

Universidad de las Ciencias Informáticas

“Facultad 4”



Título: “Propuesta de un Plan de Aseguramiento de la
Calidad para el Proyecto Banco”

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

Autores:

Dayannis Estrada Duarte
Elizabeth Yaimy Romero Quevedo

Tutor:

Ing. Geiser Arcio Pérez Rivas

Ciudad de la Habana, junio/2008

“Año del 50 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autores:

Dayannis Estrada Duarte

Elizabeth Yaimy Romero Quevedo

Tutor:

Geiser Arcio Pérez Rivas

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado “Propuesta de un Plan de Aseguramiento de Calidad para el Proyecto Banco”, constituye una guía general para todos los miembros del proyecto responsables de garantizar la calidad, por tal motivo el Equipo de Aseguramiento de la Calidad en el proyecto considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

Totalmente: X

Parcialmente en un: %

La Universidad de las Ciencias Informáticas realiza incontables esfuerzos por lograr productos de alta calidad que satisfagan las expectativas de los clientes; pero a pesar de que el interés es de todos, en varias ocasiones los esfuerzos por garantizar la calidad aparecen cuando el producto se está terminando, de ahí que surjan inconformidades que pudieran ser evitadas desde una fase inicial en la vida del software.

En este Trabajo de Diploma se definen técnicas, metodologías, cronogramas de trabajos, roles, herramientas, artefactos involucrados, en fin, todo un estilo de trabajo que garantiza el aseguramiento de la calidad del producto en el Proyecto Banco desde la fase de inicio en el ciclo de vida del desarrollo del software hasta su liberación, respetando los objetivos básicos del proyecto este persigue la obtención de un producto que satisfaga enteramente las necesidades del cliente.

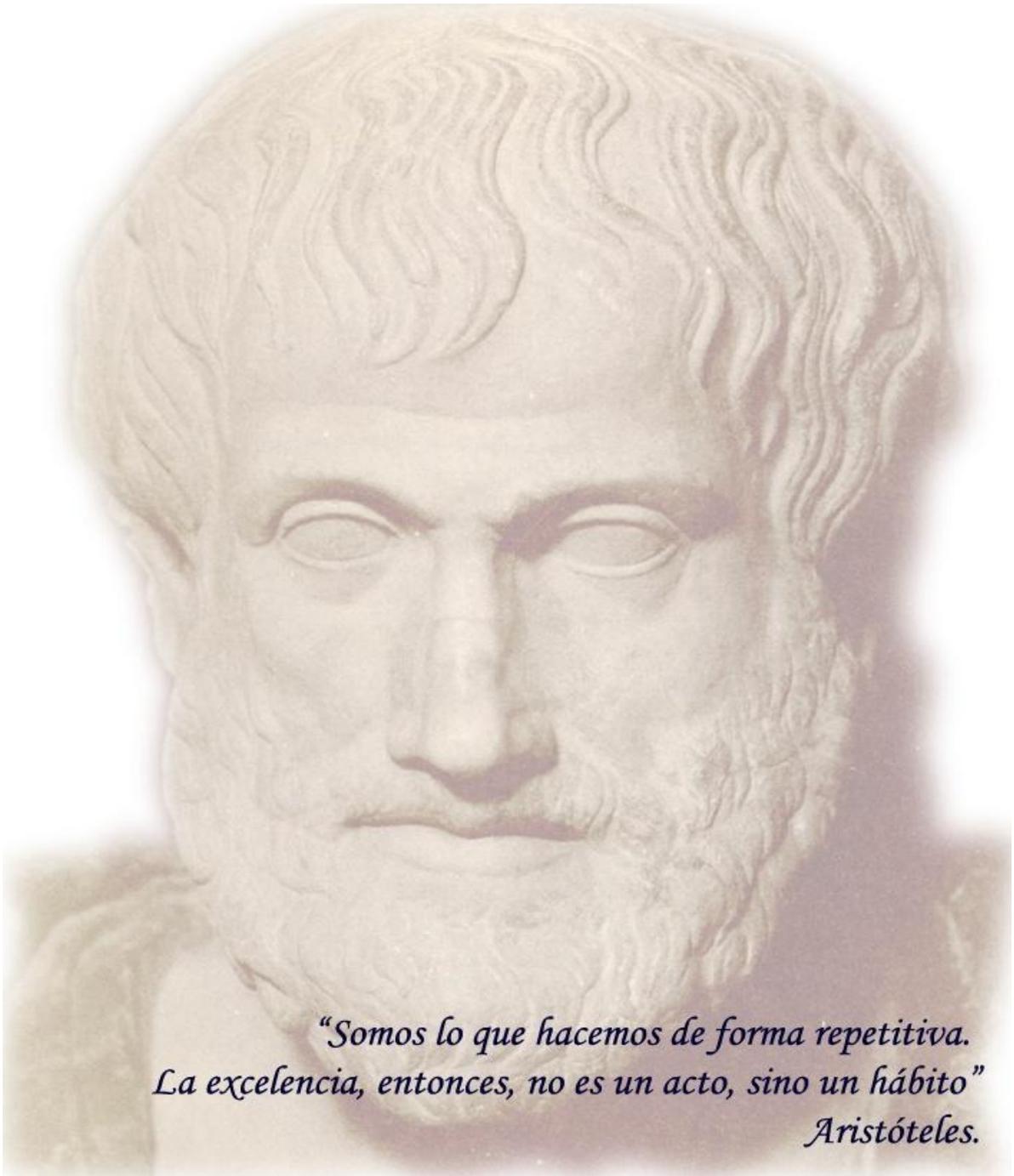
Y para que así conste, se firma la presente el primer día del mes de junio del año 2008.

Ing. Mayté Concepción Sigler

Representante de la entidad

Jefe del Equipo de Aseguramiento de la Calidad.

Cargo



*“Somos lo que hacemos de forma repetitiva.
La excelencia, entonces, no es un acto, sino un hábito”
Aristóteles.*

AGRADECIMIENTOS

A la Revolución por crear este Proyecto Futuro y permitirnos participar en él;

A todos los profesores que contribuyeron con nuestra formación a lo largo de la carrera;

A nuestro tutor por su ayuda e ideas...

A los miembros del Proyecto Banco, especialmente a Mayté y Lissette.

De Elizabeth:

Le doy muchas gracias a mi mamá por apoyarme en todo tiempo y por darme su amor incondicionalmente,

A mi papá,

A mi familia por estar presente en todos los momentos en que los necesité,

A mis tíos, tías, primas y primos por su cariño,

A todas mis amigas y amigos por entenderme aún cuando no lo merecía,

Por último y no por eso menos importante a dios pues por él soy lo que soy.

De Dayannis:

A mami por su ejemplo de responsabilidad, que ha sido mi meta a seguir durante mi formación profesional;

A papi por transmitirme optimismo y hacerme creer que todo va a salir bien;

A toda mi familia que me quiere y se preocupa por mí;

A Carmen, Thays y Roque por estar a mi lado durante estos años de universidad, brindándome su cariño y apoyo;

A mis amigos de siempre, los que me han acompañado, incondicionalmente, desde el inicio de la carrera.

De Elizabeth:

Dedico este trabajo a mi mamá porque siempre me dio su amor en los momentos más difíciles.

A mi abuelita querida.

A mi papá.

De Dayannis:

A mami y papi porque durante todos mis años de estudio confiaron plenamente en mí.

RESUMEN:

El aseguramiento de la calidad es uno de los temas actuales dentro de la disciplina de Ingeniería de Software y desde sus inicios ha sido considerado como uno de los más importantes. El establecimiento de un plan de aseguramiento de la calidad constituye la actividad fundamental para complementar dicho tema en un proyecto software.

Para los proyectos productivos, desarrollados en la Universidad de las Ciencias Informáticas, la elaboración e implementación de un Plan de Aseguramiento de la Calidad no es una tarea opcional, sino que representa una responsabilidad de obligatorio cumplimiento, establecida por el Departamento de Calidad del Software, que de no ser cumplida desacredita el buen funcionamiento del proyecto.

Con el desarrollo del presente trabajo se concibe la propuesta de un Plan de Aseguramiento de la Calidad para el Proyecto Banco y de esta forma estar acorde con las exigencias establecidas para dicho proyecto; siempre aclarando que ese no significa el principio básico de su elaboración, sino como tal su implementación. El plan pretende entregar una guía a seguir por los miembros del equipo de calidad del proyecto, asumiendo como objetivo esencial el hecho de minimizar el problema existente de obtener un software confiable que sustente el mayor grado de calidad posible.

Hacer uso de este plan facilita verificar que durante el desarrollo del producto se cumpla con los criterios de calidad establecidos y, el hecho de implementar todas las actividades propuestas para el aseguramiento de la calidad, posibilita garantizar una presencia mínima de riesgos o incidencias, para finalmente lograr satisfacer las expectativas del cliente.

PALABRAS CLAVES:

calidad, aseguramiento, plan, guía

ÍNDICE DE CONTENIDO

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 6

1.1 Introducción 6

1.2 Conceptos relacionados con la calidad..... 6

 1.2.1 Calidad del software 7

 1.2.2 Modelo de evaluación de Calidad 8

 1.2.3 Estándar de Calidad 8

 1.2.4 Norma de Calidad..... 9

 1.2.5 Aseguramiento de la Calidad 9

 1.2.6 Plan de Aseguramiento de la Calidad 10

1.3 Métricas..... 11

1.3.1 Factores que determinan la calidad del software 12

 1.3.2 Factores de Calidad ISO/IEC 9126 12

1.4 Costos de calidad 13

1.5 Estudio de estándares y modelos de calidad..... 15

 1.5.1 Ventaja del uso de estándares y modelos para asegurar la calidad del software 17

 1.5.2 Enfoques del aseguramiento de la calidad planteados por el modelo CMMI 18

 1.5.3 Enfoques del Aseguramiento de la calidad planteados por el estándar ISO/IEC 15 504 . 19

 1.5.4 Enfoques del Aseguramiento de la calidad planteados por el Estándar IEEE 730 20

 1.5.5 Enfoques del Aseguramiento de la calidad planteados por el estándar ISO 9000-3 21

 1.5.6 Propuesta del estándar o modelo a usar para asegurar la calidad del producto software 22

1.6 Modelos de Madurez de Procesos..... 23

 1.6.1 Proceso de Software Personal (PSP) 23

 1.6.2 Proceso de Software en Equipo (TSP) 24

1.7 Proceso Unificado de Desarrollo del Software. Aspectos identificados para el aseguramiento de la calidad..... 24

1.8 Criterios de Roger Pressman sobre el aseguramiento de la calidad 25

1.9 Aplicación del Plan de Aseguramiento de la Calidad en los proyectos software 27

 1.9.1 En el mundo 27

 1.9.2 En Cuba..... 27

 1.9.3 En la Universidad de las Ciencias Informáticas (UCI) 28

1.10 ¿Por qué un Plan de Aseguramiento de la Calidad es único para cada proyecto software? . 30

1.11 ¿Implementar un Plan de Aseguramiento de la Calidad minimiza el problema existente? 32

1.12 Conclusiones..... 33

CAPÍTULO 2: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN..... 34

2.1	Introducción.....	34
2.2	Descripción de las secciones correspondientes al Plan de Aseguramiento de la Calidad.....	35
2.2.1	Introducción.....	35
2.2.1.1	Propósito.....	35
2.2.1.2	Alcance.....	35
2.2.1.3	Definiciones, Acrónimos y Abreviaturas.....	35
2.2.1.4	Referencias.....	35
2.2.1.5	Resumen.....	35
2.2.2	Objetivos de Calidad.....	36
2.2.3	Gestión.....	36
2.2.3.1	Organización.....	36
2.2.3.2	Tareas y Responsabilidades.....	36
2.2.4	Documentación.....	36
2.2.5	Métricas.....	36
2.2.6	Estándares y Guías.....	36
2.2.7	Plan de Revisiones y Auditorías.....	37
2.2.7.1	Tareas generales de Revisiones y Auditorías.....	37
2.2.7.2	Cronograma.....	37
2.2.7.3	Organización y responsabilidades.....	37
2.2.7.4	Resolución de problemas y actividades de corrección.....	37
2.2.8	Herramientas, Técnicas y Metodologías.....	38
2.2.9	Pruebas y Evaluación.....	38
2.2.10	Flujo de Trabajo.....	38
2.2.11	Herramientas, Técnicas y Metodologías.....	38
2.2.12	Resolución de Problemas y Acción Correctiva.....	38
2.2.13	Registros de Calidad.....	38
2.2.14	Entrenamiento.....	39
2.3	Aspectos relacionados con la forma de solución para diferentes secciones del Plan.....	39
2.3.1	Forma de solución para establecer los Objetivos de Calidad del Proyecto.....	39
2.3.2	Estructura de Organización propuesta para el equipo de calidad del proyecto.....	42
2.3.3	Estrategia para determinar las tareas y responsabilidades propuestas.....	43
2.3.4	Especificaciones de las métricas propuestas en el plan.....	44
2.3.5	Justificación de la selección de los estándares y modelos a establecer durante las diferentes etapas del proyecto.....	47
2.3.6	Forma de solución para elaborar el Plan de Revisiones y Auditorías.....	52
2.3.7	Especificación de los tipos de pruebas propuestas para desarrollar el Plan de Pruebas.....	58
2.3.8	Solución para el flujo de trabajo de las actividades de aseguramiento de la calidad.....	60
2.3.9	Justificación de las Herramientas, Técnicas y Metodologías propuestas.....	64
2.3.10	Solución para establecer la Resolución de Problemas y Acción Correctiva.....	67

2.3.11	Especificaciones de los registros de calidad.....	69
2.3.12	Especificaciones sobre las actividades de entrenamiento	70
2.4	Conclusiones	70
CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS OBTENIDOS		71
3.1	Introducción.....	71
3.2	Beneficios de aplicar métricas	71
3.3	Descripción de los pasos a seguir para la aplicación de las métricas que fueron usadas	72
3.3.1	Formulación.....	73
3.3.2	Colección.....	74
3.3.3	Análisis	75
3.3.4	Interpretación.....	78
3.3.5	Realimentación	79
3.4	Otros resultados	81
3.5	Uso de Herramientas CASE para incrementar la calidad.....	85
3.6	Conclusiones.....	87
CONCLUSIONES		88
RECOMENDACIONES.....		89
REFERENCIAS BIBLIOGRÁFICAS		90
ANEXOS.....		93
ANEXO 1: PLAN DE ASEGURAMIENTO DE LA CALIDAD PARA EL PROYECTO BANCO.....		93
ANEXO 2: Encuesta.....		115
ANEXO 3: Datos de la medición del porcentaje de adecuación de un proceso modelado con los criterios evaluados		115
ANEXO 4: Cantidad de defectos inyectados de cada tipo por cada proceso modelado		118
ANEXO 5: Porcentaje de defectos obtenidos por cada proceso modelado		120
ANEXO 6: Plantilla de No Conformidades.....		122
ANEXO 7: Plantilla general para la elaboración de las Listas de Chequeo.....		125
ANEXO 8: Plantilla para la elaboración del Plan de Pruebas		129
ANEXO 9: Fórmula para las métricas.....		138
GLOSARIO		147
ÍNDICE DE TABLAS		
Tabla 1: Distribución de estándares por etapa de aplicación		48

Tabla 2: Datos de la aplicación de las métricas, Adecuación Funcional y Conformidad con la Usabilidad	78
Tabla 3: Rango para la interpretación de las métricas: Adecuación funcional y Conformidad con la usabilidad.....	78
Tabla 4: Rango para la interpretación del por ciento de adecuación de un proceso del negocio modelado con los criterios evaluados	82
Tabla 5: Rango para la interpretación de la aceptabilidad de defectos por procesos	84

ÍNDICE DE FIGURAS

Figura 1: Representación del concepto de Métrica	11
Figura 2: Representación del coste relativo a corregir un error en etapas posteriores de desarrollo del SW.....	15
Figura 3: Resultados de la encuesta aplicada sobre temas de aseguramiento de la calidad	30
Figura 4: Representación de las acciones a seguir durante la Formulación	73
Figura 5: Resultados de aplicar la métrica Conformidad con la usabilidad	79
Figura 6: Resultados de aplicar la Métrica Adecuación Funcional.....	80
Figura 7: Resultados de la medición del por ciento de adecuación de un proceso del negocio a los criterios evaluados	82
Figura 8: Representación de los datos de defectos inyectados durante el modelado de los procesos del negocio	84
Figura 9: Resultados de la interpretación de la cantidad de defectos por procesos	84

INTRODUCCIÓN

El hombre en su afán de desarrollo persigue mejorar su calidad de vida, con tal fin no sólo se ha centrado en crear innovaciones que le aporten beneficios, también mediante esta forma de evolución van aparejados los procesos de calidad seguidos para el avance tecnológico. La industria del software, a pesar de ser muy joven, ha adoptado con gran empeño los principios de calidad indispensables para el desarrollo de sus productos. Una explicación a esta realidad se fundamenta en la amplia competencia que existe a nivel mundial en este campo.

Actualmente cualquier empresa de software que se haya propuesto como meta satisfacer a sus clientes, mantener una imagen reconocida entre sus semejantes, o de una forma más lucrativa obtener grandes ganancias, simplemente está obligada a asegurar la calidad de sus proyectos. Este es un proceso que se ha convertido en un objetivo estratégico para las organizaciones.

Los aspectos relacionados con la calidad en los proyectos de software, no sólo hacen referencia a la tecnología utilizada o a los conocimientos de las personas responsables de su desarrollo, además, se trata de mejorar y asegurar todos los procesos involucrados para satisfacer las expectativas del cliente al entregar un producto final estable y confiable, el cual se ajuste a las especificaciones funcionales que hayan sido documentadas. A esto se suma las expectativas del cliente referentes a riesgos, prestaciones, plazos de entrega y costos.

La calidad en un producto software es medible; pero al ser inmaterial se hace difícil determinarla. Aunque sí existen algunos indicadores que generalmente son utilizados tales como: la presencia mínima o nula de defectos en el producto, el cumplimiento detallado de los requisitos y el grado de satisfacción del cliente; vale destacar que visiblemente para los usuarios finales la calidad es una característica inherente del producto y ahí es donde se medirá. En etapas más tempranas de desarrollo del producto sólo pueden medirse los recursos y los procesos.

A lo largo del ciclo de vida de un software, la calidad ha de ser implementada de forma paralela desde la planificación hasta la producción del mismo. Muchas empresas trabajan con la ideología de ignorar los procesos necesarios en el aseguramiento de la calidad durante todas las etapas de desarrollo de un proyecto software para luego medir la calidad una vez obtenido el producto. El hecho de no asegurar la calidad desde un inicio equivale a una mala práctica que consecuentemente llevará a la posibilidad real de errores y a provocar la insatisfacción del cliente. Esto también equivale a sufrir

costosos gastos de tiempo y de recursos debido a que los errores encontrados a última hora pueden implicar cambios desde la base de desarrollo del software.

Las tecnologías de la información y las comunicaciones (TIC) son especialmente importantes para empresas u organizaciones, atendiendo a que facilitan la mayoría de las actividades realizadas en ellas. Estas nuevas tecnologías para los procesos de producción contribuyen a evaluar la calidad de los productos y reducir los costos. El Estado Cubano ha logrado identificar desde muy temprano la conveniencia y necesidad de dominar e introducir en la práctica social las TIC para adquirir una cultura digital como una de las características representativas del hombre moderno.

En este contexto el sistema bancario cubano ha experimentado una serie de cambios durante los últimos años y con la estrategia de automatización trazada se han obtenido importantes logros en el desarrollo de sistemas informáticos bancarios. El Banco Central de Cuba (BCC) representa la autoridad rectora, en un segundo nivel están un grupo de instituciones financieras no bancarias y la conformación además de ocho tipos de bancos con sus particularidades específicas.

Actualmente el BCC utiliza el SABIT (Sistema Automatizado para la Banca Internacional de Comercio) creado por la Empresa de Servicios Informáticos para el Banco Central (SIBANC), como software informático para realizar todas sus operaciones financieras de procesamiento de datos. Este sistema contable si bien ofrece un conjunto de beneficios, no satisface íntegramente las necesidades del cliente, lleva en explotación mucho tiempo a través de diferentes versiones (MS-DOS, Visual FoxPro y SQL), fue construido con tecnología obsoleta y la información estadística que brinda no es suficiente.

A la facultad se le ha dado la tarea de desarrollar un software contable con la marcada estrategia de modernizar la informatización de todas las operaciones en las entidades bancarias y financieras del país de una forma flexible y configurable. Con el objetivo de que este proyecto sea viable no sólo es suficiente disponer del esfuerzo, dedicación y conocimientos de sus desarrolladores; también es necesario establecer una estrategia de aseguramiento de la calidad que ayude a prevenir y controlar los problemas, además de los riesgos usuales de los proyectos software, aunque es válido aclarar que dicha estrategia no implica inevitablemente una garantía de éxito.

Dentro de las acciones que se pretende seguir se incluye realizar un Plan de Aseguramiento de la Calidad específico para el proyecto, que contiene las actividades de aseguramiento de la calidad a ejecutar tales como: selección de los estándares a utilizar para prevenir defectos, los productos a

revisar, las métricas a usar, los procedimientos a seguir en la obtención de los productos, identificación de problemas para luego informarlos a sus desarrolladores y por último continuar su seguimiento hasta llegar a corregirlos. No es menos cierto que inicialmente, llevar a cabo estas acciones provoca un aumento de los costos de desarrollo del producto, pero en compensación se debe reducir en gran medida los costos de su mantenimiento.

Dada la situación planteada anteriormente surge el siguiente **problema científico**: ¿Cómo definir un adecuado proceso de aseguramiento de la calidad para el Proyecto Banco mediante la elaboración de una propuesta de un Plan de Aseguramiento de la Calidad? En aras de lograr una alternativa de solución, el presente trabajo propone minimizar el problema existente partiendo de la propuesta de un Plan de Aseguramiento de Calidad que facilite verificar que el producto desarrollado cumpla con los criterios de calidad establecidos y garantice además, la ausencia de riesgos o incidencias. Razón por la cual el **objeto de estudio** queda enmarcado en los procesos de aseguramiento de la calidad; y delimitando el **campo de acción** en los procesos de aseguramiento de la calidad para la elaboración del Plan de Aseguramiento de la Calidad del Proyecto Banco.

