

***Universidad de las Ciencias Informáticas***

***Facultad 7***



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

***Título: Propuesta de Procedimiento para el Aseguramiento de la  
Calidad del Software en los proyectos productivos de la Facultad 7.***

***Autor: María Lilia Rodríguez Batista***

***Tutor: Ing. Yohanler Toirán González***

***Ciudad de la Habana, junio de 2007***

## **Declaración de Autoría**

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

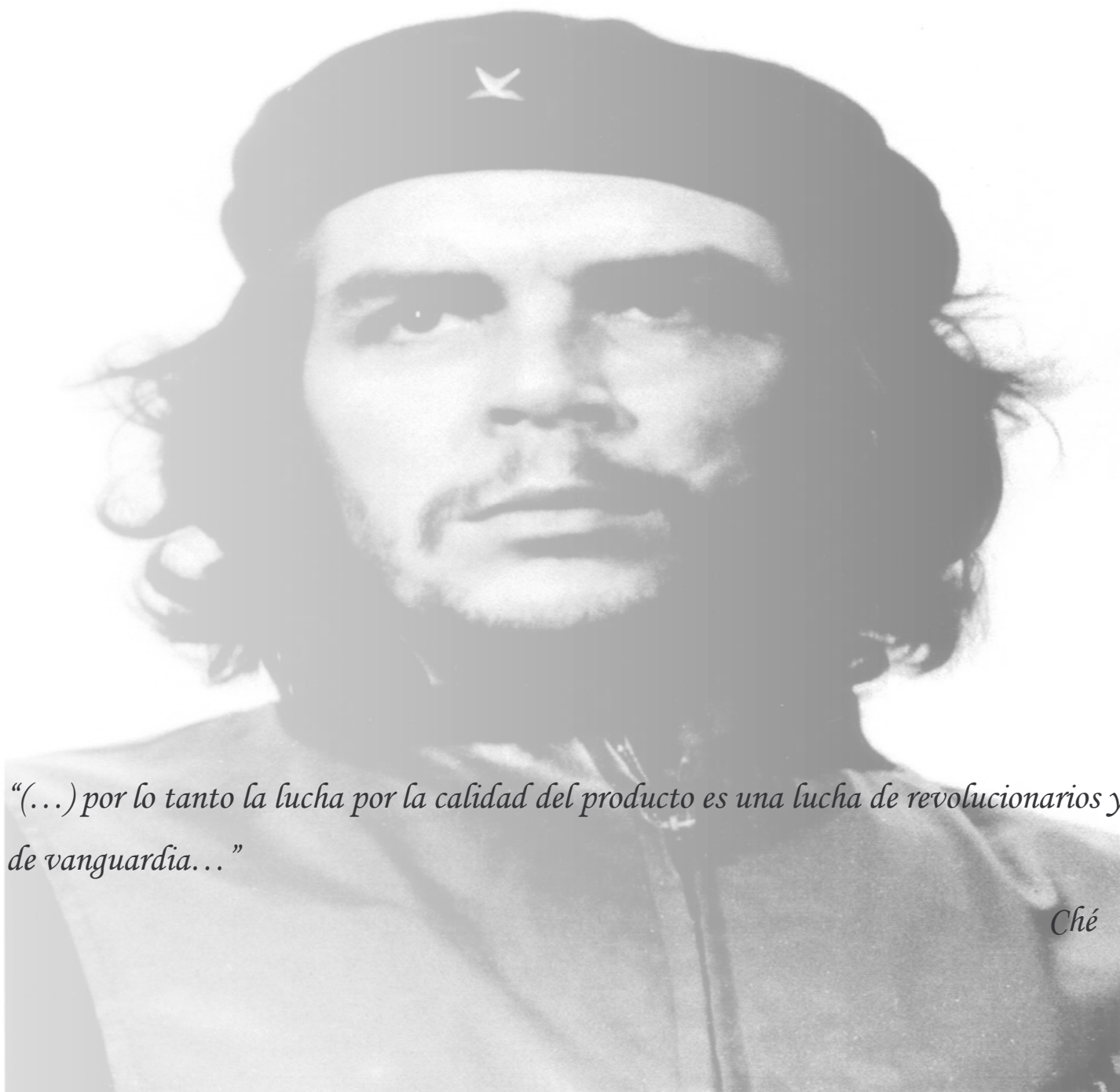
Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autor: María Lilia Rodríguez Batista

Tutor: Ing. Yohanler Toirán González

---

---



*“(...) por lo tanto la lucha por la calidad del producto es una lucha de revolucionarios y de vanguardia...”*

*Ché*

## *Agradecimientos*

*A la Revolución y a nuestro Comandante por facilitarme los medios para llegar hasta aquí.*

*A mis padres, tíos(as), primos(as), mis abuelitas queridas y a Elio por el apoyo incondicional que me han dado y porque han hecho posible que este sueño se haga realidad.*

*A los buenos vecinos que he tenido y en especial a Niurka y toda su familia por estar presentes cuando los he necesitado.*

*A mi mamita querida por estar siempre a mi lado, por guiarme por el camino correcto, por confiar en mí, por su entrega, su cariño, porque eres única y sin ti no sería nadie.*

*A la UCI y a mis profesores por guiarme durante estos cinco años.*

*A Yohanler por su dedicación y ayuda.*

*A la profesora Pura por su paciencia y apoyo.*

## *Dedicatoria*

*...A mi familia...*

*A mis abuelas que me han hecho grande.*

*A mi mamá que es la luz que me ilumina y la razón de mí existir.*

*A Patricia, Andy y a la pequeñita Bia que son los hermanitos que nunca tuve.*

*A todos mis compañeros y amigos en especial a Yaritza, Dulce, Yanelys, Yara, Lily, Yeni, Maylen, Dayanis, Roque, Yani y Jaquelin que me han acompañado durante estos cinco años; me han brindarme su amistad incondicional y porque las llevo presente en mi corazón.*

## **Resumen**

El presente trabajo se realizó en la Facultad 7 de la Universidad de Las Ciencias Informáticas, con el objetivo de elaborar un procedimiento para el Aseguramiento de la Calidad del Software (SQA). Este se basa en el Proceso Unificado de Desarrollo de Software (RUP). En el se explica mediante una secuencia lógica de pasos todas las actividades de SQA que deben realizarse en las cuatro fases de desarrollo que propone esta metodología.

En la investigación se recogen conceptos de distintos autores y durante la misma se realizó un diagnóstico acerca de los problemas existentes en la producción de software, para ello se emplearon diferentes técnicas como fue la revisión de documentos, entrevistas, encuestas entre otras, concluyendo que el aseguramiento de la calidad es una actividad de protección, que debe aplicarse a lo largo de todo el proceso de desarrollo del software.

Se proponen como principales actividades de SQA la planificación, supervisión, construcción y evaluación de la calidad. Se recomienda que los procesos de gestión de la configuración, gestión de riesgos y prueba; se realicen en todas las fases de desarrollo y se debe tener en cuenta que si el producto revisado en cada fase no cumple con la calidad requerida, no se puede pasar a la siguiente.

El trabajo tiene un valor teórico-práctico y metodológico pues se generaliza el concepto de calidad como una necesidad inmediata para la institución, proponiéndose diferentes aspectos que se deben cumplir una vez que se implemente la investigación durante la producción.

## Índice

<b>Introducción</b> .....	1
<b>Capítulo1: Fundamentación Teórica</b> .....	4
1.1 Introducción .....	4
1.2 La Industria de Software y su desarrollo mundial.....	4
1.3 Producción de Software en Cuba .....	6
1.3.1 Evolución de la calidad del software en Cuba.....	8
1.4 Calidad.....	9
1.4.1 Estándares y Modelos Internacionales para la calidad del software .....	10
1.4.2 Conceptos de calidad según destacadas figuras.....	12
1.4.3 Conceptos de calidad según los Modelos y Estándares de calidad.....	14
1.5 Gestión de la Calidad .....	15
1.5.1 Sistemas de gestión de la calidad .....	15
1.5.2 Principios de gestión de la calidad .....	16
1.5.3 Modelos integradores de la gestión de la calidad en la empresa.....	17
1.6 Aseguramiento de la Calidad del software .....	19
1.6.1 Equipo de Aseguramiento de la Calidad .....	21
1.7 Actividades de aseguramiento de la calidad del software.....	22
1.8 Revisiones del Software .....	24
1.8.1 Revisión Técnica Formal (RTF).....	24
1.8.2 Pruebas del Software.....	25
1.9 Conceptos de procesos.....	28
1.9.1 El Proceso de desarrollo de software.....	29
1.10 Conclusiones parciales .....	33
<b>Capítulo2: Caracterización y Diagnóstico a los proyectos productivos de la Facultad 7</b> .....	34
2.1 Introducción .....	34
2.2 Caracterización de la UCI .....	34
2.2.1 La producción de software en la UCI y en la Facultad 7.....	36
2.2.2 Evolución de la producción de software en la UCI.....	39

2.2.3 Necesidad de la producción.....	40
2.2.4 Desarrollo de software en los proyectos productivos.....	41
2.3 Producción de software en la Facultad 7 .....	43
2.4 Aplicación de Técnicas para el diagnóstico.....	45
2.5 Diagnóstico de la producción de software en la Facultad 7 .....	49
2.6 Conclusiones parciales .....	52
<b>Capítulo 3: Propuesta de Procedimiento para el Aseguramiento de la Calidad del Software en los Proyectos Productivos de la Facultad 7 .....</b>	<b>53</b>
3.1 Introducción .....	53
3.2 Propuesta de procedimiento de SQA .....	54
3.2.1 Propósito .....	57
3.2.2 Objetivos .....	57
3.2.3 Responsabilidades.....	57
3.2.4 Principales Actividades de SQA .....	60
3.2.4.1 Planificación de la Calidad .....	61
3.2.4.2 Construcción de la Calidad.....	74
3.2.4.3 Supervisión de la Calidad.....	75
3.2.4.4 Evaluación de la calidad del producto.....	80
3.2.5 Medidas .....	81
3.2.6 Documentos referenciados.....	81
3.3 Conclusiones parciales .....	82
<b>Conclusiones.....</b>	<b>82</b>
<b>Recomendaciones .....</b>	<b>83</b>
<b>Referencia Bibliográfica.....</b>	<b>84</b>
<b>Bibliografía .....</b>	<b>88</b>
<b>Anexos .....</b>	<b>91</b>
<b>Glosario de Términos.....</b>	<b>107</b>



## ***Índice de Figuras***

<b>Figura 1.1</b> Hilo conductor del marco teórico referencial.....	4
<b>Figura 1.2</b> Evolución histórica de la gestión de la calidad. ....	10
<b>Figura 1.3</b> Transformaciones mediante procesos. ....	28
<b>Figura 1.4</b> Capas de la Ingeniería de Software.....	31
<b>Figura 2.1</b> Hilo conductor del capítulo 2. ....	34
<b>Figura 2.2</b> Relación de las áreas de proyecto de las diferentes facultades. ....	36
<b>Figura 2.3</b> Principales entidades y empresas para las que se produce software en la universidad. ....	37
<b>Figura 2.4</b> Principales diferencias entre metodología ágil y metodología robusta. ....	43
<b>Figura 2.5</b> Relación de los proyectos productivos de la Facultad 7 por área temática. ....	44
<b>Figura 3.1</b> Hilo conductor del procedimiento de SQA. ....	53
<b>Figura 3.2</b> Flujo central del procedimiento de SQA. ....	56
<b>Figura 3.3</b> Relación de actividades de aseguramiento de la calidad con sus respectivos entregables. ....	62
<b>Figura 3.4</b> Propuesta de plantilla para identificar las responsabilidades por roles. ....	66
<b>Figura 3.5</b> Propuesta de plantilla para la documentación mínima requerida. ....	67
<b>Figura 3.6</b> Propuesta de plantilla para la agenda del Plan de SQA. ....	69
<b>Figura 3.7</b> Propuesta de plantilla para la agenda del Plan de Pruebas. ....	74

## ***Índice de Tablas***

<b>Tabla 2.1</b> Tamaños de muestra por estratos de la población. ....	48
<b>Tabla 2.2</b> Resultados de los proyectos que solo se dedican a la producción de software. ....	50
<b>Tabla 2.3</b> Resultados de los proyectos que utilizan procedimientos de SQA.....	50
<b>Tabla 2.4</b> Resultados sobre la calidad del producto que se desarrolla. ....	50
<b>Tabla 2.5</b> Resultados sobre la satisfacción del cliente con el producto que se desarrolla.....	51
<b>Tabla 2.6</b> Resultados de la metodología de desarrollo que se utilizan en los diferentes proyectos. ....	51
<b>Tabla 2.7</b> Resultados del conocimiento de métodos de prueba del software.....	51
<b>Tabla 2.8</b> Resultados sobre las técnicas de control de la calidad que se realizan en los proyectos. ....	51
<b>Tabla 2.9</b> Resultados sobre los principales procesos que se realizan en los diferentes proyectos. ....	51



# *Introducción*

### **Introducción**

Se viven tiempos de transición, la era industrial poco a poco deja espacios a una nueva, que se caracteriza por el uso intensivo de la información y el conocimiento. Se muestran cambios en todos los patrones tradicionales de la actividad humana, apoyados en el conocimiento y las Tecnologías de la Información y las Comunicaciones (TIC).

El impacto del software (SW) en nuestra sociedad y en la cultura, continúa siendo profundo. Al mismo tiempo que crece su importancia, la comunidad del SW trata continuamente de desarrollar tecnologías que hagan más sencillo, rápido y menos costosa la construcción de software de alta calidad.

Uno de los problemas que se afrontan actualmente en la rama de la informática es la calidad del SW. Desde la década del 70, este tema ha sido motivo de preocupación para ingenieros, investigadores y comercializadores de SW, los cuales han realizado gran cantidad de investigaciones con dos objetivos fundamentales: ¿Cómo obtener un SW con calidad? y ¿Cómo evaluar la calidad del software?

Cuba es un país donde el desarrollo de software es aun incipiente. Es por ello que una de las principales tareas del Gobierno Cubano es desarrollar nuestra Industria del SW. No solamente con el fin del desarrollo de sistemas para la informatización de la sociedad sino también por los beneficios de insertarnos en el mercado a nivel mundial por su perspectiva económica. Para lograr esta tarea es necesario obtener productos con calidad, lo que implica la utilización de metodologías, procedimientos y estándares para el desarrollo de SW 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 SW.

En la Universidad de las Ciencias Informáticas (UCI) se desarrollan SW que son de gran ayuda a la vida económica, política y social del país. Para ello las diferentes facultades se dividen en perfiles vinculados a las ramas de la economía del país, pero el objetivo principal es uno: lograr un producto de alta calidad. El Aseguramiento de la Calidad del Software (SQA) es aun un tema novedoso para la UCI.

Exactamente en la Facultad 7 se han detectado una serie de problemas detectados al culminar el desarrollo del software por lo que surge la siguiente **situación problemática**:

- No se tiene una visión de calidad desde que comienza la concepción del producto.

- No se asegura que se realicen todos los procesos involucrados en el desarrollo de software que influyen en la calidad del producto.
- No se tiene en cuenta el control estricto de la calidad del producto en las diferentes fases de desarrollo de software.
- No se realizan actividades de aseguramiento de la calidad del software en ninguna de las etapas de desarrollo del producto.
- Inexistencia de un procedimiento para el aseguramiento de la calidad del software en los proyectos productivos de la Facultad 7.

Dada esta problemática el **problema a resolver** es el siguiente: ¿Cómo asegurar la calidad del software en los proyectos productivos de la Facultad 7?

A partir de este problema se enmarca como **objeto de estudio** el proceso de aseguramiento de la calidad en los proyectos productivos en la UCI, y el **campo de acción** abarca el proceso de aseguramiento de la calidad en los proyectos productivos de la Facultad 7.

De acuerdo a lo mencionado anteriormente el **objetivo general** de la investigación es:

Elaborar un procedimiento para asegurar la calidad del software que se desarrollan en los proyectos productivos de la Facultad 7.

La **idea a defender** es la siguiente: La elaboración de un procedimiento para el aseguramiento de la calidad del software, garantizará la producción de software con la calidad requerida.

Para dar cumplimiento al objetivo de este trabajo investigativo se trazaron las siguientes **tareas**:

1. Hacer un análisis valorativo de los procesos llevados a cabo en la universidad para el aseguramiento de la calidad del software en las distintas facultades.
2. Analizar el estado del arte sobre los diferentes conceptos emitidos por autores acerca de la calidad del software.
3. Evaluar el contenido de la información obtenida sobre la actividad que se investiga, establecer un diagnóstico de las tendencias actuales y tomar posición al respecto.

4. Elaborar un procedimiento basado en los aspectos estudiados durante la revisión bibliográfica y que de respuesta al problema planteado.

Este trabajo tiene una estructura de tres capítulos; El capítulo I donde se realiza una fundamentación teórica, con el objetivo de estudiar del estado actual de la calidad del software, para ello se investiga sobre las tendencias de la calidad del software, seguida por el Modelo Integrado de Capacidad y Madurez (CMMI) en su última versión (1.2), las normas ISO (Organización Internacional para la normalización) y el Estándar IEEE(Instituto de Ingenieros Eléctricos y Electrónicos), también se estudia los diferentes conceptos emitidos por autores acerca de la calidad del software. En el capítulo II se hace una caracterización de la universidad, para luego hacer un diagnóstico a la producción de software en la facultad 7. Finalmente en el capítulo III se hace la propuesta del procedimiento para el aseguramiento de la calidad del software, dicho procedimiento tiene como objetivo asegurar la calidad de la producción de software en los diferentes proyectos de la facultad 7.



*Capítulo 1*

## Capítulo 1: Fundamentación Teórica

### 1.1 Introducción

El marco teórico referencial que se desarrolla en este capítulo está dirigido a recoger los aspectos más significativos relacionados con las principales temáticas abordadas en diferentes fuentes bibliográficas, y los criterios emitidos por autores acerca del estado del arte que presenta el tema estudiado. Haciendo un análisis valorativo del estado actual de la Industria de Software (ISW) en el mundo y en Cuba; explicando la situación de la Industria Cubana de Software (ICSW) antes de surgir la UCI. También fue necesario revisar diferentes conceptos acerca de calidad, y calidad del software, gestión de la calidad, aseguramiento de la calidad como un elemento dentro de la gestión y las actividades de aseguramiento de la calidad en la producción de software; apoyados en el hilo conductor de la figura 1.1 que muestra la organización que se siguió en el capítulo.

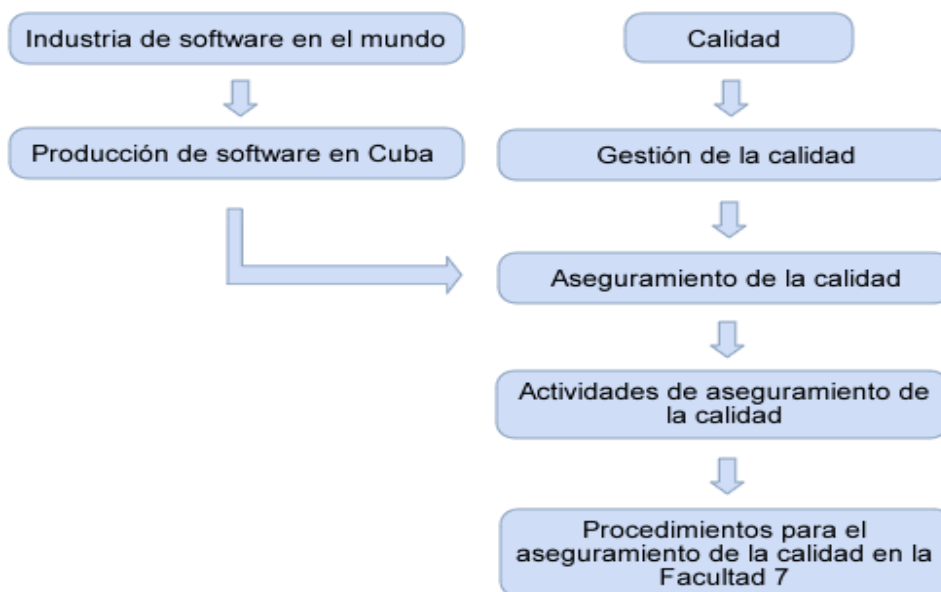


Figura 1.1 Hilo conductor del marco teórico referencial.

### 1.2 La Industria de Software y su desarrollo mundial

La industria del SW es la industria que involucra la investigación, desarrollo, comercialización y distribución de software. En nuestros días vivimos una revolución informática y los paradigmas con los



cuales veíamos la economía hace 15, 20, 30 años no son los mismos. El mercado de software mundial está estimado actualmente en \$600 billones de dólares.

El desarrollo de software constituye un sector de gran importancia mundial, se encuentra en el centro de todas las grandes transformaciones; sobre todo si se considera que los grandes temas del momento, como lo son la economía digital, la evolución de las empresas y la administración del conocimiento, se resuelven con software. La ISW interviene en todos los procesos que habilitan a la "nueva economía", se le considera una industria blanca que no contamina y que genera fuentes de trabajo bien remuneradas. Actualmente el SW representa una oportunidad para la Industria puesto que la misma se considera multibillonaria, además no requiere grandes inversiones y la materia prima es el capital humano. Existen países que ya han alcanzado grandes éxitos como: Irlanda, India e Israel e inclusive algunos de América Latina como México, Colombia, Chile, Brasil y Costa Rica y otros como China, Canadá, Estados Unidos y España. Con altas demandas en los sectores bancario, telecomunicaciones, salud, educación y servicios públicos. [1]

El Crecimiento de la ISW en la India es de un 50% existen alrededor de 1000 compañías de software, donde se alcanzó 30 mil millones en 1981 y 400 mil millones en 1995, actualmente exportan 6 billones. Estudios hechos estiman que para el año 2008 India exportará 57 billones de dólares en software y servicios asociados y que dicha industria dará empleo a 4 millones de personas. Cuentan con aproximadamente 250,000 empleados altamente calificados. El software que producen tiene una alta calidad debido a que la mayoría de sus empresas han alcanzado el nivel 5 de CMM (Modelo de capacidad y madurez). [2]

Brasil tiene una participación del mercado latinoamericano del 50%, representado por US \$ 1,900 millones; seguido por México, con una participación del 17%; y Argentina, con el 11%. En la lista, Colombia que factura alrededor de US \$190 millones al año y Venezuela ocupan respectivamente el cuarto y el quinto lugar. Sus mercados, sumados, representan el 50% del mercado argentino del software. En el caso particular de Colombia, según señalan informes, la competencia interna más importante proviene de Estados Unidos (70%), seguido por Alemania, Reino Unido, México y Canadá.

En sentido general existen grandes logros sin embargo aún la industria presentan graves problemas como: la carencia de personal capacitado, existe 885,000 vacantes en todo el mundo y un 15% promedio de rotación de empleados.[3]

A pesar del interés gubernamental y del indiscutible desarrollo de la ISW en los últimos años, las estadísticas internacionales no son alentadoras. Informes de instituciones dedicadas al análisis de software muestran los siguientes indicadores [4]:

- 25% de los proyectos de software son abortados.
- Se liberan productos a sus clientes con remanentes del 15% de defectos.
- Muchas empresas gastan de 30% a 44% de su tiempo y dinero en trabajos sobre software ya liberado.
- Se cumplen las planificaciones de tiempo solamente el 53% de las veces.

El mercado del software cada día evoluciona y se incorporan más y más empresas, por tanto la competencia se ha hecho necesaria y la fórmula para llegar al éxito es producir software de calidad utilizando estándares mundiales de calidad (CMMI, ISO) entre otros. La calidad del software debe ser demostrable y esto se puede lograr a través de certificaciones que es una herramienta de credibilidad de cara a los mercados internacionales.

### **1.3 Producción de Software en Cuba**

Cuba es un país donde el desarrollo de software es aun incipiente. Es por ello que una de las principales tareas del Gobierno Cubano es desarrollar la Industria del Software. Lo que ha traído consigo la creación de variadas estrategias con el fin de elevar la producción y calidad del software cubano.

La Industria Cubana del Software está llamada a convertirse en una significativa fuente de ingreso para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. La promoción de la industria cubana del software en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. El continuar la producción sostenida de software de alta calidad en prestaciones, imágenes y

soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación.

La industria cubana de software ha alcanzado notables logros. Se han desarrollado software para todos los aspectos de la economía cubana, incluyendo las telecomunicaciones y el mayor éxito se ha logrado en el área de equipamiento médico, donde el producto ha demostrado eficacia incrementando la calidad de la atención de salud en el país y han tenido aceptación en el exterior.

Según informes del *Christian Science Monitor*: «Hay [en el año 2000] 30 compañías dedicadas al desarrollo de software donde hace tres años no existía ninguna. Las exportaciones totales para el año 2000 fueron de (14 millones de dólares), representan un crecimiento de un 650% con respecto a 1999»

Para fomentar al desarrollo de esta industria se prioriza en el país, la enseñanza masiva de la computación en todo el sistema educacional, se cuenta con 26 politécnicos de informática, parte de los cuales son nuevos o fueron completamente remozados, donde se preparan alrededor de 40 mil técnicos medios, a los que se unen más de 11 800 estudiantes de nivel superior, de ellos 10 mil de la Universidad de Ciencias Informáticas. [5]

La UCI pretende ser la vanguardia del desarrollo de las empresas de software en Cuba y de llevar la informatización a todos los sectores de la sociedad: Salud, Educación, Cultura, Deporte, Turismo, Prensa, etc. Regir y propiciar un avance tecnológico y de la industria del software en Cuba y convertir la industria del software en un renglón fundamental de la economía e insertarnos en el mercado internacional, por lo que el reto de nuestra universidad es producir software de alta calidad.

