento que tiene Ud. de la temática que se investiga. Encierre en un círculo la que más ha influido.

No.	Fuentes de argumentación	Grado de influencia		
		Alto	Medio	Bajo
1.-	Análisis realizado por Ud.			
2.-	Experiencia.			
3.-	Trabajos de autores nacionales.			
4.-	Trabajos de autores extranjeros.			
5.-	Su propio conocimiento del tema.			
6.-	Su intuición.			

**Anexo 12.** Cuestionario para Expertos.

Usted ha sido seleccionado por su conocimiento en calidad de software, como experto para evaluar los resultados de la propuesta. Por lo cual se le pide llenar este cuestionario donde los datos obtenidos contribuirán al resultado exitoso de esta tesis.

1. ¿Qué relevancia le confiere al aseguramiento de la calidad del software?

Alta  Muy alta  Media  Baja  Muy baja

2. ¿Considera importante la utilización del procedimiento definido para asegurar la calidad en el Proyecto Solución para la Gestión de Entidades?

Si  No  No se.

3. ¿Considera usted que aumentaría la calidad del proceso de desarrollo del proyecto durante los flujos de trabajo Modelación del negocio, Requerimientos y Análisis y Diseño si se sigue el procedimiento definido?

Si  No  No se.

4. ¿Considera usted que con el desarrollo de las etapas definidas en el procedimiento se complementa el aseguramiento de la calidad?

Si  No  No todas.

4.1 En caso que su respuesta sea "No todas", marque con una cruz cuales no.

<b>Etapas</b>	
Establecer el Grupo de Aseguramiento de la calidad y sus funciones.	
Identificar estándares, procedimientos y lineamientos a utilizar.	
Establecer y mantener el Plan de Aseguramiento de la calidad.	
Establecer y mantener listas de chequeos.	
Desarrollar actividades de control de la calidad.	
Efectuar registro de los productos de trabajo resultados del proceso de Aseguramiento de la calidad.	
Revisar los resultados de las actividades de aseguramiento de la calidad con la gerencia del proyecto y los involucrados.	
Desarrollar acciones correctivas.	

5. ¿Considera usted que es posible cumplir con las actividades propuestas en el procedimiento definido de aseguramiento de la calidad para el proyecto Solución para la Gestión de Entidades?

Si  No  No todas.

5.1. En caso que su respuesta sea “No todas”, marque con una cruz cuales no.

<b>Actividades</b>	
Conformar Grupo de SQA.	
Asignar Roles y Responsabilidades del grupo de SQA.	
Asignar recursos al grupo de SQA.	
Capacitar al Equipo Aseguramiento de la calidad.	
Determinar estándares de calidad.	
Seleccionar lineamientos de calidad.	
Conformar el Plan de aseguramiento de la calidad.	
Conformar las listas de chequeos.	
Conformar el Equipo de Revisiones y Pruebas.	
Revisar y Probar la calidad de los Productos Software existentes.	
Realizar revisión Administrativa.	
Revisar los Artefactos.	
Revisar Proceso.	
Realizar Pruebas.	
Realizar Auditoría Interna del Producto y el Proceso.	

6. ¿Cree necesario realizar otras actividades para asegurar la calidad dentro del proyecto Solución para la Gestión de Entidades durante los flujos de trabajo Modelación del negocio, Requerimientos y Análisis y Diseño?

Si  No  No se.

6.1. En caso que su respuesta sea “Si”, mencione cuales.

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

7. ¿Cómo evalúa el procedimiento definido?

Muy bueno  Bueno  Regular  Malo  Muy Malo.

8. ¿Cree que se pueden presentar dificultades en el proyecto Solución para la Gestión de Entidades de la UCI durante la utilización del procedimiento?

Si  No.

8.1. En caso que su respuesta sea "Sí", mencione cuales.

---

---

---

Para finalizar, queremos expresarle que sus criterios y opiniones se manejarán de forma anónima, además le agradecemos por anticipado su valiosa colaboración y estamos seguros que sus sugerencias contribuirán a perfeccionar el proceso propuesto.

Muchas gracias por su cooperación y le pedimos disculpas por las molestias ocasionadas.

### GLOSARIO DE TÉRMINOS

**Control:** es el acto de delimitar responsabilidad y autoridad con el fin de liberar la gerencia de detalles innecesarios, conservando los medios para asegurarse de que los resultados sean satisfactorios.

**Triggers:** Son parecidos a los constraints, pero proveen mayor flexibilidad pues el usuario puede especificar un conjunto de acciones complejas a ejecutarse cuando el trigger se dispare. Los constraints actúan cuando se viola lo que ellos indican, pero el usuario no puede especificar acción alguna con esa violación, sino que el DBMS simplemente envía un mensaje de error e impide que se realice la operación que produjo la violación.

**Constraints:** Son especificaciones del DDL de SQL que se aplican a columnas o tuplas de una tabla. Algunos tipos especiales de constraints son: UNIQUE, NOT NULL, REFERENCES, CHECK; éste último permite especificar una amplia gama de reglas, como por ejemplo, rangos de valores y listas de valores, entre otros. Es conveniente que cada constraint tenga un nombre, pues cuando el constraint es violado el sistema indica su nombre y se puede saber exactamente qué falló.

**Aseguramiento:** Acción y efecto de asegurar.

**Stakeholders:** Personas u organizaciones que están activamente implicadas en el negocio ya sea porque participan en él o porque sus intereses se ven afectados con los resultados del proyecto. Pueden ser los propietarios, la dirección, quienes financian, los clientes, los trabajadores, los proveedores, la competencia, la comunidad local, etc.

**Artefacto de software:** Cualquier cosa que resulte del proceso de desarrollo de software; por ejemplo: documentos de requisitos, especificaciones, diseños, software, etc.

**Trials:** Ensayos. Pruebas (juicio).

**Iteración:** conjunto de periodos de tiempo dentro de un proyecto, en el cual usted produce una versión ejecutable del producto, estable, junto con cualquier otra documentación de soporte, instalación de scripts o similar, necesarios para usar esta liberación.

**Falibilidad:** riesgo a errar por una persona o entidad.

**Multimodular:** es una característica que expresa que algo esta compuesto por muchos módulos.

**Interoperar:** Posibilitar el intercambio eficaz de información entre sistemas, de forma que se optimice su gestión.

**Reingeniería:** es una herramienta gerencial mediante la cual se rediseñan radicalmente los procesos medulares de una empresa, para lograr mejoras dramáticas en productividad, tiempos de ciclo y calidad.

**Enterprise Resource Planner:** Sistemas de planificación de recursos empresariales.

**Customer Relationship Management:** Administración de la relación con los clientes. Referencia a una estrategia de negocio basada principalmente en la satisfacción de los clientes.

**Supply chain management:** Proceso de planificación, ejecución y control de las operaciones de la cadena de suministro tan eficientemente como sea posible.