El **objetivo general** trazado para darle solución al problema identificado sería: definir los procesos involucrados para elaborar un Plan de Aseguramiento de la Calidad. Con el propósito de darle cumplimiento al mismo se derivan los siguientes **objetivos específicos**:

1. Investigar sobre las normas, estándares y modelos empleados para el aseguramiento de la calidad y cómo son aplicados en los proyectos software.
2. Analizar los procesos llevados a cabo para el aseguramiento de la calidad según estándares, normas, modelos y directivas.
3. Analizar los problemas de calidad que han existido y actualmente existen en los proyectos productivos de la universidad.
4. Elaborar el Plan de Aseguramiento de la Calidad para el Proyecto Banco.
5. Obtener resultados de la aplicación del plan durante el avance del proyecto.

El trabajo queda sustentado en la siguiente **hipótesis** “Si se hace una correcta definición de los procesos de aseguramiento de la calidad para el Proyecto Banco, será posible garantizar una mayor calidad en los procesos que se desarrollen y en el producto final”. Con el propósito de dar confirmación a dicha hipótesis se propone realizar las siguientes **tareas de investigación**:

1. Estudiar los conceptos y términos de calidad empleados por las empresas productoras de software tanto nacional como internacionalmente.
2. Estudiar diferentes estándares, modelos y normas que han sido establecidos a nivel mundial para guiar el buen desarrollo de un producto software y certificar su calidad.
3. Estudiar las herramientas que son empleadas durante el desarrollo de algunos procesos para el aseguramiento de la calidad.
4. Definir los requerimientos de calidad del producto que se quiere obtener.
5. Definir los procesos que posibiliten un adecuado aseguramiento de la calidad para el Proyecto Banco.
6. Establecer un conjunto de actividades de aseguramiento de la calidad para intentar garantizar la calidad del producto durante todo su ciclo de desarrollo.
7. Determinar, luego de haber realizado las investigaciones pertinentes y según las especificaciones del proyecto, el modelo de calidad a seguir, estándares y normas para asegurar la calidad desde un inicio.
8. Elaborar una propuesta de métricas a usar para medir en las diferentes etapas del proyecto, el cumplimiento de los requisitos de calidad establecidos.
9. Establecer una propuesta de los tipos de revisiones y auditorías necesarias para asegurar la calidad del producto y documentarlas.
10. Determinar los distintos tipos de pruebas más convenientes para los posibles casos de prueba.

El presente trabajo se divide en 3 capítulos, a continuación se presenta el nombre del capítulo y su objetivo desde un contexto global:

Capítulo #1: “FUNDAMENTACIÓN TEÓRICA”. El objetivo de este capítulo es abordar conceptos generales y básicos que permiten comprender temas relacionados con el aseguramiento de la calidad de un proyecto software. Incluye un estado del arte en el ámbito nacional, internacional y a nivel de universidad, de las normas, metodologías y modelos vinculados al tema, para luego hacer una selección adecuada de cada uno de ellos. También se exponen los aspectos que determinan la necesidad de la implementación de un Plan de Aseguramiento de la Calidad.

Capítulo #2: “CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN”. Este capítulo tiene como principal objetivo proponer una solución para intentar asegurar la calidad en el Proyecto Banco, a través

del desarrollo de un Plan de Aseguramiento de la Calidad. Para ello se estudia a profundidad el campo de acción, de ahí que es posible definir y describir los procesos identificados, para luego establecer las acciones propuestas con las cuales se pretende alcanzar la calidad del producto desarrollado.

Capítulo #3: “ANÁLISIS DE LOS RESULTADOS OBTENIDOS”. El objetivo del presente capítulo es mostrar un conjunto de resultados y conclusiones generales de la aplicación, según el avance del proyecto, de las actividades de aseguramiento de la calidad que se proponen en el plan. Realizar un análisis de los beneficios que aporta a la calidad del software el uso de métricas para medir atributos del software y los beneficios de aplicar herramientas CASE como apoyo a los procesos seguidos durante el desarrollo del software y a la vez para incrementar la calidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Durante el desarrollo de un producto software el objetivo fundamental, si se pretende satisfacer las expectativas del cliente, debe ser el logro de un producto con calidad. En la mayoría de los proyectos software se subestiman las acciones para alcanzar el aseguramiento de la calidad y se dedican los esfuerzos de hoy en arreglar lo que se hizo mal ayer; así se explica por qué razón existen tantas inconformidades de los clientes con el producto final. Partiendo de este análisis no es difícil concluir en que es vital asegurar la calidad en el proceso de desarrollo del software, de esta forma, el producto alcanzará elevados índices de desempeño y cumplirá íntegramente las necesidades del cliente. En vista a lograr estas premisas, se recomienda como estrategia profundizar en la madurez de la Ingeniería de Software, sin dejar de olvidar que consecuentemente con ello viene aparejado el hecho de centrar la atención en la mejora de los procesos como forma de optimizar los niveles de calidad.

El presente capítulo contiene una descripción actualizada, partiendo de lo general a lo específico del estado del arte, relacionada con definiciones y conceptos significativos para el tema de aseguramiento de la calidad; proporcionando una visión más amplia referente a normas, estándares y metodologías empleadas de forma tal que se pueda determinar cómo estas se vinculan con el tema tratado. Posteriormente, se presenta un estudio de los principales modelos y estándares utilizados mundialmente para la administración de la calidad, con el objetivo de adquirir sólidos argumentos y lograr obtener una selección adecuada según las especificaciones del proyecto. Contiene además, los resultados de la investigación referida a la aplicación por parte de empresas productoras de software del Plan de Aseguramiento de la Calidad, primero desde un punto de vista internacional, seguido del nivel nacional hasta concluir al nivel base del contexto donde se desarrolla este trabajo, que en este caso sería la Universidad de las Ciencias Informáticas.

1.2 Conceptos relacionados con la calidad

Antiguamente al referirse a la calidad se decía que estaba relacionada con las personas involucradas directamente en la producción, se verificaba al final del ciclo de trabajo y la mayoría de los problemas estaban relacionados con las áreas técnicas. Todo esto aumentaba el costo en la ejecución del proyecto y los problemas de calidad conducían a excusas y culpas.

Actualmente estos criterios han cambiado, ahora se plantea que la calidad es responsabilidad de todos, la mayoría de los problemas al respecto están relacionados con las áreas gerenciales; sin embargo, los problemas de calidad conducen a soluciones que involucran a todo el equipo, indudablemente se ahorra costos y aumenta el negocio. Es muy importante que se planifique antes de empezar a producir y ocurre durante todo el ciclo de vida del proyecto.

La Real Academia de la Lengua Española proporciona las siguientes definiciones para el término calidad:(ESPAÑOLA, 2008)

1. Propiedad o conjunto de propiedades inherentes a algo que permiten juzgar su valor.
2. Buena calidad, superioridad o excelencia.
3. Condición o requisito que se pone en un contrato.
4. Importancia o gravedad de algo.

La Organización Internacional para la Estandarización (ISO) define calidad según la Norma ISO 8402: 1994, como: “Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer necesidades explícitas o implícitas”.(ISO, 1994)

Según la Norma ISO 9000: 2000, es el grado en el que un conjunto de características inherentes cumple con los requisitos.(ISO, 2000)

1.2.1 Calidad del software

Se ha evolucionado mucho en los enfoques de la calidad del software. A finales de los sesenta cada vez que se quería afrontar la mejora del software todo se dirigía a la mejora del código. Ya en los años ochenta se comenzó a considerar los aspectos referentes a especificaciones, diseño, evaluación y gestión del software. Pese a estos cambios, persiste la problemática debido al incremento del nivel de complejidad del software y consecuentemente no se alcanzan los niveles de calidad deseados.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.(VEGA LEBRÚN CARLOS, 2000)

Según Pressman, la calidad de software es: “La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente

documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.(PRESSMAN, 2001)

Según el Institute of Electrical and Electronics Engineers la calidad de software es: “El grado con que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.(SOCIETY, 1991)

Según la norma ISO 8402 es: “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”.(ISO, 1994)

Se puede decir, a modo de conclusión, que la calidad del software está estrechamente ligada a la adecuación del software lo más ideal posible con los requisitos establecidos por el usuario.

1.2.2 Modelo de evaluación de Calidad

Los Modelos de Calidad son herramientas que guían a las Organizaciones a la Mejora Continua y la Competitividad dando las especificaciones de qué tipos de requisitos deben de implementar para poder brindar productos y servicios de alto nivel. (MELVIN, 2008)

Un modelo de calidad es un conjunto de características y las relaciones entre ellas que proporcionan una base para especificar requisitos y evaluar la calidad. Desde otro punto de vista es un marco conceptual que especifica una serie de conceptos medibles y sus relaciones.

Los mismos contribuyen a medir la calidad del software a través de las características más importantes del proceso del ciclo de vida y/o midiendo las métricas que dan el grado de calidad del software. Por lo que también se define como un conjunto de buenas prácticas durante el ciclo de vida del software, orientado a los procesos de gestión y desarrollo de proyectos software indicando qué hacer, no cómo hacerlo.

1.2.3 Estándar de Calidad

Según la ISO, un estándar es "un conjunto de acuerdos documentados que contiene especificaciones técnicas u otros criterios precisos para ser usados constantemente como reglas, lineamientos o definiciones de características. Todo esto con la finalidad de asegurar que los materiales, productos, procesos y servicios son óptimos para su propósito".(ISO, 2000)

Un estándar de Ingeniería de Software es una regla o base de comparación que se utiliza para medir aspectos del software tales como: calidad, productividad, duración, esfuerzo y costo. Son creados por diferentes individuos u organizaciones para armonizar las especificaciones de productos, interfaces, procesos, terminologías, etc., abarcando de esta forma una serie de tópicos de interés para instituciones y países.

Los estándares de calidad son los criterios establecidos para certificar que un producto, proceso y servicio obedece al objetivo para el que fue desarrollado. (CASAL, 2007)

La norma ISO-8402 define estándar de calidad como: La totalidad de rasgos y características de un producto, proceso o servicio que sostiene la habilidad de satisfacer estados o necesidades implícitas. (ISO, 1994)

1.2.4 Norma de Calidad

Las normas son un modelo, un patrón, ejemplo o criterio a seguir. Una norma es una fórmula que tiene valor de regla y tiene por finalidad definir las características que debe poseer un objeto y los productos que han de tener una compatibilidad para ser usados a nivel internacional. (SANS, 1998)

Una norma de calidad es un documento, establecido por consenso y probado por un organismo reconocido (nacional o internacional), que proporciona para un uso común y repetido, reglas, directrices o características para las actividades de calidad o sus resultados, con el fin de conseguir un grado óptimo de orden en el contexto de la calidad. (COMERCIO.ES, 2008)

Las normas internacionales de calidad son redactadas de acuerdo con las reglas de las directivas de la ISO (Organización Internacional de Estándares) y la IEC (Comisión Electrotécnica Internacional).

1.2.5 Aseguramiento de la Calidad

El aseguramiento de la calidad no es más que el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto software cumplirá con los requisitos de calidad establecidos. No está completo a menos que los requisitos de calidad reflejen totalmente las necesidades del cliente; para ser efectivo se requiere una evaluación continua de los factores que afectan la calidad, además de evaluaciones periódicas.

Según la Norma ISO 9000:2000, el aseguramiento de la calidad es: “la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de calidad”. (ISO, 2000)

También se puede definir como el esfuerzo total para plantear, organizar, dirigir y controlar la calidad en un sistema de producción con el objetivo de dar al cliente productos con la calidad adecuada. Es simplemente en pocas palabras, asegurar que la calidad sea lo que debe ser.

Sridharan indica que mientras el software que se está desarrollado reúne los requerimientos y su desempeño es el esperado, es preciso que se supervisen las actividades de desarrollo del software y su rendimiento, en distintas oportunidades durante cada fase del ciclo de vida. (QUALITRAIN, 2001) Ese es uno de los propósitos esenciales del aseguramiento de la calidad.

Wieggers ha planteado tres aspectos muy importantes con relación al aseguramiento de la calidad del software: (MENDOZA, 2003)

- La calidad no se puede probar, se construye.
- El aseguramiento de la calidad del software no es una tarea que se realiza en una fase particular del ciclo de vida de desarrollo.
- Las actividades asociadas con el aseguramiento de la calidad del software deben ser realizadas por personas que no estén directamente involucradas en el esfuerzo de desarrollo.

Básicamente, el aseguramiento de la calidad del software consiste en la revisión de los productos y su documentación relacionada para verificar su cobertura, corrección, confiabilidad y facilidad de mantenimiento, incluyendo la garantía de que un sistema cumpla las especificaciones y los requisitos para su uso y desempeño deseado.

1.2.6 Plan de Aseguramiento de la Calidad

Según el Diccionario de la Real Academia de la Lengua Española una de las definiciones de plan es: proyecto, programa de las cosas que se van a hacer y de cómo hacerlas. (ESPAÑOLA, 2008)

Plan de Calidad: Documento que establece las prácticas, los recursos y la secuencia de las actividades relativas a la calidad de un producto, proyecto o contrato.

Entonces, un Plan de Aseguramiento de la Calidad es una guía que permite asegurar los procesos que se desarrollan en una empresa, indicando la necesidad de definir y describir todos aquellos requisitos

que la empresa ha de cumplir para desarrollar con eficacia y corrección un producto conforme a la calidad exigida por el cliente.

Con el propósito de implementar los criterios establecidos por los sistemas de calidad en cada proyecto, es vital desarrollar un Plan de Aseguramiento de la Calidad particular. Este plan, genéricamente, deberá contener los siguientes aspectos:

- Objetivos de calidad del proyecto y enfoque para su consecución.
- Documentación referenciada en el plan.
- Gestión del aseguramiento de la calidad.
- Documentación de desarrollo y de control o gestión.
- Estándares, normas y prácticas que hay que cumplir.
- Actividades de revisiones, auditorías y pruebas.

1.3 Métricas

Las actividades de medición de la calidad del software no representan una tarea fácil para los encargados de desempeñar este tipo de trabajo. Inicialmente es primordial identificar las entidades y atributos de interés que se desea medir. Vale señalar que una entidad es un objeto o un evento, y un atributo son las características o propiedades del software a medir. Además, en un producto software se pueden encontrar tres tipos de clases entidades, cuyos atributos van a ser medidos, estas son: recurso, proceso y producto.

El concepto de métrica se puede definir desde un punto de vista muy práctico: Forma de medir + una escala.

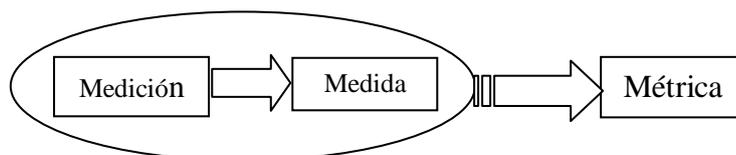


Figura 1: Representación del concepto de Métrica

Los términos: métricas, medición y medida aunque están familiarizados no tienen igual significación.

Medición: proceso por el cual se asignan números o símbolos a atributos o entidades en el mundo real de forma tal que los describa de acuerdo a reglas claramente definidas. (SANZ, 1998)

Medida: proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto. (PRESSMAN, 2001)

Paul Goodman define las métricas de software como: “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos”. (GOODMAN, 2004)

De forma más sencilla y acertada el IEEE define métrica como: “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado”. (SOCIETY, 1991)

Las métricas son utilizadas como estándares de medidas para juzgar los atributos de algo que está siendo medido, aportando información necesaria durante la toma de decisiones técnicas, logrando así obtener conocimientos relativos para centrar la atención en lo que necesita ser mejorado, darle solución y contribuyendo de esta forma al aseguramiento de la calidad del producto.

1.3.1 Factores que determinan la calidad del software

Para abordar esta temática hay que partir del hecho que los factores que determinan la calidad del software se pueden dividir en dos grandes grupos según McCall. En un primer grupo se pueden agrupar los factores en dependencia de la forma en que se consigue su medida, a la vez, este grupo se divide en otros dos que incluirían aquellos factores que pueden ser medidos directamente y los que solo pueden ser medidos indirectamente. En un segundo grupo se especifican los factores enunciados por McCall que están centrados en tres aspectos básicos de un producto software: características operativas, capacidad de soportar cambios, adaptabilidad a nuevos entornos. Conjuntamente con estas clasificaciones más generales, también hay otros tipos de clasificaciones como los factores de calidad FURPS (las siglas significan: funcionalidad, facilidad de uso, fiabilidad, rendimiento y capacidad de soporte) dados por Hewlett Packard y la propuesta por la norma ISO 9126.

Señalar que el conjunto de factores de calidad antes mencionados no representan los únicos que existen, aunque sí son de los más reconocidos mundialmente, y cuando se hace la planificación del producto software es preciso determinar previamente cuáles de dichos factores van a ser considerados como requisitos a la vez.

1.3.2 Factores de Calidad ISO/IEC 9126

Los factores de calidad ISO/IEC 9126 y sus atributos fueron seleccionados para determinar métricas de calidad para las actividades del proceso de desarrollo del producto software, pues es una norma cubana presentada en 1992; define 6 características de la calidad del producto y proporciona además 29 sub-características en total, las cuales se adaptan perfectamente a las necesidades de medición según el producto que será desarrollado. A continuación las mismas son mencionadas:

Funcionalidad: Presencia de una serie de funciones y propiedades específicas establecidas (idoneidad, precisión, interoperabilidad, seguridad, conformidad con la funcionalidad).

Confiabilidad: Capacidad del software para conservar su nivel de procesamiento bajo determinadas condiciones en un período de tiempo (madurez, tolerancia ante fallos, recuperabilidad, conformidad con la fiabilidad).

Usabilidad: Esfuerzo preciso para el uso y el valor de uso, por un grupo determinado de usuarios (comprensibilidad, cognoscibilidad, operabilidad, atractivo, conformidad con la usabilidad).

Eficiencia: Correspondencia entre el nivel de acción del software y el conjunto de recursos empleados bajo ciertas condiciones (rendimiento, utilización de recursos, conformidad con la eficacia).

Mantenibilidad: Esfuerzo aplicado con motivo de hacer modificaciones específicas (diagnosticabilidad, flexibilidad, contrastabilidad, estabilidad, portabilidad, mantenibilidad, conformidad en la mantenibilidad).

Portabilidad: Habilidad del software para ser trasladado de un ambiente a otro (adaptabilidad, instalabilidad, coexistencia, reemplazabilidad, conformidad con la portabilidad).

1.4 Costos de calidad

Se entiende por Costos de Calidad al dinero destinado para obtener la calidad requerida. La misma no se consigue por casualidad ni accidentalmente, sino que todo debe ser planeada en actividades, medida y garantizada. Además, se puede decir que los costos de calidad son aquellos en que se incurren por el cumplimiento de un conjunto de requisitos de un producto o servicio adecuado a satisfacer las necesidades implícitas o explícitas de los clientes y los costos asociados por no cumplir estos requisitos. (CAIRO, 1997)

Desde un punto de vista menos rígido, la calidad no significa un costo en el total sentido de la palabra, sino una inversión ejecutada día a día y que espera recoger frutos a largo plazo. La reducción de costos se produce porque con un adecuado proceso de desarrollo que asuma la calidad del software

como un factor central, es posible descubrir tempranamente los errores y dicho sea de paso, los errores son más costosos cuanto más tarde se descubren. Existen diversas clasificaciones para los costos de calidad, entre las más habituales se refieren las de la Sociedad Americana para el Control de la Calidad (sus siglas en inglés: ASQC), que establece una clasificación y análisis de los costos de calidad.

- Costos de prevención.
- Costos de evaluación.
- Costos de fallos internos.
- Costos de fallos externos.

Medir el costo de calidad representa una práctica fundamental en un programa de acciones para el mejoramiento de la calidad. Se invierte para alcanzar índices notables de calidad mediante la prevención, o sea, cuando se evitan errores y mediante la evaluación que es cuando se verifica que no existan errores. El costo de la calidad lo conforman 2 componentes inversamente relacionados: uno está representado por lo que se destina en obtener buena calidad y el otro de forma contraria está representado por lo que es pagado por no lograrla. El primer componente se decide por las acciones que determinan los desarrolladores del producto software, como las actividades de revisiones periódicas y constantes de las aplicaciones, las cuales se deben integrar a sus actividades cotidianas. El segundo componente no se decide por los desarrolladores del software sino que se manifiesta en las fallas del producto. Estas fallas pueden ser determinadas por los desarrolladores (fallas internas) o por los clientes (fallas externas).

Un factor muy importante para reducir el costo del proyecto lo determina la temprana detección y corrección de errores mediante Revisiones Técnicas Formales (RTF). Si se detectan errores durante el proceso de desarrollo se puede prevenir que se conviertan en defectos después de entregado el software. Por ejemplo, según datos publicados la densidad de defectos (número de defectos/ tamaño del producto) oscila entre 2 y 60 defectos/KNCSS (estas siglas significan mil instrucciones de código fuente sin comentarios).

Pressman, en su libro Ingeniería de Software un Enfoque Práctico, muestra un ejemplo del impacto sobre el costo de la detección anticipada de errores. Por ejemplo, si un error descubierto durante el

diseño cuesta 1.0 unidad monetaria, el mismo error descubierto justo antes de que comienza la prueba costará 6.5 unidades, durante la prueba 15 unidades y después de la entrega entre 60 y 100 unidades.

La siguiente figura, extraída del libro, muestra estos valores de forma más detalla:(PRESSMAN, 2001)

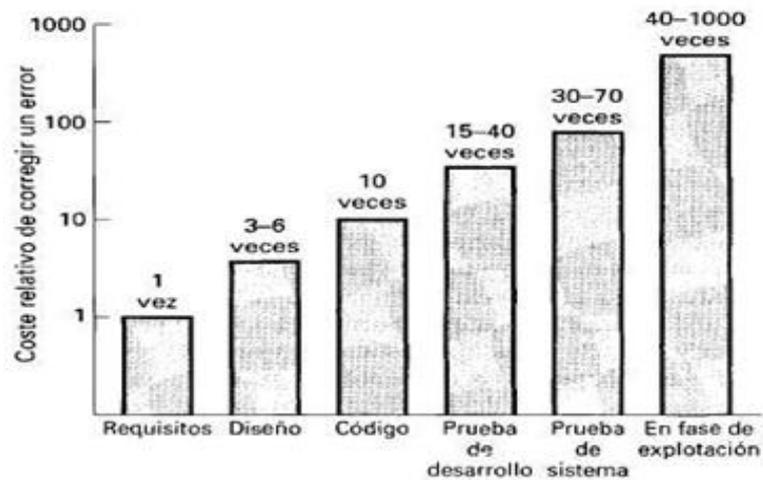


Figura 2: Representación del coste relativo a corregir un error en etapas posteriores de desarrollo del SW

Los costos económicos de la función de aseguramiento de la calidad en el software se ha estimado que varían entre un 2.5 y 5 por ciento del costo total de un proyecto de desarrollo de un producto de software. (VEGA LEBRÚN CARLOS, 2000) Sin embargo, las aplicaciones de costos de calidad han mostrado que es posible reducir incluso los costos totales de un proyecto si se simplifica y estima correctamente la inversión dedicada al aseguramiento de la calidad.