### **Situación de la Industria de SW en Cuba al surgir la UCI**

En el año 2001, existían en el Ministerio de la Informática y las Comunicaciones (MIC) 24 empresas o entidades que tenían en su objeto social la producción de SW y servicios informáticos, Esta estructura sufrió diversas reorganizaciones. Actualmente hay 8 entidades en el MIC que desarrollan esta actividad. El mercado nacional y en frontera se abarcaba sin un ordenamiento, habiendo duplicidad de esfuerzos en muchos productos y servicios. La demanda nacional no se satisfacía, solo en el 2004-2005 es que se promulgan resoluciones ordenando que la venta de SW y servicios informáticos a empresas estatales se realizaría en moneda nacional. [6]

Exportaciones decrecientes hasta el año 2004. El pico se produjo en el 2001 con 8 MM USD. La presencia fundamental se logró con el SW asociado a equipos médicos de alta tecnología. Bajos niveles de explotación del software nacional en el país. El negocio para las empresas era representar productos. No existencia de un sistema que garantizara la calidad de los productos con óptica de industria. Desarrollo artesanal. Falta de implantación de metodologías y/o herramientas de programación de calibre “industrial”. Fuerte dependencia de las personas que desarrollaban los productos. Descalificación o pérdida del personal (“robo de cerebros”). [7]

Graduados con buena formación pero en número insuficiente. No se utilizaba correctamente la fuerza de técnicos medios. Poco desarrollo de productos para ambientes de software libre. Conceptos de soporte y mantenimiento atrasados sin explotar las posibilidades de soporte online (en línea. La falta de integración real trabajando alrededor de un objetivo han primado los intereses individuales. [8]

### **1.3.1 Evolución de la calidad del software en Cuba**

Los centros de educación superior han apoyado siempre el desarrollo de las técnicas para lograr la calidad del software en el país, sin embargo la industria se ha quedado rezagada al respecto, excepto honrosas excepciones como el de el Instituto Central de Investigaciones Digitales (ICID) que actualmente la calidad de los producto es certificada por Lloyd Register, aunque sin muchos avances actualmente en cuanto a software) y la Empresa de Producción y Desarrollo de Software de Calidad (SOFTCAL) que en los años 90 desarrolló un sistema de documentación y otro de control de calidad nacionales. [9]

La creación de la UCI y su concepción productiva ha dado un vuelco en los últimos años el tema de la calidad del software, al convertir al mismo cada vez en mayor medida en un interés gubernamental.

Se crea también por el Ministerio de Informática y las Comunicaciones el Grupo Nacional de Expertos en Calidad de Software (GNECS), con la UCI al frente, que mide el nivel en que se encuentra en general la industria de software en el país y se dispone a dictar en el futuro normativas y guías técnicas para el desarrollo del tema a lo largo y ancho del territorio nacional. Uno de los momentos claves en el desarrollo por la UCI del tema ha sido el inicio de la constitución del Laboratorio Nacional de Certificación de la Calidad de Software (CALISOFT) que va desarrollando pruebas y evaluaciones técnicas, así como documentos metodológicos para ganar la experiencia necesaria para acreditarse como un

centro independiente a nivel del país e incluso la región centroamericana. Algunas empresas desarrollan actualmente aunque no muy avanzados procedimientos para controlar la calidad por ejemplo [10]:

- Normas Ramales: Trabajos de preproducción de pruebas de aceptación e introducción a la producción, desarrolladas por el Instituto Nacional de Sistemas Automatizados y Técnicas de Computación (INSAC), 1980
- Metodología de Introducción de Logros, INSAC, 1985
- Sistema Ramal de Aseguramiento de la Calidad, INSAC, 1989
- Metodología de Desarrollo de Sistemas de Calidad, SOFTCAL, 1991
- Sistema de ingeniería y control de software, desarrollado por el Centro de Estudios de Ingeniería de Sistemas (CEIS), perteneciente al Instituto Superior Politécnico José Antonio Echeverría (ISPJAE), 1995
- Sistema de procedimientos de calidad en la Universidad de las Ciencias Informáticas (UCI) donde existe actualmente un Laboratorio de certificación, grupo de auditoría y revisiones, aseguramiento de la calidad y métricas, estas actividades las realiza el proyecto CALISOFT (Centro de Calidad del Software).

### **1.4 Calidad**

Hablar de calidad es algo que hace la mayoría de las organizaciones, pero dar calidad a los clientes es algo que muy pocas se aseguran de hacer. Los sistemas de calidad son aplicables y necesarios para todo tipo de organizaciones, todas tienen como principal objetivo darle al cliente lo que solicita para que, de esta forma, la rentabilidad de la empresa esté garantizada. Por otro lado, existe confusión de lo que significa calidad en las empresas.

El término calidad es un concepto “escurridizo”, fácil de visualizar y sin embargo difícil de medir, algunos autores la definen como un término subjetivo para el cual cada persona tiene su propia definición la cual puede ser a la vez, absoluta y relativa.

Aragón, Neida sugiere que cada país y empresa, defina qué es calidad para él o ella, de acuerdo a sus características específicas de competencia y rentabilidad defendiendo esta definición con vistas a

alcanzarla, lo cual mejoraría las condiciones específicas sociales del mismo, al obtener una adecuada rentabilidad. Se considera la calidad como una ciencia porque tiene principios, conceptos y definiciones que la hacen diferente a la hora de estudiarla. [11]

Para la Gestión de Calidad de Software a nivel mundial se han seguido principalmente dos tendencias: la primera a seguir las reglas implantadas por las oficinas internacionales de estandarización para los productos y servicios a través de las normas ISO y el estándar IEEE, y la segunda a seguir las creadas específicamente para el mundo del software como CMMI y SPICE estos modelos permiten la certificación del producto con un sello de calidad permitiendo así que las empresas alcancen un alto prestigio en el mercado [2]. La mejora de la calidad a través de los años está dada por el desarrollo de actividades de control de la calidad como una forma de detectar los defectos y de actividades de aseguramiento de la calidad para prevenir los defectos. (Ver Figura 1.4)

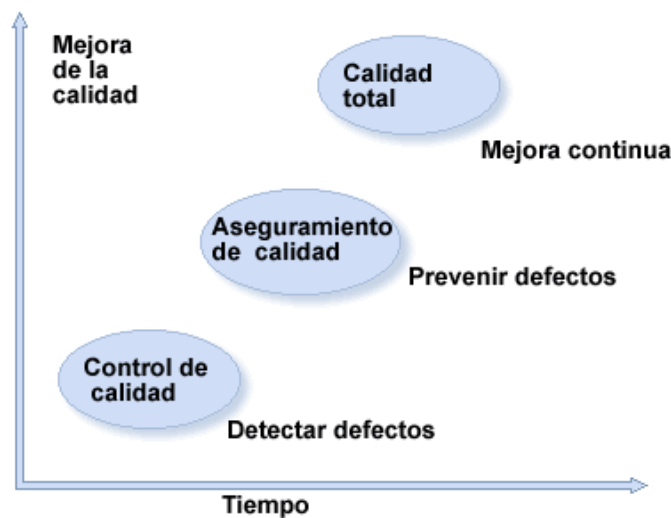


Figura 1.2 Evolución histórica de la gestión de la calidad.

### 1.4.1 Estándares y Modelos Internacionales para la calidad del software IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)

A la hora de hablar de los estándares para la Calidad de Software es prácticamente imposible dejar de mencionar el estándar IEEE para el Plan de Aseguramiento de la Calidad de Software. Este aborda

aspectos como la administración, la documentación, el control de código, etc. Dentro de la documentación le adjudica gran importancia al Plan para la Gestión de Configuración de Software. Dicho plan trata todo lo referente a la asignación de las responsabilidades, la identificación de las actividades que se realizarán durante todo el proceso, la identificación de la configuración, el reconocimiento de los elementos de configuración, el control de la configuración, el acceso a las bibliotecas, la aprobación o desaprobación de un cambio y la implementación del cambio de ser aprobado (todas estas incluidas entre las tareas de la Gestión de Configuración)

### ***CMMI V1.2 (Modelo Integrado de Capacidad y Madurez)***

CMMI es un modelo de la madurez de la mejora de proceso para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que se deben realizar a lo largo de todo el ciclo de vida del producto, realizando las actividades de desarrollo y mantenimiento del software. CMMI para el desarrollo (CMMI-DEV) proporciona una solución integrada comprensiva para las actividades del desarrollo y del mantenimiento aplicadas a los productos y a los servicios.

CMMI es un modelo creado por el SEI (Software Engineering Institute), un organismo fundado por el Ministerio de Defensa de Estados Unidos y la Carnegie Mellon University. [12]

CMMI se aplica a 4 disciplinas distintas:

- Ingeniería de Sistema: Cubre la construcción de un sistema con o sin software
- Ingeniería de Software: Cubre la construcción de soluciones software
- Integración de productos y procesos de desarrollo: Cubre la relación a largo plazo con el cliente.
- Relación con proveedores: Cubre los procesos relacionados con la subcontratación de partes del sistema

CMMI en su versión actual que es la 1.2 tiene dos representaciones una escalonada y otra continua y cada una tiene definido niveles de madurez y niveles de capacidades, mas adelante en este trabajo se explicará estas representaciones.

### ***ISO (Organización Internacional de Normalización)***

ISO es una red de institutos nacionales de estándares de 156 países que promueve la normalización internacional para facilitar el intercambio de bienes y servicios de aplicaciones. ISO 9000 es el nombre genérico con el que coloquialmente se designa a una familia de normas de aseguramiento de la calidad. La serie ISO 9000 está formada por cinco documentos, tres de ellos son modelos de aseguramiento de la calidad, específicamente el 9001, el 9002 y el 9003. Los otros dos son simples lineamientos que sirven de apoyo. [13]

**ISO 9000** Principios y conceptos, lineamientos para su selección y utilización.

**ISO 9001** Modelo de aseguramiento de la calidad, aplicable al diseño, desarrollo, fabricación, instalación y servicio.

**ISO 9002** Modelo de aseguramiento de la calidad, aplicable a la fabricación y a la instalación.

**ISO 9003** Modelo de aseguramiento de la calidad, aplicable a la inspección y ensayos finales.

**ISO 9004** Principios y conceptos, lineamientos para la gestión de calidad y elementos del sistema de calidad.

### **1.4.2 Conceptos de calidad según destacadas figuras**

El concepto de calidad ha evolucionado en el tiempo y en dependencia de la profesión base de la persona que la estudie y la utilice como herramienta en la gerencia de las industrias:

Deming en 1986 la define como un "predecible grado de uniformidad, a bajo costo y útil para el mercado". Lo cual es lógico teniendo en cuenta que es matemático y tratará siempre de cerrar las tolerancias buscando una mayor uniformidad del proceso. [14]

Juran, ingeniero eléctrico, hace varias definiciones de la calidad a lo largo de su carrera, desde "aptitud para el uso o propósito", hasta 1993 en que aporta ya no una sino dos definiciones de calidad, una que se refiere al producto "calidad es el conjunto de características de un producto que satisfacen las necesidades de los clientes y en consecuencia hacen satisfactorio el producto" y otra que se refiere a la organización "la calidad consiste en no tener deficiencias". No hay la menor duda de que para obtener calidad es preciso tener una organización que trabaje con calidad. [15]



Crosby en 1994 puntualiza que calidad es “entregar a los clientes y a nuestros compañeros de trabajo productos y servicios sin defectos y hacerlo a tiempo”. En este caso, considera dos tipos de clientes los internos y externos e involucra en la definición su filosofía de producir con cero defectos. [16]

Conway, consultor de calidad y discípulo de Deming, 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". Este autor en su definición hace referencia a la necesidad de observar la calidad del trabajo y desarrollar un sistema adecuado para obtenerla. [17]

Feigenbaum, presidente de la Academia Internacional de la 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". [18]

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 estos requisitos. [19]

La calidad del software es definida por Roger S. Pressman *como 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.* [20]

La anterior definición sirve para hacer hincapié en tres puntos importantes:

1. Los requisitos del software son la base de las medidas de la calidad. La falta de concordancia con los requisitos es una falta de calidad.
2. Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen esos criterios, casi siempre habrá falta de calidad.
3. Existe un conjunto de requisitos implícitos que a menudo no se mencionan (por ejemplo: el deseo por facilitar el uso y un buen mantenimiento). Si el software se ajusta a sus requisitos explícitos pero falla en alcanzar los requisitos implícitos, la calidad del software queda en entredicho.

A partir de esta definición vale la pena señalar que no sólo afecta la calidad el incumplimiento de los requisitos del cliente y los explícitamente definidos por la ingeniería de software, sino que los requisitos implícitos también deben ser considerados. Teniendo en cuenta que muy pocas veces el cliente está en condiciones reales de explicitar todo lo que se puede esperar del producto, muchas veces por desconocimiento y otras por la asunción tácita de muchas funcionalidades.

“La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. [21]

### **1.4.3 Conceptos de calidad según los Modelos y Estándares de calidad**

#### **Conceptos de calidad según la ISO:**

La ISO, define la calidad como la ausencia de deficiencias: "Es la totalidad de aspectos y características de un producto o servicio que se refieren a su capacidad para satisfacer necesidades dadas en la adecuación de sus objetivos".

La ISO 8402 [1994] define la calidad como "totalidad de las características de una entidad que le confieren la aptitud para satisfacer necesidades establecidas o implícitas" esta definición considera una entidad no solamente el producto o servicio que se vende sino también, una persona, una organización, un sistema, en otras palabras la amplía a todo lo que hace la calidad. [22]

La ISO 9000 [2000] plantea que 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. [23]

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

Según las ISO, la calidad no es más que la capacidad de un conjunto de características inherentes de un producto, sistema o proceso, para satisfacer los requisitos de los clientes.

#### **Conceptos de calidad según CMMIV1.2:**

El Instituto de Ingeniería de Software (SEI) en su modelo CMM define la calidad como:

- El grado en el cual un sistema, componente o proceso cumple con los requisitos especificados.
- El grado en el cual el sistema, componente o proceso cumple con las expectativas del cliente o usuario.

CMMI se asienta en el mismo principio expuesto para CMM: La calidad de un producto o de un sistema es en su mayor parte consecuencia de la calidad de los procesos empleados en su desarrollo y mantenimiento. [24]

El modelo CMMI parte de la premisa de que la calidad del software que se produce depende de la calidad de los procesos empleados en su producción.

### **1.5 Gestión de la Calidad**

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, el aseguramiento 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 y el aseguramiento de la calidad.[25]

Control de calidad: “Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio”. [26]

Aseguramiento de la calidad: “Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio satisfará los requerimientos dados sobre calidad”. [27]

#### **1.5.1 Sistemas de gestión de la calidad**

Sistema de gestión de la calidad: “Conjunto de la estructura de la organización, de responsabilidades, procedimientos, procesos y recursos que se establecen para llevar a término la gestión de calidad”. [28]

Según la norma ISO 9000:2000 sistema es “un conjunto de elementos mutuamente relacionados que interactúan.” Un sistema de gestión es aquel “sistema para establecer la política y los objetivos y para lograr dichos objetivos”. [29]

CMMI plantea que un sistema de gestión de la calidad es el conjunto de elementos interrelacionados de una empresa u organización, por los cuales se administra de forma planificada la calidad de la misma, en la búsqueda de la satisfacción de sus clientes. Entre dichos elementos, los principales son: la estructura de la organización, sus procesos, sus documentos y sus recursos. [30]

### **1.5.2 Principios de gestión de la calidad**

*ISO 9000:2000 definen gestión de la calidad:* como las actividades coordinadas para dirigir y controlar una organización en lo relativo a la calidad. Está presente en todas las etapas del proceso de producción de bienes o de servicios y en ella se incluyen, en el marco del sistema de calidad las actividades siguientes: la política, la planificación, el control, el aseguramiento y el mejoramiento de la calidad. Las actuales ISO 9000:2000 identifican ocho principios de gestión de la calidad que deben ser utilizados por la alta dirección con el fin de conducir a la organización hacia una mejora en el desempeño, los cuales son [31]:

1. Enfoque al cliente: las organizaciones dependen de sus clientes y por lo tanto deberían comprender las necesidades actuales y futuras de los clientes, satisfacer los requisitos de los clientes y esforzarse en exceder las expectativas de los clientes.
2. Liderazgo: los líderes establecen la unidad de propósito y la orientación de la organización. Ellos deberían crear y mantener un ambiente interno, en el cual el personal pueda llegar a involucrarse totalmente en el logro de los objetivos de la organización.
3. Participación del personal: el personal, a todos los niveles, es la esencia de una organización y su total compromiso posibilita que sus habilidades sean usadas para el beneficio de la organización.
4. Enfoque basado en procesos: un resultado deseado se alcanza más eficientemente cuando las actividades y los recursos relacionados se gestionan como un proceso.

5. Enfoque de sistema para la gestión: identificar, entender y gestionar los procesos interrelacionados como un sistema, contribuye a la eficacia y eficiencia de una organización en el logro de sus objetivos.
6. Mejora continua: la mejora continua del desempeño global de la organización debería ser un objetivo permanente de ésta.
7. Enfoque basado en hechos para la toma de decisión: las decisiones eficaces se basan en el análisis de los datos y la información.
8. Relaciones mutuamente beneficiosas con el proveedor: una organización y sus proveedores son interdependientes, y una relación mutuamente beneficiosa aumenta la capacidad de ambos para crear valor.

Es necesario asegurarse de que lo que se hace en la empresa, solo tiene sentido si es para darle al cliente lo que solicita. La única manera de hacer esto es a través de los procesos, que deben estar elaborados de manera simple, de tal forma que constituyan la principal herramienta de trabajo del personal, a todos los niveles y en todas las áreas de trabajo.

### **1.5.3 Modelos integradores de la gestión de la calidad en la empresa**

#### **Modelo CMMIV1.2**

CMMI tiene dos representaciones la escalonada y la continua. La representación continua ofrece la flexibilidad máxima para la mejora del proceso, permite una organización para mejorar los procesos diferentes a las proporciones diferentes. Si se conoce los procesos que necesitan ser mejorado en la organización y se entiende las dependencias entre las áreas del proceso descritas en CMMI, se debe usar la representación continua.

La representación escalonada ofrece una manera sistemática, estructurada de acercarse a la mejora del proceso, prescribe un orden para llevar a cabo las áreas del proceso según niveles de madurez que definen el camino de mejora para una organización del nivel inicial al nivel perfeccionando. Si no conoce por dónde empezar y qué procesos debe escoger para mejorar, se debe usar la representación escalonada.

Para la representación *continua* se definen 6 *niveles de capacidad* (0-5) que son un medio para mejorar los procesos que corresponden a un área de proceso dada incrementalmente y para la representación *escalonada* se define 5 *niveles de madurez* (1-5); estos niveles son un medio de predecir los resultados generales del próximo proyecto. [32]

Los 6 niveles definidos en CMMI para medir la capacidad de los procesos en una representación continua son:

- 0.- **Incompleto**: El proceso no se realiza, o se realiza parcialmente.
- 1.- **Ejecutado**: El proceso se ejecuta y se logra su objetivo, es decir satisface las metas específicas del área de procesos.
- 2.- **Gestionado**: Además de ejecutarse, el proceso se planifica, se revisa y se evalúa para comprobar que cumple los requisitos.
- 3.- **Definido**: Además de ser un proceso "gestionado" se ajusta a la política de procesos que existe en la organización, alineada con las directivas de la empresa.
- 4.- **Cuantitativamente gestionado**: Además de ser un proceso definido se controla utilizando técnicas cuantitativas.
- 5.- **Optimizado**: Además de ser un proceso cuantitativamente gestionado, de forma sistemática se revisa y modifica o cambia para adaptarlo a los objetivos del negocio.

Los 5 niveles definidos en CMMI para medir la madurez de los procesos en una representación escalonada son:

- 1.- **Inicial**: La organización normalmente no proporciona un ambiente estable para apoyar los procesos.
- 2.- **Gestionado**: La organización ha asegurado que se planifiquen los procesos y se ejecutaron de acuerdo con la política.
- 3.- **Definido**: Se caracterizan bien los procesos y se entienden, y se describen las normas, procedimientos, herramientas, y métodos.

- 4.- **Cuantitativamente gestionado:** la organización establece los objetivos cuantitativos para la calidad y la actuación de los procesos.
- 5.- **Optimizado:** La organización mejora sus procesos basados en una comprensión cuantitativa de las causas comunes de variación inherente en los procesos continuamente. [CMMI,2006]

### **Normas ISO**

Los tres modelos ISO de aseguramiento de la calidad (9001, 9002, 9003), no fueron escritos para ninguna industria en particular. Son genéricos y la intención es que se puedan adaptar a cualquier tipo de industria.

Para acometer la gestión de la calidad con resultados intermedios que permitan a la empresa ir obteniendo madurez en la medida en que avanza la implantación de las normas, se propone definir 5 niveles, ya no específicos al proceso de desarrollo del producto como da cuenta el CMMI, sino referentes a la gestión de la calidad a nivel de toda la organización: Nivel inicial, Nivel de proyecto, Nivel de gestión total de calidad, Nivel de certificación, Nivel de referencia.

### **1.6 Aseguramiento de la Calidad del software**

El aseguramiento de la calidad abarca todas aquellas actividades o prácticas que se realizan con el objetivo de asegurar un cierto nivel de calidad en el producto desarrollado que se realizan de forma independiente al equipo de desarrollo. [33]

*¿Qué es?* no es suficiente hablar por hablar diciendo que la calidad del software es importante, tienes que: (1) definir explícitamente lo que significa calidad del software; (2) crear un conjunto de actividades que ayuden a que todo producto de la ingeniería de software presenta alta calidad; (4) utilizar métricas para desarrollar estrategias que mejoren el proceso del software y, como consecuencia, mejoren la calidad del producto final. [34]

*¿Quién lo hace?* Todo el que esté relacionado con el proceso de ingeniería del software es responsable de la calidad. [35]

*¿Por qué es importante?* Lo puedes hacer correctamente o lo puedes repetir. Si un equipo de software aplica la calidad a todas las actividades de la ingeniería de software, reducirá la cantidad de trabajo

repetido que deba realizar. Esto supondrá costes más bajos, y lo que es más importante, mejorará el tiempo de llegada al mercado. [36]

*¿Cuáles son los pasos?* Antes de que se pueda iniciar las actividades de aseguramiento de la calidad del software, es importante definir la calidad del software a un número diferente de niveles de abstracción. Una vez que comprendes lo que es calidad, un equipo de software debe identificar un conjunto de actividades de aseguramiento de la calidad del software que eliminarán los errores de los productos realizados antes de que ocurran. [37]

*¿Cuál es el producto obtenido?* Para definir una estrategia de SQA para el equipo del software se ha creado un plan de SQA. Durante el análisis, diseño y codificación, el producto principal de SQA es un breve informe de la revisión técnica formal (RTF). Durante las pruebas se realizan los planes y procedimientos de prueba. También se puede generar otros productos de trabajo relacionados con la mejora de procesos. [38]

*¿Cómo puedo asegurar que lo he hecho correctamente?* Encontrar los errores antes que pasen hacer defectos, esto es, trabajar para mejorar tu eficiencia en la eliminación de errores, reduciendo de este modo la cantidad de trabajo repetido que tiene que hacer tu equipo de software. [39]

ISO 9000:2000 define el aseguramiento de la calidad: como parte de la gestión de la calidad dirigida a inspirar confianza en que se han cumplido los requisitos de la calidad. [40]

CMMI v1.2 define el aseguramiento de la calidad del software como: *Un medio planificado y sistemático para asegurar (por parte de la administración) que los estándares, prácticas, procedimientos y métodos definidos sean aplicados.* [41]

El modelo CMMI define los siguientes objetivos del aseguramiento de la calidad:

- **Evaluar Objetivamente Procesos y Productos:** 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.



Algunos desarrolladores de software continúan creyendo que la calidad del software es algo por lo que deben preocuparse una vez que se ha generado el código. Esta forma de pensar está muy lejos de la realidad.

El Aseguramiento de la Calidad del software es una actividad de protección, que se aplica a lo largo de todo el proceso del software. El aseguramiento de la calidad del software consiste en la auditoria y las funciones de información de la gestión. El objetivo del aseguramiento de la calidad es proporcionar la gestión para informar de los datos necesarios sobre la calidad del producto, por lo que se va adquiriendo una visión más profunda y segura de que la calidad del producto está cumpliendo sus objetivos. Por supuesto, si los datos proporcionados mediante el aseguramiento de la calidad identifican problemas, es responsabilidad de la gestión afrontar los problemas y aplicar los recursos necesarios para resolver aspectos de calidad.

### **1.6.1 Equipo de Aseguramiento de la Calidad**

El equipo de SQA sirve como representación del cliente en casa. Es decir, la gente que lleva a cabo el SQA debe mirar el software desde el punto de vista del cliente. ¿Satisface de forma adecuada el software los factores de calidad? ¿Se ha realizado el desarrollo del software de acuerdo con estándares preestablecidos? ¿Han desempeñado apropiadamente sus papeles las disciplinas técnicas como parte de la actividad de SQA? El grupo de SQA intenta responder a estas y otras preguntas para asegurar que se mantiene la calidad del software. [42]

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, el promedio está en una persona de Aseguramiento de la de Calidad por cada 15 a 40 personas en el equipo de desarrollo. Las áreas que caen bajo la responsabilidad del grupo de Aseguramiento de la Calidad son tres [43]:

- Las metas y objetivos: asegurar que las metas de la organización y los objetivos del usuario se están satisfaciendo.
- Los métodos: asegurar que las actividades de desarrollo de software siguen los procedimientos establecidos, estándares, las políticas de la organización y las guías de trabajo y recomendaciones disponibles.

- Rendimiento: asegurar que se optimiza la utilización del hardware y software, que son económicos, eficientes y efectivos.

### **1.7 Actividades de aseguramiento de la calidad del software**

El aseguramiento de la calidad del software comprende una gran variedad de tareas, asociadas con dos constitutivos diferentes los ingenieros de software que realizan trabajo técnico y un grupo de SQA que tiene la responsabilidad de la planificación de la calidad de forma tal que se asegure la calidad del software, supervisión, mantenimiento de registros, análisis e informes-. Los ingenieros de software afrontan la calidad (y realizan el aseguramiento de la calidad) aplicando métodos técnicos sólidos y medidas, realizando revisiones técnicas formales y llevando a cabo pruebas de software bien planificadas.