Como resultado de todo lo planteado anteriormente, la conclusión más sensata para los desarrolladores de software correspondería a mantenerse siempre conscientes de que la vía más rápida y económica de producir software es haciéndolo bien; mientras más recursos se dediquen a una correcta gestión de aseguramiento de la calidad menos probabilidades de pérdida se sufrirá al finalizar y entregar el producto al cliente.

1.5 Estudio de estándares y modelos de calidad

Para comenzar a abordar esta temática sería bueno enmarcarse en el contexto histórico de que los primeros modelos convencionales de aseguramiento de la calidad surgieron en los ámbitos de las

industrias americanas de defensa naval y nuclear. El estándar más antiguo del área de Ingeniería de Software fue publicado en 1972 y corresponde al estándar militar estadounidense sobre el aseguramiento de calidad. Más adelante lo siguió en 1976 el estándar sobre los planes de aseguramiento de calidad de IEEE. Desde entonces esta organización ha generado 29 estándares (hasta 1996) y otros 13 están en preparación.

Anteriormente se hizo mención a los conceptos de modelo y estándar de calidad; pero considerando su importancia en su uso posterior durante la elaboración del plan, no sería redundante dejar claro cuándo emplear la definición de modelo y cuándo la de estándar.

Diferencia fundamental: (SANTOS, 2008)

- Un modelo es algo conceptual que pertenece al campo de lo subjetivo, conceptualizado por el hombre.
- Una norma (o estándar) es algo concreto, un documento que contiene reglas, requisitos, lineamientos y recomendaciones.

Los modelos de calidad del software están conformados por un conjunto de procedimientos que hacen uso de herramientas, técnicas y metodologías, estos sirven de guías para mantener una trayectoria de mejoras continuas desde el inicio de un proyecto hasta su fin. Los estándares de calidad son más específicos, como reglas que describen los criterios a medir para lograr un objetivo.

Mundialmente se han desarrollado una serie de modelos de calidad para diferentes productos y procesos software, además de la gran diversidad de normas y estándares genéricos existentes para fomentar la competitividad y garantizar la calidad de las aplicaciones siguiendo un Plan de Aseguramiento de la Calidad. De ahí que es posible lograr que se cumplan los requisitos a implementar especificados por el cliente en aras de obtener un producto al nivel de calidad requerido.

Las normas, directivas, modelos y estándares son básicamente las siguientes:

- Familia de normas ISO 9000 y en especial, la ISO 9001 y la ISO 9000-3.2: 1996 Quality Management and Quality Assurance Standards.
- ISO 8402: 1994.
- IEEE 730/1984, Standard for Software Quality Assurance Plans.
- IEEE Std. 1028: 1989, IEEE Standard for Software Reviews and Audits.

- El Plan General de Garantía de Calidad del Consejo Superior de Informática (MAP).
- CMM. Capability Maturity Model.
- ISO/IEC JTC1 15504. SPICE. Software Process Improvement and Capability Determination.
- Modelo de EFQM. Modelo de la Fundación Europea de Gestión de Calidad.

Más adelante se hará referencia a varios de estos estándares mencionados, teniendo en consideración qué elementos plantean para ser usados en las actividades de aseguramiento de la calidad.

1.5.1 Ventaja del uso de estándares y modelos para asegurar la calidad del software

La necesidad de desarrollar un producto software de calidad que satisfaga la necesidad de los usuarios ha hecho que se tenga en cuenta que la calidad debe evaluarse usando un modelo de calidad que tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquisidores y usuarios finales. (ISO, 2001) En tal caso se han presentado una serie de modelos y estándares para incentivar la aplicación de este proceso en las organizaciones. Teniendo en cuenta que la competencia entre las empresas desarrolladoras de software cada día se hace más fuerte, convenientemente se ha logrado que dichas instituciones se preocupen por optimizar sus productos. El uso en el mercado de estas propuestas ha dado la posibilidad a las empresas informáticas de ajustarse a una nueva forma de trabajo que procura alcanzar la satisfacción del cliente, contar con una mayor visibilidad, además de un control de la calidad superior de los procesos y productos finales.

Estos modelos convierten la calidad del software en algo concreto que se puede medir, definir y planificar. Ayudan a mejorar la comunicación entre usuarios, dirección y técnicos. Al cliente le place mucho más saber que el software que va a adquirir está certificado por alguna metodología que controle la calidad en todas las etapas de vida de un software, esto da una mayor confianza en el resultado del trabajo esperado. (ALARCÓN, 2004)

El uso sistemático de estándares de ingeniería de software puede mejorar significativamente la calidad del software que produce una organización. (CORONAS, 1997) Vale señalar el hecho que con la aplicación de los estándares se reducen las posibilidades de cometer errores lamentables que hasta pueden llegar a ser mortales en determinados tipos de software, incrementado de esta forma la fiabilidad del software.

Consecuentemente al reducir los defectos también se reducen los costos del software. Identificar un error en la misma fase donde se ocasionó es menos costoso de solucionar que posteriormente, donde el mismo pudo haber acarreado consigo otros más complejos. La ventaja principal radica en que, al emplear estándares durante todo el ciclo de desarrollo del software, la probabilidad de encontrar un defecto en la fase donde se originó es más elevada que la de encontrarlo en etapas de pruebas finales.

Los modelos de evaluación y mejora de procesos y su estandarización han tomado un papel determinante en la identificación, integración, medición y optimización de las buenas prácticas existentes en la organización y desarrollo software. (MANUEL DE LA VILLA, 2005) El uso de estándares es indispensable pues permite:

- Unificar la codificación, no sólo del equipo de trabajo sino de todos los proyectos que se realicen.
- Ahorro en el tiempo de desarrollo del sistema, sobre todo en Diseño.
- Código completo, preciso, legible y sin ambigüedades.
- Aumentar la eficiencia del grupo de desarrollo, disminuyendo el tiempo de codificación y diseño.
- Mayor productividad en la realización y diseños de pruebas.
- Facilitar el mantenimiento posterior de los programas.
- Incrementar la calidad del producto final.

A modo de conclusión es posible señalar que el uso de estándares ha permitido un entendimiento más claro, o por lo menos más consistente entre los actores participantes de la industria del software. De forma paralela también agregar que los estándares de calidad para el software ayudan a entender las relaciones existentes entre disímiles características de un producto software y especifican una serie de criterios de desarrollo como referencia a la vía de aplicación de la Ingeniería de Software. Si no son seguidos dichos criterios las probabilidades de que se amplíe el margen de error y por consiguiente no se logre garantizar de calidad son una de las consecuencias más importantes que se pueden citar producto a un inadecuado desarrollo del software.

1.5.2 Enfoques del aseguramiento de la calidad planteados por el modelo CMMI

CMMI (Modelo Integral de Capacidad y Madurez) es un modelo que inicialmente se dio a conocer en diciembre del 2000 por el SEI (Instituto de Ingeniería de Software), es tratado a través de dos representaciones: continua y escalonada. En el caso de la representación continua las áreas de

proceso se concentran en cuatro categorías según su propósito. La categoría de Soporte es la que incluye el Área de Aseguramiento de la Calidad de Procesos y Productos (PPQA) proporcionando prácticas para evaluar objetivamente procesos, productos y servicios.

La representación escalonada clasifica las empresas en niveles de madurez que sirven para conocer el grado de experiencia de los procesos desarrollados para producir software. En el nivel 2 el proyecto es gestionado y controlado a medida que se lleva a cabo su desarrollo. Para alcanzar este nivel es necesario establecer diferentes áreas de procesos cuyos objetivos debe lograr la organización, dentro de los que se incluye el Aseguramiento de la Calidad como punto de referencia para facilitar personas y gestión en vista a lograr que los procesos y mecanismos de trabajo satisfagan los procesos.

Esta área de procesos especifica una serie de acciones planeadas y de carácter permanente que son precisas para alcanzar determinados criterios de calidad requeridos, siempre teniendo presente que la planificación debe concebirse desde antes del desarrollo del software y no durante su transcurso.

Para conseguirlo es preciso realizar las siguientes acciones:

- Evaluar objetivamente la ejecución de los procesos, los elementos de trabajo y servicios contra las descripciones de procesos, estándares y procedimientos.
- Identificar y documentar los elementos no conformes.
- Proporcionar información a las personas que están usando los procesos y a los gestores de los resultados de las actividades del aseguramiento de la calidad.
- Asegurar que los elementos no conformes son arreglados.

Sin lugar a dudas, esta es un área de proceso clave para el desarrollo del software siguiendo las políticas planteadas por el modelo CMMI, lamentablemente muchas empresas no le dan la importancia requerida a la hora de implantar dicho modelo.

1.5.3 Enfoques del Aseguramiento de la calidad planteados por el estándar ISO/IEC 15 504

ISO/IEC 15504 (SPICE, Software Process Improvement and Capability Determination) es un estándar bidimensional, en una de sus dimensiones, la que se encuentra enfocada a los Procesos propios del Ciclo de Vida y orientada además a la categoría de suministrador, aparecen indicaciones directas a la calidad y su aseguramiento, así como a su medida. Existen 3 clasificaciones básicas para la arquitectura de dichos procesos, en una de ellas, la de Soporte, se hace referencia expresa a

proporcionar el aseguramiento para que productos y procesos de un proyecto satisfagan los requerimientos especificados y se adhieran a los planes establecidos.

El objetivo básico vinculado al aseguramiento de la calidad que plantea este estándar es: garantizar que los productos y actividades cumplan con las normas, procedimientos y requisitos. Para ello se han identificado las siguientes actividades:

- Selección de normas del proyecto. Ayuda a seleccionar las normas y procedimientos que se emplean en el proyecto.
- Revisar las actividades de ingeniería de software. Revisar estas actividades contra los planes y las normas y los procedimientos seleccionados.
- Auditar el producto. Auditoría de los productos de software frente a las normas y los procedimientos seleccionados.
- Informar resultados. Informe sobre los resultados de las actividades mencionadas.

1.5.4 Enfoques del Aseguramiento de la calidad planteados por el Estándar IEEE 730

La norma IEEE 730 fue creada por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, sus siglas en inglés) en 1989. Constituye un estándar para desarrollar un Plan de Aseguramiento de Calidad de Software y su propósito consiste en proveer una uniformidad mínima para la preparación y el contenido de dicho plan; pero siempre teniendo en consideración que es específico para cada proyecto.

La adopción de este estándar según las regulaciones permite su aplicación abalado por uno o más estándares IEEE o ANSI referentes al aseguramiento de la calidad, definiciones y otras materias. La existencia del mismo no debe ser traducida para prohibir un posible contenido adicional de un Plan de Aseguramiento de la Calidad.

Según lo especificado por la norma el aseguramiento de la calidad consiste en un planeado y sistemático patrón de todas las acciones necesarias para proveer la adecuada confianza de las conformidades de los artículos o productos para establecer los requerimientos técnicos.

Estar en concordancia con el Estándar IEEE 730 implica que un Plan de Aseguramiento de Calidad de Software desarrollado para un proyecto dado debe incluir un conjunto de secciones ordenadas en la secuencia referida; pero en caso contrario, dada la flexibilidad de este estándar, una tabla debe ser provista con referencia cruzada de las sub-secciones cambiadas. Además si no aparece la información

pertinente para una sección, la siguiente aclaración debe aparecer debajo de la sección de la cabecera: “Esta sección no es aplicable a este plan”, junto con la apropiada razón de la exclusión. Algunos de los materiales pueden aparecer en otros documentos. En ese caso la referencia a dicho documento debe ser hecha en el cuerpo del (SQAP). En otro caso el contenido de cada sección del plan debe ser especificado directamente o por referencia a otro documento.

1.5.5 Enfoques del Aseguramiento de la calidad planteados por el estándar ISO 9000-3

La norma ISO 9000-3 (Sistema de la calidad. Modelo para el aseguramiento de la calidad en la inspección y los ensayos finales) pertenece a la familia de la serie ISO 9000 de un total de 5 estándares internacionales. Esta norma es requerida por todas las compañías desarrolladoras de software en el sentido de que es la que provee las especificaciones de cómo aplicar la ISO 9001 al desarrollo, implementación y mantenimiento del software.

Entre sus características generales se refiere al control de la calidad, el cual debe ser aplicado en todas las fases de producción del software. Se debe definir un Sistema de Calidad que incluirá los elementos que han sido acordados entre el proveedor y el cliente, además, la empresa debe asegurarse que se ponga en práctica dicho sistema.

La ISO 9000-3 es una guía que está formada por un conjunto de cláusulas que indican cómo aplicar sus exigencias. Dentro de las cláusulas especificadas son de interés la 4.2 “Sistema de Calidad” y la 5.5 “Planificación de la Calidad”. En el caso de la 4.2 se requiere de una planificación y documentación del sistema de calidad, lo cual es conocido como “Plan de Garantía de Calidad del Software”. Para la cláusula 5.5 se toma como punto de referencia la metodología de Medidas de Calidad instituidas en el Estándar IEEE 1061 para establecer los objetivos de calidad.

Otro aspecto a tener en cuenta según las políticas de este estándar para el aseguramiento de la calidad es que se puede clasificar en dos tipos: interno o externo, considerando si la evidencia de la calidad del producto o del proceso se manifiesta a la gerencia del proveedor o del cliente. Dicho proceso implementa diferentes actividades como:

- Implementación del proceso.
- Aseguramiento del producto.
- Aseguramiento del proceso.

- Aseguramiento del sistema de calidad.

1.5.6 Propuesta del estándar o modelo a usar para asegurar la calidad del producto software

En la actualidad existen más de 250 diferentes estándares de ingeniería de software elaborados por diferentes organismos de estandarización, todos con diferentes grados de detalle, cobertura y aplicabilidad. Generalmente, el propósito, el enfoque y el nivel de adaptabilidad de estos estándares varían grandemente, lo que dificulta el proceso de selección de los estándares adecuados a una organización. (JENKINS, 2003) Por tanto es difícil discernir a la hora de adoptar técnicas de gestión de la calidad y procedimientos de certificación de los sistemas de calidad; aunque para el desarrollo de este trabajo existe la ventaja de que es posible descartar todos aquellos modelos que no incluyan aspectos relacionados con el aseguramiento de la calidad dentro de sus regulaciones.

Se recomienda como punto de partida a tener en cuenta para la selección de un modelo o estándar de calidad, considerar los aspectos puntualizados seguidamente: (SCALONE, 2006)

- La complejidad del proceso de diseño.
- La madurez del mismo.
- La complejidad del proceso de producción.
- Las características particulares del producto o servicio.
- La seguridad del producto o servicio y las consecuencias de su fallo.
- Las consideraciones económicas de los aspectos anteriores.

Sin embargo, a pesar de que esa es una buena práctica, en este trabajo se va a omitir la consideración de los aspectos antes mencionados ya que inmediatamente después de hacer un análisis al respecto, se llegó a la conclusión de que la rigurosidad formal de los modelos provoca que se haga difícil adoptar un modelo de calidad específico para empresas, organizaciones o instituciones que están dispuestas a invertir recursos en conseguir una certificación; y determinar un modelo que se adapte exactamente a sus necesidades es casi imposible.

En tal caso se propone: Desarrollar un modelo propio. O sea una vez realizado un estudio previo de la situación específica del proyecto, según los flujos y procesos definidos para su desarrollo, optar por el o los estándares que más se adecuen al problema, adaptándolos y no adoptándolos en la empresa.

Siempre puntualizando que esta técnica es factible al considerar que un modelo de calidad es una descomposición jerárquica y su elaboración puede abordarse de diferentes formas.

Para pequeñas y medianas empresas dicha práctica es más efectiva que dedicar recursos, tiempo y esfuerzo en implementar un estándar de referencia global como CMMI o SPICE con evaluaciones costosas que requieren una fuerte inversión económica; los cuales publican una amplia documentación sobre qué es lo que tienen que lograr, pero no dicen cómo, por lo que asumirlos es sumamente complicado. Eso es sin mencionar el hecho de que muchas empresas pretendiendo acreditarse con alguno de estos modelos, olvidan el verdadero objetivo por el que fueron creados, o sea, mejorar la calidad del proceso de desarrollo del software y no el fin de alcanzar un mayor nivel de madurez que en realidad ha de verse como un fin consecuente.

1.6 Modelos de Madurez de Procesos

1.6.1 Proceso de Software Personal (PSP)

PSP fue desarrollado en el Instituto de Ingeniería de Software (SEI) por Watts Humphrey. Representa un marco de trabajo diseñado para enseñar a los ingenieros de software a realizar mejor su trabajo, por lo cual constituye un proceso muy útil para certificar a un individuo en desarrollar software de calidad, con el nivel de madurez requerido, a la vez que brinda disciplina en las prácticas individuales de ingeniería de software

PSP está estructurado por 4 tipos de procesos según el orden de ejecución, el tercero corresponde al Proceso Personal de Calidad, aquí se revisa la codificación y el diseño, predicciones de costos y tiempo de desarrollo, mostrando cómo estimar y planificar el trabajo, cómo controlar el rendimiento frente a esos planes y cómo mejorar la calidad de lo que se produce, siendo este el objetivo primordial.

Principios del PSP:

- La calidad de un sistema de software está regida por la calidad del proceso utilizado para desarrollarlo.
- La calidad de un sistema de software está regida por la calidad de sus peores componentes.
- La calidad de un componente de software está regida por el individuo que lo desarrolló.
- Esto está regido por el conocimiento, disciplina y compromiso del individuo.

Y para complementar estos principios plantea que todos los ingenieros de software deberían planificar siempre su trabajo, regirse por lo establecido; manteniendo así el uso disciplinado de buenas prácticas. Como resultado obtendrían productos de alta calidad, ajustados a los plazos establecidos y a su presupuesto; contribuyendo de esta forma al aseguramiento de la calidad del software desde la base del desarrollo.

1.6.2 Proceso de Software en Equipo (TSP)

TSP es un proceso para formar y guiar equipos de ingenieros que desarrollan software. Provee un marco de trabajo para el equipo basado en los principios de PSP, o sea, TSP es complementario a PSP y se puede aplicar tanto a pequeña como a gran escala.

Las metas básicas de TSP son: producir un software de calidad efectiva, ejecutar un proyecto productivo bien dirigido y terminar en tiempo. Para alcanzar estos propósitos una de las estrategias planteadas por TSP se refiere a establecer estándares de métricas para calidad y desempeño. Además, uno de los principales roles que propone TSP es el director de proceso/calidad, este es el encargado de asegurar que todo el equipo de desarrollo use de manera adecuada TSP para producir un producto libre de defectos.

En TSP el principal enfoque para alcanzar la calidad se enmarca en el manejo de defectos. Para ello, la fase de Prueba, dentro de sus ocho fases, ha sido determinada para el manejo de la calidad. Durante esta etapa se llevan a cabo las pruebas de sistema. Planeamiento, es otra de las fases para agrupar los procesos y prácticas de ingeniería. Desarrollar un Plan de Calidad representa una de las tareas que se especifica para dicha fase.

No sería atrevido asegurar que utilizando correctamente TSP es posible obtener productos de calidad a tiempo y dentro del presupuesto establecido; exigiendo un conjunto de parámetros que el equipo deberá cumplir para finalmente asegurar la calidad del producto desarrollado.

1.7 Proceso Unificado de Desarrollo del Software. Aspectos identificados para el aseguramiento de la calidad

Durante la investigación realizada para el desarrollo del presente trabajo se concluyó proponer el Proceso Unificado de Desarrollo del Software (RUP) como la metodología de desarrollo del software a seguir para asegurar la calidad por todos los beneficios que aporta al tema, y además porque también

es la misma determinada por el proyecto. Como punto de partida y referente al tema central de este trabajo, es válido aclarar que dicha metodología incluye una plantilla como guía para elaborar un Plan de Aseguramiento de la Calidad, el cual debe ser implementado durante todas las etapas del ciclo de vida software desarrollado.

Uno de los cinco principios en que está basado RUP es: “Enfocarse en la calidad”. O sea, asegurar y controlar la producción de software de alta calidad representa un objetivo planteado por esta metodología para satisfacer las necesidades de los clientes en un plazo y presupuesto predecible. En tal caso, pretende enfocarse en la calidad durante todo momento del ciclo de vida del desarrollo del software y no al final de cada iteración. El aseguramiento de la calidad se construye dentro del proceso en todas las actividades, involucrando a todos los participantes, utilizando medidas y criterios objetivos; permitiendo así, detectar e identificar los defectos de forma temprana.

RUP divide el proceso de desarrollo del software en 4 fases durante todo el ciclo de vida: Concepción, Elaboración, Construcción y Transición, cada una de ellas cuenta con sus respectivos flujos de trabajo que se van ejecutando de forma iterativa. Uno de estos flujos es el de Prueba, que se aplica durante todo el ciclo de vida de desarrollo del software y constituye de gran interés para los temas de la calidad puesto que las pruebas se enfocan principalmente en la evaluación y el aseguramiento de la calidad del producto. El hecho de efectuar pruebas en cada iteración, asegura la calidad del software que se está implementando, dando paso a la posibilidad real de eliminar disímiles errores antes de pasar a una siguiente etapa.

RUP facilita una guía específica de mejores prácticas en diferentes áreas claves. Entre las mejores prácticas determinadas por la Ingeniería de Software y seguidas por esta metodología está la Verificación de la Calidad, proporcionando las listas de verificación necesarias de revisión y plantillas de documentos que pueden ser aplicables como estándares de proyectos. Los desarrolladores de RUP están conscientes que encontrar y reparar los problemas de software es entre 100 y 1000 veces más caro después de su liberación.

A modo de conclusión, por todo lo antes mencionado, se ratifica usar RUP como metodología ideal de desarrollo de software a seguir, con el fin de lograr un mayor aseguramiento de la calidad para el producto desarrollado por el Proyecto Banco.

1.8 Criterios de Roger Pressman sobre el aseguramiento de la calidad

Pressman, en su libro *Ingeniería de Software un Enfoque Práctico*, plantea que: La garantía de la calidad o aseguramiento de la calidad, consiste en la auditoría y las funciones de información de la gestión. (PRESSMAN, 2001) Es una actividad esencial en cualquier empresa que elabora productos que van a ser usados por otros. SQA es un diseño de acciones planificado y sistemático que se requiere para asegurar la calidad del software. Plantea además, que es una "Actividad de protección" que se aplica a lo largo de todo el proceso de ingeniería del software con el objetivo de proporcionar la gestión para informar de los datos necesarios sobre la calidad del producto. Si estos datos identifican algún tipo de problema, inmediatamente se debe afrontar la situación y aplicar los recursos necesarios para resolverlos.

Los sistemas de software son cada vez más importantes en la sociedad moderna y están creciendo rápidamente en tamaño y complejidad. Forzado por la competencia y la tendencia a ciclos de vidas más cortos, la calidad de los productos debe ser cada vez mayor. Aparte de las técnicas modernas de especificación, diseño e implementación de software la introducción de un proceso de testeo es vital para asegurar un nivel de calidad apropiado.

Pressman considera que el aseguramiento de la calidad del software comprende una gran variedad de tareas asociadas, dentro de ellas se encuentra 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 garantía de calidad realizadas por el equipo de ingeniería de software y el grupo SQA son gobernadas por el plan. El plan identifica:

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

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

1.9 Aplicación del Plan de Aseguramiento de la Calidad en los proyectos software

1.9.1 En el mundo

Actualmente, en los proyectos software que se llevan a cabo en todo el mundo se está tomando muy seriamente el aseguramiento de la calidad según se incrementa la competencia del mercado; por tal razón se han trazado diversas estrategias para la evolución y logro de esta meta, una de las tareas identificadas como primordiales es establecer un Plan de Aseguramiento de la Calidad.

Ejemplos de instituciones internacionales que aplican este tipo de plan son:

- La compañía Clickway, líder en América Latina en el desarrollo de software, la cual se dedica a proveer soluciones informáticas, reconoce entre sus principales objetivos asegurar la calidad de sus productos, para ello tienen implementado un plan de aseguramiento de la calidad donde registran todos los cambios del proyecto y trazan toda una estrategia para asegurar su calidad.
- El proyecto de software abierto para el desarrollo de la Web móvil “MyMobileWeb”, tiene implementado un Plan de Aseguramiento de la Calidad.
- Netgrup es una consultora dedicada a la gestión de estructura de motores de datos y medición de experiencia de usuario. Desarrolla sus proyectos mediante una metodología establecida, en su etapa de planificación tienen determinado la realización de un Plan de Aseguramiento de la Calidad.
- La empresa informática nicaragüense, CASA, para garantizar la calidad establece la realización de un Plan de Aseguramiento de la Calidad.
- La Universidad de Chile en el Departamento de Ciencias de la Computación tiene un taller de proyectos de gestión de la calidad donde se ofrece un programa donde enseña cómo realizar un Plan de Aseguramiento de la Calidad para un proyecto.

1.9.2 En Cuba

La calidad del software cubano, en el momento que se desarrolla este trabajo, no se puede ver como un punto fuerte de la Industria Cubana del Software (ICSW), sino como algo que está naciendo recientemente con perspectivas de mejoras continuas; sin embargo, dada la importancia de este punto durante el desarrollo del software, tampoco se ha dejado totalmente rezagado, por lo que no es menos cierto que con relación al aseguramiento de la calidad, en muchas empresas y proyectos de

producción nacional se han implementado y se encuentran en ejecución los planes de aseguramiento de la calidad.

Se pueden citar como abanderados del tema: al Laboratorio de Química Inorgánica del INHEM, esto le aportó muchos beneficios pues permitió mantener la integridad de los datos emitidos y por tanto, satisfacer los requisitos establecidos previamente; al Centro de Biofísica Médica (CBM) de la Universidad de Oriente que determinó implementar un plan para asegurar su calidad y a la Empresa de Telecomunicaciones de Cuba (ETECSA) que también se decidió por aplicar mejoras continuas de procesos de software haciendo uso de un Plan de Aseguramiento de la Calidad.

A raíz de todo lo planteado anteriormente surgió la necesidad de verificar, de alguna forma, la óptima calidad de los productos que van saliendo al mercado. La creación de Calisoft (Centro de Calidad del Software) en el 2004 es un ejemplo de los intentos que están surgiendo por controlar la calidad y lograr entrar exitosamente al mercado internacional.

1.9.3 En la Universidad de las Ciencias Informáticas (UCI)

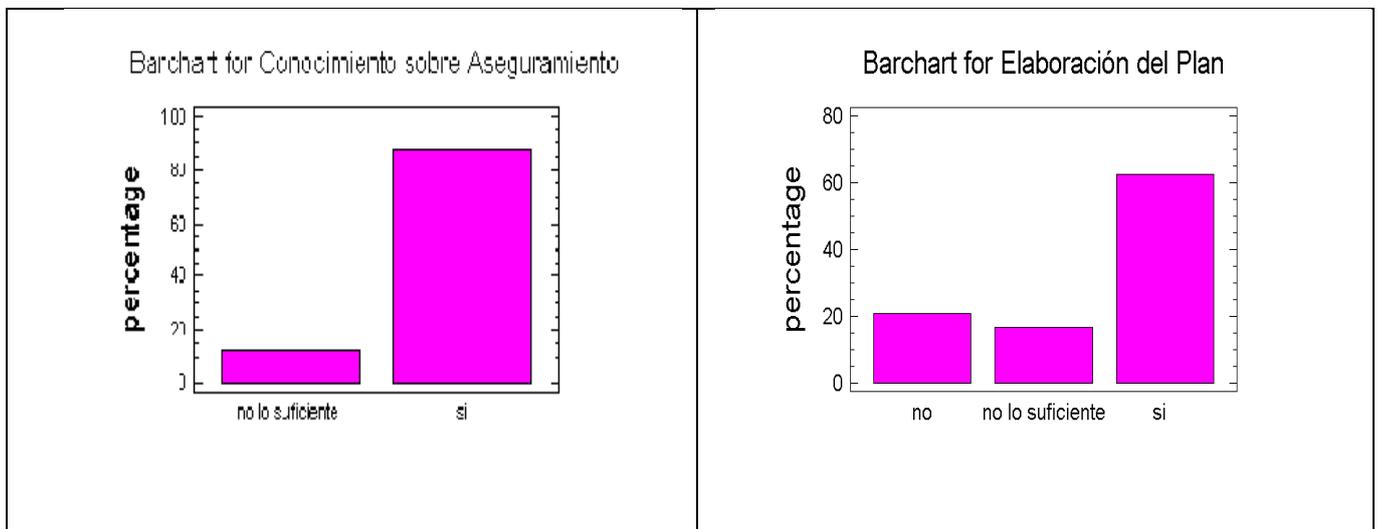
En la UCI se han tomado diferentes medidas para obtener productos con máxima calidad, y sobre todo crear un nivel de conciencia entre los miembros de sus proyectos con relación a la prioridad que representa este tema en aras de conquistar un reconocimiento en la ICSW. Llevar a cabo una meta como esta no ha sido un proceso acelerado, sino que en el transcurso de los cinco años de fundada la universidad, poco a poco se ha ido avanzado por este camino, logrando así un nivel de madurez superior durante el desarrollo del software.

Para tener una idea de cómo ha sido el proceso, es básico señalar que: como primer paso se comenzaron a aplicar los modelos de calidad por los miembros de calidad de cada proyecto, seguidamente en cada facultad se estableció un grupo de calidad conformado por aproximadamente 30 estudiantes, los cuales realizan pruebas durante el desarrollo del software a proyectos de otras facultades para finalmente ser entregado al Departamento Central de Calidad donde se le aplica una revisión final; a pesar de haber dado estos pasos, todavía no se ha logrado asumir en toda su magnitud el Plan de Aseguramiento de la Calidad, aún considerando las exigencias establecidas por la universidad y sólo un conjunto de proyectos están conscientes de la importancia de su desarrollo y aplicación.

En estos momentos existen en la UCI 122 proyectos productivos distribuidos entre sus 10 facultades, de ellos 24 son de exportación, lo que representa aproximadamente un 20% del total. Se decidió aplicar una encuesta a dichos proyectos (el modelo de la encuesta se incluye en el Anexo 2), con el fin de determinar algunas estadísticas generales vinculadas al aseguramiento de la calidad en los proyectos de la universidad. La razón para determinar el tamaño de esta muestra representativa tiene su explicación en la similitud que existe entre los proyectos de exportación y el Proyecto Banco. Los encuestados fueron los líderes de proyectos o responsables de calidad y los resultados obtenidos fueron los siguientes:

- El 87.5% (21) conoce sobre los temas de aseguramiento de la calidad, y el 12.5% (3) restantes solo tienen aisladas ideas.
- De los que conocen, el 90% (19) domina los conceptos relativos a qué es un Plan de Aseguramiento de la Calidad.
- De estos 19, solo el 79% (15) tiene implementado el plan dentro del proyecto, siendo usado en su totalidad tan solo por el 42% (8), de forma no lo suficiente por el 32% (6) y el restante 5% (1) ni lo tenía en cuenta.

En las gráficas se muestran de forma más clara los resultados anteriores llevados a por cientos con relación al total de la muestra.



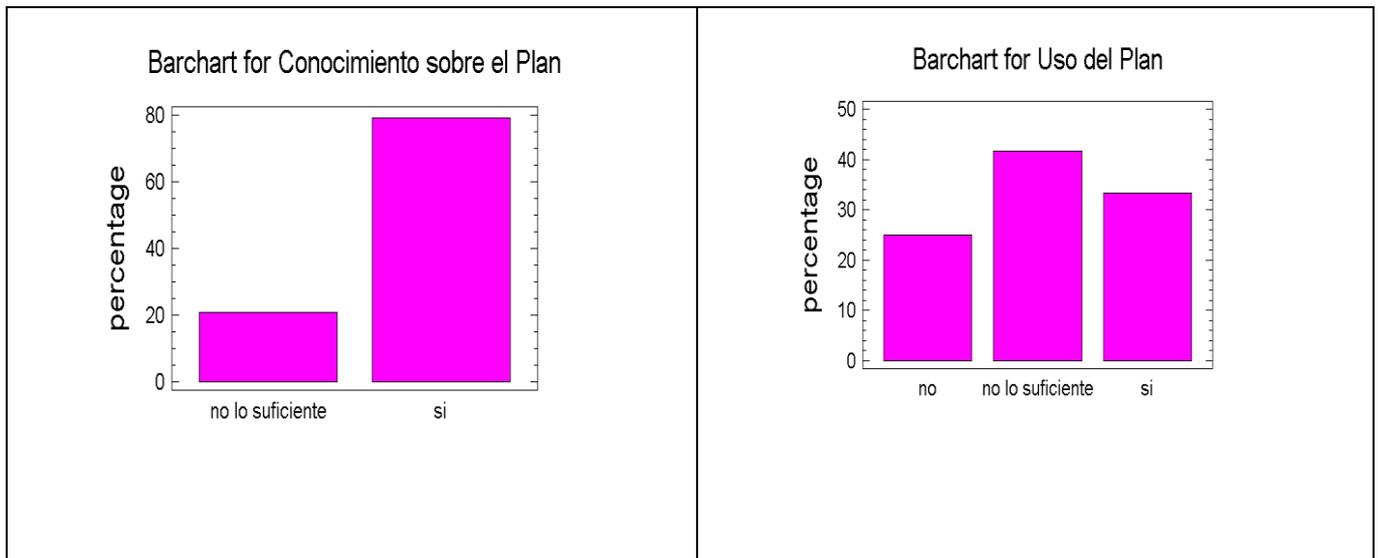


Figura 3: Resultados de la encuesta aplicada sobre temas de aseguramiento de la calidad

Al efectuar un análisis de los datos se puede concluir en que, sólo la tercera parte de los proyectos usan de forma efectiva un Plan de Aseguramiento de la Calidad, razón por la cual se justifica el por qué la mayoría de los proyectos que llegan al Laboratorio de Pruebas de Liberación pasan hasta por 8 iteraciones de pruebas para encontrarse en condiciones mínimas de ser entregados a un cliente, eso sin contar con el gran número de productos que son abordados por serios fallos de funcionalidades, existencia de errores ortográficos en documentos oficiales e inestabilidad en las versiones. Por otro lado resaltar como aquellos proyectos que de forma activa desempeñan actividades de aseguramiento de la calidad guiados por su plan, son liberados en a penas 2 o 3 iteraciones, nunca son abortadas sus pruebas y obtienen excelentes resultados frente a sus clientes.

1.10 ¿Por qué un Plan de Aseguramiento de la Calidad es único para cada proyecto software?

En el mundo del software se fomenta el desarrollo basado en la reutilización, desde esta perspectiva surge la polémica de: ¿Por qué no usar un Plan de Aseguramiento de la Calidad elaborado con anterioridad para otro proyecto software? Al responder a esta interrogante se pretende demostrar, a la vez, el carácter único que tiene cada plan.

Para comenzar a fundamentar el planteamiento anterior es esencial partir del análisis de que: *un Plan de Aseguramiento de la Calidad es la contraparte del proceso de desarrollo de un producto y cada*

producto tiene un proceso de desarrollo único. A dicho proceso se le incluyen una serie de aspectos a considerar, exclusivos en un proyecto para obtener un producto determinado, tales como:

- Categoría del proyecto (Portal, Gestión, Multimedia, Realidad Virtual).
- Tipo de aplicación (Web, Desktop).
- Objetivos del producto.
- Metodología de desarrollo del software.
- Roles involucrados en el aseguramiento de la calidad.
- Cantidad de procesos definidos.
- Definición de los flujos de trabajo (si se fusionaron).
- Herramienta de modelado visual utilizada.
- Lenguaje de programación utilizado.
- Sistema gestor de base de datos seleccionado.

Es muy difícil que dos proyectos coincidan en todos esos aspectos; si hipotéticamente existieran dos proyectos a los que se le ha dado la tarea de desarrollar productos iguales y uno de ellos realiza un Plan de Aseguramiento de la Calidad, no es válido reutilizarlo de forma íntegra en el otro, pues gran parte de los aspectos tratados en el mismo, son propios de cada proyecto. Por ejemplo, el alcance especifica los proyectos con los que está involucrado cada plan y esto es diferente en cada caso, al igual que los objetivos de calidad trazados a partir de los objetivos generales del proyecto. Es imposible que dos proyectos tengan los mismos objetivos, en tal caso estarían desarrollando el mismo producto y ni aún así el plan podría ser el mismo, ya que no se habría elaborado para ser aplicado por las mismas personas.

Con relación a la organización del equipo de calidad, es poco probable que dos proyectos tengan los mismos roles ya que estos son determinados teniendo en cuenta la importancia que les conceda al proyecto la dirección de la facultad, ahí entrarían aspectos como la cantidad de recursos físicos designados, cantidad de estudiantes y profesores seleccionados, etc. Habría que considerar también si los proyectos tienen igualdad de recursos para seleccionar las herramientas a utilizar. A esto se puede añadir que los registros de calidad son específicos para cada proyecto, al igual que el lugar donde serán almacenados y el establecimiento de por cuánto tiempo estarán disponibles. Por último, en cada proyecto se deben trazar diversas actividades de entrenamiento obedeciendo a las necesidades

propias de su equipo de calidad, por lo que la capacitación está en dependencia de los conocimientos requeridos por sus miembros para lograr ejecutar con eficiencia las actividades propuestas en el plan.

Analizando esta temática desde un punto de vista específico para la universidad y con objetivo de probar los argumentos planteados, se realizaron entrevistas a los 15 proyectos (de la encuesta mencionada en el Epígrafe 1.11 del presente capítulo) que efectivamente tenían elaborado un Plan de Aseguramiento de la Calidad.

En las entrevistas se obtuvo información sobre el proceso que desarrollan dichos proyectos, para luego compararlo con el que está siguiendo y tiene planificado desarrollar el proyecto Banco. Se abordaron una serie de preguntas descriptivas del proceso como las que ya se mencionaron. Los resultados fueron los esperados, o sea, la gran mayoría coincidió con las particularidades del proyecto Banco en varios de los aspectos, pero ninguno coincidió con la totalidad. A partir de ahí, la idea de utilizar uno de esos planes queda anulada, pero es válido aclarar que sí es posible que sean usados como guías para desarrollar algunos de los puntos comprendidos en el plan dada la concordancia entre diferentes criterios característicos de los productos y los procesos definidos por los proyectos.

Por tanto, se puede afirmar que: un Plan de Aseguramiento de la Calidad es propio para cada proyecto al considerar la imposibilidad de que un plan realizado para asegura la calidad de su producto sea aplicable a otro a otro proyecto; y a modo de reafirmación, aunque se quisiera realizar un mismo producto por dos grupos diversos no es posible que sean trazadas las mismas estrategias.

1.11 ¿Implementar un Plan de Aseguramiento de la Calidad minimiza el problema existente?

La implementación del plan ayuda a corregir los errores en una etapa temprana del desarrollo del software, pues proporciona diversos mecanismos que favorecen la garantía de la calidad, evitando que con el avance del proyecto estos errores sean más costosos y difíciles de eliminar.

Un ejemplo de ello es en el proyecto SIGEP (más conocido como Prisiones) de la facultad, donde en investigaciones realizadas se pudo confirmar que aplicando el plan que tenía elaborado se consiguió detectar el 75% de los fallos antes que el producto pasara al Laboratorio de Pruebas de Liberación. Logrando de esta forma entregas efectivas al cliente, en tiempo y por ende, un costo de producción mucho menor y una reducción considerable de posibilidades de pérdidas de presupuesto por la extendidas sesiones de pruebas que sufren la mayoría de los proyectos.

Uno de los puntos exigidos en el plan lo representa el establecimiento de un conjunto de actividades que van comprobando el desarrollo de un producto confiable y con calidad en cada una de las etapas del ciclo de vida del software. Si estas actividades son cumplidas correctamente, los errores se irán eliminando y no se arrastrarán a la siguiente etapa, de lo contrario es mucho más difícil detectar y erradicar el error.

Aplicando correctamente un Plan de Aseguramiento de la Calidad es posible afirmar que el producto final incluirá menos errores latentes, resultando así un menor esfuerzo y un menor tiempo durante las actividades de prueba y soporte, por lo que se podrán reducir los costos de mantenimiento y a la par el costo total del ciclo de vida del software.

Además, implementar el plan permite verificar si se están cumpliendo los objetivos del producto pues a partir de ellos se establecen los objetivos de calidad a cumplir como uno de los aspectos esenciales del plan; de esta manera se proporciona una mayor fiabilidad y, por lo tanto, una mayor satisfacción del cliente con el producto final.

Como punto final, para darle respuesta a la interrogante planteada en este epígrafe se puede concluir en que: la realización y aplicación de un Plan de Aseguramiento de la Calidad en un determinado proyecto permite garantizar que el producto que desarrolla el mismo se beneficie con una calidad superior. En síntesis, el plan, básicamente lo que permite es garantizar que en cada etapa de desarrollo del software se esté realizando un producto con una elevada calidad.

1.12 Conclusiones

En el presente capítulo se realizó un estudio de los conceptos básicos relacionados con los temas de calidad, su aseguramiento, sus costos, las métricas y el plan que es usado con el objetivo de garantizar la calidad durante los procesos de desarrollo del software. Se investigó sobre los diversos modelos de evaluación de la calidad y estándares, enfatizando en las ventajas de su uso, en los enfoques que abordan sobre el aseguramiento de la calidad; y se llegó a determinar que no es recomendable escoger uno solo, sino hacer una selección para luego tomar los aspectos de interés de cada uno. También se realizó un estudio a nivel internacional, nacional y local con relación a la aplicación del plan en los proyectos software. Se logró justificar por qué el plan tiene carácter único y finalmente se demostró cómo su implementación ayuda a resolver los problemas relacionados con la calidad del software, aunque no siempre se tienen en cuenta y es subestimado.

CAPÍTULO 2: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

2.1 Introducción

Según todo lo visto en el capítulo anterior, es natural que se piense en el aseguramiento de la calidad como una necesidad con matices de carácter obligatorio; a pesar de que ya por fuerza se haya aterrizado a la realidad y, los desarrolladores y clientes son conscientes de que no existe una calidad perfecta o absoluta. Sin embargo, es válido aclarar que: enmarcándose en un contexto dado y cumpliendo con una serie de acciones convencionales delimitadas para lograr este fin, sí es posible alcanzar una calidad necesaria y suficiente.

Las acciones indicadas a desarrollar para asegurar la calidad del software son:

- Establecimiento de un sistema de calidad.
 - Gestión de la calidad.
 - Planificación de la calidad.
 - Definición de políticas de calidad.
- Uso de técnicas de verificación y validación del software.
 - Revisiones e inspección de los productos de software.
 - Pruebas de programas.
- Gestión de la Configuración del Software.
 - Uso de normas y estándares de calidad.
 - Evaluación y mejoramiento de los procesos de software.

El presente capítulo es el que sustenta el mayor peso en el trabajo desarrollado, de forma general aquí es donde se describe paso a paso cómo son llevadas a cabo todas las acciones mencionadas anteriormente dentro de un Plan de Aseguramiento de la Calidad y, con un carácter más específico, se propone una solución de cómo implementar dicho plan para el Proyecto Banco. Aclarar a la vez que, el plan que se presenta es sólo una propuesta, por lo que no quiere decir que sea este, exactamente, el que usarán los miembros del proyecto. En tal caso se recomienda tomarlo como una propuesta dinámica y flexible que da la posibilidad de estar sujeta a cambios, mejoras y actualizaciones, siempre y cuando se adapte a las condiciones del proyecto y a las eventualidades que puedan surgir durante su avance.

El plan no se mostrará en este capítulo de forma inmediata, sino al final en el Anexo 1, lo cual no significa que se deba referenciar como un documento independiente, por el contrario, estará compuesto, además, por otros documentos que se anexarán también concluyendo el último capítulo.

2.2 Descripción de las secciones correspondientes al Plan de Aseguramiento de la Calidad

A continuación se describe de forma genérica los elementos que deben incluirse en cada acápite del plan; teniendo en cuenta las especificaciones de la plantilla propuesta por el DCS, que fue la utilizada para la elaboración del Plan de Aseguramiento de la Calidad del Proyecto Banco.