Las reglas del grupo de SQA tratan de ayudar al equipo de ingeniería del software en la obtención de un producto final de alta calidad. Estas actividades se enfrentan con el establecimiento de un *plan de SQA* para un proyecto. El plan se desarrolla durante la planificación del proyecto y es revisado por todas las partes interesadas. Las actividades de aseguramiento de la calidad realizadas por el equipo de ingeniería del software y el equipo de SQA son gobernadas por el plan. El plan identifica [44]:

- Evaluaciones a realizar.
- Auditorias y revisiones a realizar.
- Estándares que se pueden aplicar al proyecto.
- Procedimiento para información y seguimiento de errores.
- Documentos producidos por el grupo de SQA.
- Realimentación de información proporcionada al equipo de proyecto del software.

#### **Papel del equipo de SQA.**

*Participación en el desarrollo de la descripción del proceso de software del proyecto.* El equipo de ingeniería de software selecciona un proceso para el trabajo que se va a realizar. El grupo de SQA revisa la descripción del proceso para ajustarse a la política de la empresa, los estándares internos del software,

los estándares impuestos externamente (por ejemplo: ISO 9001), y a otras partes del plan de proyecto del software.

*Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso de software definido.* El grupo de SQA identifica, documenta y sigue la pista de las desviaciones desde el proceso y verifica que se han hecho las correcciones. [45]

*Auditoria de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software.* El grupo de SQA revisa los productos seleccionados; identifica, documenta y sigue la pista de las desviaciones; verifica que se han hecho las correcciones, e informa periódicamente de los resultados de su trabajo al gestor del proyecto. [46]

*Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.* Las desviaciones se pueden encontrar en el plan del proyecto, en la descripción del proceso, en los estándares aplicables o en los productos técnicos. [47]

*Registrar lo que no se ajuste a los requisitos e informar a sus superiores.* Los elementos que no se ajustan a los requisitos están bajo seguimiento hasta que se resuelven. [48]

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

El Aseguramiento de la calidad del software engloba [49]:

- Un enfoque de gestión de calidad.
- Tecnología de Ingeniería de Software efectiva (métodos y herramientas).
- Revisiones técnicas formales que se aplican durante el proceso del software.
- Una estrategia de prueba multiescalada.
- Un control de la documentación del software y de los cambios realizados.
- Un procedimiento que asegure un ajuste a los estándares de desarrollo de software.
- Mecanismos de medición y de generación de informes.

### **1.8 Revisiones del Software**

El trabajo técnico necesita ser revisado porque errar es humano. La segunda razón por la que necesitamos revisiones es que, aunque la gente es buena descubriendo algunos de sus propios errores, algunas clases de errores se le pasan más fácilmente al que los origina que a otras personas. Una revisión es una forma de aprovechar la diversidad de un grupo de personas para:

1. Señalar la necesidad de mejoras en el producto.
2. Confirmar las partes del producto en las que no es necesaria una mejora.
3. Conseguir un trabajo de mayor calidad.

El papel del grupo de SQA es decisivo en la producción de un software de alta calidad, puesto que los mismos deben de participar en el desarrollo de la descripción del proceso de software del proyecto, realizar revisiones de las actividades de ingeniería del software para verificar su ajuste al proceso de software, asegurar que las desviaciones del trabajo y los productos del software se documentan y registrar lo que no se ajuste a los requisitos e informar a los superiores. Además de estas actividades, el grupo de SQA coordina el control y la gestión de cambios y ayuda a recopilar y analizar las métricas del software.

#### **1.8.1 Revisión Técnica Formal (RTF)**

Una revisión técnica formal (RTF) es una actividad que asegura la Calidad del Software y que es llevada a cabo por los profesionales de la ingeniería de software.

Los objetivos de la RTF son [50] :

1. Descubrir errores en la función, la lógica o la implementación de cualquier representación del software.
2. Verificar que el software bajo revisión alcance sus requisitos.
3. Asegurar que el software haya sido representado de acuerdo con ciertos estándares predefinidos.
4. Conseguir un software desarrollado de forma uniforme
5. Hacer que los proyectos sean más manejables.

La RTF es una actividad colectiva que permite ampliar la visión sobre lo que se revisa, situación que se profundiza al ser aplicada por distintos niveles y especialidades de profesionales a distintos elementos que componen el software, lo cual permite; por una parte que los profesionales que recién se incorporan al

equipo de trabajo puedan observar los diferentes enfoques del análisis, diseño e implementación del software, además que sirve para promover la seguridad y la continuidad, ya que varias personas se familiarizan con partes del software que de otro modo no hubiesen visto nunca. Las revisiones técnicas formales permiten establecer un marco común para la definición de distintas actividades de revisión durante el desarrollo del software, este mecanismo debe contemplar todos los flujos de desarrollo y mantenimiento del software incluyendo cada una de sus fases. El mecanismo más común para su implementación es la reunión de revisión, la cual deberá regirse, para asegurar su éxito, por una buena planificación, control y, sobre todo, por la participación dedicada de todos y cada uno de los involucrados. [51]

Independientemente del formato que se elija para la RTF, cualquier reunión de revisión debe acogerse a las restricciones expuestas a continuación:

- Debe convocarse para la reunión (normalmente) entre tres y cinco personas.
- Se debe preparar por adelantado, pero sin que requiera más de dos horas de trabajo a cada persona.
- La duración de la reunión de revisión debe ser menor de dos horas.

Con estas condiciones sumamente restrictivas para la forma en que estas revisiones se realicen, trae consigo que cada RTF se centre en una parte muy delimitada del software total. De esta forma al limitar el centro de atención de la RTF la probabilidad de descubrir errores es mayor. El centro de atención de las revisiones es un producto de trabajo (por ejemplo, una porción de una especificación de requisitos, modelo de casos de uso, un diseño detallado de un componente, el código fuente de un componente o módulo, etc.). [52]

### **1.8.2 Pruebas del Software**

Las pruebas del software son un elemento crítico para el aseguramiento de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. El flujo de trabajo de pruebas añade valor al producto descubriendo sus errores. El objetivo específico de este flujo es encontrar cuantos más errores posibles. Las pruebas pueden demostrar la presencia de errores, no su ausencia. [53]

Objetivos de las pruebas:

1. La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
2. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
3. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Nuestro objetivo es diseñar pruebas que sistemáticamente saque a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y de esfuerzo. Si la prueba se lleva a cabo con éxito descubrirá errores en el software. Como ventaja secundaria la prueba demuestra hasta qué punto las funciones del software parecen funcionar de acuerdo con las especificaciones y parecen alcanzarse los requisitos de rendimiento, de alguna manera indican la calidad del software como un todo [54]

### **Principios de la prueba:**

Antes de la aplicación de métodos para el diseño de casos de prueba efectivos, se deberá entender ciertos principios básicos que guían las pruebas del software [55]

1. La prueba puede ser usada para mostrar la presencia de errores, pero nunca de su ausencia.
2. La principal dificultad del proceso de prueba es decidir cuando parar.
3. Evitar casos de pruebas no planificados, no reusables y triviales a menos que el programa sea verdaderamente sencillo.
4. Una parte necesaria de un caso de prueba es la definición del resultado esperado.
5. Los casos de pruebas tienen que ser escritos no solo para condiciones de entrada válidas y esperadas sino también para condiciones no válidas e inesperadas.
6. Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
7. El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.
8. Las pruebas deberían empezar por “lo pequeño” y progresar hacia “lo grande”.
9. Con la excepción de las pruebas de unidad e integración, un programa deberá ser probado por la persona u organización que lo desarrolló.

10. las pruebas deberían ser realizadas por un equipo independiente.

### **Métodos de Prueba.**

#### **El método de caja blanca:**

La prueba de la caja blanca del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

Requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código.

Mediante los métodos de prueba de la caja blanca, el ingeniero de software puede obtener casos de prueba que garanticen que [56] :

1. Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
2. Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsa.
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales.
4. Se ejerciten las estructuras internas de datos para asegurar su validez.

#### **El método de caja negra:**

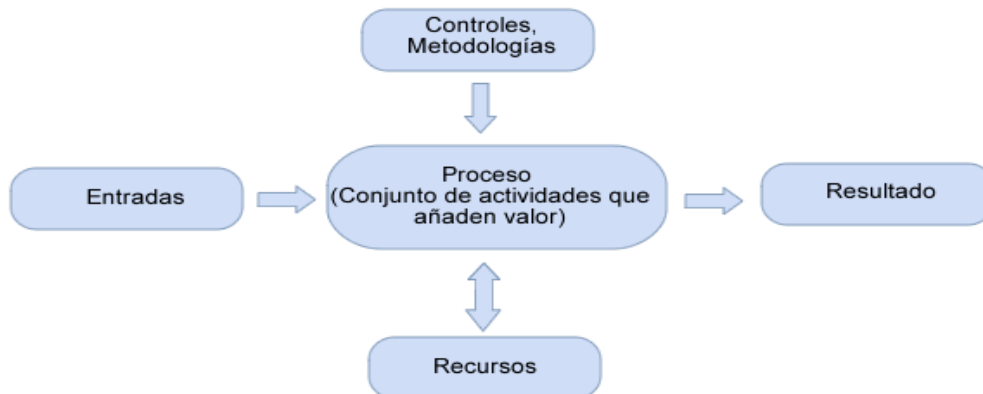
La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, o sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad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

Características de Pruebas de la caja negra [57] :

- Verifican las especificaciones funcionales y no consideran la estructura interna del programa.
- Es hecha sin el conocimiento interno del producto.
- No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

### 1.9 Conceptos de procesos

Un proceso transforma elementos de entrada en elementos de salida utilizando para ello mecanismos (recursos) regulados mediante controles. ( Ver figura 1.3)



**Figura 1.3** Transformaciones mediante procesos.

Los elementos de salida y el control (procedimiento pueden ser tangibles o intangibles). Se puede emplear un sistema de medición para recopilar informaciones y datos para analizar el desempeño del proceso y/o las características de los elementos de entrada y los elementos de salida. Los procesos se pueden combinar en una red que permita lograr colectivamente un propósito planificado. Conway [1988], plantea que el elemento de salida de un proceso puede ser el elemento de entrada de otro proceso. De esta forma, los procesos se pueden combinar en una cadena. No existe producto y/o servicio sin un proceso. De la misma manera no existe proceso sin un producto o servicio. [58]

Según Nogueira Rivera et al., [2004] un *proceso* es una secuencia ordenada y lógica de *actividades repetitivas* que se realizan en la organización por una persona, grupo o departamento, con la capacidad de transformar una entradas (*inputs*) en salidas o *resultados* programados (*outputs*) para un *destinatario* (dentro o fuera de la empresa que lo ha solicitado y que son los clientes de cada proceso) con un valor agregado. [59]

De la ISO 9000:2000 se toma la definición de:

*Proceso*: conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados. [60]



Procesos de desarrollo de software: define “Quién”, “Qué”, “Cuándo y “Cómo” hay que realizar las cosas para alcanzar un determinado producto de software.

### **Principales factores para la identificación y selección de los procesos:**

- Influencia en la satisfacción del cliente.
- Los efectos en la calidad del producto/ servicio.
- Influencia en factores claves de éxito.
- Influencia en la misión y estrategia.
- Cumplimiento de requisitos legales o reglamentarios.
- Los riesgos económicos y de insatisfacción.
- Utilización intensiva de recursos.

Deming plantea que el éxito de cualquier proyecto de mejora depende de la actitud que asumen los directivos, quienes han de crear valores y expectativas que sean claros y visibles. Los directivos deben de reconocer la importancia que tiene otorgarles a los empleados responsabilidades y autoridades para que actúen según su propio juicio hacia la consecución de los objetivos, desarrollando la creatividad e innovación en la búsqueda de soluciones, en los marcos de una cultura de trabajo en equipo, en un ambiente de confianza. [61]

### **1.9.1 El Proceso de desarrollo de software**

Howard Baetjer, comenta sobre el proceso: Como el software, al igual que el capital, es el conocimiento incorporado, y puesto que el conocimiento está inicialmente disperso, el desarrollo del software implícito, latente e incompleto en gran medida, es un proceso social de aprendizaje. El proceso es un diálogo en el que se reúne el conocimiento y se incluye en el software para convertirse en software. El proceso proporciona una interacción entre los usuarios y los diseñadores, entre los usuarios y las herramientas de desarrollo, y entre los diseñadores y las herramientas de desarrollo [tecnología]. Es un proceso interactivo donde la herramienta de desarrollo se usa como medio de comunicación con cada iteración del diálogo se obtiene mayor conocimiento de las personas involucradas. Realmente, construir software de computadora es un proceso de aprendizaje iterativo, y el resultado, es el conjunto del software reunido, depurado y organizado mientras se desarrolla el proceso.

*¿Qué es?* Cuando trabaja para construir un producto o un sistema, es importante seguir una serie de pasos predecibles- el mapa de carretera que te ayuda a obtener el resultado oportuno de calidad-. El mapa de carretera a seguir es llamado proceso del software. [62]

*¿Quién lo hace?* Los ingenieros de software y sus gestores adaptan el proceso a sus necesidades y entonces lo siguen. Además las personas que han solicitado el software tienen un papel a desempeñar en el proceso del software. [63]

*¿Por qué es importante?* Porque proporciona estabilidad, control y organización a una actividad que puede, si no se controla volverse caótica. [64]

*¿Cuáles son los pasos?* A un nivel detallado, el proceso que adaptemos depende del software que estemos construyendo. Un proceso puede ser apropiado para crear software de un sistema de aviación, mientras que un proceso diferente por completo puede ser adecuado para la creación de un sitio Web. [65]

*¿Cómo puedo estar seguro de que lo echo correctamente?* Hay una cantidad de mecanismos de evaluación del proceso del software que permiten a las organizaciones determinar la madurez de su proceso del software. Sin embargo, la calidad, oportunidad y viabilidad a largo plazo del producto que esta construyendo son los mejores indicadores de la eficiencia del proceso que estamos utilizando. [66]

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 tiene la misión de transformar los requerimientos de los usuarios 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. [67]

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. [68]

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. [69]

**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 termino 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.

### Proceso, métodos y herramientas

La Ingeniería del software es una tecnología multicapa. Como muestra la figura 1.4 cualquier enfoque de ingeniería (incluida ingeniería del software) debe apoyarse sobre un compromiso de organización de calidad.

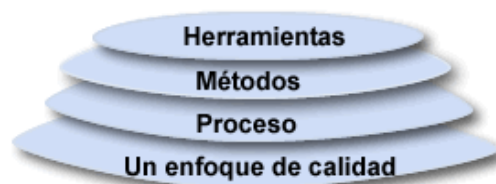


Figura 1.4 Capas de la Ingeniería de Software.

El fundamento de la ingeniería del software es la capa de proceso. El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de La Ingeniería de Software. El proceso define un marco de trabajo para un conjunto de áreas claves de proceso (ACPs) que se deben establecer para la entrega efectiva de la tecnología de la ingeniería del software. Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen productos del trabajo (modelos, documentos, datos, informes, formularios, etc.), se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente. Los métodos de la ingeniería del software indican cómo, construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería del software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología que incluyen actividades de modelado y otras técnicas descriptivas. [70]

Las *herramientas de* la Ingeniería del software proporcionan un enfoque automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por una herramienta la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado *ingeniería de software asistida por computadora (CASE)*. [71]

### **Producto y Procesos**

Si el proceso es débil, el producto final va a sufrir indudablemente. Aunque una dependencia obsesiva en el proceso también es peligrosa. Toda actividad humana puede ser un proceso, pero cada uno de nosotros obtiene el sentido de autoestima ante estas actividades que producen una representación o ejemplo que más de una persona puede utilizar o apreciar, una u otra vez, o en algún otro contexto no tenido en cuenta. Es decir, los sentimientos de satisfacción se obtienen por volver a utilizar nuestros productos por nosotros mismos o por otros. [72]

### **1.10 Conclusiones parciales**

- La Industria de Software es un mercado que evoluciona día a día, perfeccionándose cada vez más, donde se incrementan nuevas empresas productoras de software con grandes potenciales.
- La fórmula para llegar al éxito es producir software de alta calidad y demostrarla a través de las certificaciones que ofrecen los modelos de calidad que constituyen una herramienta de credibilidad, de cara a los mercados internacionales.
- El Aseguramiento de la Calidad del software es una actividad de protección, que debe estar presente en todas las etapas de desarrollo de cualquier producto; y debe existir en cualquier organización que esté encaminada a obtener una producción con la calidad deseada.



*Capítulo 2*

## Capítulo 2 Caracterización y Diagnóstico a la producción

### Capítulo 2: Caracterización y Diagnóstico a los proyectos productivos de la Facultad 7

#### 2.1 Introducción

En este capítulo se realiza una caracterización de la UCI, donde se analiza cómo se lleva a cabo la principal tarea en la universidad que es la producción de software, específicamente en la Facultad 7 la cual es el objeto de estudio de la investigación. Para dar cumplimiento a estos objetivos se realiza un diagnóstico con vista a detectar las necesidades e insuficiencia que presenta la producción del software desde su fase inicial, donde se emplearon diferentes técnicas como es la entrevista a diferentes personas relacionadas con la producción de software en la facultad y en la universidad, revisión de documentos internos, encuesta a los principales integrantes de los proyectos de la Facultad 7, entre otras; lo que facilita detectar las deficiencias que presenta la producción de software en la facultad, relacionadas con la calidad del producto que se desarrolla, durante su elaboración y entrega. Para ello se describe el presente hilo conductor que muestra la estructura del capítulo. (Ver figura 2. 1)

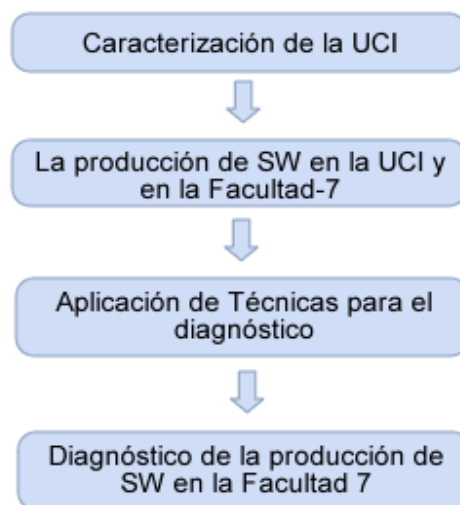


Figura 2.1 Hilo conductor del capítulo 2.

#### 2.2 Caracterización de la UCI

La Universidad de las Ciencias Informáticas es un programa de la Revolución en el marco de la Batalla de Ideas. Su primer curso se inició el 23 de septiembre de 2002. Es una universidad atípica precisamente porque fue creada sobre la base del nuevo concepto de universidad productiva, logrando una fuerte vinculación Universidad-Empresa. Está formada por estudiantes y profesores de las diferentes provincias

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

de nuestro país y su misión fundamental es: formar profesionales, comprometidos con su Patria, altamente calificados en la rama de la informática; así como producir software y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación.

Está constituida por 10 facultades y en cada facultad hay alrededor de un promedio de 2000 estudiantes, completando en este año 2007 una matrícula de 10 000 mil estudiantes alcanzando así el quinto año y la primera graduación en la universidad.

En la UCI la producción es un problema social, político, y económico, el cien por ciento de los estudiantes y profesores están vinculados a la producción participando en proyectos de alto valor tanto para el mercado nacional, como internacional, se plasma la concepción de que la docencia se realice desde la producción. Para lograr esta tarea se asigna a cada facultad un perfil y en dicho perfil se desarrollan varias áreas de proyectos con el objetivo de obtener un software que contribuye a la economía del país y al proceso de informatización de la sociedad cubana. (Ver figura 2.2)

La UCI representa la visión del futuro de nuestro país y así lo manifestó nuestro comandante en jefe en su visita en el año 2002 al referirse a la universidad y a los profesionales que se formaran en la misma de la siguiente manera: "La UCI, sería mejor decir la informática, se convertirá en una poderosísima fuerza científica, económica e incluso política, para el país... ustedes son los profesionales más importantes que vamos a formar en cuanto a perspectiva económica."

"Las producciones intelectuales serán el sustento fundamental de Cuba. La idea es convertir la informática en una de las ramas más productivas y portadoras de recursos para la nación..."

"Nuestra sociedad será una sociedad de trabajadores intelectuales. Dando vueltas y meditando – no es mucho lo que todavía hemos pensado y profundizado –, parece ser que aquí está la cantera principal."



## Capítulo 2 Caracterización y Diagnóstico a la producción

FACULTADES	ÁREAS DE PROYECTO SEGÚN PERFILES
Facultad #1	Gobierno (Sistemas Postales, Sistemas de Identificación) y Sistema de Gestión Académica
Facultad #2	Telecomunicaciones y Telemática (Comunicaciones Móviles); Redes y Seguridad Informática; Sistemas de Análisis de Información y Gestión de Emergencias
Facultad #3	Sistemas Jurídicos y Sistemas de Información Empresarial
Facultad #4	Sistemas Financieros y Tributarios Factoría de Software
Facultad #5	Realidad Virtual Control Automatizado
Facultad #6	Bioinformática; Cadena de Suministros Alimentación
Facultad #7	Sistemas Médicos Procesamiento de Imágenes
Facultad #8	Software Educativo y Multimedia, Deporte, Cultura. Sistemas Inteligencia Policial
Facultad #9	Matemática Aplicada; Sistemas Industriales Teleformación
Facultad #10	Software Libre y Plataformas de Productividad Sistemas de Gestión de Contenidos (Informatización de la Prensa)

Figura 2.2 Relación de las áreas de proyecto de las diferentes facultades.

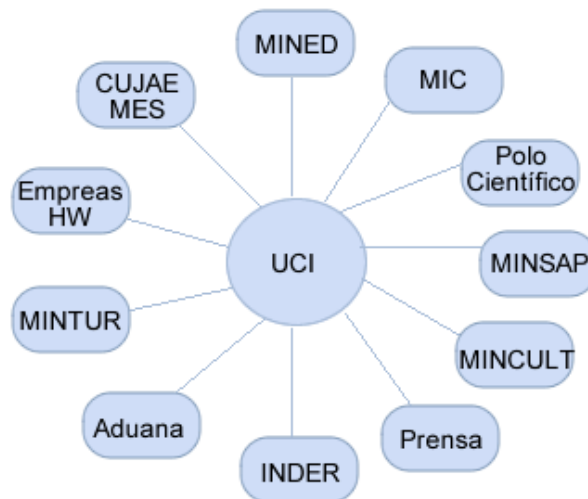
### 2.2.1 La producción de software en la UCI y en la Facultad 7

La Producción: proceso central, alrededor del cual se articulan los demás procesos fundamentales de la universidad: formación de pre y postgrado, investigación, u obtención de grados y categorías científicas. En la UCI la formación de los estudiantes se hace desde la producción, para esto se diseña la vinculación del estudiante a los proyectos desde el 1er año, en roles mas sencillos, hasta el 5to curso ya en roles de alta complejidad. El objetivo fundamental de las producciones es obtener proyectos para la informatización

## Capítulo 2 Caracterización y Diagnóstico a la producción

del país, la informatización de la vida de la propia universidad, y la exportación; estos se desarrollan fundamentalmente en la UCI, no en las entidades clientes. Los proyectos se organizan, gestionan, ejecutan y dirigen en las Facultades y estas asumen los proyectos según las áreas en que se hacen expertas; en cada facultad hay un Vicedecano de Producción, y especialistas al frente de los temas de Arquitectura y Calidad, que asesoran al Vicedecano, además funciona un Consejo de Producción. [73]

La UCI es una universidad de todos. Un programa de la Revolución diseñada para lograr resultados concretos, actualmente la universidad produce para diferentes entidades y órganos del país las cuales están representadas por diferentes empresas. (Ver figura 2.3)



**Figura 2.3** Principales entidades y empresas para las que se produce software en la universidad.

Para gestionar el tema de la producción se crea una Infraestructura Productiva (IP) como órgano metodológico, regulador, controlador y balancista, que presta servicios generales a todas las Facultades. La IP es la encargada de gestionar todos los procesos involucrados con la producción de software en la universidad está estructurada por diferentes colectivos de dirección: consejo de producción, consejo de dirección, consejo de técnicos y consejo de calidad, además existe también la dirección de informatización que es la encargada de convertir la universidad en una ciudad digital y cuatro áreas de dirección de producción donde cada una de estas áreas tienen la tarea de asumir la asesoría metodológica y el control de los procesos asociados a la producción de software de las diferentes facultades para garantizar la

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

correcta ejecución de proyectos de producción a través de la definición y estandarización de procedimientos y procesos productivos. [74]

En sentido general las diferentes direcciones de producción en la universidad tienen los siguientes objetivos:

- Proporcionar a las facultades metodologías, procesos y procedimientos definidos para llevar a cabo la ejecución de los proyectos productivos de la universidad.
- Definir y establecer indicadores y mecanismos de control a los proyectos.
- Efectuar chequeos y evaluaciones periódicas a los proyectos productivos de las facultades según los indicadores establecidos.
- Velar por la eficiente administración de los recursos humanos y materiales en cada proyecto.
- Participar en las negociaciones con los clientes que soliciten servicios de la universidad.
- Efectuar chequeos periódicos con los clientes para medir el avance del proyecto, el grado de satisfacción del cliente y la muestra de los resultados alcanzados.
- Brindar el servicio de consultoría a los proyectos en temas de Gestión de Proyectos Informáticos, Metodologías de Desarrollo y Arquitectura de Software.
- Impartir cursos de capacitación en los temas de Gestión de Proyectos Informáticos, Metodologías de Desarrollo y Arquitectura de Software.

La Dirección de Informatización de la Universidad de las Ciencias Informáticas , perteneciente a la Infraestructura Productiva, tiene como misión: dirigir, organizar, coordinar, chequear, diseñar y definir la Informatización de todos los procesos internos en cada una de las áreas que rigen la vida de la universidad, desde la perspectiva de una Ciudad Digital, logrando una total integración de todas las entidades, flujos y/o procesos, basado en el funcionamiento armónico de la tecnología y los servicios informáticos. Proporcionando además el uso ordenado y masivo de las tecnologías con que cuenta esta importante institución. Convirtiéndose en el prototipo para la Informatización de la Sociedad Cubana.

La dirección de calidad de la UCI es la encargada de certificar el software desarrollado.