2.2.1 Introducción

2.2.1.1 Propósito

La siguiente sección especifica el propósito que se persigue con la elaboración del Plan de Aseguramiento de Calidad, lo cual sería describir, de forma general, cómo se asegurará la calidad del producto, a través de los artefactos, herramientas y procesos con los que se pretende alcanzar los objetivos de calidad del proyecto.

2.2.1.2 Alcance

Esta sección hace referencia al alcance del Plan de Aseguramiento de Calidad elaborado según la guía o las guías seleccionadas, puntualizando además, si es necesario adaptarlas a las condiciones específicas del proyecto. Incluye también los proyectos con los que se involucra dicho plan.

2.2.1.3 Definiciones, Acrónimos y Abreviaturas

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

2.2.1.4 Referencias

En esta sección se presenta un listado de documentos a los que se hace referencia en el Plan de Aseguramiento de la Calidad y que por su estrecho vínculo con el trabajo, se puede decir que son de obligatoria consulta.

2.2.1.5 Resumen

El siguiente apartado debe especificar una síntesis de los aspectos abordados en el Plan de Aseguramiento de la Calidad, facilitando al lector una panorámica general de las actividades a desarrollar para alcanzar la calidad del producto software.

2.2.2 Objetivos de Calidad

En este punto se debe incluir los requerimientos de proyecto que están alineados con los requerimientos de calidad, vale señalar que los objetivos de calidad definidos parten de los objetivos básicos del proyecto, asegurando así su cumplimiento.

2.2.3 Gestión

2.2.3.1 Organización

En esta sección se plantea cómo estará estructurado el equipo de calidad del proyecto, se identifican los roles, se da una breve descripción al respecto y, se detallan los conocimientos mínimos que debe poseer cada miembro para poder desempeñar un rol específico.

2.2.3.2 Tareas y Responsabilidades

Aquí se proponen las tareas que se realizarán para garantizar la calidad del producto a desarrollar, estas se presentan teniendo en cuenta el avance del proyecto, dentro de ellas se incluyen revisiones, auditorías y pruebas; además se especifican el o los roles responsables en ejecutarlas.

2.2.4 Documentación

En esta sección se da un listado de los documentos específicos del proyecto, que fueron consultados y que sirvieron de utilidad en la confección del Plan de Aseguramiento de la Calidad.

2.2.5 Métricas

La siguiente temática aborda las definiciones de métricas que serán utilizadas para seguir el comportamiento de los defectos introducidos durante el desarrollo del producto y, de esta forma tomar acciones que prevean errores, además de acelerar el proceso de revisiones consecuentemente con ello.

2.2.6 Estándares y Guías

El propósito de esta sección es definir los estándares, prácticas, convenciones y guías utilizadas para lograr los objetivos de calidad definidos en el plan, los cuales han de ser aplicados por el equipo de desarrollo del proyecto al cual se hace referencia.

2.2.7 Plan de Revisiones y Auditorías

2.2.7.1 Tareas generales de Revisiones y Auditorías

En el siguiente punto se indican las tareas a realizar por el equipo SQA para armar su propia infraestructura y las tareas de control de los productos y procesos críticos respecto a la calidad del producto, incluyendo además, cómo se llevarán a cabo cada una de ellas. Describe brevemente cada revisión y auditoría que se efectuará y, para cada tipo, se identifica los artefactos del proyecto que serán el asunto de la revisión o auditoría.

Se requiere como mínimo que se realicen revisiones e inspecciones de los requerimientos y del diseño, lo mismo que revisiones al terminar cada etapa. Las revisiones son discusiones de los artefactos propuestos y las inspecciones se hacen sobre los artefactos terminados que se presentan.

2.2.7.2 Cronograma

Aquí se establece un cronograma a seguir para todas las revisiones y auditorías de SQA planificadas. Este debe incluir las revisiones y auditorías programadas en las fechas principales del proyecto, así como revisiones que son provocadas por la entrega de artefactos del proyecto. El cronograma de SQA puede estar incluido dentro de un cronograma general del proyecto. En ese caso, indica de qué forma se identifican las tareas de SQA dentro de las tareas del proyecto.

2.2.7.3 Organización y responsabilidades

El presente punto está enfocado en delimitar las responsabilidades de los roles definidos dentro del proyecto. En esta sección se listan los grupos específicos o individuos a ser involucrados en cada una de las actividades de revisión y de auditoría identificadas; y además, se describe brevemente las tareas y responsabilidades de cada uno.

2.2.7.4 Resolución de problemas y actividades de corrección

De forma general esta sub-sección describe los procedimientos para informar y manejar problemas identificados durante las revisiones y auditorías que sean efectuadas en las diferentes etapas del proyecto.

2.2.8 Herramientas, Técnicas y Metodologías

Aquí se detallan las herramientas, técnicas o metodologías específicas que serán usadas para llevar a cabo las actividades de revisión y de auditoría identificadas en el plan. Se debe describir el proceso explícito a seguir para cada tipo de revisión o auditoría; y además, especificar las listas de chequeo a utilizar.

2.2.9 Pruebas y Evaluación

En el siguiente apartado se detalla cómo el equipo de calidad realizará los distintos tipos de pruebas al producto que se desarrolla, debe describirse de forma específica, para ello se propone realizar un Plan de Pruebas concreto para el proyecto.

2.2.10 Flujo de Trabajo

Esta sección se incluye para identificar los procesos que son llevados a cabo por el equipo de calidad para asegurar la calidad del producto software durante los flujos de trabajos de Pruebas, Revisiones y Auditorías. Además, también se hace referencia al flujo general que vincula las actividades de calidad desarrolladas durante todo el ciclo de vida del software y su interacción con los flujos de trabajo particulares del proyecto.

2.2.11 Herramientas, Técnicas y Metodologías

Aquí se define un listado general de todas las herramientas, técnicas y metodologías seleccionadas a ser empleadas para apoyar el desarrollo de las actividades encaminadas a garantizar la calidad del software durante todo su ciclo de vida.

2.2.12 Resolución de Problemas y Acción Correctiva

En este punto, al ser detectado algún error, problema o incongruencia en cualquiera de las etapas de revisiones y pruebas, se levantará un documento (Informe de No Conformidades) que será relacionando con la no conformidad encontrada. De forma más detallada este espacio: Describe los procedimientos (o hace referencia a ellos) a ser utilizados para reportar, monitorear y resolver problemas identificados durante el desarrollo del software con el fin de manejar desviaciones, de lo real respecto a lo esperado.

2.2.13 Registros de Calidad

Aquí se describe la documentación que se requiere que el grupo de SQA produzca durante el tiempo de vida del proyecto, o sea, descripciones de varios registros de calidad, incluyendo cómo y dónde cada tipo de registro se guardará y por cuánto tiempo. Además de describir la identificación, recolección, clasificación, almacenamiento, mantenimiento y depuración de los registros de calidad.

2.2.14 Entrenamiento

En esta sección se establecerá un listado referente a las actividades de entrenamiento que son necesarias, para lograr que el equipo de calidad logre ejecutar con óptimos resultados las actividades de aseguramiento de la calidad establecidas en el plan.

2.3 Aspectos relacionados con la forma de solución para diferentes secciones del Plan

2.3.1 Forma de solución para establecer los Objetivos de Calidad del Proyecto

Partiendo de los objetivos básicos del proyecto se identificaron un conjunto de objetivos de calidad; antes de hacer referencia a ellos, vale destacar que, dichos objetivos se establecieron según los criterios de calidad propuestos por la norma cubana ISO/IEC 9126 a la que se hizo mención en el Epígrafe 1.3.2 del Capítulo # 1. De forma general la estrategia para definir los objetivos de calidad fue:

➤ *Asegurar que se logre agrupar de manera flexible y configurable todas las operaciones que desarrollan los diferentes bancos del país.* El atributo del software para definir este objetivo, enfocado a la calidad y relacionado con la Norma ISO/9126 estaría vinculado con la *Funcionalidad*. Según esta norma, la *Funcionalidad*, es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo determinadas condiciones. El software que se pretende desarrollar debería ser capaz de agrupar de manera flexible y configurable todas las operaciones bancarias del país. A partir de esta característica, se relacionan además dos sub-características: *Idoneidad* e *Interoperabilidad*. La primera está muy ligada al concepto de *Funcionalidad* en el sentido de que hace referencia a la capacidad para mantener un conjunto apropiado de funciones, para las tareas y objetivos de los usuarios. En el caso de la segunda, también se enfoca a la capacidad, pero desde la perspectiva que el producto software pueda interactuar recíprocamente con uno o más sistemas especificados; o sea, se desea lograr que el sistema a desarrollar pueda interactuar con otros sistemas para el intercambio de información.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- *Verificar que se adquiriera la parametrización de las funcionalidades, logrando flexibilidad y configuración del sistema.* Según la Norma ISO/9126, una característica que es posible seleccionar para definir este objetivo desde el punto de vista de la calidad sería la *Funcionalidad*, porque partiendo de lo planteado en su contenido, el software debe ser capaz de proporcionar funciones que satisfacen las necesidades específicas del cliente, en este caso sería la parametrización de las funcionalidades. De ahí, también es posible relacionar el presente objetivo con la necesidad de que el producto final se enfoque en la *Idoneidad*. Orientándose en la definición de esta sub-característica, el objetivo de calidad sería: verificar la capacidad del software para mantener la parametrización de las funcionalidades de una forma apropiada, o sea flexible para desempeñar las tareas especificadas por los usuarios. Otra característica que se vincula con este objetivo es la *Usabilidad* que plantea la capacidad del producto software para ser comprendido y utilizado por el usuario. Si se logra la parametrización de las funcionalidades, el software sería menos complejo y por consiguiente más comprensible para el usuario, además de que se ganaría en flexibilidad, durante la configuración del sistema, y se cumpliría con la sub-característica de *Operabilidad*, permitiéndole al usuario operarlo y controlarlo.
- *Estandarización de la información que se maneja y registra entre los diferentes bancos.* Un atributo software para definir este objetivo enfocado a la calidad, según la Norma ISO/9126 serían la *Funcionalidad*. El mismo fue seleccionado porque se pretende asegurar la capacidad del software para proporcionar la estandarización de la información manejada por el sistema; las sub-características correspondientes a este atributo según el objetivo mencionado serían: *Idoneidad* e *Interoperabilidad*. La *Idoneidad* se puede ver desde la perspectiva que el software sea capaz de mantener la estandarización de la información manejada entre los diferentes bancos y de esta forma poder desarrollar otras tareas de interés para el usuario como, ofrecer informaciones estadísticas generalizadas. La *Interoperabilidad* se identifica desde el punto de vista que el objetivo se refiere al manejo y registro de la información entre diferentes bancos, o sea, que el software sea capaz de interactuar recíprocamente con uno o más sistemas para realizar funciones.
- *Disminuir los tiempos de respuestas en los servicios.* Para dicho objetivo del proyecto, es posible identificar como características del software a la *Funcionalidad* en aras de definir el objetivo de calidad correspondiente según la Norma ISO/9126. La *Funcionalidad* porque se puede ver enfocada considerando la necesidad de asegurar la capacidad del software para proporcionar tiempos de respuesta más cortos, bajo las condiciones de circunstancias especificadas por los usuarios. Con ello

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

viene aparejada la sub-característica de *Idoneidad* refiriéndose al hecho de lograr la capacidad del software para mantener, en dependencia de los servicios, los cortos tiempos de respuestas ya referidos anteriormente. Otra sub-característica que se relaciona es la *Precisión* pues se pretende además, asegurar la capacidad del software para que estos tiempos de respuestas posean un grado de exactitud elevado en dependencia de las necesidades establecidas.

➤ *Mejorar la centralización de la información para la toma de decisiones por parte de la directiva del banco.* El atributo de software que es posible definir para este objetivo, enfocado a la calidad, según la Norma ISO/9126, sería la *Funcionalidad*. Partiendo de lo planteado por la norma para este atributo, se puede analizar el objetivo desde la perspectiva de asegurar la capacidad del software para proporcionar mejoras en la centralización de la información según las condiciones establecidas por los usuarios. La sub-característica correspondiente con este objetivo sería la de *Idoneidad*. Enfocando su definición a dicho objetivo se concluye que: se desea lograr la capacidad del software para mantener las mejoras continuas de centralización de la información y así poder desempeñar satisfactoriamente la toma de decisiones, según los criterios de interés por parte de la directiva de los usuarios que usan el software.

➤ *Contar con una documentación bien detallada de todos los procesos del negocio bancario, además de la documentación del sistema.* Según la Norma ISO/9126, es posible seleccionar la *Usabilidad* como una característica de calidad del software de necesario cumplimiento, tomando como punto de partida el objetivo antes mencionado y su relación con la definición de dicho atributo. Una vez realizado ese análisis, se traza como objetivo de calidad, asegurar la capacidad del producto software a ser comprendido, aprendido y utilizado, contando con una documentación bien detallada de los procesos seguidos durante su desarrollo y, con una documentación del sistema incluida que haga atractivo su uso. Para esta característica, se pueden vincular las sub-características de *Comprensibilidad* y *Cognoscibilidad* teniendo en cuenta las definiciones planteadas por la norma y el presente objetivo. La *Comprensibilidad* es posible cuando se enfoca el objetivo de calidad a conseguir la capacidad del producto software para permitirle al usuario aprender su aplicación mediante la documentación de los procesos; y la *Cognoscibilidad* cuando se enfoca el objetivo de calidad en asegurar la capacidad del sistema para permitirle al usuario operarlo y controlarlo, a través de una documentación detallada del sistema.

➤ *Garantizar la seguridad que el sistema ejecute las operaciones bancarias en dependencia del nivel de acceso, tal cual ha sido especificado para cada uno de los usuarios.* Un atributo del software para definir este objetivo, enfocado a la calidad y relacionado con la Norma ISO/9126 estaría relacionado con la *Funcionalidad*. Partiendo de lo planteado en su contenido, el software debe ser capaz de garantizar la seguridad de las operaciones del sistema teniendo en cuenta los requisitos especificados por los clientes. Para esta característica, se puede vincular la sub-característica de *Seguridad (Informática)*. Enfocando su definición al objetivo quedaría como la capacidad del producto software para proteger las operaciones bancarias, o sea para que personas o sistemas desautorizados no puedan leer o modificar los mismos, y por otra parte las personas o sistemas autorizados tengan el acceso correspondiente a ellos.

2.3.2 Estructura de Organización propuesta para el equipo de calidad del proyecto

Existen casi tantas estructuras de organización de personal para el desarrollo de software como organizaciones que se dedican a ello. La mejor estructura de equipo depende del estilo de gestión de una organización, el número de personas que compondrá el equipo, sus niveles de preparación y la dificultad general del problema. (PRESSMAN, 2001)

Los participantes de un proyecto software constituyen el factor más importante para desarrollar con éxito los procesos de ingeniería de software y por ende el éxito del proyecto. Sin embargo, a este punto no se le da la atención requerida por parte de las empresas desarrolladoras de software. La composición de un proyecto software debe tomarse como una tarea primordial de la que se necesita un estudio previo. Según Pressman para que esta actividad se ejecute de forma eficaz, el equipo del proyecto necesita organizarse de manera que maximice las habilidades y capacidades de cada persona. De igual forma esta política se propone aplicar para determinar la estructura del equipo de calidad del Proyecto Banco.

Con motivo de conformar esta estructura de una forma más eficiente se realizó una investigación relacionada con los diferentes tipos de roles que se pudieran proponer. Para la selección de dichos roles se escogieron los que presenta RUP, tomando en consideración aquellos que se adecuan al desarrollo y características propias del proyecto; y partiendo además de la forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo) proporcionada por esta metodología.

Persiguiendo una genuina evolución del proyecto se expone la base de conocimientos mínimos que deberá poseer la persona que ejerza cada uno de los roles propuestos para lograr ejecutar su trabajo con eficiencia, en conjunto con algunas de las tareas responsabilizadas a desarrollar por cada rol. Luego de algunas indagaciones sobre el comportamiento actual de diferentes proyectos de desarrollo de software en la universidad, se llegó al consenso de que: la persona que ejerza cada rol, específicamente, será la encargada de responder ante cualquier exigencia u obligación. De esta forma se garantiza que el equipo continúe desenvolviéndose normalmente si uno de los miembros, por cualquier motivo, deja de ejercer la función que realizaba al colocar a otra persona a ocupar dicho rol; y a la vez se disminuyen los riesgos de no haber cumplido satisfactoriamente con su trabajo.

Finalmente los roles propuestos fueron: Responsable de Calidad, Diseñador de Pruebas, Revisor Técnico, Probador. El rol Responsable de Calidad es el más importante de los especificados, siempre aclarando que RUP considera que la calidad es compromiso de todos los miembros del equipo sin concentrarse únicamente en una sola persona. El mismo, es el que debe verificar que se realicen las actividades relacionadas con el aseguramiento de la calidad, y quien asigna las tareas a desempeñar dentro del equipo de calidad, además de comprobar su correcto cumplimiento.

2.3.3 Estrategia para determinar las tareas y responsabilidades propuestas

Según Pressman, *la calidad de un sistema, aplicación o producto es tan buena como los requisitos que describen el problema, el diseño que modela la solución, el código que conduce a un programa ejecutable y las pruebas que ejercitan el software para detectar errores*; razón por la cual en cada flujo de trabajo del software se debe ir chequeando, rigurosamente, el trabajo de los desarrolladores intentando asegurar su efectividad. Ahí es donde entra en acción el equipo de calidad del proyecto asumiendo una serie de tareas que deberán realizar conscientes de la importancia de las mismas. De forma general, el equipo de calidad responde por la ejecución de cada una de esas actividades, pero las responsabilidades básicas se distribuyeron de forma individual teniendo en cuenta el tipo de actividad en cuestión y el rol dentro del equipo de calidad capacitado para desempeñarla.

Con el objetivo de proponer las actividades a desarrollar por el equipo de calidad del proyecto para asegurar la calidad del producto, se tomó como referencia las actividades específicas por Pressman en su libro *Ingeniería de Software un Enfoque Práctico*. Las Actividades de SQA según Pressman son:

- Participación en el desarrollo de la descripción del proceso de software del proyecto.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- Revisión de las actividades de ingeniería del software para verificar su ajuste al proceso de software definido.
- Auditoría de los productos de software designados para verificar el ajuste con los definidos como parte del proceso del software.
- Asegurar que las desviaciones del trabajo y los productos del software se documentan y se manejan de acuerdo con un procedimiento establecido.
- Registrar lo que no se ajuste a los requisitos e informar a sus superiores.

Todas estas actividades se ven reflejadas de una forma u otra en las tareas propuestas en el Plan de Aseguramiento de la Calidad para el Proyecto Banco, siempre aclarando que, también le fueron incluidas las actividades de pruebas. En total se especifica la realización de 10 RTF, 2 tipos de auditorías y 4 tipos de pruebas. Más adelante en los Epígrafe **2.3.6** y **2.3.7** se abordarán con un mayor nivel de detalle estos tipos de revisiones, auditorías y pruebas.

De las 10 revisiones, 4 se realizarán de forma general para el alcance del proyecto, las cuales serían: Revisión de los procesos del negocio, Revisión de la captura de requisitos, Revisión de los casos de uso del sistema, Revisión de la gestión de la configuración; las restantes se realizarán por cada módulo definido. De los 2 tipos de auditorías, la correspondiente a la Auditoría de Procesos incluye la realización de: Auditoría a la Arquitectura, Auditoría al Diseño, Auditoría al Diseño Gráfico, Auditoría a la Base de Datos, Auditoría a la Gestión de Configuración. El otro tipo de auditoría corresponde a la Auditoría al Sistema de Calidad. Todas se ejecutarán concluyendo la implementación de cada módulo al igual que las Pruebas de Caja Negra; y ya finalizados la totalidad de los módulos se procederá a realizar las Pruebas de Integración del Sistema, las Pruebas de Seguridad y las Pruebas de Aceptación del Cliente.

Finalmente, con la realización de estas actividades se pretende brindar la posibilidad de avalar un adecuado desarrollo durante la elaboración del producto en sus diferentes etapas.

2.3.4 Especificaciones de las métricas propuestas en el plan

La validez y la confiabilidad son los dos temas más importantes de la calidad. Estos términos deben estar bien claros antes de proponer, usar y analizar una métrica. Lograr que la medición del software sea válida y consistente requiere de un contexto dado donde sea posible determinar un conjunto de procedimientos para modelar atributos y definir métricas. Preguntas tales como: ¿Por qué se mide?,

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

¿Qué se mide?, ¿Cómo se mide?, ¿Qué métricas son necesarias?, es inevitable formularse para situar el marco de medición y evaluar su idoneidad.

El estándar IEEE 1061 plantea que, para desarrollar un juego de métricas para un proyecto, se debe crear una lista de factores de calidad que son importante para el mismo.(SOCIETY, 1998) Se asocia con cada factor de calidad una métrica directa lo que sirve como una representación cuantitativa de un factor de calidad.

Una *línea base de métricas* es una recopilación de métricas que sirve para establecer indicadores. La selección de las métricas, para establecer una línea base de métricas se efectuó basándose en los objetivos, o sea, el propósito de desarrollar este trabajo radica en proponer un Plan de Aseguramiento de la Calidad para el Proyecto Banco, luego, de forma más específica, el propósito de dicho plan parte de proponer una pauta general a seguir, centrada en asegurar la calidad del software desarrollado.

Tomando como referencia estas ideas se plantea implementar *métricas de calidad* al software con el objetivo de medir los procesos y el producto; lo cual significa un medio para evaluar la calidad del proyecto software en general. Las *métricas de calidad*, proporcionan una idea de cómo se ajusta el software a los requisitos explícitos e implícitos del cliente; es decir, cómo se va a medir para que el sistema se adapte a los requisitos solicitados por el cliente.

Según Pressman las métricas de la calidad del software, se centran en el proceso, en el proyecto y en el producto. Además, mediante el desarrollo y análisis de una línea base de métricas de calidad, una organización puede actuar en aras de corregir las áreas del proceso del software que son la causa de los defectos del software, evaluando la calidad en tiempo real.