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

En la IP se brindan diferentes servicios; los servicios tecnológicos son los que facilitan las tecnologías y la arquitectura de la misma, así como la estandarización y la arquitectura de información; por otro lado los servicios de calidad realizan las actividades de aseguramiento de la calidad a través de un grupo de personas que realizan diferentes auditorías y revisiones en el laboratorio de certificación; los servicios de informatización que tienen la misión de crear plataformas y soporte de software: así como los servicios de comunicación visual encargados de gestionar información visualmente esta tarea la realizan fundamentalmente tres grupos: creativo, producción, realización y por último también brinda servicios legales de gestión de la propiedad Intelectual así como asesoría especializada.

Cada negocio es un nuevo modelo de proyecto; cualquier proyecto productivo está compuesto por [75]:

- Personal de la UCI y una empresa productora de SW (negocia el cliente).
- Expertos funcionales.
- Equipo de desarrollo (altamente comprometido).
- Condiciones logísticas y tecnológicas adecuadas.

La encargada de la comercialización con otros países es la empresa de comercio exterior ALBET (alternativa bolivariana para la exportación de tecnologías); creada en el 2005 es una Sociedad Mercantil Cubana, de la UCI, representa la “cara comercial” para todos los proyectos de exportación coordinados a través de la Infraestructura Productiva y en el marco del Programa de la Revolución para la Producción de Software. Además Integra esfuerzos de múltiples organizaciones y empresas para garantizar las soluciones tecnológicas integrales. [76]

### **2.2.2 Evolución de la producción de software en la UCI**

Curso 2002-2003: No existían proyectos ni claridad de cómo iba a enfrentar la UCI el tema de la producción. Se estudiaban los Parques Tecnológicos. Se gana una licitación en México, pero el proyecto no resulta dada la inmadurez de la organización. [77]

Curso 2003-2004: Se buscan proyectos en todos los organismos del país. Gran dispersión de pequeños proyectos. La fuerza de estudiantes está en 1er y 2do año. No están claras las prioridades. Primeros

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

intentos de organización metodológica y de una estructura organizativa para la producción. Experiencia de impacto en Venezuela. [78]

Curso 2004-2005: Se crea la Vicerrectoría Primera para unir la atención a los procesos de formación y producción. Se crean en la IP direcciones de SW para la Salud, SW Educativo, Calidad, Exportación de SW, Comunicación Visual (Diseño). Los proyectos más importantes se organizan y dirigen desde la IP. Entidades productivas externas se ubican en la IP como por ejemplo softel. Se definen conceptos para la producción de la UCI. Se crea ALBET (Alternativa Bolivariana para la Exportación de Tecnología). Se comienzan grandes proyectos en el marco del Convenio de Colaboración Cuba-Venezuela. [79]

Curso 2005-2006: Se consolida la IP, se definen las Facultades como las ejecutoras de los proyectos y la IP como la estructura que se responsabiliza con el proceso productivo. La UCI enfatiza su papel como coordinadora y desarrolladora de proyectos en el marco de la Alternativa Bolivariana para las América (ALBA). Propuestas de alianzas y negocios. [80]

### **2.2.3 Necesidad de la producción**

Proceso productivo real: entendiendo este no como la mera actividad de hacer el software (de programarlo) sino de asistir al conjunto de actividades que lo garantizan, que van desde un proceso organizacional de división del trabajo por roles, desarrollo de bases tecnológicas que implican enfrentarse a problemas desconocidos (al menos parcialmente), hasta el echo de lograr un trabajo integrado en equipo. Luego el estudiante no se enfrenta a ejercicios académicos, sino estos quedan superados por una práctica real, donde los problemas se corresponden a situaciones prácticas determinados por la necesidad.

La formación desde la producción tiene como objetivo formar en los estudiantes las competencias necesarias para desempeñarse de manera eficaz en el desempeño de un rol dentro de un equipo de desarrollo de Software.

Al mismo tiempo ayuda a que se consoliden las bases de la industria de software, propiciando: mecanismos de gestión del conocimiento, sistemas de certificación, cursos de capacitación ajustados a los intereses de los proyectos, etc.

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

El aprendizaje autónomo aparece en este contexto con una mayor presencia, la necesidad de alcanzar una gran experiencia en poco tiempo hace necesario que se utilicen múltiples métodos de enseñanza: presenciales, semipresenciales y a distancia.

El profesor visto en un sentido amplio debe convertirse en tutor, ya que las funciones del profesor en un grupo de proyecto superan tanto al profesor que esta detrás de una plataforma como al clásico profesor de un aula, esencialmente por las funciones educativas. En consecuencia el colectivo de profesores pasa a ser junto a las direcciones de las organizaciones el colectivo educativo por excelencia.

Es precisamente en el plano educativo y ético donde este modelo manifiesta todas sus potencialidades.

En los proyectos es donde los estudiantes tienen mayor permanencia, en ellos se desarrollan las habilidades y aptitudes que luego este reproducirá en su vida profesional, ya que la mayoría una vez graduado se mantendrán trabajando en un proyecto y capacitándose de manera continua. Igualmente es donde se puede lograr mayor interacción entre estudiantes y profesores, convirtiéndose este en el patrón más cercano en el que los estudiantes se reflejan.

Es importante que cada profesor y en especial los Guías y Jefes de Proyecto, se apropien de técnicas para desarrollar los valores que deseamos y necesitamos que estén formados en nuestros estudiantes en relación con nuestro proyecto social socialista, nuestras condiciones sociales y nuestra UCI.

Cada estudiante debe sentirse comprometido con el rol que desempeña, dentro de su proyecto debe ser capaz de aplicarlo eficientemente y en el tiempo que se le ha propuesto, por eso es muy importante que nuestros estudiantes adquieran una conciencia y un sentido de pertenencia, y que valoren la importancia de la producción de software para nuestro país.

Estos valores sólo pueden ser detectados, valorados y formados a partir del ejemplo, en la interacción de la convivencia de profesores y estudiantes, pues sólo en un ambiente natural es donde se manifiestan todas las aristas de la personalidad como la sinceridad y la correspondencia de los parlamentos con las acciones y compartir e integrar estos esquemas de valores. [81]

### **2.2.4 Desarrollo de software en los proyectos productivos**

El Proceso fundamental que se lleva a cabo en la universidad es el proceso de desarrollo de software:

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

Un proceso es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, explica los pasos necesarios para terminar el proyecto.

Un proceso define: “Quién”, “Qué”, “Cuándo y “Cómo” hay que realizar las cosas para alcanzar un determinado producto de software.

Para llevar a cabo los diferentes procesos dentro del desarrollo de software se utilizan metodologías y herramientas de desarrollo de software. De forma general en los proyectos de la universidad se utiliza una sola metodología, excepto en algunos casos que utilizan más de una metodología.

Una metodología se encarga de elaborar estrategias de desarrollo de software que promueven prácticas adoptivas centradas en las personas o los equipos, orientados hacia la funcionalidad y la entrega, de comunicación intensiva que requiere implicación directa del cliente.

Las metodologías para desarrollo de software más utilizadas en la universidad son: RUP y XP (Programación Extrema).

RUP tiene tres características fundamentales es interactivo e incremental, dirigido por casos de usos y centrado en la arquitectura. Además define nueve procesos fundamentales para obtener un software con alta calidad; los seis primeros se consideran flujos ingenieriles modelación del negocio, levantamiento de requisitos, análisis y diseño, implementación, prueba y despliegue, Los tres restantes se consideran flujos de apoyo: planificación de proyecto, gestión de la configuración y ambiente para realizar las diferentes actividades. En cada uno de estos flujos de trabajo, RUP define cuatro fases: fase de inicio, elaboración, construcción y transición, cada fase culmina con un hito; el de la primera fase es tener definidos claramente los objetivos de la organización, en de la segunda definir la arquitectura del sistema, el de la tercera tener la capacidad operacional inicial del sistema y el de la última tener el *release* del sistema

La característica fundamental de XP es que su principal proceso está enfocado a la programación, además un aspecto significativo es que el cliente forma parte del equipo de desarrollo, también define actividades de planificación, diseño y pruebas.

RUP es una metodología robusta y XP es una metodología ágil, las principales semejanzas entre ambas metodologías es que son un instrumento fundamental para el desarrollo de software, que aseguran la

## Capítulo 2 Caracterización y Diagnóstico a la producción

calidad del software implementado, además definen roles para realizar las diferentes tareas en cada fase de desarrollo; sin embargo entre ambas metodologías existen varias diferencias. (Ver figura 2.4)

METODOLOGÍA ÁGIL	METODOLOGÍA ROBUSTA
Se basan en heurística para la producción de software	Se basan en Normas
Aceptan y fomentan el cambio	Mayor o menor resistencia a los cambios
El cliente forma parte del equipo	El cliente no forma parte del equipo, solo se tiene encuentros planificados con el cliente, entrevistas, etc
Equipo de trabajo pequeño o mediano	Equipos grandes mayores de 15-20 miembros.
Procesos con pocas reglas	Procesos con muchas normas.
Poca documentación Poco análisis y diseño	Mucha documentación Mucho análisis y diseño
Conceden poca importancia a la arquitectura de los sistemas	Conceden mucha importancia a la arquitectura de los sistemas

**Figura 2.4** Principales diferencias entre metodología ágil y metodología robusta.

En la universidad cada proyecto desarrolla software con distintas características, de forma general los más comunes que se desarrollan son software de aplicaciones, además de multimedia, entre otros.

Una actividad fundamental en el desarrollo del software es la implementación, para ello en nuestra universidad se utiliza varios lenguajes de programación por ejemplo C #, C++, Java, PHP, entre otros.

### 2.3 Producción de software en la Facultad 7

La producción de software en la facultad siete está vinculada al sector de la salud, su objetivo es elaborar y mantener software que permitan contribuir al proceso de informatización en dicho sector. Esta responsabilidad es de todo los trabajadores de la facultad, a través de la dirección del vice decano de



## *Capítulo 2 Caracterización y Diagnóstico a la producción*

producción, el cual organiza la producción por áreas temáticas, facilitando que se creen por varios grupos diferentes proyectos. Cuando surge un nuevo proyecto se discute en la IP, entonces se decide si se acepta o no, en caso positivo se lleva el proyecto a la facultad y se le asigna un área temática. Por lo general cada proyecto tiene como promedio 70 estudiantes y 4 profesores, los que tienen asignado un laboratorio con el equipamiento necesario, constituyendo este su puesto de trabajo. En este momento en la facultad existen 12 proyectos vinculados a la salud y otros proyectos que tienen otra misión como por ejemplo, el portal de GPI (Grupo de Procesamiento de Imágenes y Señales) y el proyecto GICAC (Grupo de Investigación, Control y Aseguramiento de la calidad). [82]

En la figura 2.5 aparece la relación de los proyectos productivos de la Facultad 7 por áreas temáticas:

ÁREA TEMÁTICA	PROYECTO
Sistema de Gestión Hospitalaria	Hospitales LIS:Laboratory Information System FMS:Pharmacy Management System
Sistema de Apoyo a la Salud	Balance de Materiales Control Sanitario Internacional Docencia y Estadísticas
Sistema Especializado	Nefrología Fisioterapia SIUM: Sistema integral de urgencia médica
Procesamiento de Imágenes y Señales	Cassandra PACS Portal de GPI Diana
Atención Primaria de Salud	APS
Calidad del software	GICAC: Grupo de Investigación, Control y Aseguramiento de la Calidad

**Figura 2.5** Relación de los proyectos productivos de la Facultad 7 por área temática.

El proyecto GICAC (Grupo de Investigación, Control y aseguramiento de la calidad de la Facultad siete) perteneciente al área temática de calidad del software, está formado por estudiantes de segundo, tercero, cuarto y quinto año de la facultad. Una vez concluido el software perteneciente a cada proyecto, GICAC es

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

el encargado de revisarlo aplicando varias pruebas para asegurar la calidad de estos productos, y luego es pasado a la dirección de calidad que es la encargada de certificar el software

Las pruebas que se aplican hasta el momento son de caja negra; los integrantes del proyecto realizan diferentes casos de pruebas con el objetivo de encontrar errores, una vez detectada las no conformidades del producto, se registran y se le comunica al equipo de desarrollo del software. Este proceso es muy importante porque permite que el software antes de ser entregado al cliente este libre de errores, y agiliza el proceso de obtención del certificado de calidad.

### **2.4 Aplicación de Técnicas para el diagnóstico**

Las principales técnicas empleadas que permitieron fundamentar la investigación de este trabajo fueron las siguientes:

#### **La entrevista**

Es un método de recopilación de datos empíricos, un proceso de comunicación entre dos o más personas, generalmente de forma oral donde las preguntas al entrevistado se hacen por vía directa, se deben desarrollar preguntas que permitan respuestas precisas.

#### **La revisión de documentos**

Se utiliza para recoger la información que se encuentra registrada en un documento establecido. Se utilizaron los listados de los perfiles por facultades, registros de proyectos por área temática de la facultad siete, etc.

#### **La encuesta**

Es una técnica que tiene como objetivo obtener cierta información deseada de un sujeto (trabajador) preseleccionado de antemano dentro de una muestra representativa, por medio de una conversación directa cuyas pautas vienen indicadas en un guión o cuestionario que ha sido previamente diseñado y probado en una muestra piloto o muestra inicial.

Se realiza cuando la información que se realiza puede ser obtenida a partir de la respuesta que una persona o varias puedan dar a un cuestionario preelaborado, y las mismas están dispuestas a colaborar con la investigación.

## Capítulo 2 Caracterización y Diagnóstico a la producción

---

La encuesta es semejante a la entrevista pero escrita, donde a través de un conjunto de preguntas se pretende obtener una información sobre el mundo interior del encuestado o su percepción del fenómeno que se investiga, por lo que no puede ser obtenida por observación.

### Muestreo Aleatorio Estratificado

Existen pocas áreas donde el impacto del desarrollo de la estadística se haya hecho sentir más que en la ingeniería. La estadística ha venido a ser una herramienta vital para ingenieros, les permite comprender fenómenos sujetos a variaciones y predecirlos o controlarlos eficazmente. En este trabajo se utilizó para determinar la cantidad de población que será encuestada, y el tamaño de la muestra es decir la cantidad de encuestas que se realizarán.

El Muestreo Aleatorio Estratificado consiste en subdividir la población en subpoblaciones de tal forma que la unión de ellos será la población, y la intersección de cualquiera de los dos dará como resultado el conjunto vacío, es decir, no tendrán elementos comunes.

A las subpoblaciones se les llamará estratos, se tratará de conformar estos de modo que los elementos dentro de ellas sean homogéneos. El tamaño de la muestra se distribuirá entre los estratos, en función de distintos criterios pero lo que caracteriza al MAE (Muestreo aleatorio estratificado) es que la selección de la muestra de cada estrato se hará bajo el procedimiento de muestreo irrestricto aleatorio y se realizará independientemente de los diferentes estratos.

Es recomendable estratificar en función del tamaño de las unidades y distribuir la muestra proporcionalmente al número de unidades de los estratos.

### Notación

Se supone que la población consta de  $n$  unidades y están distribuidas en  $L$  estratos, constituyen una partición de la población; se representará por  $N_h$  en el  $n_i$  de  $u$  en el estado  $h$ -ésimo, de aquí:

$$N = N_1 + N_2 + \dots + N_h + \dots + N_l$$

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

Total de la población: 
$$x_h = \sum_1^{N_h} x_{hi} \quad \text{media} \quad \bar{x}_h = \frac{1}{N_h} \sum_i^{N_h} x_{hi}$$

Fracción de muestreo del estrato: 
$$f_h = \frac{n_h}{N_h}$$

Si el tamaño de la muestra de los estratos se distribuye proporcionalmente al número de unidades en el estrato, es decir se cumple que:

$$n_h = n \frac{N_h}{N} \quad \text{Para todo } h. \text{ En este caso se dice que la distribución de la muestra se ha hecho con asignación proporcional.}$$

Para la determinación del tamaño de muestra de una población con  $S^2$  desconocida se puede determinar mediante la expresión:

$$n = \frac{\left( \frac{Z_{1-\frac{\alpha}{2}}}{d} \right)^2 P(1-P)}{1 + \frac{1}{N} \left( \frac{Z_{1-\frac{\alpha}{2}}}{d} \right)^2 p(1-P) - \frac{1}{N}}$$

Donde  $1-\alpha$  es el nivel de confianza,  $d$  es el error absoluto,  $Z_1$  percentil de la distribución normal,  $P$  Proporción de la población y  $N$  tamaño de la muestra.

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

Realizando el cálculo del tamaño de muestra para un nivel de confianza  $1 - \alpha = 90\%$ , con un error absoluto  $d = 0.10$ , se obtiene el percentil de la distribución normal  $Z_{1-\alpha/2} = 1.64$  y se asume como proporción de la población  $P = 0.5$ ,  $N = 323$  que es la cantidad total de todos los integrantes de los proyectos de la facultad; se obtiene que:

$$n = \frac{\left(\frac{1.64}{0.10}\right)^2 0.5(1-0.5)}{1 + \frac{1}{323} \left(\frac{1.64}{0.10}\right)^2 0.5(1-0.5) - \frac{1}{323}} = \frac{67.24}{1.2052} = 55,79 \approx 56$$

Se deben aplicar un total de 56 encuestas.

Conociendo que  $n = 56$ ,  $N_h$  es la cantidad de integrantes en cada proyecto,  $N = 323$  que es el total de integrantes de los diferentes proyectos.

Calculando el tamaño de muestra por estrato (se considera estrato a cada proyecto) aplicando

$$n_h = n \frac{N_h}{N}; \text{ resultan los tamaños de muestra por estrato que se muestran en la tabla 2.6}$$

Estratos	Cantidad	Tamaño de la Muestra
GPI	81	14
APS	80	14
Hospitales	70	12
Docencia	35	6
Nefrología	17	3
Balance de Materiales	14	2
SIUM	13	2
Fisioterapia	13	2

**Tabla 2.1** Tamaños de muestra por estratos de la población.

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

---

### **2.5 Diagnóstico de la producción de software en la Facultad 7**

#### **Resultados de la encuesta aplicada a los integrantes de los proyectos productivos de la Facultad 7**

Para detectar algunas deficiencias existentes en los proyectos productivos relacionadas con el aseguramiento de la calidad del trabajo, se aplicaron diferentes técnicas para el diagnóstico, entre ellas está la encuesta realizada a los integrantes de los diferentes proyectos productivos de la facultad la cual fue aprobada por la vicerrectora primera de la universidad la compañera Alina Ruiz. (Ver anexo 1)

La muestra hallada por regiones representa el 17,28; 17,5; 17,14; 17,14; 17,64; 14,28; 15,38; 15,38 % de los integrantes (estudiantes y profesores) de los diferentes proyectos de la facultad. A estos grupos se les aplicó la encuesta relacionada con el aseguramiento de la calidad en los proyectos productivos de la facultad 7 en la que se recoge diferentes opiniones acerca de cómo se manifiesta la gestión de la calidad (aseguramiento y control) del software que se desarrolla; así como los principales procesos que se llevan a cabo en el desarrollo del software, con vista a detectar los principales problemas que están afectando la calidad de los productos y de los procesos en los proyectos productivos de Facultad 7.

Se decidió segmentarlos por regiones para evitar que los diferentes procesos de desarrollo del software que presentan cada proyecto puedan influir significativamente en los resultados, estratificándose entonces por las distintas regiones.

Los resultados obtenidos revelan un cierto desconocimiento en sentido general de los principales estándares y modelos para la gestión de la calidad, sin embargo todos los integrantes de los proyectos tienen bien definidos los conceptos de calidad y aseguramiento de la calidad del software (SQA). Los proyectos que sólo se dedican a la producción de software representan un 85,8%; mientras los que aparte de producir se dedican a otras actividades solo el 14,2%.

El 3, 57 % de los encuestas creen que la calidad del producto que desarrollan es excelente, el 82, 14% buena, el 1,78% regular y el 12,5% no sabe como es la calidad del software que desarrollan. El 78, 63 % de los encuestados plantean que el cliente se siente satisfecho con el servicio que brindan mientras que el 16,07% creen que no y solo el 5,30% no sabe si el cliente se siente satisfecho. El 94,64% de los proyectos utilizan para el desarrollo de software la metodología RUP y el 3,57% XP y existe en la facultad un solo proyecto que aparte de estas dos también utilizan otras lo que representa el 1,78%.

## Capítulo 2 Caracterización y Diagnóstico a la producción

De los diferentes procesos de desarrollo de software se puede decir que el único que se realiza correctamente es el proceso de planificación de las tareas, pero los procesos de gestión de la configuración y los procesos de PSP (procesos de software personal) y TSP (proceso de software en equipo) en muchas ocasiones no se tienen en cuenta en los proyectos productivos, y las veces que se tienen en cuenta no se ejecutan correctamente.

El 94,65% de los encuestados plantean que en el proyecto se realiza la planificación de tareas y el 5,35% no; el 46,42% realizan el proceso de gestión de la configuración, mientras que el 53,57% no; el 35,71% realizan el PSP y el 64,28% no; el 39,28% realizan el TSP y el 60,71% no. En sentido general podemos afirmar que en los proyectos productivos no existe un procedimiento de aseguramiento de la calidad pues el 94,64% de los encuestados plantean que no y solo el 5,35% plantea que si pero que no está definido aún. El 19,64% de los encuestados plantean que en el proyecto que pertenecen se realizan técnicas de control de la calidad y que conocen métodos de prueba y el 80,35% plantea que no. A continuación aparecen tabulados los resultados de la encuesta con mayor precisión a través de las siguientes tablas.

Procedimiento SQA	Total	Porcentaje (%)
si	48	85,8
no	8	14,2

**Tabla 2.2** Resultados de los proyectos que solo se dedican a la producción de software.

Procedimiento SQA	Total	Porcentaje (%)
si	3 no definidos	5.35
no	53	94.64

**Tabla 2.3** Resultados de los proyectos que utilizan procedimientos de SQA.

Calidad del producto	Total	Porcentaje (%)
Excelente	2	3.57
Bien	46	82.14
Regular	1	1.78
No sabe	7	12.5

**Tabla 2.4** Resultados sobre la calidad del producto que se desarrolla.

## *Capítulo 2 Caracterización y Diagnóstico a la producción*

Satisfacción del cliente	Total	Porcentaje (%)
si	44	78,63
no	9	16,07
no sabe	3	5,30

**Tabla 2.5** Resultados sobre la satisfacción del cliente con el producto que se desarrolla.

Metodología de Desarrollo	Total	Porcentaje (%)
RUP	53	94,64
XP	2	3,57
Otras	1	1,78

**Tabla 2.6** Resultados de la metodología de desarrollo que se utilizan en los diferentes proyectos.

Métodos de prueba	Total	Porcentaje (%)
si	11	19.64
no	45	80.35

**Tabla 2.7** Resultados del conocimiento de métodos de prueba del software.

Téc. de control Q	Total	Porcentaje (%)
si	11	19.64
no	45	80.35

**Tabla 2.8** Resultados sobre las técnicas de control de la calidad que se realizan en los proyectos.

Procesos		Total	Porcentaje (%)
Planificación de tareas.	si	53	94.64
	no	3	5.35
Gestión de la configuración	si	26	46.42
	no	30	53.57
PSP	si	20	35.71
	no	36	64.28
TSP	si	22	39.28
	no	34	60.71

**Tabla 2.9** Resultados sobre los principales procesos que se realizan en los diferentes proyectos.



### **2.6 Conclusiones parciales**

- El proceso productivo en la universidad, no está dedicado solamente a programar software, también debe asistir al conjunto de actividades que lo garantizan. Estas van, desde el proceso organizacional de división del trabajo por roles, el desarrollo de bases tecnológicas que implican enfrentarse a problemas desconocidos, hasta el hecho de lograr un trabajo integrado en equipo.
- El objetivo fundamental de la producción en la UCI está dedicado a desarrollar proyectos para la informatización del país. Así como la informatización de la vida de la propia universidad, y la exportación. Convirtiéndose la universidad en una de las instituciones más productivas y aportadoras de recursos para el país.
- Los proyectos productivos de la Facultad 7 no tienen definido un procedimiento para el aseguramiento de la calidad del software.

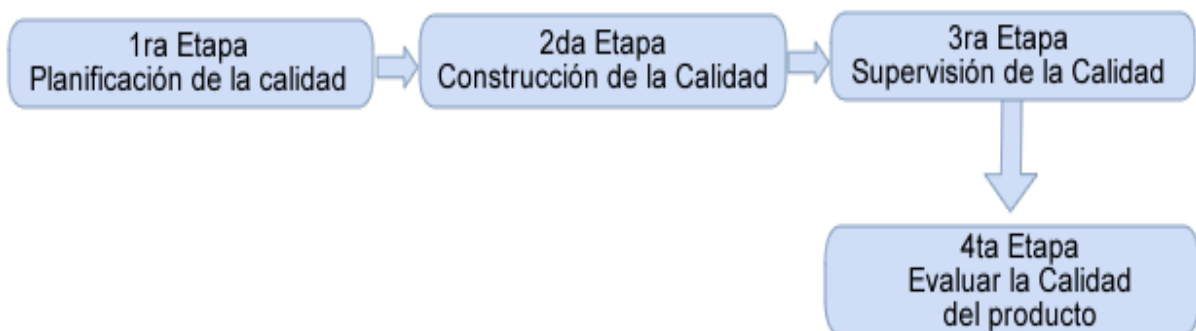


*Capítulo 3*

### Capítulo 3: Propuesta de Procedimiento para el Aseguramiento de la Calidad del Software en los Proyectos Productivos de la Facultad 7

#### 3.1 Introducción

Este capítulo está dirigido a explicar la propuesta de procedimiento de Aseguramiento de la Calidad del Software el mismo se puede aplicar en los diferentes proyectos productivos de la Facultad 7. Se describen todos los pasos a seguir para el correcto desarrollo de dicho procedimiento, y se referencia toda la documentación requerida para la realización del mismo. Se propone seguir cuatro etapas para de esta forma desarrollar exitosamente este procedimiento; en la primera etapa se debe planificar la calidad, en la segunda etapa construir la calidad y la tercera supervisar la calidad, una vez que el producto final es aceptado por el cliente, se realiza la cuarta etapa y última que es evaluar la calidad del producto final que ha sido desarrollado. (Ver figura 3.1)



**Figura 3.1** Hilo conductor del procedimiento de SQA.

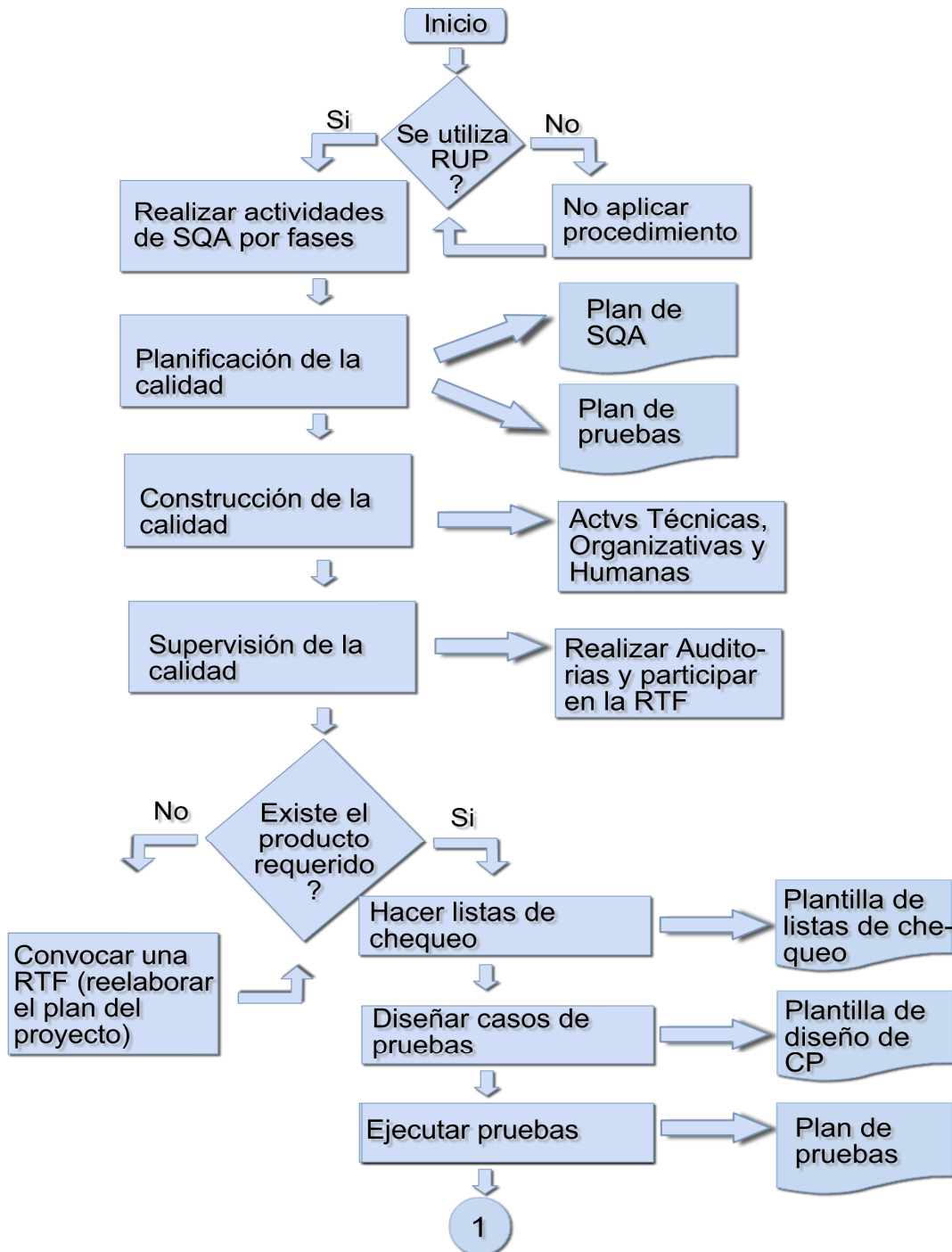
### 3.2 Propuesta de Procedimiento de SQA

Este procedimiento está dirigido a asegurar la calidad en las producciones de software de los proyectos productivos de la Facultad 7; el mismo está basado en la metodología RUP por ser esta la más utilizada en los proyectos de software, según la encuesta aplicada.

Se propone realizar una serie de actividades de SQA que se deben realizar en cada una de las 4 fases de desarrollo que propone RUP (Inicio, Elaboración, Construcción, y Transición). Para ver estas actividades consultar los anexos (2, 3, 4 y 5) donde se detalla el funcionamiento de dicho procedimiento para cada una de estas fases respectivamente.

El procedimiento propuesto tiene concordancia directa con el concepto estudiado de SQA que define CMMI V1.2 porque en el mismo se planifica la calidad de forma sistemática, mediante el Plan de SQA y el Plan de Prueba. El plan de SQA es realizado por el administrador de la calidad de forma sistemática para asegurar que los estándares, prácticas, procedimientos y métodos definidos por parte del equipo de desarrollo de software sean aplicados de forma eficiente, así como los procesos vinculados al desarrollo del producto. En este procedimiento se definen roles para cumplir con las actividades de aseguramiento de la calidad del software.

En la siguiente figura se detalla a través de un flujo central los pasos a seguir para desarrollar el procedimiento propuesto. Teniendo en cuenta las cuatro etapas definidas para asegurar la calidad del software, se deben seguir cuatro pasos fundamentales la planificación, construcción y supervisión de la calidad, por último evaluar la calidad del producto. Se propone seguir este flujo en cada una de las fases de desarrollo de software. (Ver figura 3.2)



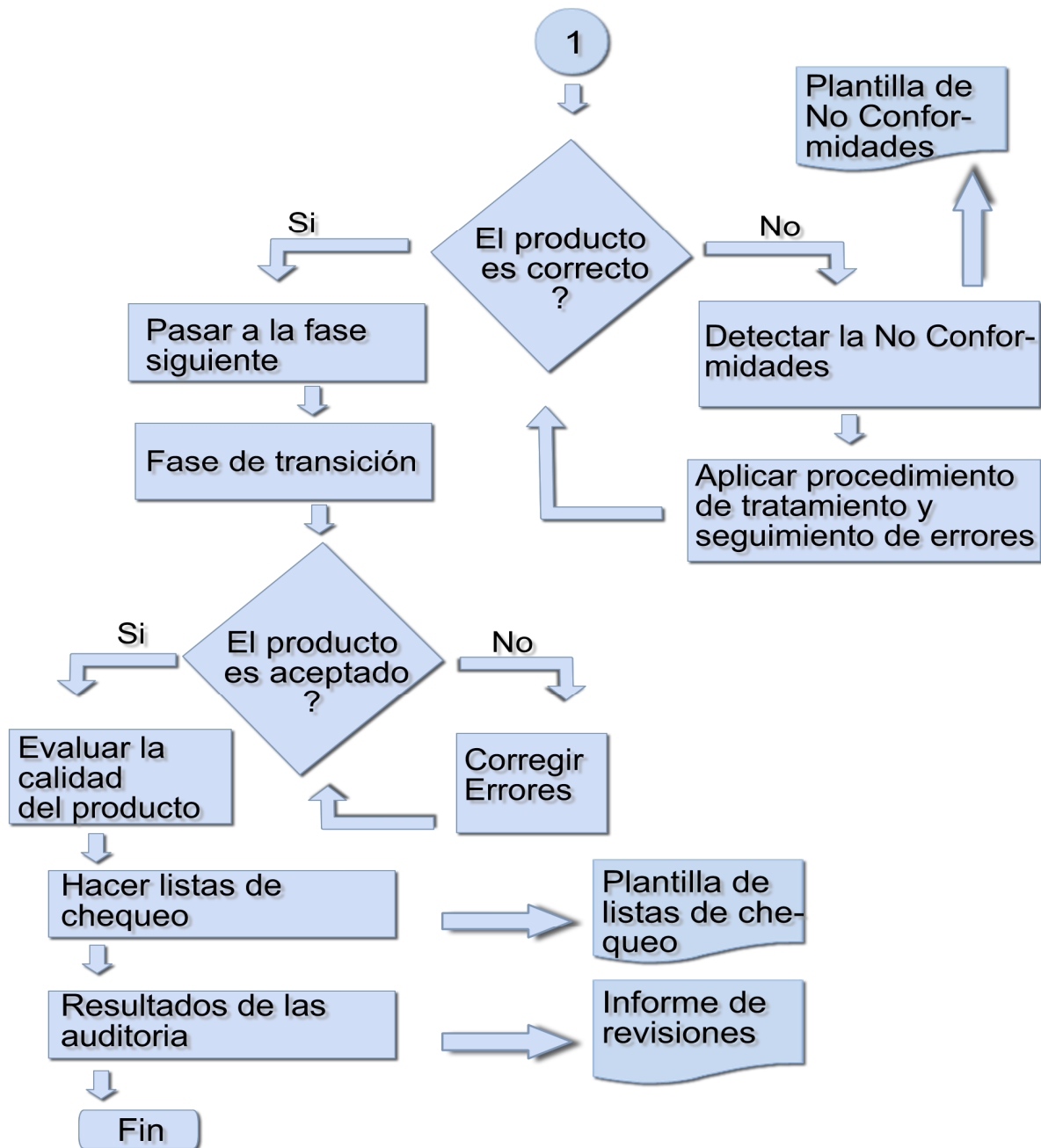


Figura 3.2 Flujo central del procedimiento de SQA.

### 3.2.1 Propósito

Definir los pasos a seguir para lograr alcanzar un nivel máximo de calidad en el software desarrollado y en los procesos involucrados.

### 3.2.2 Objetivos

Asegurar la calidad del software que se produce en los diferentes proyectos de la facultad 7.

### 3.2.3 Responsabilidades

**Aclaraciones:** Se propone que el equipo de aseguramiento de la calidad debe ser diferente al equipo de desarrollo del software, de esta forma se detectan con facilidad los posibles errores que pueden afectar en la calidad del software.

Si el equipo de desarrollo es de 15 a 40 personas debe existir un equipo de aseguramiento de la calidad.

Si el equipo de desarrollo es de 41 a 65 personas deben existir dos equipos de aseguramiento de la calidad.

Si el equipo de desarrollo es de 65 ó mas personas deben existir tres equipos de aseguramiento de la calidad.

Para el equipo de aseguramiento de la calidad se proponen los siguientes **roles**:

**Administrador de la calidad:** Es el responsable de asegurar la calidad del producto y de los procesos involucrados en el desarrollo del software; además debe lograr que las pruebas tengan éxito por tanto es defensor de las prueba y de la calidad, planificación y administración de recursos, resolución de problemas que impiden las pruebas. Además participa en la evaluación de calidad del producto final.

**Especialista en tecnologías:** Es el responsable de proponer los métodos y herramientas de ingeniería de software eficientes y óptimos.

**Analista de Prueba:** Es el 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. Tiene la responsabilidad de 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 los casos de prueba y asegurar su implementación exitosa. Identifica las técnicas apropiadas, herramientas e instrucciones para implementar las pruebas necesarias y encauzar los recursos correspondientes para las pruebas.

**Probador:** Es el responsable de ejecutar las pruebas en las diferentes fases de desarrollo del software, pero además es el encargado de registrar el resultado de la prueba.

A continuación se describe de una forma detallada las actividades que debe cumplir cada persona según los roles definidos:

*El administrador de calidad es el responsable de:*

1. 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.
2. 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 SQA.
3. Definir las métricas que se utilizarán para medir las propiedades de calidad que se identifiquen; las mismas se insertan en el plan de SQA.
4. Revisar las entregas semanales que hace el equipo de desarrollo de software, debe asegurar que los productos se realicen en la fecha señalada 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.
5. Revisar la calidad del producto y hacer un resumen de la situación actual del producto que se está revisando; este resumen se introduce en el informe de revisión.
6. Elaborar el Plan de SQA y el Plan de pruebas; en la fase inicial del desarrollo del proyecto el administrador de la calidad elabora el plan de SQA y le da un seguimiento sistemático en las restantes fases, sin embargo debe elaborar un plan de prueba para cada fase de desarrollo.

*El analista de pruebas es el responsable de:*

1. Hacer entrevistas y visitas a los clientes durante todo el ciclo de desarrollo del software.



2. 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.

El diseñador de prueba es el responsable de:

1. 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.
2. Elaborar las listas de chequeo que se realizarán en las diferentes fases.

El probador es el responsable de:

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

El equipo de aseguramiento de la calidad es responsable de:

1. Participar en el desarrollo de la descripción del proceso de software del proyecto, el equipo de desarrollo de software selecciona un proceso para el desarrollo del producto y el equipo de SQA revisa la descripción del proceso para ajustarse a la política que se sigue en la universidad a partir de la metodología RUP y los estándares internos del software.
2. Revisar las actividades de ingeniería del software para verificar su ajuste al proceso de software definido.
3. Realizar auditorías a los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software
4. Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
5. Registrar lo que no se ajuste a los requisitos e informar los problemas.

6. Realizar actividades de construcción de la calidad, con el objetivo de evitar que el equipo de desarrollo de software cometa errores y se obtenga el producto en el tiempo señalado y con una alta calidad; esta actividad requiere de la elaboración de un informe de actividades constructivas. Se propone realizar tres tipos de actividades constructivas. Actividades técnicas: incluye por ejemplo la aplicación de principios, técnicas y herramientas de ingeniería de software, organizativas: incluye por ejemplo la aplicación de modelos de procesos o planes y humanas: incluye por ejemplo la formación del personal y la motivación
7. Participar en las revisiones técnicas formales que desarrolla el equipo del proyecto.
8. Evaluar la calidad del software producido. Teniendo en cuenta los resultados obtenidos en las diferentes pruebas y revisiones aplicadas cuando el producto es entregado al usuario final y el mismo es aceptado, pues se concluye este procedimiento de SQA dando una evaluación de calidad al producto final.

### **3.2.4 Principales Actividades de SQA**

Las principales actividades que se desarrollan en este procedimiento son:

Planificación de la calidad: Seleccionar y clasificar las propiedades de calidad que se van a establecer como requisitos, con respecto al producto y con respecto al proceso, así como los mecanismos de control de calidad.

Construcción de la calidad: Actividades constructivas que sirven para evitar la introducción de errores.

Supervisión de la calidad: Supervisar y corregir el trabajo que se está realizando con el objetivo de llegar a satisfacer los requisitos establecidos.

Evaluar la calidad del producto final.

Las actividades de aseguramiento de la calidad están dirigidas a garantizar que se realicen los diferentes procesos en cada una de las fases de desarrollo del software y se obtenga un producto al culminar cada fase; de esta forma en la última fase, el software obtenido tendrá una alta calidad y el cliente se sentirá satisfecho con el mismo. Para ello se propone realizar las actividades de SQA en cada una de las fases de desarrollo.

### 3.2.4.1 Planificación de la Calidad

Como actividades de planificación se propone: Realizar el Plan de SQA y el Plan de pruebas. Primeramente el administrador de la calidad y el analista de pruebas deben entrevistarse con los clientes para identificar las necesidades de los mismos, esta actividad culmina con un listado de necesidades del cliente. De esta forma se puede seleccionar y clasificar las propiedades de calidad e insertarlas en el Plan de SQA. Identificar los mecanismos de control de calidad e insertarlas en el plan de pruebas. Se debe tener claro que este plan de pruebas es realizado por el equipo de SQA y es totalmente diferente al plan de prueba que defina el equipo de desarrollo de software.

- **Elaborar el Plan de SQA**

*Para el formato del Plan de SQA se utilizó la plantilla del estándar IEEE y el mismo tiene la siguiente estructura:*

*1 Propósito:* Redactar el propósito que tiene el Plan de aseguramiento de la calidad.

*2 Acrónimos y Abreviaturas:* Definir las diferentes abreviaturas o acrónimos que se utilizaran en el plan para una mejor comprensión de todas las personas que lo lean.

*3 Referencias:* Referenciar la bibliografía que se utilice para realizar el plan.

*4 Gestión:* Especificar los elementos de la organización que tienen influencia sobre la calidad del software, como está conformada la línea de gestión de calidad en el proyecto de software, de quien es la autoridad y responsabilidad por la calidad del software, la lista de tareas cubiertas por este plan (incluye la planificación de la calidad, e identificar las propiedades de calidad que se deben medir) y las responsabilidades por cada tarea.

*4.1 Organización:* El administrador de calidad es el responsable de realizar la gestión que asegura que el proceso establecido sea realmente implementado y que los productos de ese proceso cumplan con los criterios de calidad establecidos en este plan. La gestión de calidad es una disciplina de gestión, junto con Gestión de proyecto y Gestión de la configuración del software. Las disciplinas de gestión brindan soporte a las disciplinas ingenieriles (Modelación del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue) y se realizan en forma paralela a ellas.

4.2 *Tareas*: En esta sección se describen las tareas de aseguramiento de la calidad a realizar, indicando para cada una: el entregable asociado. A continuación aparece una propuesta de las principales tareas que se deben realizar (Ver figura 3.3).

Actividad	Entregable Asociado
Planificación de la Calidad	Plan de SQA
Identificar Propiedades de Calidad	Plan de SQA
Revisión de Calidad de Producto	Informe de Revisión
Revisión de Entregas semanales	Informe de Revisión
Revisión Técnica Formal (RTF)	Informe de Revisión
Planificación de las pruebas por fases	Plan de Pruebas

**Figura 3.3** Relación de actividades de Aseguramiento de la Calidad con sus respectivos entregables.

4.2.1 *Planificación de Calidad*: Se planifica cómo se medirá la calidad, especificando las pruebas que se realizan en qué fase y quién las realiza y los resultados de dicha prueba; además se especifica según las propiedades de calidad a través de qué métricas se medirá dicha calidad. Se propone seguir el siguiente formato.

*Fase*: Especificar la fase de desarrollo (inicio, elaboración, construcción, transición)

*Atributos de calidad del producto*: Identificar las propiedades de calidad que se medirán en la fase especificada anteriormente.

*Mecanismo*: Identificar cómo se medirá la propiedad de calidad descrita anteriormente, puede ser mediante métricas, pruebas y listas de chequeo.

*Métrica*: Nombre de la métrica para medir los atributos de calidad.

*Tipo de Métrica*: Se especifica si la métrica es interna. Externa o de calidad en uso.

*Prueba*: Nombre de las prueba que se realizarán

*Responsable de la Prueba*: El Responsable del equipo de SQA en que se ejecute la prueba (puede ser más de una persona).

*Listas de chequeo:* Especificar el objetivo de las listas de chequeo, así como exactamente que parte del producto se chequeará.

Se propone utilizar los atributos de calidad del software que define la norma cubana ISO 9126 para las propiedades de calidad del software (la funcionalidad, la confiabilidad, la usabilidad, la eficiencia, la mantenibilidad y la portabilidad). Éstos son además divididos en sub-características. Las sub-características pueden medirse por métricas internas o externas.

**Funcionalidad:** La capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas. [83]

- **La idoneidad:** La capacidad del software para mantener un juego apropiado de funciones para las tareas especificadas y los objetivos del usuario.
- **La precisión:** La capacidad del software para proporcionar efectos o resultados correctos o convenidos.
- **La interoperabilidad:** La capacidad del software para actuar recíprocamente con uno o más sistemas especificados.
- **La conformidad:** La capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares.

**Confiabilidad:** La capacidad del software para mantener su nivel de ejecución cuando se usa bajo las condiciones especificadas. [84]

- **La madurez:** Capacidad del software de evitar una avería como resultado de haberse producido un fallo del software.
- **La tolerancia ante fallos:** Capacidad del software de mantener un nivel de ejecución específico en caso de fallos del software o de infracción de sus interfaces especificadas.
- **Posibilidad de recuperación:** Capacidad del software de restablecer su nivel de ejecución y recobrar los datos directamente afectados en caso de avería.

- **La conformidad:** La capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares. [85]

**Usabilidad** (facilidad de uso): La capacidad del software ser comprendido, aprendido, utilizado y de ser amigable para el usuario, cuando se emplee bajo las condiciones especificadas. [86]

- **La comprensibilidad:** La capacidad del producto de software para permitirle al usuario entender si el software es conveniente, y cómo puede usarse para las tareas particulares y condiciones de uso.
- **La cognoscibilidad:** La capacidad del producto del software para permitirle al usuario aprender su aplicación.
- **La operabilidad:** La capacidad del producto del software para permitirle al usuario operarlo y controlarlo.
- **La atracción:** La capacidad del producto del software de ser amigable para el usuario.

**Eficiencia:** La capacidad del software para proporcionar la requerida ejecución, en relación con la cantidad de recursos usados, bajo las condiciones declaradas. [87]

- **El crono-comportamiento:** La capacidad del software para proporcionar una respuesta apropiada y los tiempos de procesamiento.
- **La utilización de recursos:** La capacidad del software para usar los recursos apropiados en un plazo de tiempo adecuado cuando el software realiza su función bajo las condiciones declaradas.
- **La conformidad:** La capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la eficiencia.

**Mantenibilidad:** La capacidad del software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptación del software a los cambios en el ambiente, y en los requisitos y las especificaciones funcionales. [88]

- **La diagnósticabilidad:** La capacidad del producto del software ser diagnosticado para detectar deficiencias o causas de fracasos o averías en el software, o para las partes para ser modificadas para ser identificadas. [89]
- **La mutabilidad:** La capacidad del producto del software para permitir llevar a cabo una modificación especificada.
- **La estabilidad:** La capacidad del software para minimizar los efectos inesperados de las modificaciones del software.
- **La contrastabilidad:** La capacidad del producto del software para permitir validar el software modificado.
- **La conformidad:** La capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la mantenibilidad.

**Portabilidad:** La capacidad de software ser transferido de un ambiente a otro. [90]

- **La adaptabilidad:** La capacidad del software de ser modificado para los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con este propósito para el software considerado.
- **La instalabilidad.:** La capacidad del software ser instalado en un ambiente especificado.
- **La coexistencia.:** La capacidad del software para coexistir con otro software independiente en un ambiente común que comparte los recursos comunes.
- **La remplazabilidad;** La capacidad del software ser usado en lugar de otro software especificado en el ambiente de ese software.
- **La conformidad:** La capacidad del software para adherirse a las normas que se le apliquen o convenciones que se relacionan con la portabilidad.

### 4.2.2 Responsabilidades

El equipo de SQA es el responsable de realizar las actividades y entregables mencionadas en la sección anterior. Como parte de las actividades se revisarán los productos que se consideren relevantes para la calidad del producto y del proceso. A continuación se identifican esos productos y el responsable de cada

producto, que será la referencia en caso de que dicho producto necesite correcciones ver formato en la figura 3.4, [Se sugieren algunos productos como ejemplo, deberá agregar a esta lista todos los productos que no estén y que se revisarán en el transcurso del proyecto.]

Actividad	Rol del Responsable	Responsable
Especificación de requerimientos		
Modelo de casos de usos		
Modelo de Diseño		
Interfase de Usuario		
Estándar de Implementación		
·		
·		

**Figura 3.4** Propuesta de plantilla para identificar las responsabilidades por roles.

**5 Documentación:** Especificar los documentos que dirigen el desarrollo del proyecto y que deberán ser revisados como parte de las actividades de aseguramiento de la calidad. Para cada documento se propone utilizar el siguiente formato: se indica el objetivo del documento, la plantilla, norma y/o estándar que se usa para elaborar el documento y el contenido mínimo que debe tener dicho documento. (Ver figura 3.5)

**5.1 Documentación Mínima Requerida:** Listar todos los documentos que deben ser revisados, se propone los siguientes documentos aunque usted debe agregar los que crea necesarios que no aparezcan aquí.

- Especificación de Requerimientos
- Diseño del Sistema y Descripción de la Arquitectura
- Documentación de Usuario
- Plan de Configuración
- Plan de Proyecto



Objetivo	Estándar	Contenido Mínimo
[Se especifica el objetivo de dicho documento, el papel que representa dentro del desarrollo del producto final ]	[Detallar la plantilla, norma o estándar que se sigue para la realización de dicho documento]	[Especificar el contenido mínimo de dicho documento]

**Figura 3.5** Propuesta de plantilla para la documentación mínima requerida.

**6 Estándares y Métricas:** Identificar los estándares utilizados en el proyecto (documentación, codificación, prueba, diseño, etc.). Para cada entregable que se revise se debe indicar el estándar o plantilla que se utiliza para su creación. Detallar las métricas de calidad que se aplicarán al producto y proceso.

**6.1 Estándares:** Para cada entregable que se entregue en las distintas etapas de desarrollo del software se debe identificar el estándar que se utiliza.

**6.2 Métricas:** Se describe para cada propiedad de calidad identificada, las métricas que se utilizarán para medir dicha propiedad de calidad.

### 7 Revisiones

**7.1 Descripción:** En esta sección se definen los tres tipos de revisiones (Revisión de Calidad de Producto, Revisión de entregas semanales y Revisión Técnica Formal), sus objetivos y mecanismos.

#### 7.1.1 Revisión de Calidad de Producto

**Objetivo:** Revisar los productos que se definieron como claves para asegurar la calidad. Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.

**Mecanismo:** Se revisan los productos para verificar que cumplan con los estándares (sección 6) y con los objetivos de calidad.

Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones.

Como salida se obtiene el Informe de revisión, que contiene todas las desviaciones o defectos encontrados durante la revisión. Este informe debe ser distribuido a los responsables del producto y se debe asegurar que ellos son conscientes de las desviaciones o discrepancias encontradas y de las acciones correctivas que deben realizar.

### 7.1.2 Revisión de Entregas Semanales

*Objetivo:* Revisar los artefactos que se generan semanalmente. Detectar desviaciones en los objetivos de calidad definidos e informar a los responsables para que sean corregidas.

*Mecanismo:* Se revisan los productos para verificar que cumplan con los estándares definidos anteriormente y con los objetivos de calidad.

Como salida se obtiene el Informe de revisión.

### 7.1.3 Revisión Técnica Formal (RTF)

El equipo de SQA participa en las RTF como una forma de supervisar la calidad del producto.

*Objetivo:* Descubrir errores en la función, la lógica ó la implementación de cualquier producto del software, verificar que satisface sus especificaciones, que se ajusta a los estándares establecidos, señalando las posibles desviaciones detectadas.

*Mecanismo:* Es un proceso de revisión riguroso, su objetivo es llegar a detectar lo antes posible, los posibles defectos o desviaciones en los productos que se van generando a lo largo del desarrollo. Por esta característica se adopta esta práctica para productos que son de especial importancia.