En el plan fueron seleccionadas un conjunto de métricas de aplicación a ser sintetizadas durante un nivel inicial de maduración las cuales son descritas a continuación:

Métrica para la medición de la característica funcionalidad

➤ *Métricas de idoneidad*: Las métricas externas de idoneidad deben ser capaces de medir un atributo, como es la ocurrencia de un funcionamiento insatisfactorio o la ocurrencia de una operación insatisfactoria. Un funcionamiento u operación insatisfactoria puede ser:

- Funcionamiento u operación que no se desempeña de la forma especificada en el Manual de Usuario o la Especificación de Requisitos.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- Funcionamiento u operación que no provee una salida aceptable o razonable al tomar en consideración un objetivo específico de las tareas del usuario.

➤ *Métricas de exactitud:* Las métricas externas de precisión deben ser capaces de medir un atributo relacionado con la frecuencia con que los usuarios se encuentren con la ocurrencia de una falta de exactitud o de precisión, como puede ser:

- Resultados incorrectos o imprecisos causados por datos inadecuados; por ejemplo: un dato con pocos dígitos significativos para un cálculo de precisión;

- Inconsistencia entre el procedimiento de operación actual y el descrito en el manual de operación;

- Diferencias entre el resultado actual y el razonablemente esperado producto de una tarea ejecutada durante la operación.

➤ *Métricas de interoperabilidad:* Las métricas externas de interoperabilidad deben ser capaces de medir un atributo, como es el número de funciones o la ocurrencia de la menor incomunicación que involucre a datos y comandos o instrucciones que sean transferidos entre el producto de software y otros sistemas, otros productos de software u otros equipos a los cuales está conectado.

Métricas para la medición de la característica usabilidad

➤ *Métricas de comprensibilidad:* Las métricas externas de comprensibilidad deben ser capaces de valorar cómo un nuevo usuario podría comprender:

- si el software es idóneo para la aplicación a la cual se destina, y

- cómo el software puede ser usado para una tarea en particular.

➤ *Métrica de conformidad con la usabilidad:* Las métricas externas de conformidad con la usabilidad deben ser capaces de evaluar la adherencia del software a las regulaciones, normas, convenciones, guías y estilos relativos la usabilidad.

Métricas del proceso

Además se proponen otras métricas que no son de calidad, pero que sí pretenden medir objetivamente los productos, los servicios del trabajo y los procesos desarrollados. Estas están referidas a las *métricas de procesos* que son algo más que una simple medida de algún atributo del proceso de software ya que aportan una visión profunda de la eficiencia de los procesos, tales como: esfuerzo,

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

errores detectados durante la entrega, errores detectados por los usuarios finales, productos entregados, etc.; todo esto con fines estratégicos para identificar mejoras. Las mismas son:

- *Eficacia de la Eliminación de Defectos (EED)*: es una medida de la habilidad de filtrar las actividades de la garantía de calidad y de control, al aplicarse a todas las actividades del marco de trabajo del proceso. Considerada globalmente para el proyecto, aunque también puede utilizarse para medir la habilidad de un equipo para encontrar errores antes de pasar a la siguiente fase.
- *Grado de efectividad en identificar las no-conformidades (ENC)*: Esta métrica muestra el grado de efectividad existente para encontrar las no conformidades, debido a que se miden las no conformidades encontradas contra las no conformidades que fueron correctamente encontradas o sea las que eran reales.
- *Porcentaje de efectividad para resolver las no-conformidades (PER)*: Esta métrica brinda el porcentaje de efectividad en resolver las no conformidades basándose en las no conformidades que fueron resueltas y las que fueron detectadas.
- *Capacidad del procedimiento especificado para evaluar los procesos (C)*: Esta métrica muestra un resultado cuantitativo de la capacidad del procedimiento definido para las evaluaciones, de esta forma es posible contar con un parámetro para decidir si es efectiva o no la evaluación realizada.

El procedimiento que se plantea para la aplicación de las métricas parte de los principios básicos para la medición sugeridos por Pressman: (PRESSMAN, 2001).

1er paso: Formulación

2do paso: Colección

3er paso: Análisis

4to paso: Interpretación

5to paso: Realimentación

Un aspecto importante para obtener un mayor apoyo en el proceso de medición, es asegurarse que todos los miembros del proyecto comprendan que el resultado obtenido será usado para lograr los objetivos básicos de la organización y no con el fin de evaluar el rendimiento individual.

2.3.5 Justificación de la selección de los estándares y modelos a establecer durante las diferentes etapas del proyecto.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

En el plan se proponen un conjunto de estándares a ser aplicados en diferentes etapas de desarrollo del software así como en todo su ciclo de vida. Esta propuesta puede variar si el equipo de desarrollo se reúne y decide establecer otros tipos de estándares, por lo que representa una guía muy flexible a tener en cuenta.

Etapa de Aplicación	Estándar
Captura de Requisitos	IEEE 830; ISO/IEC 12 207; ISO/IEC 9126 – 1
Diseño Gráfico	IEEE 1016
Arquitectura	IEEE 1471
Implementación	Codificación en J2EE
Pruebas y Métricas	ISO/IEC 9126 – 2; ISO/IEC 9126 – 3; ISO/IEC 9126 – 4
Documentación	ISO 12 119
Durante todo el ciclo de vida del desarrollo del SW.	IEEE 828; IEEE 1028; IEEE 730; RUP; LCS, ISO/IEC 12 207

Tabla 1: Distribución de estándares por etapa de aplicación

Seguidamente se presentan detalles significativos de dichos estándares que justifican la validez de su elección para cada etapa de desarrollo.

➤ **LCS:** Los Lineamientos de Calidad del Software, emitidos por la Dirección de Calidad del Software de la Infraestructura Productiva (IP) son de obligatorio cumplimiento y aplicación en los proyectos software desarrollados en la Universidad de las Ciencias Informáticas. De ahí se justifica que la selección de dicho estándar para todo el ciclo de vida del Proyecto Banco no es de carácter opcional. Durante la especificación de los lineamientos se tiene en cuenta un punto para definir estándares de código y documentación partiendo de la necesidad vital de que, los miembros del equipo posean conocimientos de los modelos de calidad, con la determinación de regirse por aquellos que fueron seleccionados y considerando evidentemente su concordancia con las particularidades del proyecto. Otros de los puntos planteados de interés lo constituyen el desarrollo de un Plan de Pruebas y un Plan de Aseguramiento de la Calidad, siendo este último el punto clave para el presente trabajo de diploma.

➤ **RUP:** En el Epígrafe 1.7 del Capítulo #1 se concluyó que: se ratificaba la elección realizada por el equipo de desarrollo del proyecto de usar el Proceso Unificado de Desarrollo del Software como

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

metodología a seguir, partiendo de los diferentes aspectos que plantea la misma para el aseguramiento de la calidad. Entonces, la elección de RUP como estándar de desarrollo del software a aplicar durante todo el ciclo de vida del proyecto, es obligatoria y viene justificada desde esta perspectiva.

- *IEEE 730*: El presente estándar, como bien fue analizado en el Epígrafe **1.7.4** del Capítulo #1, representa una guía detallada para la elaboración de un Plan de Aseguramiento de la Calidad, proponiendo una descripción específica de todos los elementos que debe poseer dicho plan. Fue creado por una de las organizaciones más prestigiosas a nivel mundial. Muchas de las empresas internacionales y la mayoría de las de EEUU, abanderadas en la industria del software, se certifican por él. En tal medida se seleccionó esta plantilla propuesta por la IEEE como referencia, además de la propuesta por RUP, para elaborar el Plan de Aseguramiento de la Calidad del Proyecto Banco.
- *ISO/IEC 9126-1*: Norma conocida también como Tecnología de la información – Calidad del producto software – Modelo de calidad. La selección de esta norma para definir los objetivos de calidad tiene su explicación partiendo, primeramente, de que es una norma cubana publicada por la ISO para la evaluación de productos software, características de calidad y guías para su uso. Proveen un marco para especificar los requisitos de calidad para el software, permitiendo el intercambio entre las diversas capacidades del producto software. Las características de calidad pueden estar dirigidas a especificar los requisitos funcionales y no funcionales de los clientes y usuarios. Además, son aplicables a todo tipo de software. En el Epígrafe **1.3.2** del Capítulo #1 se hizo referencia específica a todas esas características.
- *ISO/IEC 9126- (2,3)*: Se propone aplicar la segunda y tercera parte de la norma pues estas definen las métricas para la medición cuantitativa de la calidad tanto externa como interna del software, respectivamente, en términos de las características y sub-características definidas en la primera parte, por lo que se deben utilizar en conjunto. Las métricas internas miden el software en sí, las métricas externas miden el comportamiento del sistema.
- *ISO/IEC 9126-4*: La cuarta parte de norma también es de necesaria aplicación pues provee calidad en el uso de métricas para la medición de los atributos definidos en la primera parte, sin embargo, las métricas que figuran en esta norma no están destinadas a ser usadas de forma exhaustiva. Los desarrolladores, administradores de calidad o evaluadores pueden seleccionar las

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

métricas propuestas por esta norma para la definición de requisitos, la evaluación de productos de software, la medición de aspectos de calidad, entre otros fines.

- *IEEE 830*: Es una norma internacional conocida además como *Prácticas recomendadas para las especificaciones de requisitos de software*. En su contenido propone diversos criterios para determinar los requerimientos planteados por los clientes; de ahí que se haya tomado este aspecto como punto de partida para su selección y aplicación en la especificación de los requisitos de software del proyecto. Además, se propone el uso de esta norma pues la organización IEEE es una de las más prestigiosas y conocidas mundialmente.
- *ISO/IEC 12 207*: Esta norma internacional nombrada también como: *Tecnología de la Información, Procesos del Ciclo de Vida del Software* está destinada a facilitar la confección de metodologías del ciclo de vida y de elaboración del software en el nivel organizacional y de proyecto. Se propone para que sea utilizada durante todo el ciclo de vida del software y la misma proporciona a los desarrolladores una estructura común para facilitar la integración de los productos y servicios. Se escoge esta norma, además, pues fue confeccionada por la Organización Internacional para la Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC), instituciones que constituyen el sistema especializado para la normalización a nivel mundial.
- *IEEE 1016*: Esta norma es conocida también como: *Práctica Recomendada para la Descripción del Diseño de Software*, en su contenido describe la información necesaria y las recomendaciones para que una organización desarrolle descripciones de diseño de software, proporcionando así guías como facilidades durante el desarrollo de la documentación del diseño (Documento de Diseño del Software, DDS). Aporta también una serie de beneficios pues esta práctica no se limita sólo a metodologías específicas del diseño, la gestión de la configuración o la garantía de la calidad. Teniendo en cuenta estos aspectos, e incluyendo, además, que la organización creadora es muy prestigiosa a nivel internacional, se propone usar dicha norma.
- *IEEE 1471*: La presente norma es denominada también como *Prácticas Recomendadas para la Descripción de Arquitectura de Software*, se ocupa de las actividades de la creación, análisis y logística de la arquitectura de sistemas de software. El propósito de esta práctica es el de facilitar la expresión y comunicación de arquitecturas, y por tanto, sentar las bases para la calidad y el costo mediante la normalización de elementos y habilidades para la descripción arquitectónica. Para ello define un

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

conjunto de requisitos de contenido en las descripciones arquitectónicas que pueden aplicarse en el contexto de una organización con métodos, procesos, técnicas y herramientas definidas. Se hace la propuesta de esta norma pues aportaría valiosos beneficios a la descripción de la arquitectura y considerando, además, que la organización creadora de dicha norma está acreditada internacionalmente.

- *IEEE 828*: El IEEE define este estándar para establecer los contenidos mínimos que deben aparecer en el Plan de Gestión de Configuración del Software (el plan puede ser un documento aislado o formar parte de otro documento); en las diferentes secciones se debe plantear *quién* podrá modificar *qué*, *cuándo* y *cómo*. Así se pretende mantener una documentación detallada del control de las versiones. Se eligió dicho estándar porque es muy fácil de aplicar, por ejemplo, los componentes del plan son descritos profundamente, el nivel de detalle es tal que se especifica tanto para usuarios principiantes como para expertos, por lo que no necesita documentación adicional, y sobre todo, trata la flexibilidad como parte de sus especificaciones.
- *IEEE 1028*: Esta es la norma para el comentario de software. Su objetivo radica en definir los requerimientos para los procesos de revisión y auditorías, en tal caso define cinco tipos de revisiones de software, junto con los procedimientos necesarios para la ejecución de cada tipo de examen. Incluye tanto al producto como al proceso de software. Especifica una tabla en la que se señalan los principales procesos para conseguir objetivos de calidad.
- *ISO 12 119*: Norma cubana conocida como *Tecnología de la Información. Paquetes de Software. Requisitos de Calidad y ensayos/pruebas*. Fue seleccionada porque en ella se establecen parámetros para la documentación, en tal caso se dan especificaciones para el Documento de requisitos, Descripción del producto, Documentación para el usuario, Documentación del paquete (como procesadores de texto, hojas de cálculo, programas de bases de datos, paquetes gráficos; refiriéndose a los requisitos de calidad para ellos y dando instrucciones de cómo probarlos), además de los requisitos para los ensayos/pruebas.
- *Codificación en J2EE*: Norma para convenciones o estándares de código. Se propone esta norma para estandarizar el trabajo de implementación desempeñado por los programadores, implantando políticas referentes a la forma en que se nombran las clases, métodos, objetos, comentarios, etc. Esto facilita en gran medida una mayor uniformidad, organización y entendimiento en

la forma de codificar entre todo el equipo de programadores. Cada proyecto puede definir su norma; pero una vez que esté definida se deberá cumplir con exactitud y para verificar su correcto acatamiento se establecerán listas de chequeo.

2.3.6 Forma de solución para elaborar el Plan de Revisiones y Auditorías

La sección del plan correspondiente al Plan de Revisiones y Auditorías representa uno de los puntos claves para intentar prevenir los fatales errores que muchas veces son hallados al producto final, pues las acciones que aquí se propone desarrollar van encaminadas a encontrar defectos, o asegurar su ausencia, en los artefactos generados durante todo el ciclo de vida del software.

Esta sección dispone la realización de un Plan de Revisiones y Auditorías, para ello es básico dejar lo suficientemente claro la diferencia existente entre los conceptos de Revisiones y Auditorías. Las diferencias más importantes entre las revisiones y las auditorías son las siguientes: (ANTONIO, 2002)

- Las revisiones se llevan a cabo desde las primeras fases del desarrollo, mientras que las auditorías se llevan a cabo en las fases finales.
- El objetivo de las revisiones es detectar defectos, mientras que el objetivo de las auditorías es certificar conformidad e identificar desviaciones.

Además, las revisiones son más flexibles, las auditorías no porque generalmente se efectúan por personas ajenas al proyecto y van dirigidas a evaluar la concordancia de los procedimientos establecidos con la realidad, de esta manera adquieren un carácter legal.

A pesar de las notables diferencias, existen varios puntos en común entre estas actividades vistas como técnicas de grupo, por ejemplo se establecen con el objetivo de darle seguimiento al control de la calidad durante el desarrollo del proyecto, ambas deben realizarse mediante procedimientos previamente definidos, y deben documentar correctamente los resultados obtenidos en ellas. Su misión principal radica en conseguir que la responsabilidad del control de la calidad no recaiga sólo sobre el propio desarrollador. Tanto las revisiones como las auditorías se llevan a cabo durante un período de tiempo delimitado y con recursos finitos. Se propone, incluso, establecer un método común para su realización: el uso de listas de comprobación o de chequeo teniendo en cuenta las amplias ventajas que ofrece su uso.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

A partir de las premisas mencionadas se presentaron un conjunto de tareas generales de revisiones y auditorías que fueron determinadas para inspeccionar artefactos generados durante todo el ciclo de vida del desarrollo del software. Puntualizar además que, se tomó en cuenta la Lista de Riesgos del proyecto como punto de partida para enfocar algunas de las tareas de revisiones y auditorías mencionadas, a identificar posibles errores ya previstos.

Considerando las tareas que fueron definidas en el Epígrafe **1.3.2** del Anexo 1 correspondiente al Plan de Aseguramiento de la calidad para ser desarrolladas por los miembros del equipo de calidad, se llegó a la conclusión de que, se realizarán los siguientes tipos de revisiones y auditorías:

Dentro de los tipos de revisiones se ejecutarán las *Inspecciones* que son aquellas en las que los protagonistas estudian los artefactos detalladamente guiados por sus desarrolladores y van chequeando a la vez, en cada paso, el cumplimiento de los criterios establecidos en una lista de comprobación con objetivo de encontrar defectos. Según Pressman estas revisiones han sido categorizadas como RTF y son el filtro más efectivo desde el punto de vista de garantía de calidad. Para tener una idea más precisa a continuación se mencionan los objetivos básicos de una RTF. (LOUDEN, 2004)

- Descubrir errores en la función, la lógica o la implementación de cualquier representación del software;
- Verificar que el software bajo revisión alcanza sus requisitos;
- Garantizar que el software ha sido representado de acuerdo con ciertos estándares predefinidos;
- Conseguir un software desarrollado de forma uniforme y;
- Hacer que los proyectos sean más manejables.

La clasificación de los tipos de auditorías se determinó, inicialmente según la procedencia del auditor, o sea, si el auditor pertenece a la entidad y realiza la auditoría en nombre de la propia entidad, se estaría en presencia de una *auditoría interna*, por el contrario para las *auditorías externas*, el auditor no pertenece a la entidad, sino a otra parte interesada de la organización.

Se propone realizar la *Auditoría del Sistema de Calidad* como parte de una auditoría externa con el objetivo de comprobar la efectividad y completitud del propio Sistema de Calidad establecido. Este tipo de auditoría externa no sólo trata de poner de manifiesto la existencia de un Sistema de Calidad correctamente documentado, sino también que dicho sistema es conocido por toda la organización. De

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

forma más específica esta propuesta de auditoría será vista en este plan como *Auditoría de Control de Revisiones*, con la estrategia de obtener información objetiva sobre el funcionamiento del sistema de calidad al comprobar la adecuada realización de las revisiones planificadas (que serían las RTF). El estado o evaluación que alcanzó el artefacto revisado en la RTF auditada, se registrará en la Plantilla de Informe Final de Auditoría, como parte de los datos de realización de la misma.

Dentro de las auditorías internas se propone desarrollar como tipo de auditoría la *Auditoría del Proceso* cuyo objetivo es evaluar el proceso de desarrollo o de gestión, conjuntamente con su completitud y efectividad, determinando así qué no está acorde y dónde se puede mejorar. Se comprueba que las características del proceso cumplen con las especificaciones que de él se esperan tales como:

- Qué debe hacerse.
- Quién debe llevarlo a cabo.
- Dónde y cómo debe ser hecho.
- Qué materiales, equipamientos y documentos son necesarios.
- Cómo debe ser dirigido y registrado.

En este epígrafe se propone el procedimiento definido para la realización de las actividades de control que se plantea realizar en el Plan de Revisiones y Auditorías.

Procedimiento para el desarrollo de las revisiones

Cada una de las revisiones propuestas será planificada, y el cronograma habrá sido informado a todos los miembros del proyecto que están implicados en ellas con una semana de antelación como mínimo. Además, para cada revisión se desarrollará una reunión, la cual estará controlada por un jefe de revisión, que es el responsable de dirigir y asesorar todo el proceso. Se propone que estas reuniones, independientemente del tipo de RTF que se efectúe en ellas, mantengan las mismas restricciones planteadas por Pressman que se especifican a continuación:

- Deben convocarse para la revisión (normalmente) entre tres y cinco personas;
- Se debe preparar por adelantado, pero sin que requiera más de dos horas de trabajo a cada persona; y
- La duración de la reunión de revisión debe ser menor de 2 horas.

Es necesario delimitar el centro de atención de la revisión, o sea, partiendo de que la duración de la reunión no debe llegar a las dos horas, se hace evidente que el objetivo de la revisión sea una parte

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

pequeña del producto, como por ejemplo el diseño de un módulo o una parte de su código fuente, así la probabilidad de descubrir errores será más elevada.

Se plantea desarrollar listas de chequeos para cada producto o artefacto a ser revisado, de esta forma se organizará mejor el proceso de revisión. Cada error encontrado será documentado (debe quedar escrito bien claro, para ello se propone el uso de la Plantilla del Informe de No Conformidades) y señalado a los miembros del equipo profesionalmente, sin imposiciones, de forma tal que no se cree animosidad entre los desarrolladores, dado que posteriormente ellos se encargarán de la resolución de los problemas. Por lo que se puntualiza que los revisores nunca intentarán resolver los problemas que encontraron, pero sí están facultados para proponer soluciones o posibles mejoras, las cuales se tendrán en cuenta de la misma forma que los criterios dados por los desarrolladores.

Para obtener mejores resultados en un proceso de revisión y una mayor organización, se debe dejar bien claro para cada tipo de revisión los siguientes puntos:

- Su objetivo.
- Qué producto es el que evalúa.
- Sus propósitos.
- Cuál es el elemento organizativo responsable de llevar a cabo la revisión.
- Cuáles son los elementos organizativos que deben tomar parte en la revisión.
- Cuáles son los requisitos de revisión.
- Dónde deben documentarse los resultados de revisión.

Finalmente, para concluir el proceso de revisión se propone la aplicación de métricas a través de las cuales se logre medir diferentes características de interés, y de esta forma obtener estadísticas que permitan evaluar la calidad del artefacto o proceso chequeado.

Procedimiento para el desarrollo de las auditorías

Para la realización de las auditorías se proponen un conjunto de pasos a seguir según las especificaciones planteadas por la Norma ISO 9004 con el objetivo evaluar la concordancia de los procedimientos establecidos con la realidad.

- Planificación

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

Consiste en definir los objetivos de la auditoría que se pretende realizar y su alcance. Para ello se propone darles respuesta a las siguientes interrogantes:

¿Por qué se realiza?; Es una auditoría de rutina o para resolver problemas concretos.

¿Para qué se realiza?; Para mejor o para conseguir una certificación, etc.

¿En qué se centra?; Definir los objetivos de la auditoría.

¿Qué va a ser auditado?; Delimitar el alcance de la auditoría.

¿Qué resultados se esperan?; Identificar situaciones problemáticas, tales como desviaciones del estado actual con respecto al estado deseado, y sugerir posibles soluciones o mejoras.

¿Cómo y dónde se van a utilizar los resultados?; Se documentarán los resultados en un informe para su posterior análisis, por lo que se debe especificar bien los resultados.

¿Quiénes son los responsables de efectuarla?; Especificar quiénes son los auditores o las personas responsables.

¿De qué forma se va a llevar a cabo?; Incluye una especificación de los datos que se van a recoger y de qué forma se hará.

¿Cuándo se va a realizar?; Fecha de ejecución.

Todos estos parámetros se deberán documentar en la Plantilla Plan de Auditoría que propone la DCS con ese propósito.

➤ Efectuar la investigación pertinente

La auditoría se inicia con una reunión de apertura de la investigación donde se exponen los datos recogidos en la Plantilla Reunión de Apertura. Se desarrolla mediante entrevistas y evaluaciones en los que se recopilan datos partiendo de las especificaciones de la lista de chequeo que ha de ser elaborada para esta actividad y la Guía Exploratoria que establece la DCS.

➤ Analizar datos obtenidos

En ocasiones, al desarrollar las auditorías se maneja una cantidad considerable de datos, por lo que para el equipo de auditores resulta complicado seleccionar aquellos datos considerados como relevantes y recurren a utilizar técnicas de análisis estadísticos. Posteriormente se realiza una evaluación en paralelo de los resultados por un grupo de evaluadores, se comparan las conclusiones obtenidas y se estudian las causas de las desviaciones significativas. Sin embargo esto se puede simplificar considerablemente al definir con antelación los aspectos que se deben auditar.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- Proponer soluciones a los problemas encontrados y posibles mejoras

En caso de identificarse alguna desviación durante el análisis de los datos estos se plasmarán en la Plantilla de No Conformidades que fue creada, además los auditores están facultados en sugerir ideas que pudieran conllevar a posibles soluciones del problema. Para ello puede convocarse a una reunión, realizarse de forma independiente o de una forma más formal documentarse en la Plantilla de Acciones Correctivas o de Mejora propuesta por la DCS.

- Elaborar y presentar un informe de resultados

Finalmente, para concluir todo el proceso se deberá escoger un responsable que documente adecuadamente el Informe Final de Auditoría (plantilla que estable la DCS), el cual será presentado a los directivos; aunque de forma general, en esta tarea debe participar todo el equipo de auditores para garantizar que no se olvide destacar puntos claves identificados tanto positivos como negativos. Puntualizar además que en este informe se deberá ofrecer una evaluación a la auditoría realizada según los resultados, la cual incluye los criterios siguientes: Aceptable, Deficiente y No Procede.

Cronograma para las actividades propuestas en el Plan de Revisiones y Auditorías

La estrategia para diseñar el cronograma que especifica las fechas de realización de las actividades de revisión y auditorías se propone partiendo de la organización, definición de tareas y responsabilidades, e incluye además la situación real por la que transita el proyecto. Al mismo tiempo que se desarrolla este trabajo el proyecto aún se encuentra en una etapa de concepción, donde no se ha logrado elaborar un cronograma de trabajo para el equipo de desarrollo, no se cuenta con la totalidad de los integrantes requeridos y esencialmente lo que se está haciendo son actividades de capacitación. Dada esta situación, se decidió proponer un cronograma tipo o genérico, tratando de estimar lo más cercano posible a la realidad el tiempo para la ejecución de dichas tareas.

Una de las columnas del cronograma se refiere a la duración de la actividad de aseguramiento, especificada en semanas y la otra al inicio de dicha actividad. Por lo que el cronograma no especifica las fechas exactas para comenzar a ejecutar cada una de las tareas de aseguramiento definidas, sino que propone desarrollarlas, como contrapartida al concluir las actividades ejecutadas por el equipo de desarrollo del software.

Teniendo en cuenta estos aspectos se puede concluir en que las fechas para las actividades de aseguramiento dependen del avance de las actividades para desarrollar el producto. Además, la duración de las actividades, es una estimación relativa porque está en dependencia de las condiciones de trabajo del equipo de calidad y las eventualidades que puedan surgir en el transcurso de las mismas.

2.3.7 Especificación de los tipos de pruebas propuestas para desarrollar el Plan de Pruebas

Según Glenford J. Meyers existen varios objetivos en las pruebas; se puede decir que la prueba es el proceso de ejecución de un programa con la intención de descubrir un error, un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces y una prueba tiene éxito si descubre un error no detectado. (MEYERS, 1979)

De una forma más abarcadora, una prueba es exitosa si descubre errores en el software, demuestra hasta dónde las funcionalidades del software se corresponden con las especificaciones y los datos que se obtienen como resultado de las mismas brindan el grado de fiabilidad del software e indican su calidad. Como bien diría Pressman: la prueba no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software. (PRESSMAN, 2001)

A continuación se especifican una serie de aspectos a considerar durante el diseño de los diferentes tipos de pruebas que fueron definidas en el Plan de Aseguramiento de Calidad.

➤ *Pruebas de Caja Negra:* Se refieren a las pruebas que se llevan a cabo sobre la interfaz del software. Es decir, 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. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software. Se centran en los requisitos funcionales del software y no se interesa por la estructura interna del sistema. Dentro de los principales objetivos están: detectar y documentar la mayor cantidad de errores posibles en el software, verificar el cumplimiento de la mayor cantidad posible de requisitos funcionales y no funcionales establecidos para el sistema; demostrar que el sistema cumple con estos requisitos solicitados y, que los clientes tengan la posibilidad de ver hasta qué punto el sistema cumple con sus expectativas; además de que los desarrolladores observen cómo se comporta la aplicación en un entorno real.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- *Pruebas de Integración del Sistema:* Son ejecutadas para asegurar que los componentes en el modelo de implementación operan correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Estas pruebas descubren errores o elementos no completos en las especificaciones de las interfaces de los paquetes. El objetivo es tomar los componentes probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño. A medida que la integración progresa, disminuye la necesidad de controladores de prueba diferentes. Durante este avance se deben identificar los módulos críticos, que son aquellos que presentan una o más de las siguientes características: varios requisitos del software, mayor nivel de control, complejo o propenso a errores, requisitos de rendimiento muy definido; los módulos críticos deben ser probados lo antes posible. Si se encuentra un error al integrar varios módulos se propone realizar una prueba de regresión, que no es más que las pruebas que se realizan después de haber solucionado los errores que fueron detectados.
- *Pruebas de Seguridad:* Se enfocan a la seguridad en el ámbito de aplicación, incluyendo el acceso a los datos y a las funciones de negocios asociadas a cada rol de usuario, la misma asegura que los usuarios sólo accedan a los procesos de acuerdo al rol asignado. El objetivo principal de esta prueba es verificar que los mecanismos de protección incorporados en el sistema lo protegerán de accesos impropios, es decir, verificar que un usuario pueda acceder solo a las funciones o datos para los cuales su rol tiene permiso únicamente a través de los puntos de ingreso apropiados y no permitir que nadie ajeno al sistema pueda acceder a él. Al finalizar esta prueba se procura que para cada rol las funciones y datos convenientes estén disponibles y todas las operaciones se realicen según lo esperado.
- *Pruebas de aceptación:* Es la prueba final antes del despliegue del sistema, su objetivo fundamental es verificar que el software está listo y puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Además, están encaminadas a asegurar el cumplimiento de los procesos del negocio, los casos de uso y los prototipos correspondientes validados en la etapa de Captura de Requisitos, así como las funcionalidades de cada uno de los módulos planificados. Se desea la comprobación de los requisitos capturados en el desarrollo del producto, creando un margen de tiempo que permita erradicar incongruencias y ajustar más el producto a las necesidades reales del cliente, y de esta forma, reducir las posibilidades de entregar un producto final poco funcional sin satisfacer las necesidades del cliente.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

La prueba de software es un elemento crítico para la garantía de la calidad, de ahí que se propone desarrollar la siguiente estrategia para todas las pruebas a ejecutar con el propósito de que se obtengan los mejores resultados (encontrar mayores defectos) al ejecutar esta actividad:

- 1ro. Planificación de las pruebas.
- 2do. Diseño de los casos de prueba.
- 3ro. Implementación de las pruebas.
- 4to. Ejecución de las pruebas.
- 5to. Evaluación de los resultados.

Las pruebas se empezarán a diseñar en la fase inicial del proyecto, desde la toma de requisitos y se irán adaptando a medida que se vaya desarrollando el software durante todo su ciclo de vida. Para la ejecución se identificaron tres roles principales: el responsable de calidad, el diseñador de pruebas y el probador. Cada prueba se especificará en el Plan de Pruebas a través de los casos de pruebas que se hayan determinado, estableciendo las condiciones de ejecución, las entradas, y los resultados esperados, además de que cada caso de prueba llevará asociado un procedimiento específico con las instrucciones para realizar la prueba, según el tipo que se trate. El Plan de Pruebas se propone desarrollar como una herramienta de calidad que debe ayudar a manejar las pruebas mencionadas, encontrar errores, realizar su seguimiento y verificar su resolución.

Finalmente, añadir que: con estos tipos de pruebas se pretende garantizar que los objetivos trazados para el desarrollo del producto cumplan con el software final entregado al cliente, asegurar el cumplimiento del alcance de cada prueba, detectar la mayor cantidad posible de No Conformidades en el tiempo planificado, documentar y analizar todos los resultados obtenidos, además de garantizar que el criterio de aceptación se mantenga dentro de límites tolerables.

2.3.8 Solución para el flujo de trabajo de las actividades de aseguramiento de la calidad

Esta sección no viene incluida en la plantilla propuesta por RUP para elaborar el Plan de Aseguramiento de la Calidad, ni en la que especifica el estándar IEEE 730, por lo cual significa un aporte que se hace en este trabajo. Se tiene en cuenta asumiendo la premisa de que: identificar los flujos de trabajos necesarios durante las actividades de aseguramiento de la calidad, facilita una visión más amplia para lograr el marcado propósito de obtener un producto final ideal para el cliente.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

Se proponen 4 flujos de trabajo; un flujo general donde se vinculan las actividades de aseguramiento de la calidad dentro de las actividades del proyecto. Un flujo de trabajo para las actividades de revisiones, uno para las actividades de auditorías, al igual que un flujo de trabajo para las actividades de pruebas; todas ellas especificadas en el Plan de Aseguramiento de la Calidad elaborado.

Flujo de trabajo para las actividades de Revisión

El flujo de trabajo para las actividades de revisiones es iniciado por el Responsable de Calidad quien tiene la responsabilidad de planificar dichas actividades; lo cual da paso al Revisor Técnico a emplear las listas de chequeo elaborada por el Diseñador de Pruebas según la etapa en que se encuentre el proyecto. A continuación, se tiene en cuenta si se encontraron o no errores durante la revisión. En caso de no identificar errores, el Responsable de Calidad procede a archivar el proceso de revisión, para que inmediatamente con esta fuente de datos el Diseñador de Pruebas logre aplicar diferentes tipos de métricas seleccionadas según lo que haya sido chequeado y lo que interese medir. Terminada la actividad el Responsable de Calidad realiza una evaluación y un resumen de los resultados al respecto, generando algún tipo de informe o documento (Ejemplo: Informe resumen de la inspección, Lista de acciones pendientes, Informe de asuntos relacionados, Informe del proceso de inspección, Informe Final) para constatar el desarrollo de la actividad, y finalizar de esta forma, el flujo de trabajo. Si por el contrario, fueron encontrados errores, el Revisor Técnico los documenta en el Informe de No Conformidades, luego el Responsable de Calidad comunica al equipo de desarrollo del proyecto los problemas encontrados, estos inmediatamente están obligados a realizar los cambios pertinentes con el objetivo de solucionar los defectos especificados, para que nuevamente el Revisor Técnico aplique las listas de chequeo. Este flujo puede convertirse en un ciclo hasta que no se encuentren más errores o el proyecto tome la decisión de obviarlos al no ser significativos y por consiguiente puedan realizarse las actividades mencionadas anteriormente, para el caso en que no fue encontrado ningún tipo de error.

Flujo de trabajo para las actividades de Auditoría

El presente flujo de trabajo es iniciado por el Responsable de Calidad una vez que haya planificado la auditoría en el Plan de Revisiones y Auditorías y corresponda darle cumplimiento. El mismo convoca y realiza una reunión de apertura donde es creado un informe que documenta los objetivos de la auditoría a realizar y su alcance. Seguidamente el Diseñador de Pruebas crea la lista de chequeo la

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

cual servirá de herramienta para evaluar los parámetros deseados. Vale señalar que las listas de chequeo para las auditorías no deberán ser creadas al detalle con el propósito de encontrar errores específicos, sino de forma tal que estén enfocadas en comprobar que se cumplan las regulaciones establecidas para validar o no su cumplimiento. El Revisor Técnico se encarga de efectuar la investigación pertinente usando la lista de chequeo creada y la Guía Exploratoria que establece la DCS, para luego analizar los datos obtenidos y proponer soluciones y posibles mejoras en la Plantilla de Acciones Correctivas o de Mejoras en caso de identificar no conformidades y que el revisor tenga los suficientes conocimientos para ello. Concluido este paso el Responsable de Calidad procede a elaborar el informe de resultados utilizando la Plantilla Informe Final de Auditoría establecida por la DCS y, como punto final de la actividad convoca a los miembros del equipo de desarrollo que tienen la responsabilidad de responder según el alcance de la auditoría para presentar y discutir los resultados obtenidos.

Flujo de trabajo para las actividades de Prueba

El siguiente flujo de trabajo, correspondiente a las actividades de prueba, es iniciado por el Responsable de Calidad facultado para establecer el desarrollo del Plan de Pruebas, para ello, el Diseñador de Pruebas comienza diseñando los casos de prueba y a continuación el Probador realiza dichas pruebas. Si no fueron encontrados errores, el Probador realiza un informe preliminar, para que luego el Responsable de Calidad elabore el informe final correspondiente, dando paso a la liberación del producto por parte del equipo de desarrollo y de esta forma terminar el flujo de trabajo. Para el caso de que sí se hayan encontrado errores durante las pruebas, el Probador realiza el Informe de No Conformidades detallando dichos aspectos, seguidamente el Responsable de Calidad informa al equipo de desarrollo los problemas existentes y estos se encargan de corregir los defectos especificados. Hechos los arreglos pertinentes, el probador procede a realizar nuevamente las pruebas, convirtiéndose este flujo de trabajo en un ciclo, similar al mencionado en el Flujo de Trabajo para las actividades de revisiones y finalizando de la misma forma mencionada anteriormente.

Flujo de trabajo general

También se modeló el flujo de trabajo general vinculando las actividades del equipo de desarrollo con las del equipo de calidad. Este es iniciado por el equipo de desarrollo al concebirse el proyecto, a continuación se comienza a poner en práctica la estrategia planificada para la gestión de la

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

configuración (que deberá desarrollarse durante todo el ciclo de vida del software) y se realizan las entrevistas al cliente para modelar el negocio, actividad que debe quedar documentada en la Plantilla de Descripción de los Procesos del Negocio.

El equipo de calidad procede a realizar la revisión a los procesos del negocio, si se encontraron no conformidades se vuelve a documentar el modelado del negocio hasta que sean erradicadas, para luego pasar a entrevistar al cliente y capturar los requisitos del sistema que serán formalmente documentados en la Plantilla de Especificación de Requisitos. Seguidamente se deberá efectuar la revisión a la captura de requisitos, si se determinaron errores, se vuelve a especificar los requisitos en el documento y en caso de no existir no conformidades son aprobados los requisitos definidos por parte del cliente.

A continuación se comienza a documentar la arquitectura y los casos de uso del sistema de forma simultánea. La primera de estas actividades queda registrada en el Documento de arquitectura de software, la segunda se registrará en el Modelo de casos de usos del sistema, posteriormente se pasa a las revisiones correspondientes a estas dos actividades y en caso de encontrarse no conformidades se vuelve a documentar.

Finalizando estas actividades se determinan los módulos principales a desarrollar en paralelo, para luego comenzar a diseñar gráficamente según las pautas establecidas y a documentar el diseño de las clases simultáneamente. Para documentar el diseño gráfico se estableció la plantilla Documento de pautas del diseño gráfico, si se encuentran no conformidades se volverá a diseñar.

Para la documentación del diseño de las clases se debe crear el Modelo de diseño y seguidamente se realizarán las revisiones correspondientes, en caso de que existan errores se procederá a documentar nuevamente el diseño de las clases. Cuando estas dos actividades estén con un rango de errores dentro de los límites establecidos se procederá al diseño de las bases de datos, de aquí surgirán el Modelo de datos y el Script de base de datos, concluida esta actividad se revisará el diseño de la base de datos y si son encontradas inconformidades según lo establecido estas deberán ser solucionadas. Completada la actividad satisfactoriamente se realizará la implementación del módulo desarrollado, a partir de ese momento se comienza a crear el código fuente para luego ser revisada la actividad de implementación, al igual que en los casos anteriores, si se encuentran errores se deberán solucionar y volver a repetir el ciclo de revisión.

Al concluir cada módulo se realizarán las Pruebas de Caja Negra y, posteriormente se deberá terminar de documentar todo lo referente al modulo desarrollado, contando con los documentos establecidos en el Expediente de Proyecto, para luego proceder con la revisión de toda la documentación del proyecto, haciendo énfasis en la documentación que es entregada al usuario, si son identificadas no conformidades con lo realizado se deberá volver a documentar y a revisar.

Finalizando la actividad anterior se realizarán auditorías generales a los procesos, se incluirán la auditoría al diseño, a la arquitectura, al diseño gráfico, a la base de datos y a la gestión de configuración, estas auditorías son internas del proyecto y serán realizadas para comprobar el cumplimiento de las normas, estándares y reglamentaciones establecidas por parte del proyecto, esta actividad se desarrollará antes de la auditoría general al sistema de calidad para que el equipo de calidad haya realizado la mayor parte de su trabajo.

Los restantes módulos se deberán ejecutar siguiendo la misma estrategia planteada. Una vez terminados los módulos se procederá con la integración de los mismos, lo cual dará paso a la revisión de la gestión de configuración y a las pruebas de integración simultáneamente. Si fueran encontrados defectos en la gestión de configuración se debe volver a documentar el plan de gestión de configuración propuesto. Ya pasadas satisfactoriamente las pruebas de integración se deben efectuar las de seguridad y concluidas estas las de aceptación del cliente. Cuando se terminen de realizar todas las pruebas y no hayan sido encontrados defectos en la gestión de configuración el cliente debe aceptar el producto final obtenido por el equipo de desarrollo del software, terminando de esta forma el flujo de trabajo.

2.3.9 Justificación de las Herramientas, Técnicas y Metodologías propuestas

Hoy en día se aplica y se incrementa continuamente el uso de un conjunto de técnicas, herramientas y metodologías que sirven de apoyo a las actividades desarrolladas para intentar asegurar la calidad del producto software. Fue necesario desarrollar un estudio característico para lograr establecer una propuesta de cada una de ellas a usar durante la implementación del plan. Producto al mismo se obtuvieron una serie de argumentos que justifican la selección establecida, los cuales se presentan a continuación:

➤ *Aplicación de estándares IEEE:* Se propone el uso de los estándares IEEE porque abarcan todos los aspectos técnicos relacionados con la Ingeniería de Software, aportando a la vez un conjunto

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

de ventajas durante sus etapas de desarrollo; seguidamente se hace referencia a estas ventajas de una forma general. Como punto de partida es válido señalar que son estándares públicos, están bien documentados y orientados al aseguramiento de la calidad a nivel de proyecto. Tienen una descripción muy específica, se concentran en la definición de procesos y son independientes unos de otros. Añadir además que están actualizados, o sea, son periódicamente revisados, por lo que no se corre el riesgo de caer en obsolescencias; y sobre todo, constituyen una buena base para definir estándares propios a la medida de la organización, una vez que hayan sido interpretados y adaptados a sus necesidades particulares.

- *Aplicación de normas ISO:* El organismo que emite estas normas goza de gran prestigio internacionalmente, es el encargado de la estandarización de normas de productos para facilitar el intercambio comercial de bienes y servicios y desarrollar la cooperación en las áreas de actividades intelectuales, científicas, tecnológicas y económicas de más de 140 países. Se propone la aplicación de las normas ISO pues aportan beneficios de calidad a las empresas, brindan el marco que permite evaluar razonablemente por parte de terceros la efectividad del sistema y su finalidad radica en coordinar las normas nacionales en equilibrio con el Acta Final de la OMC. La mayor importancia que tiene la aplicación de las normas ISO es que son normas prácticas, o sea, normas sin carácter académico, por eso han permitido su aplicación generalizada sobre todo en pequeñas y medianas empresas.
- *Aplicación de normas NC:* La Oficina Nacional de Normalización (NC), es el organismo que representa a la República de Cuba ante las Organizaciones Internacionales y Regionales de Normalización. Se propone la aplicación de estas normas pues como tienen carácter nacional no es necesario el uso de patentes, además la documentación referente a ella aparece en idioma español y especialmente porque la NC representa los intereses nacionales antes los organismos u organizaciones internacionales.
- *Aplicación de las Plantillas del Expediente de Proyecto establecidas por la DCS:* El uso de estas plantillas es obligatorio para llevar a cabo la documentación de todo proyecto desarrollado en la universidad; aunque existe un cierto grado de flexibilidad si el proyecto decide crear una plantilla propia, o sea, el proyecto presenta una propuesta que tiene que ser aprobada por la DSC. Aclarar que el proyecto es el que decide cuáles de estas plantillas se incluirán en su expediente teniendo en cuenta sus particularidades.