En la reunión participan el administrador de calidad e integrantes del equipo de desarrollo.

Se debe convocar a la reunión formalmente a los involucrados, informar del material que ellos deben preparar por adelantado, llevar una lista de preguntas y dudas que surgen del estudio del producto a ser revisado.

Como salida se obtiene el Informe de RTF.

*7.2 Requerimientos Mínimos:* En esta sección se detallan los elementos mínimos que deberán ser revisados.

En la lista siguiente aparecen algunos elementos detallados. Agregue los que se consideren necesarios, indicándose para cada uno los tipos de revisiones que se aplicarán.

Especificación de Requerimientos (Modelo de Casos de Uso, Requerimientos Suplementarios)

Modelo de Diseño y Descripción de la Arquitectura

Plan del Proyecto

Plan de Gestión de Configuración

Diseño vs. Especificación de Requerimientos

Implementación vs. Diseño

Verificación vs. Especificación de Requerimientos

**7.3 Agenda:** En esta sección se detallan todas las revisiones de calidad que se realizarán durante todo el proyecto, organizadas por fase e iteración. Es decir en una misma fase se puede realizar varias iteraciones y se deben tener en cuenta las cuatro fases de desarrollo Inicio, Elaboración, Construcción y Transición. A continuación se muestra la propuesta del formato que se debe seguir para cada una de las revisiones. (Ver figura 3.6)

*Fase I – Inicio*

Iteración I

Entregable	Fase de realización	Revisión	Tipo de revisión
[Nombre del entregable o producto a revisar]	[Fase, iteración y semana en que se debe realizar la versión del producto a revisar]	[Semana, si se quiere también la fecha, en la que se realizará la revisión del entregable o producto]	[Tipo de revisión que se realizará: Evaluación de la calidad de los productos, Revisar el ajuste al proceso o Revisión Técnica Formal ]

**Figura 3.6** Propuesta de plantilla para la agenda del Plan de SQA.

**8 Verificación:** Identificar las propiedades de calidad que no se hallan verificado, y aquellas que no estén implícitas en la sección de planificación de la calidad. Para cada una de esas propiedades definir y detallar las pruebas a realizar para asegurar su cumplimiento.

*9 Reporte de problemas y acciones correctivas:* Indicar de que manera y a quienes se reportarán los problemas encontrados en los productos y en el proceso. Indicar como se realizará el seguimiento de los problemas. Indicar de quien es la responsabilidad de tomar acciones correctivas para cada problema. Es decir se debe detallar cual es el procedimiento para información y seguimiento de errores entre el equipo de SQA y el equipo de desarrollo.

*9.1 Procedimiento para información y seguimiento de errores:* detallar el funcionamiento de dicho procedimiento; una vez que se realicen las diferentes revisiones con sus respectivas pruebas y se detecten los errores estos son anotados en la plantilla de las no conformidades, dichas no conformidades se le entregan al equipo de desarrollo del software, los mismo corrigen los errores y se vuelve a entregar el producto al equipo de aseguramiento de la calidad para una nueva revisión, y se sigue los mismos pasos hasta que el producto revisado no tenga errores. Para informar al equipo de desarrollo de software de las posibles no conformidades detectadas se utiliza una reunión de revisión.

*10 Herramientas, técnicas y metodologías:* Especificar las herramientas que se han utilizado para la realización de este plan.

*11 Gestión de configuración:* El objetivo del SQA en esta área es asegurar que se realizan las actividades de gestión de configuración establecidas en el Plan de GCS y que se realizan tal como están establecidas en el proceso. Incluye identificar los métodos para la gestión de la configuración.

*11.1 Métodos para la gestión de configuración:* En esta sección se indican los métodos que se utilizarán para mantener, almacenar, asegurar y documentar las versiones controladas identificadas en las fases de desarrollo, lo cual será definido en conjunto con el responsable de gestión de la configuración del software.

Se indican los métodos que se utilizarán para proteger el almacenamiento adecuado de los programas, documentación, etc., así como también la prevención de acceso sin autorización, daño, etc., lo cual será definido en conjunto con el responsable de gestión de la configuración del software.

*11.2 Actividades para asegurar la calidad:* Se deberán explicitar en esta sección las actividades que llevará a cabo el administrador de calidad para asegurar que la gestión de configuración se realiza como lo establece el proceso definido para el proyecto.

Se proponen algunas actividades mínimas que se deberían realizar:

- Asegurar que se generó la Línea Base del proyecto en el momento establecido en el modelo de proceso.
- Asegurar que la Línea Base del proyecto generada es correcta.
- Se verifica periódicamente que el Responsable de gestión de la configuración del software mantiene apropiadamente el control de la línea base, así como el registro completo de cambios para requerimientos, diseño, código, verificación y documentación.
- Se monitorean los procedimientos del Comité de Control de Cambios para verificar que son efectivamente realizados como se especificaron en el Plan de configuración.
- Para cada actividad que se indique, se debe especificar como se realizará y si corresponde se debe incluir en la agenda de revisiones, además de indicar en esta sección cuando se realizarán dichas actividades.

*12 Gestión de riesgos:* El objetivo del SQA en esta área es asegurar que la gestión de riesgos se realiza y se hace de forma correcta.

*12.1 Métodos para la gestión de riesgos:* Se indican los métodos y procedimientos que serán utilizados para identificar, monitorear y controlar los riesgos identificados en el proyecto.

*12.2 Actividades para asegurar la calidad:* Para evaluar que se cumple el objetivo planteado se deberán realizar actividades de SQA, se proponen siguientes actividades:

- Asegurar que se realice una evaluación de riesgos en el proyecto.
- Asegurar que la evaluación de riesgos se realice correctamente, lo cual implica:
- Que se evalúen riesgos en todas las áreas del proyecto y por personas diferentes.
- Que esa evaluación de riesgos se haga en forma individual y luego se reúnan los riesgos detectados para determinar el impacto de cada riesgo, con todos los integrantes del equipo de trabajo.
- Que se prioricen los riesgos.
- Que se planifique su eliminación, mitigación y/o contingencia.
- Que se incorporen los riesgos a los planes del proyecto para su seguimiento.

- Asegurar que se realice el seguimiento de los riesgos.
- Para cada actividad que se indique, se debe especificar como se realizará y si corresponde se debe incluir en la agenda de revisiones, agenda, de este plan, o bien indicar en esta sección cuando se realizarán dichas actividades.
- **Elaborar el Plan de Pruebas**

El analista de prueba identifica las pruebas a realizar en cada fase así como los mecanismos de pruebas y el tipo de prueba. El administrador de la calidad elabora el plan de prueba, para el cual se propone la siguiente estructura:

*1 Propósito:* Redactar el propósito que tiene el Plan de Pruebas.

*2 Acrónimos y Abreviaturas:* Definir las diferentes abreviaturas o acrónimos que se utilizaran en el plan para una mejor comprensión de todas las personas que lo lean.

*3 Referencias:* Referenciar la bibliografía que se utilice para realizar el plan.

*4 Organización:* El analista de prueba es el responsable de realizar el plan de pruebas. Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. Se debe especificar el alcance de la prueba para cada fase; se propone para que las pruebas se comporten de la siguiente forma:

*4.1 Inicio:* El desarrollo del prototipo exploratorio de demostración; se debe realizar pruebas a la documentación y no está de más ejecutar algunas pruebas exploratorias al primer prototipo de interfaz de usuario que se obtenga.

*4.2 Elaboración:* Probar los componentes ejecutables que se han implementado y que deben corresponderse con la arquitectura básica de la aplicación.

*4.3 Construcción:* Desarrollar los casos de prueba y procedimientos de prueba para hacerlos.

*4.4 Transición:* El producto en su entorno de operación por lo que es probado por usuarios reales.

*5. Niveles de Prueba:* Definir en que nivel se realizará las pruebas teniendo en cuenta las diferentes fases de desarrollo y los artefactos que se probarán. La prueba es aplicada para diferentes tipos de objetivos, en

diferentes escenarios o niveles de trabajo. Se proponen tener en cuenta los siguientes niveles de pruebas:

**Prueba de Desarrollador.** Es la prueba diseñada e implementada por el equipo de desarrollo. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad en algunos casos pueden ejecutar pruebas de integración.

**Prueba Independiente:** Es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores. El objetivo de estas pruebas es proporcionar una perspectiva diferente y en un ambiente más rico que los desarrolladores.

**Prueba de Unidad:** Es la prueba enfocada a los elementos más pequeño del software. Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera. La prueba de unidad siempre está orientada a caja blanca.

**Prueba de Integración:** Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación.

**Prueba de Sistema:** Son las pruebas que se hacen cuando el software está funcionando como un todo.

**Prueba de Aceptación:** Prueba de aceptación del usuario es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

*6 Métodos de Prueba:* Definir los métodos de prueba que se utilizarán; se propone utilizar los métodos de *caja negra* y de *caja blanca*.

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

La prueba de la **caja blanca** del software se comprueba los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el

estado del programa en varios puntos para determinar si el estado real coinciden con el esperado o mencionado.

*7 Agenda:* En esta sección se detallan todas las pruebas que se realizarán durante todo el proyecto, organizadas por fase e iteración. Es decir en una misma fase se puede realizar varias pruebas y se deben tener en cuenta las cuatro fases de desarrollo Inicio, Elaboración, Construcción y Transición. Se propone el siguiente formato para elaborar la agenda de pruebas. (Ver figura 3.7)

*Fase I – Inicio*

Iteración I

Entregable	Fase de prueba	Nivel de prueba	Método de prueba	Tipo de prueba
[Nombre del entregable o producto a probar]	[Fase, iteración y semana en que se debe probar el producto]	[Nivel en que se realizará la prueba]	[Método de prueba que se utilizará ]	[Nombre de la prueba que se realizará ]

**Figura 3.7** Propuesta de plantilla para la agenda del Plan de Pruebas.

### 3.2 4.2 Construcción de la Calidad

Las actividades de construcción de la calidad sirven para evitar la introducción de errores y las mismas culminan con un informe de actividades constructivas. En este procedimiento se propone como una de las principales actividades constructivas de tipo **organizativas**:

- Asegurar que se lleven a cabo todos los procesos vinculados al desarrollo de software de una forma eficiente, para ello se define como procesos en cada fase de desarrollo los flujos de trabajo que propone RUP realizar en cada una de estas fases, agregando el proceso de gestión de riesgos y también con la excepción que los procesos de Gestión de la Configuración, Pruebas y Gestión de Riesgos se deben llevar acabo en la cuatros fases de desarrollo.



Se propone como actividad constructiva de tipo **técnica**:

- Asegurar que se utilicen estándares de desarrollo de software para cada una de las fases, para la fase de inicio: estándares de documentación, fase de elaboración: estándares de diseño, fase de construcción: estándares de implementación y por último en la fase de transición: estándares de documentación técnica para el usuario. Además el equipo de SQA debe guiar y asesorar al equipo de desarrollo de software para la selección y la correcta aplicación de técnicas y herramientas de ingeniería de software.

Se propone como actividad constructiva de tipo **humana**:

- Formar y capacitar a los integrantes del equipo del proyecto así como motivarlos a que se comprometan en un alto grado con la calidad del producto. Cada una de las actividades mencionadas culmina con un informe de revisión.
- **Realizar Informe de Actividades Constructivas**

Este informe tiene como objetivo identificar las diferentes actividades constructivas que se realicen a lo largo de todo el ciclo de desarrollo del software para asegurar la calidad del producto estas actividades son responsabilidad del equipo de aseguramiento de la calidad para ello se identifica la actividad, el tipo de actividad, el responsable de realizarla, el alcance de la actividad es decir a quién o para quién va dirigida, los resultados esperados de dicha actividad y por ultimo una vez concluida la actividad se elabora un resumen de los resultados alcanzados después de desarrollar la actividad.

### **3.2 4.3 Supervisión de la Calidad**

Para llegar a establecer los requisitos establecidos, se necesita supervisar y corregir el trabajo que se está realizando pero se propone no esperar a la última fase de desarrollo para corregir los errores sino realizar auditorias durante todas las fases de desarrollo, así como revisiones de entregas semanales para supervisar que los diferentes artefactos se están realizando en la fecha trazada y cumplen con los principios definidos, también se proponen hacer una revisión de calidad del producto. Al culminar cada fase de desarrollo, se propone al equipo de desarrollo de software realizar una Revisión Técnica Formal y esta revisión se realiza a los artefactos que influyen notablemente en la calidad del producto. Si el

artefacto no es aprobado no se puede pasar a la siguiente fase y se vuelve hacer la revisión hasta que el producto presente la calidad requerida. Todas estas revisiones culminan con un informe de revisión.

Es importante aclarar que la RTF la realiza el equipo de desarrollo de software, el equipo de SQA sólo participa en este tipo de revisión para supervisar la calidad del producto y debe anotar las desviaciones que se detecten e informales.

- **Realizar Informe de Revisión**

El informe de revisión se identifica el artefacto que se está revisando el responsable de la elaboración de dicho artefacto y la fecha de entrega, además se llena la columna de observaciones donde el administrador de la calidad inserta todas las dificultades que detecta en dicho artefacto para luego cuando se realice la RTF con el equipo de desarrollo de software se analice si el artefacto es aprobado o no y el mismo puede determinar si se comienza con al siguiente fase o si se debe seguir en la fase actual hasta que dicho artefacto esté aprobado.

*Para la redacción del informe de revisión se propone el siguiente formato:*

**Proyecto:** Nombre del proyecto al que pertenece el registro.

**RTF:** Se debe marcar con una X esta columna si la revisión corresponde a una RTF, en caso contrario se deja sin marcar esta columna y se especifica el tipo de revisión que se está realizando.

**Fase:** Fase en la que se realiza la revisión.

**Fecha de entrega:** se especifica la fecha en que fue entregado el producto.

**Iteración:** Número de la iteración dentro de la fase en la que se realiza la revisión.

**Fecha:** Fecha de la reunión de análisis de los resultados de la revisión.

**Producto:** Nombre del artefacto o producto.

**Aprobado:** Se marca con una cruz si el producto o el artefacto son aprobados en la RTF.

**Reprobado:** Se marca con una cruz si el producto o el artefacto son reprobados en la RTF.

**Responsable:** El nombre de la persona del equipo de desarrollo que es el responsable del cumplimiento de que dicho artefacto se construya.

**Observaciones:** Revisión correspondiente a la revisión que se está registrando, además de otra información que se considere útil reflejar en el registro.

- **Realizar Listas de Chequeo**

Es responsabilidad del analista de prueba elaborar las listas de chequeo pero el probador es quien la realiza; la misma tiene como objetivo chequear la calidad de un determinado producto, y comprobar si los objetivos definidos para cada fase se están cumpliendo satisfactoriamente; se debe utilizar para chequear los productos que determinen de una forma importante la calidad del software. (Ver plantilla de Lista de chequeo en el anexo 6)

- **Diseño de Casos de Pruebas**

Para el diseño de los casos de pruebas de caja negra se siguió la plantilla que utiliza actualmente el proyecto de calidad del software de la Facultad 7.

*Diseños de casos de pruebas (caja negra)*

Es responsabilidad del diseñador de pruebas elaborar los casos de pruebas pero el probador ejecuta todos aquellos casos diseñados y llena la plantilla de diseño de casos de pruebas, en el caso de que se estén realizando pruebas de caja negra por cada caso de uso que exista debe existir un caso de prueba conformados por las posibles clases validas e inválidas que proponga el diseñador de pruebas. (Ver plantilla de diseño de casos de pruebas en el anexo 8) Se propone el siguiente formato para diseñar los casos de prueba de caja negra.

*Descripción General*

[Se hace una descripción general de los elementos que serán cubiertas por este caso de prueba y qué se medirá de cada uno de ellos y de qué manera.]

*Objetivos generales*

[Se especifican cuales serán los objetivos principales del caso de pruebas así como la relación que se establece con otros casos de prueba]

*Lista de elementos a evaluar (CPR)*

[Todos los elementos que se someterán a prueba, elemento de interfaz de Windows, requisito funcional o no funcional, mitigación de un riesgo, u tipo de prueba específico, etc.]

### *Recursos y condiciones necesarios*

[Se listan todos los recursos que se prevé serán necesarios para el correcto desarrollo del diseño y la aplicación de las pruebas]

### *Datos de los diseñadores y probadores*

[Se adicionan todas personas responsables del diseño y de la aplicación del caso de prueba]

< Nombre 1 > < Identificador 1 >

### *CPR 1*

[Se coloca el nombre del elemento que se someterá a prueba y fue listado]

### *Descripción*

[Descripción del elemento que se va a evaluar y los principales objetivos que se persiguen específicamente]

### *Flujo Central*

[Se describe la vía para llegar hasta el elemento que se desea evaluar]

### *Condiciones de Ejecución*

[Se describe lo más detalladamente posible las condiciones necesarias para que la prueba pueda llevarse a cabo con el objetivo de reproducirla tantas veces como se desee y el resultado sea coherente]

### *Iteraciones de entradas para la aplicación de las pruebas*

Ver la plantilla de diseño de casos de prueba en el anexo 8

### *Evaluación de la prueba del elemento CPR1*

[Estado del caso de prueba (CPR), que puede ser por ejemplo: propuesta, pendiente de evaluación, realizada satisfactoria, etc.]

### *Evaluación de los Resultados de la Prueba del elemento CPR1*

[Resultado de la prueba (CPR)]

Se propone utilizar las siguientes categorías para priorizar o calificar defectos:

*Crítico:* Denota una función inutilizable que causa un término anormal o una falla general, o cuando un cambio en un área de la aplicación causa un problema en otra parte.

**Severo:** Una función no actúa como fue requerido o diseñado, o un objeto de interfaz no trabaja como se muestra.

**Advertencia:** La función trabaja, pero no tan rápidamente como es esperado, o no se ajusta a las normas y convenciones.

**Cosmético:** No crítico para el funcionamiento del sistema: palabras con mala ortografía, formateo incorrecto, mensajes de error vagos o confusos o advertencias. ]

### *Registro de defectos y dificultades detectados*

Ver plantilla de no conformidades en el anexo 7.

### *Documentación de los Resultados*

[Cuando el esfuerzo de prueba esté terminado, documente los resultados y mediciones. Identifique cualquier discrepancia entre el plan y la puesta en práctica real y describa adecuadamente como aquellas discrepancias fueron manejadas]. (Ver anexo 8 plantilla de diseño de casos de prueba)

### *Diseño de Casos de Pruebas (caja blanca)*

Es responsabilidad del diseñador de pruebas elaborar los casos de pruebas pero el probador ejecuta todos aquellos casos de pruebas. Cuando se estén realizando las pruebas de caja blanca se propone seguir la técnica del camino básico, la misma permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Primeramente se debe calcular la complejidad ciclomática que es la que brinda la cantidad de casos de pruebas que se deben realizar, luego se sacan los caminos básicos que ofrece los principales caminos a seguir y los parámetros que se deben probar y por último se ejecuta los casos de prueba. Para el diseño de los casos de prueba se propone el siguiente formato:

### *Descripción General*

[Se hace una descripción general de los elementos que serán cubiertas por este caso de prueba y que se medirá de cada uno de ellos y de que manera.]

### *Objetivos generales*

[Se especifican cuales serán los objetivos principales del caso de pruebas así como la relación que ese establece con otros casos de prueba]

*Datos de los diseñadores y probadores* [Se adicionan todas personas responsables del diseño y de la aplicación del caso de prueba]

< Nombre 1 > < Identificador 1 >

*CP1* [Se pasa los parámetros que se probaron en este caso de prueba]

*Respuesta Esperada* [La respuesta después de haber sido ejecutado el caso de prueba]

*Registro de defectos y dificultades detectados* [Ver plantilla de no conformidades en el anexo 7]

### **3.2.4.4 Evaluación de la calidad del producto**

La evaluación de la calidad del producto se realizará una vez culminada la fase de transición. Si el producto es aceptado por el usuario final, el equipo de SQA se reúne para dar una evaluación final de la calidad del software desarrollado culminando con esta actividad el procedimiento propuesto.

El equipo de SQA para dar la evaluación final debe tener en cuenta los resultados de las diferentes auditorías realizadas en cada fase de desarrollo, si el software desarrollado satisface las necesidades del cliente, y si además cumple todas sus expectativas, seguidamente se analiza si el producto tiene concordancia con los requerimientos explícitamente establecidas y con los estándares de desarrollo definidos, así como el ajuste del producto con los procesos involucrados en el desarrollo de software.

Seguidamente se propone utilizar listas de chequeo para evaluar el cumplimiento del software con la sub-características de las características de calidad especificadas en el plan de SQA, se utilizará la plantilla de listas de chequeo que aparece en el anexo 6. Finalmente el equipo de SQA se reúne y teniendo en cuenta los aspectos mencionadas anteriormente, se evalúa la calidad del producto de baja, media o alta, esta actividad concluye con un informe de evaluación.

- **Realizar Informe de Evaluación**

Para el informe de revisión se propone el siguiente formato:

**Nombre del proyecto:** Especificar el nombre del proyecto

**Número del equipo de desarrollo:** Especificar la cantidad de integrantes del equipo de desarrollo

**Tipo de Software:** especificar el tipo de software desarrollado si es una aplicación de escritorio, aplicación Web entre otras.

**Cantidad de módulos:** Cantidad de módulos que tiene el software.

**Características de los clientes:** Especificar el tipo del cliente y sus características.

**Cantidad de revisiones:** La cantidad de revisiones por fases.

**Evaluación:** La evaluación de la calidad del producto otorgada, la misma será de baja, media o alta.

**Resumen:** Se hace un resumen de forma general señalando los aspectos que se analizaron para evaluar la calidad del producto, así como los resultados satisfactorios que haya logrado el equipo del proyecto.

**Observaciones:** Las principales recomendaciones que se le hace al equipo de desarrollo, y de existir dificultades leves que no influyen en la calidad del producto se deben detallar de forma precisa.

### 3.2.5 Medidas

Para este procedimiento se propone medir:

- Cantidad total de revisiones por fases
- Cantidad total de revisiones que dieron como resultado la aprobación del artefacto revisado.
- Cantidad total de revisiones que dieron como resultado la reprobación del artefacto revisado.
- Tipos de artefactos con el mayor número de reprobaciones en las diferentes fases de desarrollo.
- Tipos de artefactos con el menor número de reprobaciones en las diferentes fases de desarrollo.

### 3.2.6 Documentos referenciados

Anexo 2, 3,4 y 5 (Principales actividades de SQA por pasos para las diferentes fases de desarrollo).

Anexo 6 Plantilla de Listas de Chequeos

Anexo7 Plantilla de No Conformidades

Anexo7 Plantilla de Diseño de Casos de Pruebas

### 3.3 Conclusiones parciales

- Para realizar el procedimiento de aseguramiento de la calidad, se debe cumplir con todas las actividades descritas en el Plan de aseguramiento de la calidad, teniendo en cuenta todas las plantillas a realizar.
- El equipo de aseguramiento de la calidad debe estar en constante intercambio con el equipo de desarrollo del software.
- El procedimiento propuesto tiene concordancia directa con la definición que describe el modelo CMMI V 1.2 de aseguramiento de la calidad del software; teniendo en cuenta que este lo define como un medio planificado y sistemático.



### **Conclusiones**

- Se fundamentó la importancia del Aseguramiento de la Calidad del Software como una actividad de protección, que debe estar presente en todas las etapas de desarrollo de cualquier producto; y debe existir en cualquier organización que esté encaminada a obtener una producción con la calidad deseada.
- Se elaboró un procedimiento que tiene concordancia directa con la definición que describe el modelo CMMI V 1.2 de aseguramiento de la calidad del software, y además está basado en la metodología RUP.
- En el procedimiento elaborado se realizan actividades de protección, construcción y planificación de la calidad. Las mismas permiten que se cumplan con los estándares, procedimientos, y metodologías definidas. Además se sigue un control sistemático de los diferentes artefactos generados en las fases de desarrollo de software planificando las pruebas que se realizarán a lo largo de todo el ciclo de vida del software.

### **Recomendaciones**

- Aplicar y dar seguimiento al procedimiento propuesto en los proyectos de la Facultad 7.
- Hacer un estudio de las herramientas Case (Rational Clear Case, Rational Clear Quest) para utilizarlas en el control de cambios y versiones de los principales artefactos generados en el procedimiento.
- Realizar una capacitación previa a las personas que van a desempeñar los roles propuestos, donde se profundice en el estudio del modelo CMMI V1.2.
- Adaptar este procedimiento a otras metodologías de desarrollo de software como por ejemplo la metodología XP.
- Hacer un estudio sobre las métricas internas, externas y las métricas de calidad en uso para utilizarlas en la medición de las propiedades de calidad del software.
- Estudiar procedimientos para realizar pruebas de caja blanca, mediante la técnica del camino básico.

### Referencia Bibliográfica

1. Serrano, M.A. Gestión de procesos de Software. Introducción a CMMI y SCAMPI. 2007 [Cited; Available from: \\10.128.21.32\docs\Presentaciones\presentaciones Informática 2007.
2. Ídem a la referencia 1.
3. Ídem a la referencia 1.
4. Delgado, R., Herramienta de Apoyo a la Gestión de la Configuración. Propuesta Arquitectónica. 2006, Instituto Superior Politécnico José Antonio Echeverría: Ciudad de la Habana.
5. Ruiz, A. (2007) La UCI y la Industria Cubana de SW. [Cited; Available from: \\10.128.21.32\docs\Presentaciones\presentaciones Informática 2007.
6. Ídem a la referencia 5.
7. Ídem a la referencia 5.
8. Ídem a la referencia 5.
9. Santos, J.M., Evolución histórica de la calidad del software en Cuba. 2007: Ciudad de la Habana.
10. Ídem a la referencia 9.
11. Toirán, Y., Organización en Procesos del Departamento de Detención a Estudiantes Extranjeros. 2006, Universidad Central de Las Villas: Villa Clara.
12. CMMIv1.2. CMMI para el desarrollo. Mejora de procesos para mejores productos. 2006
13. ISO/IEC. Technology- Software product quality. 2005. [Cited; Available from: <http://www.iso.vilspa.esa.es>
14. Ídem a la referencia 11.
15. Ídem a la referencia 11.
16. Ídem a la referencia 11.
17. Ídem a la referencia 11.
18. Ídem a la referencia 11.
19. Ídem a la referencia 11.
20. Pressman, R.S., Ingeniería de Software, un enfoque práctico.5 ed. 2001, Madrid.
21. IEEE. Guide for Software Quality Assurance Planning. 1900
22. Ídem a la referencia 11.
23. Ídem a la referencia 11.

24. Ídem a la referencia 12.
25. Ídem a la referencia 11.
26. Ídem a la referencia 11.
27. Ídem a la referencia 11.
28. Ídem a la referencia 11.
29. Ídem a la referencia 11.
30. Ídem a la referencia 12.
31. Ídem a la referencia 11.
32. Ídem a la referencia 12.
33. Ídem a la referencia 20.
34. Ídem a la referencia 20.
35. Ídem a la referencia 20.
36. Ídem a la referencia 20.
37. Ídem a la referencia 20.
38. Ídem a la referencia 20.
39. Ídem a la referencia 20.
40. Ídem a la referencia 13.
41. Ídem a la referencia 12.
42. Ídem a la referencia 20.
43. Ídem a la referencia 20.
44. IEEE. Standard for Software Quality Assurance Plans. 2006
45. Ídem a la referencia 20.
46. Ídem a la referencia 20.
47. Ídem a la referencia 20.
48. Ídem a la referencia 20.
49. Ídem a la referencia 44.
50. Ídem a la referencia 20.
51. Ídem a la referencia 20.
52. Ídem a la referencia 20.

53. Colectivos de Profesores, Conferencia de Flujo de Prueba. 2006. [Cited; Available from: [http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias\\_IS2\\_05-06](http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias_IS2_05-06)
54. Ídem a la referencia 53.
55. Ídem a la referencia 53.
56. Ídem a la referencia 53.
57. Ídem a la referencia 53.
58. Ídem a la referencia 11.
59. Ídem a la referencia 11.
60. Ídem a la referencia 11.
61. Ídem a la referencia 11.
62. Ídem a la referencia 20.
63. Ídem a la referencia 20.
64. Ídem a la referencia 20.
65. Ídem a la referencia 20.
66. Ídem a la referencia 20.
67. Jacobson, I.B., G. y Rumbaugh, J, El Proceso Unificado de Desarrollo de software. 2000.
68. Colectivos de Profesores. Introducción a la Ingeniería de Software I. 2006. [Cited; Available from: [http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias\\_IS1\\_05-06](http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias_IS1_05-06)
69. Ídem a la referencia 68.
70. Ídem a la referencia 20.
71. Ídem a la referencia 20.
72. Ídem a la referencia 20.
73. Ídem a la referencia 5.
74. Ídem a la referencia 5.
75. Ídem a la referencia 5.
76. Ídem a la referencia 5.
77. Ídem a la referencia 5.
78. Ídem a la referencia 5.
79. Ídem a la referencia 5.

80. Ídem a la referencia 5.
81. Ídem a la referencia 5.
82. Solís, E., El proceso de producción de SW en la facultad 7. 2007.
83. NC ISO9126. Tecnología de la Información. Características de calidad y métricas del software. 2003 [cited; Available from: \\10.128.21.32\docs\Calidad Software\normas calidad\iso\NC ISO 9126.
84. Ídem a la referencia 83.
85. Ídem a la referencia 83.
86. Ídem a la referencia 83.
87. Ídem a la referencia 83.
88. Ídem a la referencia 83.
89. Ídem a la referencia 83.
90. Ídem a la referencia 83.

### **Bibliografía**

1. Álvarez, J. Calidad en la Ingeniería de Software. 2002
2. Benedi, A. Extracto del libro en formato digital Calidad tradicional y de SW.2006, Universidad Técnica Federico Santa María.
3. Basulto, L. Estrategias de calidad para las PYMES de desarrollo de SW. 2006
4. Calderón, Y. Metodología para la gestión de tareas investigativas en la producción de software en la UCI, 2007
5. CMMIv1.2. CMMI para el desarrollo. Mejora de procesos para mejores productos. 2006
6. Colectivos de Profesores. Introducción a la Ingeniería de Software I. 2006. [Cited; Available from: [http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias\\_IS1\\_05-06](http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias_IS1_05-06)
7. Colectivos de Profesores, Conferencia de Flujo de Prueba. 2006. [Cited; Available from: [http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias\\_IS2\\_05-06](http://teleformacion.uci.cu/mod/resource/view.php?id=10817&subdir=/Conferencias_IS2_05-06)
8. Delgado, R., Herramienta de Apoyo a la Gestión de la Configuración. Propuesta Arquitectónica. 2006, Instituto Superior Politécnico José Antonio Echeverría: Ciudad de la Habana.
9. Estévez, D. Taller de Gestión de software., Plan de SQA. 2007
10. Fernández, A. Ingeniería de SW, Métrica 3. 2007. [Cited; Available from: <http://www.csi.map.es/csi/metrica3/index.html>
11. Granja, C. Ingeniería de calidad. Métricas del software. 2007
12. IEEE. Guide for Software Quality Assurance Planning. 1900
13. IEEE. Standard for Software Quality Assurance Plans. 2006
14. ISO/IEC. Technology Software product quality. 2005. [Cited; Available from: <http://www.iso.vilspa.esa.es>
15. Jacobson, I.B., G. y Rumbaugh, J, El Proceso Unificado de Desarrollo de software. 2000.
16. NC ISO9126. Tecnología de la Información. Características de calidad y métricas del software. 2003 [cited; Available from: \\10.128.21.32\docs\Calidad Software\normas calidad\iso\NC ISO 9126.
17. Pressman, R.S., Ingeniería de Software, un enfoque práctico.5 ed. 2001, Madrid.

18. Quintana, N y Otero, R. Mejora de Procesos de Software y Coste de la Calidad 2005. [Cited; Available from: <http://WWW.navegapelis.net>.
19. Rivero, C. Certificación de procesos de desarrollo de SW. 2004
20. Ruiz, A. La UCI y la Industria Cubana de SW. 2007. [Cited; Available from: \\10.128.21.32\docs\Presentaciones\presentaciones Informática 2007.
- 21 Santos, J.M., Evolución histórica de la calidad del software en Cuba. 2007: Ciudad de la Habana.
22. Serrano, M.A. Gestión de procesos de Software. Introducción a CMMI y SCAMPI. 2007 [Cited; Available from: \\10.128.21.32\docs\Presentaciones\presentaciones Informática 2007.
23. Solís, E., El proceso de producción de SW en la facultad 7. 2007.
24. Toirán, Y., Organización en Procesos del Departamento de Detención a Estudiantes Extranjeros. 2006, Universidad Central de Las Villas: Villa Clara.



**Anexos**

**Anexo 1** Relación de las preguntas aplicadas en la encuesta sobre el Aseguramiento de la Calidad en los Proyectos Productivos de la Facultad 7.

Nombre del proyecto \_\_\_\_\_ Facultad \_\_\_\_\_

Rol del encuestado: \_\_\_\_\_ Tamaño del equipo de desarrollo \_\_\_\_\_

Misión del proyecto \_\_\_\_\_

Objetivos del proyecto: \_\_\_\_\_

1. ¿Se dedican solo a la producción de software? Si \_\_\_ No \_\_\_ ¿Qué otras actividades realizan?

2. ¿Cuántos software han fabricados? \_\_\_\_\_

3. La calidad de los productos que fabrican es: E \_\_\_ B \_\_\_ R \_\_\_ M \_\_\_

4. ¿Qué usted entiende por calidad? \_\_\_\_\_

5. ¿Considera que es necesario garantizar la calidad de los productos fabricados? Si \_\_\_ No \_\_\_ ¿Por qué?

6. Sus clientes son: a. \_\_\_ Organizaciones Nacionales desarrolladoras de software

b. \_\_\_ Organizaciones Internacionales desarrolladoras de software

c. \_\_\_ Organizaciones Nacionales que necesitan software para desarrollarse

d. \_\_\_ Organizaciones Internacionales que necesitan software para desarrollarse

e. \_\_\_ Personas individuales

7. ¿Los clientes se sienten satisfechos con el servicio que brindan? Si \_\_\_ No \_\_\_ No sé \_\_\_

8. Para conocer a cerca de la satisfacción de los clientes lo hacen a través de:

a. \_\_\_ Encuestas

b. \_\_\_ Encuentros casuales

c. \_\_\_ Encuentros planificados

d. \_\_\_ Correo electrónico

e. \_\_\_ Vía Telefónica

f. \_\_\_ Visitas al cliente

g. \_\_\_ Por el servicio de Soporte de Software h. \_\_\_ Otros

10. ¿Utiliza en el proyecto los procesos TSP y PSP? Si \_\_\_ No \_\_\_ No sé \_\_\_

11. ¿Lleva a cabo en el proyecto la disciplina de planificación de las tareas? Si \_\_\_ No \_\_\_ No sé \_\_\_

12. ¿Lleva a cabo en el proyecto la disciplina de gestión de la configuración? Si \_\_\_ No \_\_\_ No sé \_\_\_

13. ¿Utiliza la metodología RUP? Si \_\_\_ No \_\_\_. Mencione otra metodología que utiliza \_\_\_\_\_

14. ¿En el proyecto los integrantes realizan las tareas a través de roles? Si \_\_\_ No \_\_\_ No sé \_\_\_

15. ¿Conoce algún modelo o estándar que se utiliza a nivel mundial para la gestión de la calidad del SW?: Si \_\_\_ No \_\_\_ No sé \_\_\_ Si su respuesta es si:

b. Menciónelo(s) \_\_\_\_\_

c. Utiliza algún(s) de estos modelo o estándares en el proyecto: Si \_\_\_ No \_\_\_ No sé \_\_\_

d. Menciónelo(s) \_\_\_\_\_

e. ¿Utilizan algún procedimiento para el aseguramiento de la calidad del SW? Si \_\_\_ No \_\_\_

Si su respuesta es No. Mencione cómo garantizan la calidad del software en el proyecto:

16. ¿El proyecto realiza actividades de control de la calidad internamente? Si \_\_\_ No \_\_\_ Si su respuesta es Si

a. Las tarea de control de la calidad las realizan:

\_\_\_ un equipo interno del proyecto \_\_\_ cada integrante individualmente

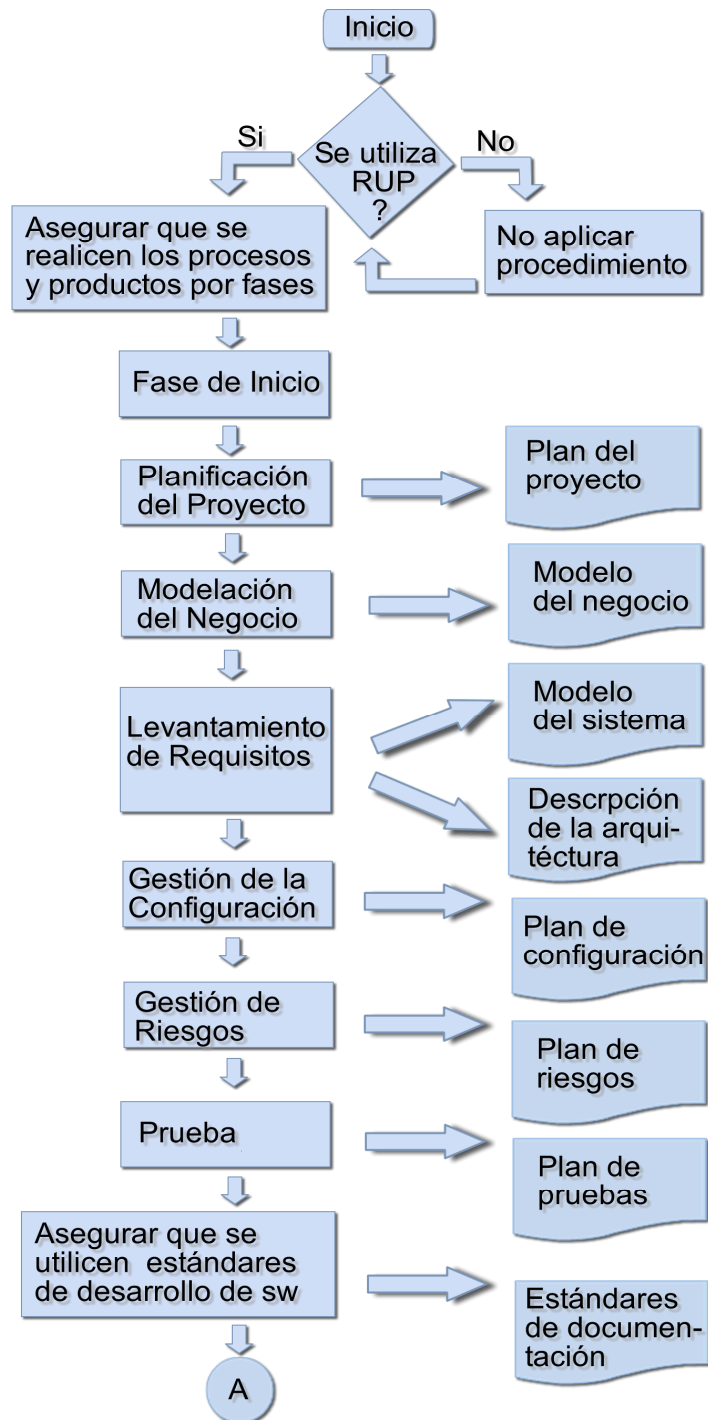
b. En el proyecto se realizan pruebas de: \_\_\_ Caja negra \_\_\_ Caja blanca

Si se realizan pruebas de caja blanca diga que métodos utiliza: \_\_\_ Prueba del camino básico \_\_\_ Prueba de condición \_\_\_ Prueba de flujo de dato \_\_\_ Prueba de bucles

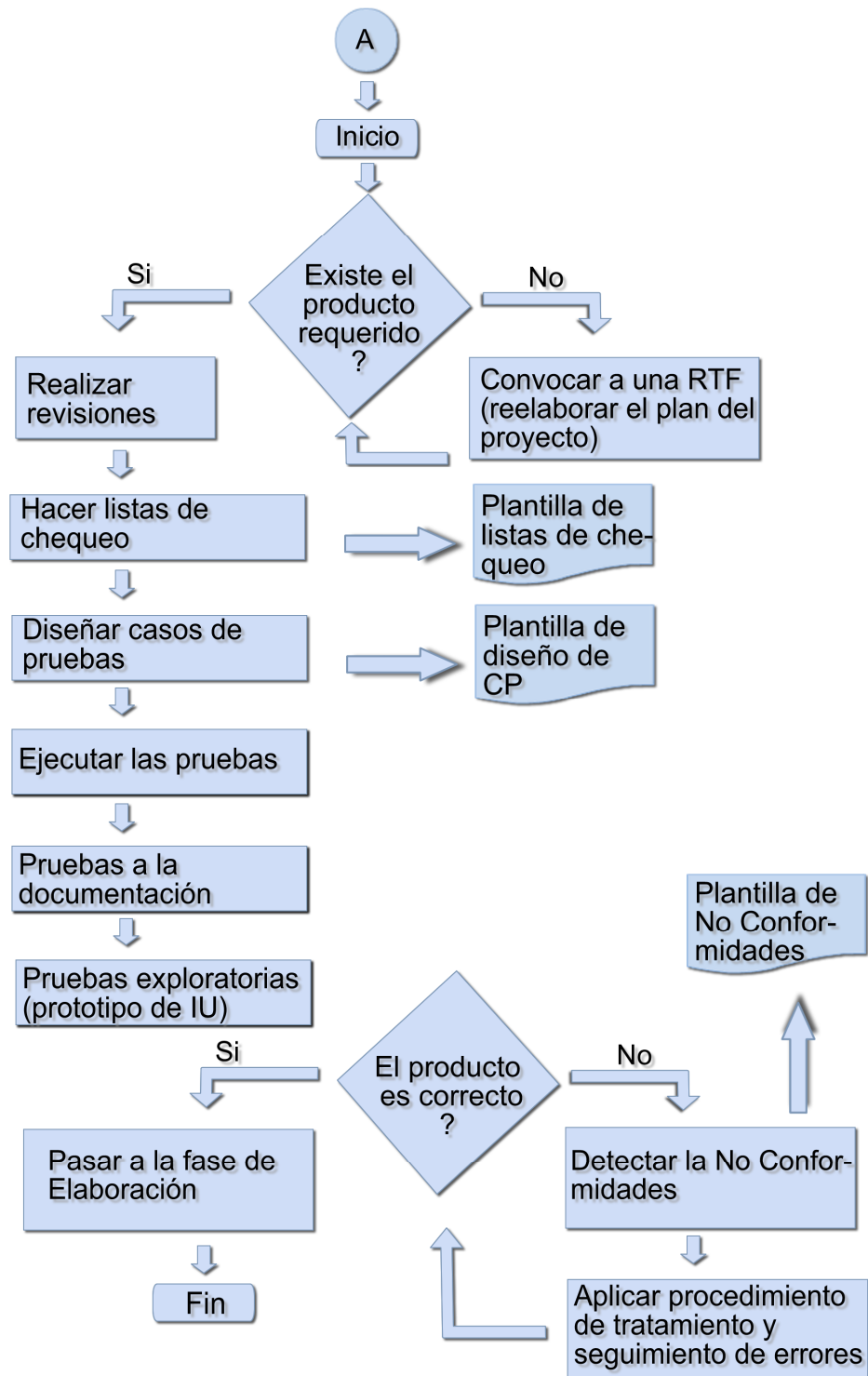
17. ¿El proyecto ha sido revisado por el equipo de calidad del software de la facultad o de la universidad? Si \_\_\_ No \_\_\_ No sé \_\_\_

Si su respuesta es Si Considera que estas revisiones son necesarias: Si \_\_\_ No \_\_\_

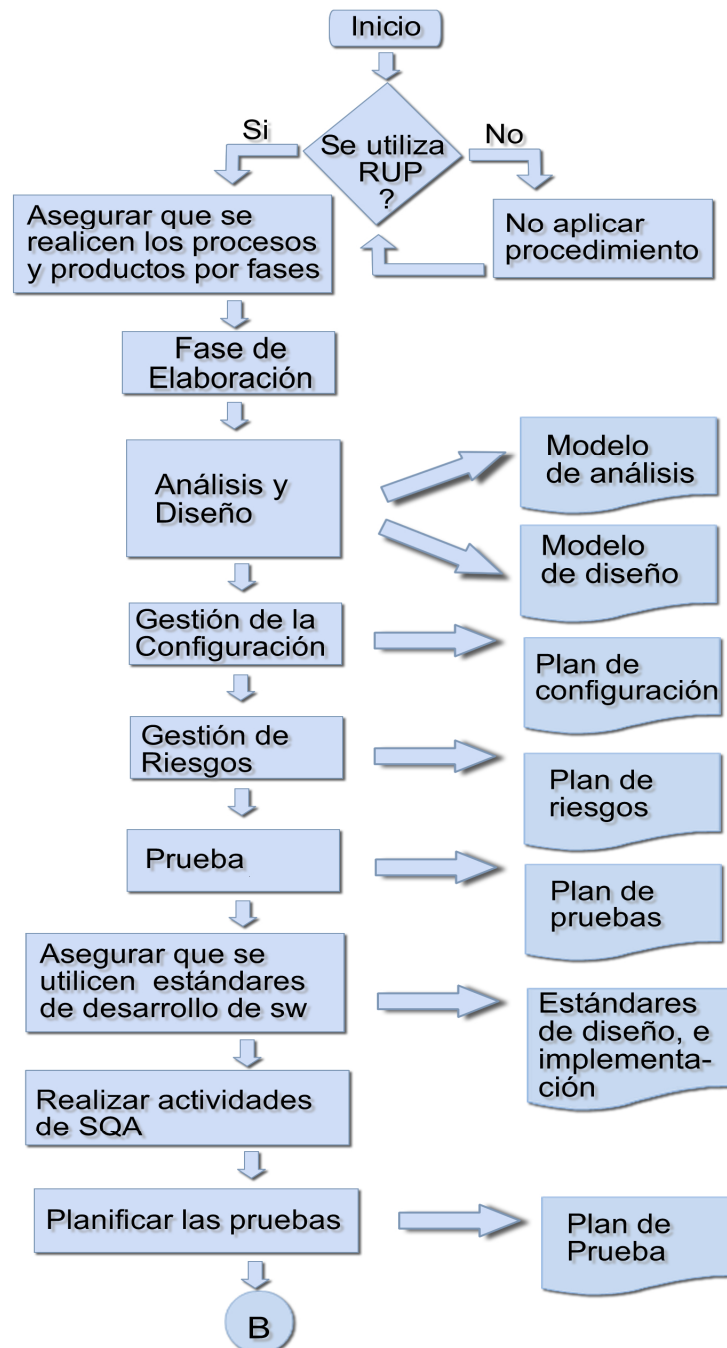
Anexo 2 Esquema 1: Principales actividades de SQA por pasos para la fase de desarrollo (inicio).

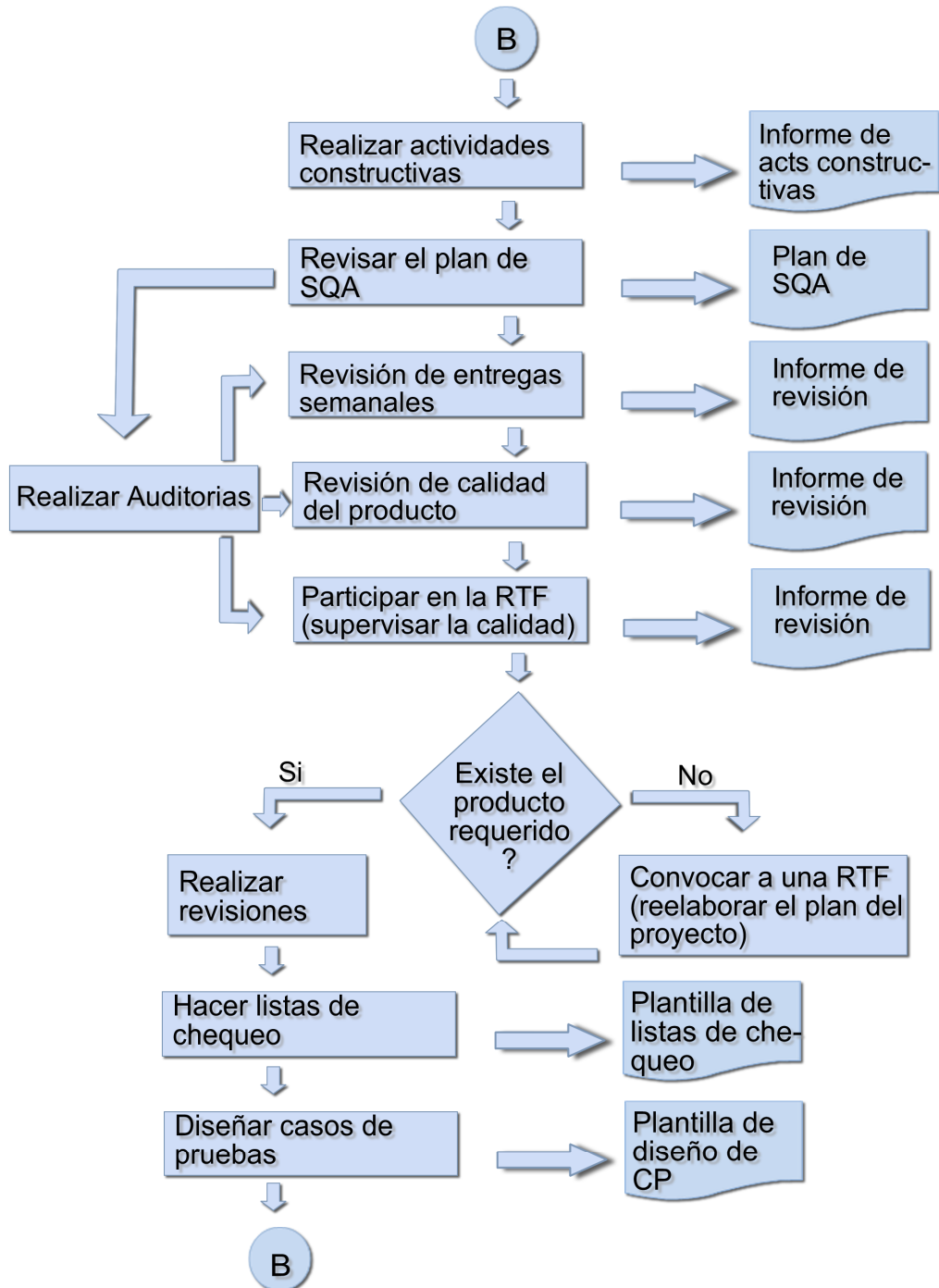


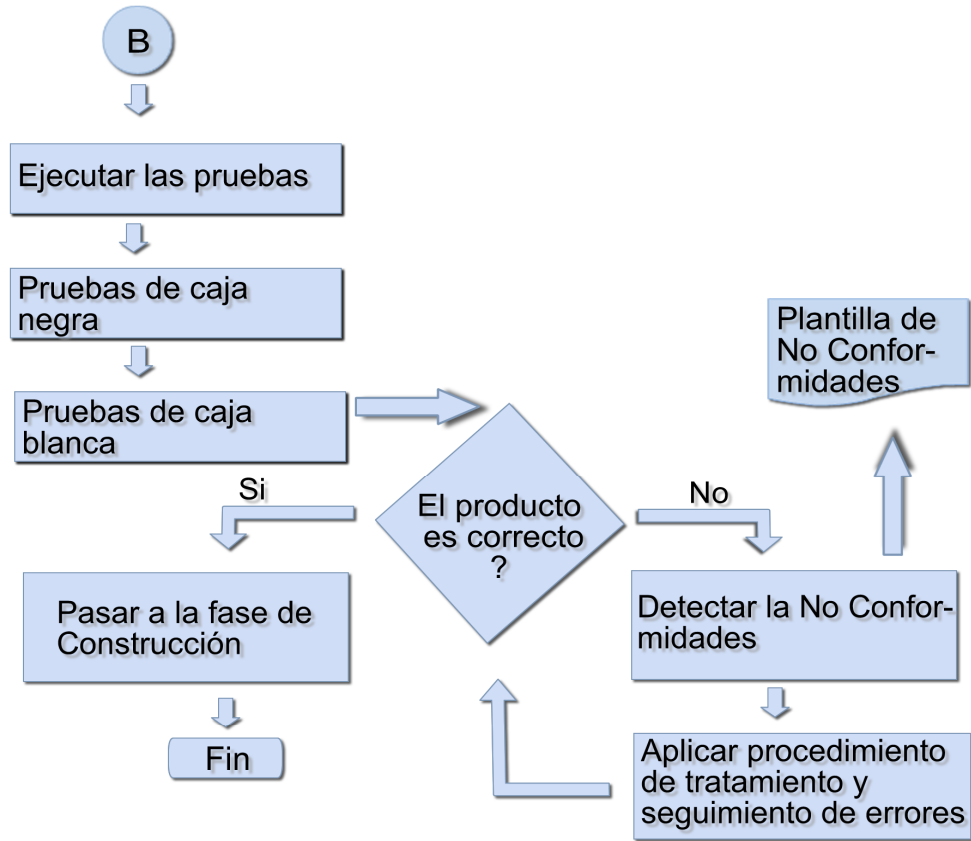




Anexo 3 Esquema 2: Principales actividades de SQA por pasos para la fase de desarrollo (elaboración).

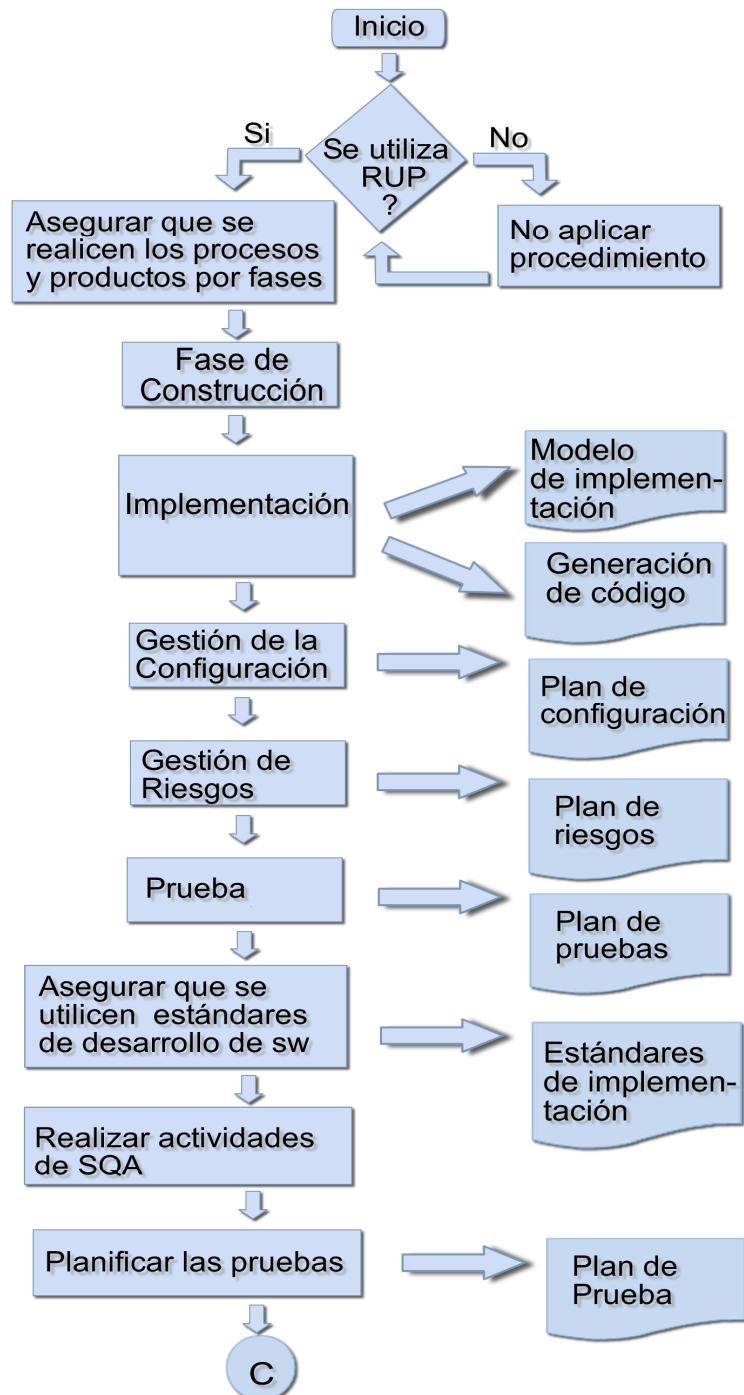


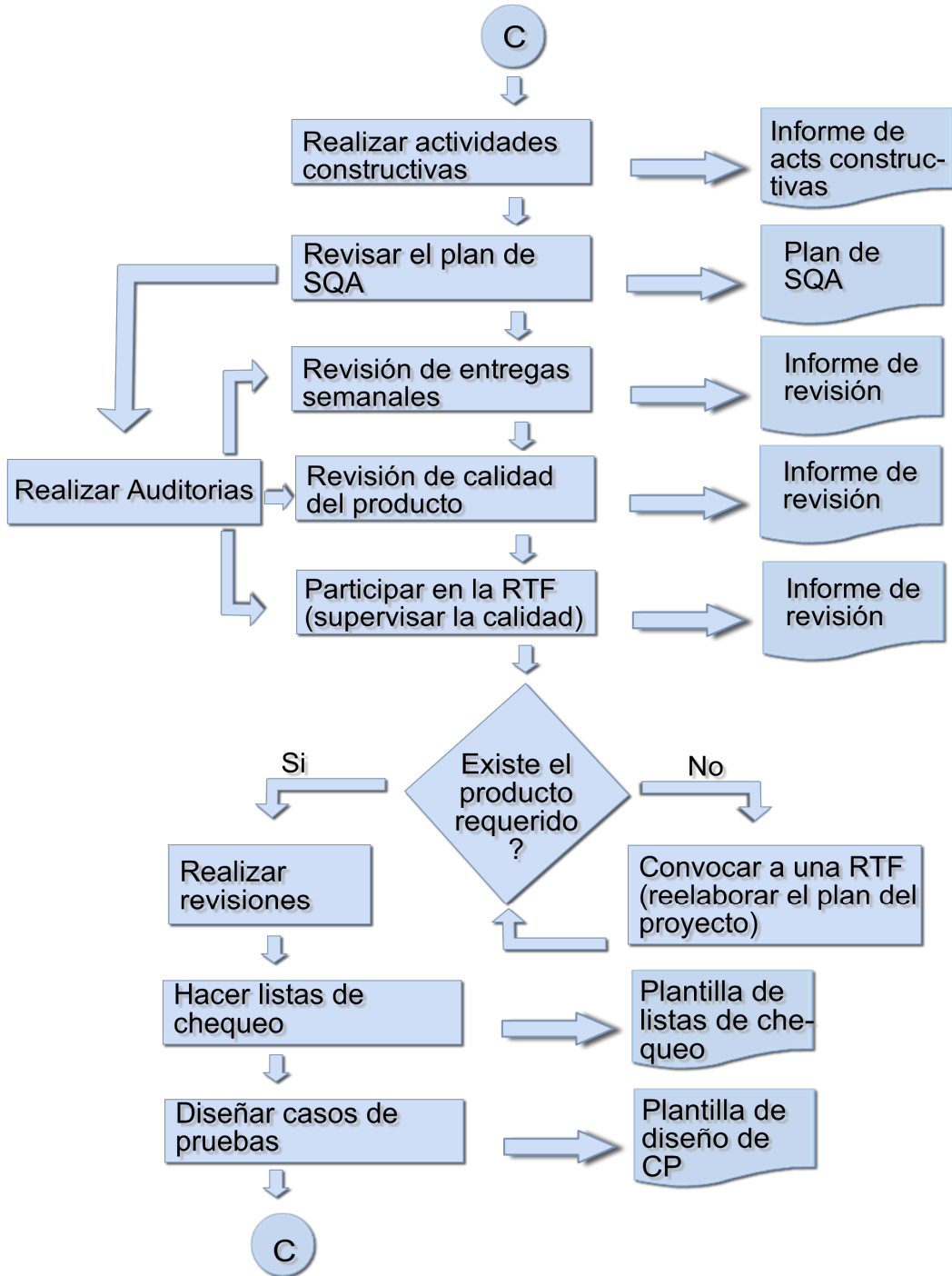


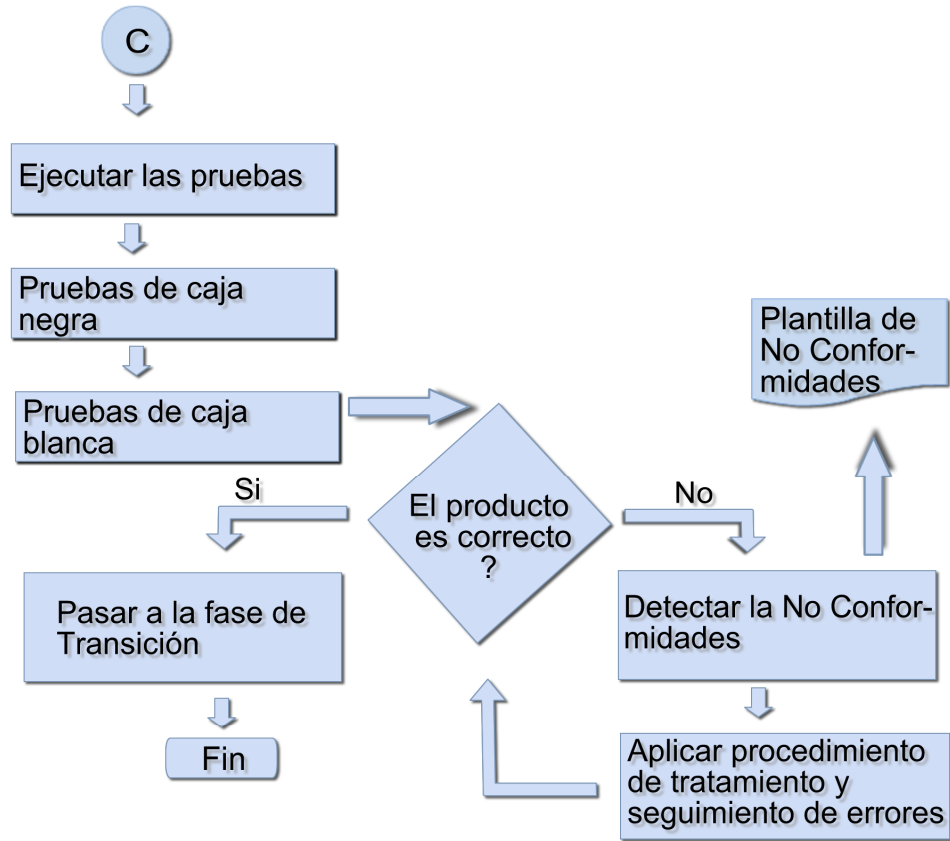




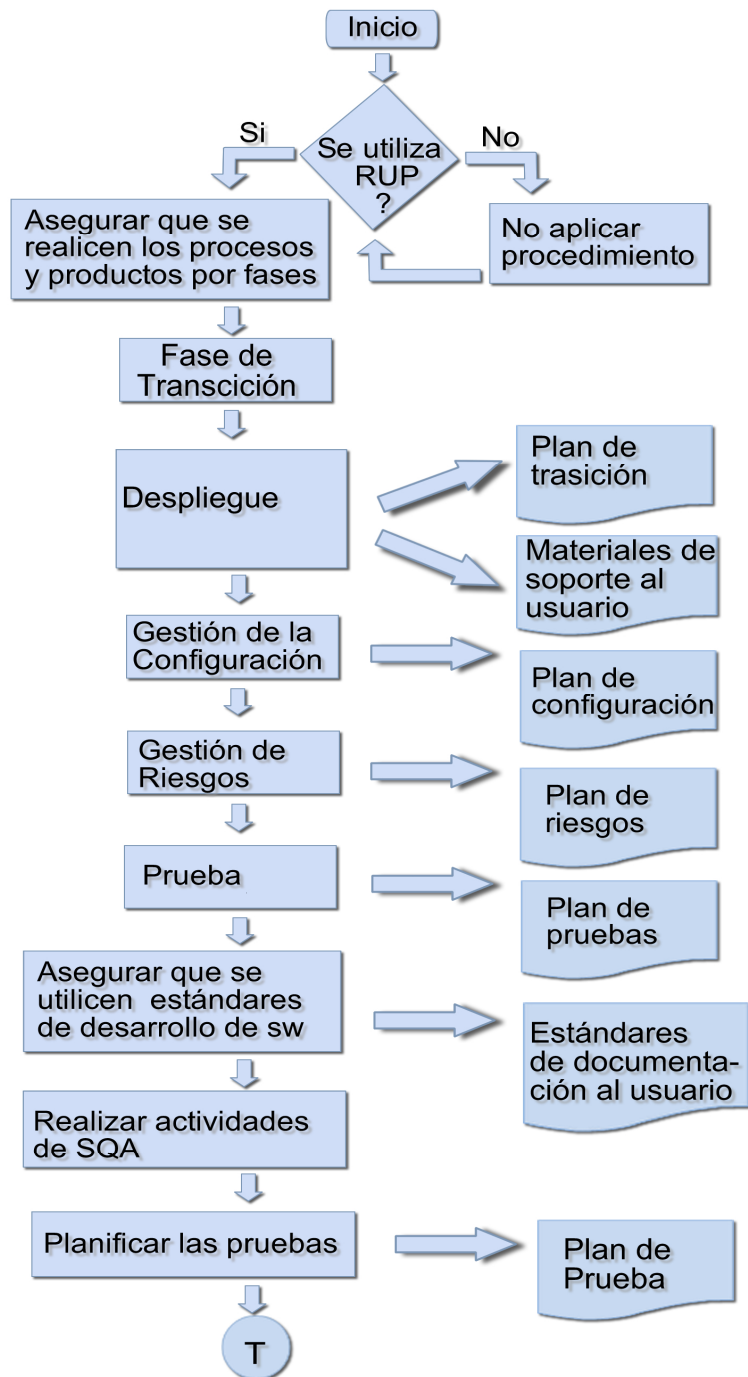
Anexo 4 Esquema 3: Principales actividades de SQA por pasos para la fase de desarrollo (construcción).

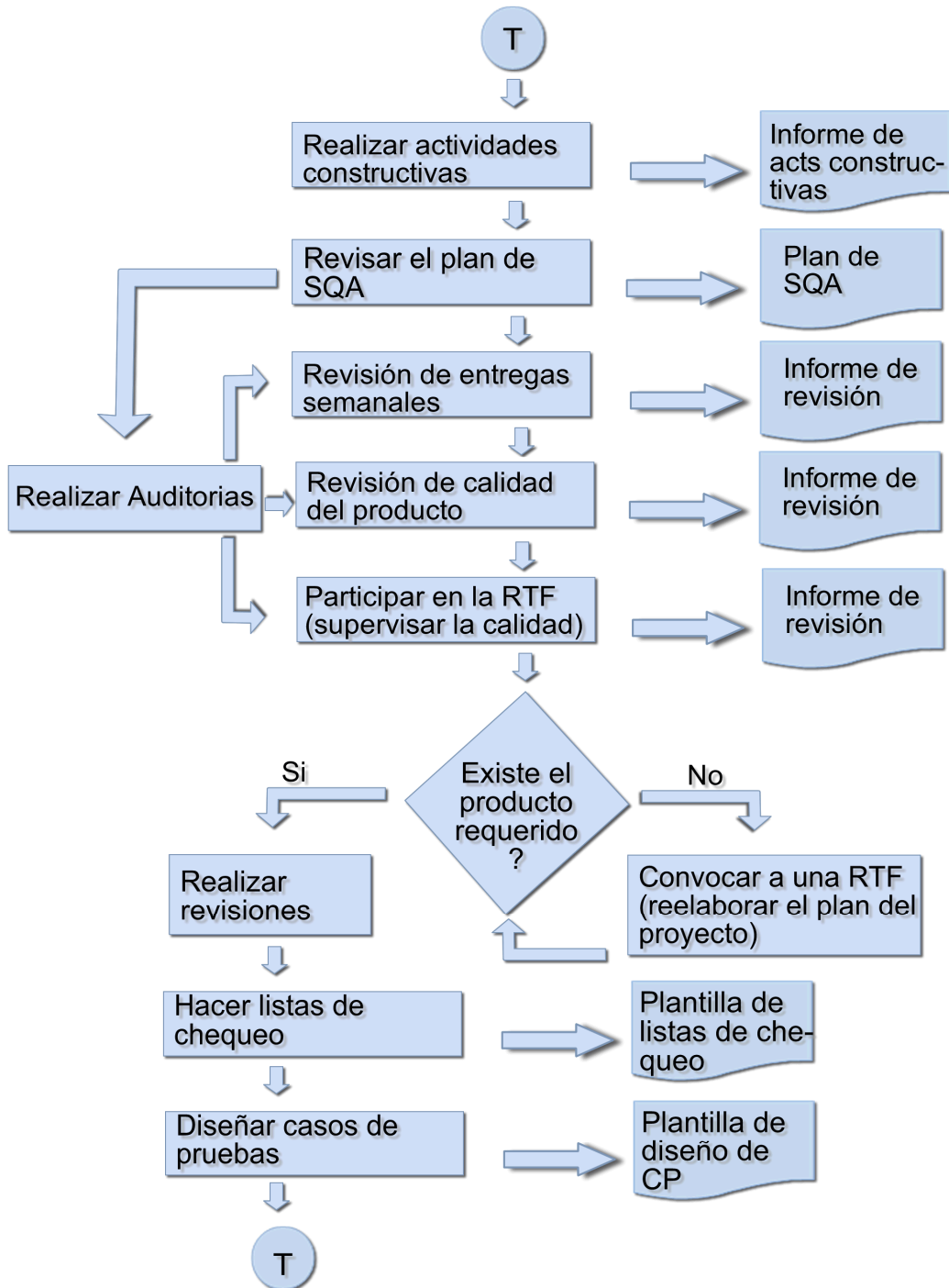


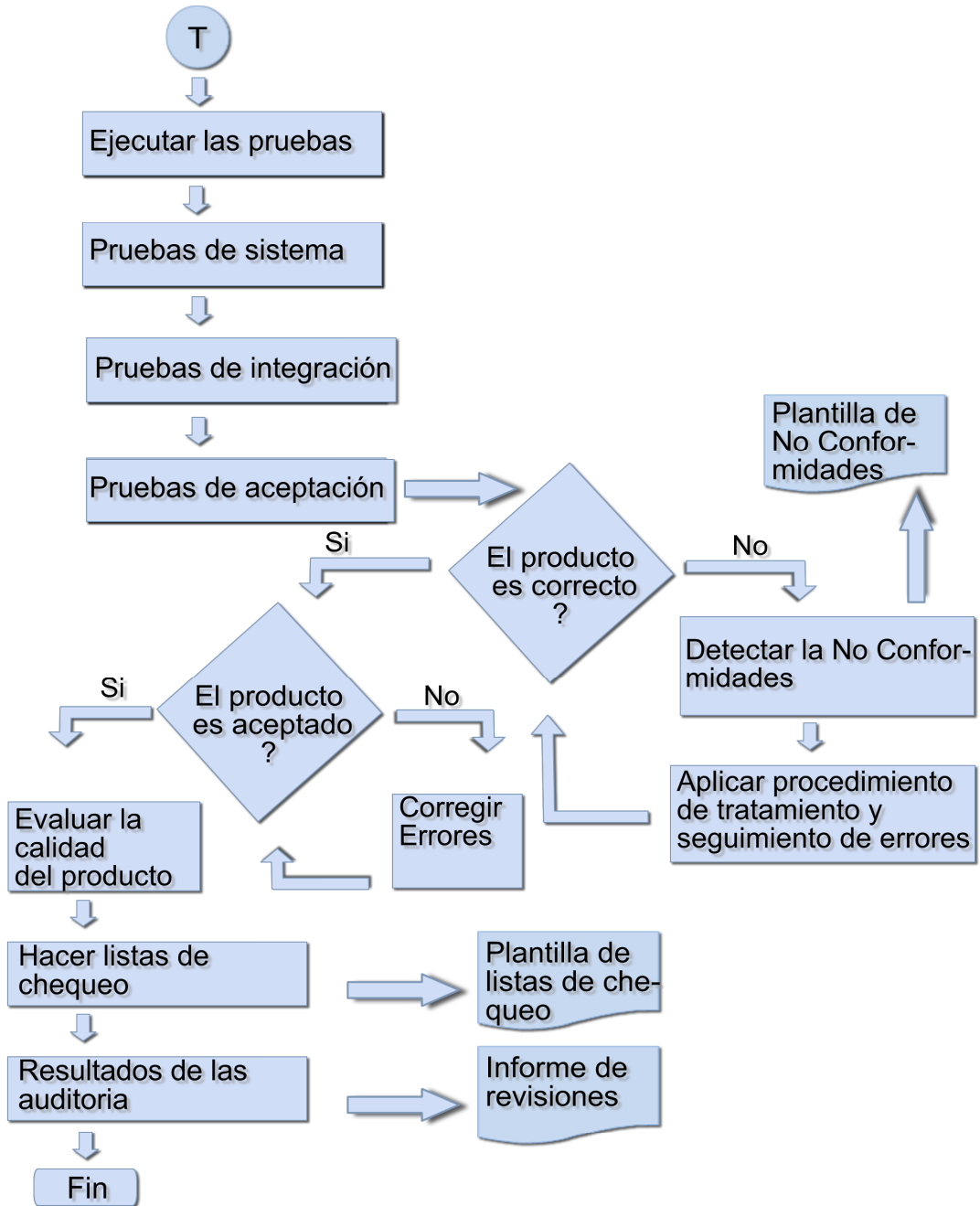




Anexo 5 Esquema 4: Principales actividades de SQA por pasos para la fase de desarrollo (transición).







**Anexo 6** Propuesta de Plantilla para Lista de Chequeo

Lista de Chequeos : Insertar la funcionalidad de la lista de chequeo				
Fecha: _____				
Forma: _____			Descripción: _____	
Analista: _____			Probador: _____	
Revisión : Insertar lo que se esta revisando				
Nivel	Actividad	E	NP	Comentario
Importancia del aspecto a evaluar	Lo que se desea chequear, se realiza en forma de pregunta	Evaluación	No procede	Es obligatorio en las respuestas negativas

**Las evaluaciones serán:**

- Malo - propiedad no disponible. (0)
- Satisfactorio - propiedad parcialmente disponible. (2)
- + Bien - propiedad disponible. (4)
- ++ Excelente - propiedad muy bien implementada. (5)

**El peso de la pregunta será:**

- !! Muy importante (5)
- ! Menos importante (3)

**Anexo 7** Propuesta Plantilla para las No Conformidades.

<b>Elemento de Configuración</b>	<b>No de la No-Conf.</b>	<b>Descripción de la no-conformidad</b>	<b>Aspecto correspondiente a su localización</b>	<b>Imp</b>	<b>Recom</b>
[Elemento de configuración del que se describe la No-conformidades, pueden existir varias sobre el mismo elemento]	1.	[Se especifica de manera clara y precisa en que consiste la no-conformidad, utilizando un lenguaje técnico y claro, no se debe hablar de manera general sino referirse a la palabra, al botón, al hipervínculo, comportamiento específico]	[Se especifica de manera clara y precisa en que lugar fue detectada la no-conformidad, de manera tal que sea fácil su hallazgo, de ser necesario se puede apoyar la explicación con imágenes anexadas]	[Se señala con una X si la no conformidad es Importante]	[Se señala con una X si la no conformidad es una recomendación]



**Anexo 8** Propuesta de Plantilla para el diseño de Casos de Prueba.

<b>Caso de Prueba</b>				
<b>Fecha</b>				
<b>Resultado</b>				
<b>Observaciones</b>				
<b>Responsables de las Pruebas</b>				
<b>Clases válidas</b>	<b>Clases inválidas</b>	<b>Resultado Esperado</b>	<b>Resultado de la prueba</b>	<b>Observaciones</b>
[Se describe detalladamente la entrada y la acción que evidenciará el comportamiento <b>correcto</b> del sistema]	[Se describe detalladamente la entrada y la acción que evidenciará el comportamiento <b>incorrecto</b> del sistema]	[Se describe detalladamente como se espera reaccione el sistema ante esta entrada específica, hasta esta columna se hace durante la etapa de diseño de la prueba]	[Durante la aplicación de la prueba se describe detalladamente que ha ocurrido realmente, cómo es que ha reaccionado el sistema, referenciado detalles, usando anexos]	[se recogen los detalles que se escapen o ajusten en las columnas anteriores]

### **Glosario de Términos**

Aseguramiento de la calidad: Actividad de prevención que se realiza para evitar introducir defectos en el producto que se desarrolla.

Calidad: La capacidad de cumplir con las características inherentes de un producto, componente de producto, o proceso para satisfacer las exigencias de los clientes.

Control de la calidad: Actividad que se realiza para detectar los defectos del producto que se desarrolla.

Complejidad Ciclomática: Es una métrica de software que proporciona una medición cuantitativa de la complejidad lógica de un programa, da el límite superior del número de pruebas que se deben realizar.

Doc: Documentación. La documentación que se genera en cada flujo de trabajo relacionada con el proceso de desarrollo de software.

Flujo de trabajo: Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

IU: Interfaz de usuario es el conjunto de componentes empleados por los usuarios para comunicarse con las computadoras.

Multimedia: es un sistema que utiliza más de un medio de comunicación al mismo tiempo en la presentación de la información, como el texto, la imagen, la animación, el vídeo y el sonido.

RTF: Revisión Técnica Formal es una actividad para asegurar la calidad del software.

Release: Versiones que se obtienen del producto.

SW: Software. Es un término genérico que designa al conjunto de programas de distinto tipo (sistema operativo y aplicaciones diversas) que hacen posible operar con la computadora.

Software Libre: Se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Stakeholder: 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, los clientes, los trabajadores, los proveedores, etc.