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- *Depuradores*: Un depurador es un programa que puede utilizarse para determinar los errores de ejecución de un programa compilado. A menudo está integrado con un compilador en un IDE. La ejecución de un programa con un depurador se diferencia de la ejecución directa en que el depurador se mantiene al tanto de la mayoría o la totalidad de la información sobre el código fuente, tal como los números de líneas y los nombres de las variables y procedimientos. (LOUDEN, 2004) La codificación se implementará en lenguaje J2EE, Java incluye un depurador en su entorno de desarrollo, JDB, que es un depurador de líneas de comandos. Para su cómoda utilización se recomienda esperar a disponer de herramientas visuales porque es un tanto complicado de utilizar.
- *Perfiladores*: Un perfilador es un programa que examina una aplicación mientras esta se ejecuta. Provee útil información de ejecución como el tiempo invertido en determinados bloques de código, el grado de utilización de la memoria y del heap, el número de instancias de cada clase que hay en memoria, etc. Para el lenguaje Java se propone el JProfiler que se obtiene como un paquete independiente. Este perfilador se caracteriza por ser muy eficaz en su tarea, posee una atractiva interfaz y sobre todo es un completo analizador de aplicaciones Java.
- *Metodología RUP*: La forma práctica de probar las funciones de un sistema es la prueba de que el sistema pueda utilizarse de manera que tenga sentido para los usuarios. (MARINESCU, 2003) Esta metodología plantea una nueva técnica que consiste en identificar los casos de pruebas como casos de usos durante el flujo de trabajo de requerimientos, o sea, mucho antes de comenzar el diseño del sistema, para luego asegurarse que el diseño concebido implementa correctamente los casos de uso. Estos casos de uso de prueba especifican una forma de probar el sistema, incluyendo las entradas, los resultados esperados, entre otras condiciones relevantes con el objetivo de verificar el flujo básico de un caso de uso determinado. Incluye el tipo de caso de prueba a desarrollar, particularmente el que sería de interés para las especificaciones de pruebas referidas en el plan indicando cómo probar un caso de uso o un escenario específico de un caso de uso (particularmente para el plan elaborado durante el desarrollo de este trabajo interesaría probar mediante Pruebas de Caja Negra).
- *Generación de casos de prueba a partir de casos de uso*: El caso de uso se representa mediante un diagrama de actividades. Se propone el uso de la notación UML para el desarrollo del proceso de prueba a partir de escenarios de uso del sistema descritos mediante diagramas (diagrama de actividades). Esta técnica consiste en inicialmente generar escenarios de casos de uso abarcando todos los posibles caminos de ejecución para cada caso de uso, aclarar que cada camino es un

escenario de caso de uso. A partir de aquí se identifican los casos de pruebas teniendo en cuenta los escenarios anteriores que se quieran ejercitar, para finalmente identificar, además, los valores a probar asociados a cada caso de prueba especificado.

➤ *Listas de Chequeo*: Las listas de chequeo favorecen no sólo el proceso de ejecución de las RTF o las auditorías en una forma dinámica y comprensible, sino también al propio examen de las actividades de control de la calidad que se realizarán como parte de las tareas que fueron determinadas en el plan. La propuesta de estas listas responde a los puntos de comprobación definidos en RUP para cada flujo de trabajo. Estas listas representan una guía flexible que aporta una serie de beneficios, como por ejemplo:

- Permiten una mejor planificación.
- Sirven de asistencia a los revisores y/o auditores para lograr un mejor desempeño durante el proceso a desarrollar.
- Permiten una recopilación de los datos que surgen durante los procesos de revisiones y auditorías.
- Constituyen un repositorio de información que puede ser usado posteriormente.

➤ *Trac*: Se propone el uso de esta herramienta para gestionar todos los registros de calidad que se generen producto a las actividades de aseguramiento de la calidad efectuadas. Como software libre permite reducir los costos de su adquisición. Su principal función es ayudar al equipo de calidad a lograr una buena documentación de los registros, manteniendo a la par una gestión de los cambios controlada.

2.3.10 Solución para establecer la Resolución de Problemas y Acción Correctiva

Para determinar el procedimiento propuesto en la sección del plan correspondiente a la Resolución de Problemas y Acción Correctiva inicialmente se partió de que:

El propósito de un sistema de Gestión de Problemas y Acciones Correctivas es: (ANTONIO, 2002)

- Asegurar que todos los problemas se documentan, se corrigen y no caen en el olvido.
- Asegurar que se evalúa la validez de los informes de problemas.
- Realimentar al desarrollador y el usuario sobre el estado de los problemas.
- Proporcionar datos para medir y predecir la calidad y fiabilidad del software.

Siguiendo estas pautas se plantea:

CAPÍTULO II: CARACTERÍSTICAS Y PROPUESTA DE LA SOLUCIÓN

- *Establecer el equipo:* Seleccionar los miembros del equipo de calidad que posean conocimientos del proceso y habilidades necesarias para encontrar problemas, según el artefacto analizado, e identificar y proponer posibles acciones correctivas.
- *Describir el problema:* Describir el problema identificando "qué está mal con qué" y detallar el problema en términos entendibles para los desarrolladores. De forma más clara, identificar el objeto y el defecto, definir el problema con la mayor precisión posible, siempre tratando con un problema a la vez.
- *Proponer la acción correctiva:* Verificar la raíz del problema, si no se tiene una propuesta de solución inmediata y definitiva, por lo menos tratar de plantear una de contención provisional que les sirva de guía a los desarrolladores para determinar la forma de solución ideal.
- *Informar problemas:* Mantener informado al equipo sobre las no conformidades que se vayan detectando con el objetivo de que no sean cometidas nuevamente por la misma persona o como medio para prevenir que otras incurran en ellas. Este paso representa una acción para asegurar la imposibilidad de que las no conformidades vuelvan a ser señaladas.

Toda acción correctiva debe quedar legalmente documentada. Se plantea que durante cualquier tipo de actividad de aseguramiento de la calidad efectuada, si es encontrado algún error en el desarrollo del software el mismo se deberá agregar inmediatamente a la Plantilla de No Conformidades correspondientes. Dicha plantilla se obtuvo del repositorio de información de la DCS, se rige por la plantilla establecida por RUP; pero le fueron realizados algunos cambios en su contenido para hacerla más asequible y menos redundante a la hora de utilizarla. Los cambios efectuados se presentaron ante las personas autorizadas, como lo establecen los lineamientos de calidad de la universidad y posteriormente fueron aprobados.

En tal caso, la nueva Plantilla de No Conformidades es un aporte del presente trabajo de diploma. La misma consta de una tabla que posee 7 columnas las cuales son: *Elemento*, *Número*, *No conformidad*, *Clasificación*, *Estado de la no conformidad* y *Respuesta del equipo de desarrollo*. En la columna nombrada *Elemento* se debe especificar el componente a utilizar ya sea una aplicación o un documento, en la *No conformidad* se plantea una descripción de la no conformidad encontrada en el proceso de revisión o auditoría, en la de *Clasificación* se indica si la no conformidad es significativa, no significativa o es una recomendación, en el *Estado de la no conformidad* se agrega si la no conformidad está resuelta, pendiente o no procede, así como la fecha en que se realizó, por último la

columna *Respuesta del equipo de desarrollo* se llenará después de la segunda iteración y se especifica la conformidad o no conformidad con lo encontrado y si no se procede con la misma a explicar las causas.

2.3.11 Especificaciones de los registros de calidad

Los registros de calidad que son generados producto a la realización de las actividades definidas en el Plan de Aseguramiento de la Calidad, representan una fuente importante de documentación como guía a consultar, o de forma más concreta son empleados para obtener información referente a los estados de los artefactos evaluados. Este epígrafe describe los tipos de registros puntualizados en el plan:

- *Plan de Pruebas*: Este documento pretende dejar constancia de los recursos disponibles (roles y responsabilidades, escenarios de pruebas, recursos de sistema); los requisitos a probar, descripción de las estrategias para los tipos de pruebas, descripción del flujo de trabajo, evaluación de los resultados de las pruebas; entre otras aspectos menos significativos.
- *Casos de Pruebas*: En este registro se anotarán aspectos tales como: el nombre del caso de prueba ejecutado, la fecha y hora de la ejecución, la salida obtenida, el resultado de comparar la salida obtenida con la salida esperada. Es satisfactoria cuando la salida obtenida está dentro del rango previsto para la salida esperada, falla menor en el caso de que la diferencia entre las salidas sea considerada como una cuestión de formato, falla grave o falla catastrófica si el sistema no completó su salida o la completó pero finalmente no resultó.
- *Listas de Chequeos*: Con relación a las listas de chequeo ya se hizo alusión anteriormente a que serán usadas como herramientas para desarrollar las actividades de revisiones y auditorías; pero a la vez servirán para dejar constancia de la realización de cada una de estas actividades, además de que en ellas se habrán documentado las evaluaciones correspondientes al artefacto chequeado según los criterios establecidos para ser usadas como fuentes de datos durante la aplicación de métricas.
- *Informe de No Conformidades*: El uso del siguiente informe tiene gran relevancia ya que será generado al identificarse alguna no conformidad en cualquiera de los tipos de actividades para el aseguramiento de la calidad que hayan sido ejecutadas. En el mismo se documentará de forma bien clara los defectos encontrados al artefacto examinado, dándole seguimiento según la respuesta

proporcionada por los miembros del equipo de desarrollo del software que sean responsables de generar el artefacto chequeado.

Finalmente se plantea que deberá aplicarse la gestión de configuración a todos los registros de calidad que sean generados durante el proyecto, verificando que esté disponible la versión actualizada de dichos documentos.

2.3.12 Especificaciones sobre las actividades de entrenamiento

Partiendo de la idea que existe una proporcionalidad directa en el hecho de que: a mayor conocimiento en el personal, mayor calidad se obtiene de su trabajo; se determinó proponer una serie de cursos de capacitación y trabajos investigativos sobre los temas de aseguramiento de la calidad. En tal caso se valoró dónde pudieran radicar los mayores problemas para lograr desempeñar satisfactoriamente las actividades de aseguramiento de la calidad propuestas, e inmediatamente hacer énfasis en la consolidación de los conocimientos requeridos, además de ampliar la cultura de calidad del equipo. Y de forma más específica, con el desarrollo de estas actividades se pretende formar una cultura sólida referente a los temas de calidad tratados en un Plan de Aseguramiento de la Calidad.

2.4 Conclusiones

Con el desarrollo de este capítulo se describieron las diferentes secciones correspondientes al Plan de Aseguramiento de la Calidad propuesto para el Proyecto Banco y se logró explicar de forma detallada cómo fueron establecidas cada una de ellas. El mismo ha de significar una obligatoria consulta para los miembros del equipo de calidad (en caso de que se decida aplicar este plan), teniendo en cuenta que se especifican las estrategias a seguir durante las actividades propuestas para asegurar la calidad, y se fundamenta además la elección de los estándares, métricas, pruebas y herramientas determinadas.

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

3.1 Introducción

Anteriormente, en el presente trabajo se hizo referencia al hecho de que es imposible medir directamente la calidad del software, pero sí indirectamente alguna de sus manifestaciones. La medición por sí misma no mejora el proceso; pero la visibilidad permite profundizar en la planificación, control, gestión y mejora. Mediante la aplicación de métricas es factible observar la calidad estadísticamente sin descartar que las mediciones de cualquier fenómeno no estén exentas a cierta posibilidad de error.

La actividad de medición se realiza para indicarle a la organización si se ha aplicado o no los métodos apropiados y adecuados para el seguimiento y cumplimiento de las actividades de aseguramiento de la calidad según su plan; debiendo demostrar y confirmar la capacidad continua de cada proceso para satisfacer su finalidad prevista y poder alcanzar los resultados planificados o establecidos.

En el momento que se desarrolla este trabajo el proyecto se encuentra en una etapa inicial donde sólo se ha podido efectuar una de las actividades propuestas en el Plan de Aseguramiento de la Calidad elaborado, la cual está referida a la revisión del modelado del negocio. A partir de aquí, el presente capítulo muestra resultados preliminares de la puesta en práctica de dicha actividad, realizando a la vez un análisis de los resultados obtenidos al aplicar algunas de las métricas propuestas en el plan. Todo esto con la intención de lograr prevenir de una forma temprana los errores cometidos. Además, incluye un análisis del uso de herramientas CASE empleadas como apoyo para controlar la calidad del software desarrollado.

3.2 Beneficios de aplicar métricas

Cuando puedas medir lo que estás diciendo y expresarlo con números ya conoces algo sobre ello; cuando no puedas medir, cuando no puedas expresar lo que dices con números, tu conocimiento es precario y deficiente: puede ser el comienzo del conocimiento, pero en tus pensamientos apenas estás avanzando hacia el escenario de la ciencia. (SOLÁNS, 2004)

Partiendo del análisis relativo a que una métrica de software representa la proyección de un conjunto de objetos en el mundo de la ingeniería de software a un conjunto de objetos en el mundo matemático, se plantea la posibilidad de que en los procesos de desarrollo de software, las métricas proveen

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

información necesaria para la toma de decisiones técnicas. Cuando un proceso se puede medir y obtener resultados cuantitativos de su estado, entonces se pueden tomar decisiones sobre el mismo y analizar la necesidad o no de algún cambio. Sin embargo; *lo que no se mide, no se puede mejorar.*(ESPAÑOL, 2008)

La aplicación de métricas proporciona generalmente:

- Las bases para la estimación ya que proveen información para tomar decisiones,
- Ayudan a determinar la complejidad,
- A predecir el tiempo invertido en el mantenimiento de un sistema,
- A comprender cuando se ha alcanzado un estado deseado de calidad,
- A tener un control del progreso del proceso, y
- Determinar el impacto que tiene los cambios.

Una de las razones principales del incremento masivo del interés en la medición software ha sido la percepción de que las métricas son necesarias para la mejora de la calidad del proceso.(GARCÍA, 2007)

De forma general las métricas del software tienen como propósito ejecutar evaluaciones durante todo el ciclo de vida del software con intenciones de comprobar si los requisitos de calidad del software se están cumpliendo. De ahí que mantener una supervisión constante de las actividades de mejora del proceso, ayuda a identificar lo que funciona y lo que no funciona y proporciona además, una visibilidad del proceso software incrementada como resultado de haber adquirido una cultura de calidad en la organización.

Finalmente, para fortalecer la idea de los beneficios que implica aplicar métricas sería bueno proporcionar algunos elementos de los costes incurridos como resultado de no medir. Por ejemplo, se incrementa la incapacidad para estimar y planificar de forma realista, determinar el progreso, evaluar la calidad, reconocer mejoras y las oportunidades para su desarrollo. Todo esto conlleva a insatisfacciones por parte del cliente y por consiguiente una pérdida de la posición competitiva para la organización en el mercado.

3.3 Descripción de los pasos a seguir para la aplicación de las métricas que fueron usadas

Los pasos que se propone seguir para la aplicación de las métricas, como fue planteado en el Epígrafe 2.3.4 del Capítulo # 2, parten de la estrategia propuesta por Pressman para los procesos de medición. No se debe omitir ninguno porque en cada uno se desarrolla una función específica para lograr obtener los resultados deseados.

3.3.1 Formulación

Formulación: Seleccionar métricas apropiadas a implementar según los atributos que se desea medir, así como la obtención de medidas. Un tipo de escala de medición de la métrica debe ser identificado para cada medida. Los tipos de escalas de las métricas son: nominal, ordinal, de intervalo, relativa y absoluta.

Teniendo en cuenta los objetivos de calidad definidos, se puede concluir que la característica de calidad de mayor peso es la funcionalidad. A partir de este análisis se pretende medir, como fue propuesto en el plan, el atributo *idoneidad* utilizando una de las métricas de idoneidad seleccionada de la norma ISO/9126.

La característica de usabilidad también entra entre los objetivos de calidad definidos, aunque el atributo de *conformidad con la confiabilidad*, que es la capacidad del producto software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la confiabilidad, no se incluye dentro de los atributos especificados; pero se ha considerado de interés para ser medido dado el enfoque y propósito de su medición.

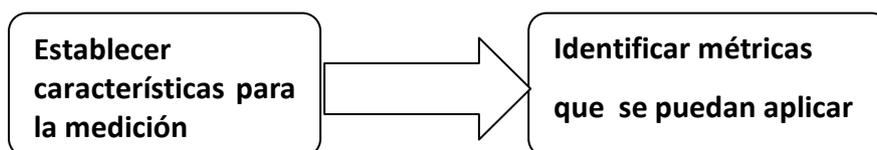


Figura 4: Representación de las acciones a seguir durante la Formulación

Cada proyecto tiene la entera libertad de seleccionar métricas concretas adecuadas para su entorno, modelos como CMM instan a las organizaciones a identificar individualmente para cada área clave del proceso, uno o más conjuntos de métricas significativas. De las métricas que fueron propuestas en el plan, serán aplicadas para una etapa inicial del proyecto (que abarca los procesos desarrollados en el modelado del negocio) las mencionadas a continuación:

Nombre de la métrica: Adecuación funcional.

Tipo de escala: Absoluta.

La métrica se propone medir ¿Cuán adecuada es la función evaluada?

Adaptando esta idea al propósito particular de la medición la interrogante quedaría de la siguiente forma: ¿Cuán adecuado es el criterio evaluado para el proceso modelado?

Nombre de la métrica: Conformidad con la usabilidad.

Tipo de escala: Absoluta.

La métrica se propone medir: ¿Cuánto se adhiere el producto de software a las regulaciones aplicables, las normas, y convenciones, relacionadas con la confiabilidad?

Adaptando esta idea al propósito particular de la medición la interrogante quedaría de la siguiente forma: ¿Cuánto se adhiere el proceso software a los criterios evaluados en el proceso modelado relacionados con la confiabilidad?

3.3.2 Colección

Colección: Determinar un mecanismo para acumular datos necesarios para poder aplicar las métricas formuladas.

Para desarrollar este paso se hizo uso de la *Lista de chequeo para el modelado del negocio* la cual fue creada concretamente para la actividad de Revisión de los Procesos del Negocio propuesta en el plan; la misma incluye una serie de aspectos enfocados a encontrar deficiencias al trabajo desarrollado por los analistas y, oportunamente los resultados obtenidos a través de su aplicación sirvieron como medio para acumular datos de los atributos que se establecieron para ser medidos.

El equipo de calidad del proyecto conformado por 10 integrantes estuvo encargado de llenar una lista de chequeo por cada uno de los procesos modelado por los analistas. Ellos llenaron la columna donde se define la evaluación con valores 0, 2, 4 y 5, según su juicio de adecuación con los criterios de interés que fueron chequeados; acumulando de esta forma los datos necesarios para la medición. Aclarar que la columna relacionada con el nivel, en este caso, fue especificada por las desarrolladoras del presente trabajo, debido a que son las que pretenden obtener resultados de los datos recolectados para el proceso de medición.

➤ Método de aplicación para la métrica *Adecuación funcional*:

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Según la norma ISO/9126 se especifica como: Número de funciones idóneas para ejecutar funciones específicas en comparación con el número de funciones evaluadas.

Luego de adaptarlo a las particularidades del mecanismo de medición quedaría como: Número de criterios evaluados idóneos para modelar el proceso especificado en comparación con el número de criterios evaluados.

➤ Método de aplicación para la métrica *Conformidad con la usabilidad*:

La norma ISO/9126 lo plantea como: Especificación de los elementos que requieren estar en conformidad, diseñe los casos de prueba de esos elementos, conduzca las pruebas funcionales para esos elementos que requieren estar en conformidad.

Adaptando este método a una esta etapa inicial de desarrollo y según el contexto propio de la medición sencillamente quedaría como: Especificación de los elementos que requieren estar en conformidad.

3.3.3 Análisis

Análisis: Hacer el cálculo de la métrica empleando fórmulas y herramientas matemáticas.

➤ Fórmula para la medición de la métrica *Adecuación funcional*:

$$X = 1 - A/B$$

A - Número de criterios evaluados en las cuales se detectaron problemas.

B - Número de criterios evaluados.

➤ Fórmula para la medición de la métrica *Conformidad con la usabilidad*:

$$X = 1 - A / B$$

A- Número de criterios especificados que requiriendo estar en conformidad no han sido desarrollados correctamente (o sea, aquellos criterios a los que se le dio nivel 5 y no fueron evaluados con el máximo valor).

B - Número total de criterios especificados que requieren estar en conformidad (criterios a los que se les dio nivel 5).

La siguiente tabla contiene el cálculo y el resultado de la aplicación de las fórmulas para cada métrica:

<i>Nombre del proceso del negocio revisado</i>	<i>Conformidad con la usabilidad</i>	<i>Adecuación funcional</i>
--	--------------------------------------	-----------------------------

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Negociación de CC	1- 6/73=0.92	1-14/106=0.87
Negociación de CC Importación	1- 3/73=0.96	1-10/106=0.91
CC Garantía	1- 1/73=0.99	1-2/104=0.99
Realizar informe de liquidez	1-0/42=1	1-0/65=1
Confeccionar posición del banco por cuenta	1-1/46=0.98	1-4/77=0.95
Cargar tipo de cambio	1- 5/45=0.89	1-7/7=0.91
Realizar compra/venta de moneda	1-0/48=1	1-0/71=1
Realizar transferencia	1-1/47=0.98	1-4/79=0.95
Conceder depósito	1- 1/49=0.98	1- 8/82=0.91
Renovar depósito concedido	1-0/47=1	1-4/77=0.95
Cobrar intereses por depósito concedido	1- 2/48=0.96	1-12/77=0.85
Tomar depósito	1-1/50=1	1-4/80=0.95
Renovar depósito tomado	1-0/43=1	1-5/76=0.94
Chequear préstamos/depósitos	1- 3/41=0.93	1- 5/69=0.93
Conceder préstamos	1- 5/46=0.89	1- 11/78=0.86
Cobrar por préstamo concedido	1- 4/46=0.91	1-7/79=0.92
Contabilizar vencimiento de un préstamo recibido/concedido	1- 5/41=0.88	1-8/67=0.89
Recibir préstamos	1- 4/43=0.91	1- 9/75=0.88
Renovar préstamo recibido/concedido	1-0/42=1	1-3/70=0.96
Avalar letras de cambio	1- 2/47=0.96	1-5/79=0.94
Conceder financiamientos contra cuenta única	1- 4/68=0.94	1- 6/98=0.94
Contabilizar gastos por transferencias	1- 2/44=0.95	1- 6/77=0.93
Apertura de la Carta de Crédito de Importación	1- 0/51=1	1-4/106=1
Enmienda de la Carta de Crédito de Importación	1- /47= 0.98	1-5/78=0.94
Vencimiento de la Carta de Crédito de Importación	1- 0/42=1	1-0/68=1
Recepción de Cartas de Crédito de Exportación	1- 2/71=0.96	1-16/104=0.86
Enmiendas de Cartas de Crédito de Exportación	1- 3/73=0.96	1-12/104=0.89
Cancelación de Cartas de Crédito de Exportación	1- 0/43=1	1-0/72=1
Ampliación de la Licencia	1- 11/73=0.85	1- 24/98=0.76
Apertura de Cuentas Corrientes o Escrow	1- 0/41=1	1-1/67=0.99
Apertura de Cuentas Ingreso o Financiamiento	1- 1/40=0.97	1-2/66=0.97
Apertura del Expediente Legal	1- 0/38=1	1-2/64=0.97
Autorizar Sobregiro	1- 10/73=0.86	1- 25/106=0.77

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Cancelar Sobregiro	1- 10/73=0.86	1- 25/106=0.77
Bloquear Cheque	1- 8/73=0.89	1-18/106=0.84
Cancelación de Firmas Autorizadas	1- 3/73=0.96	1-15/106=0.86
Certificar Cheque	1- 3/73=0.96	1-15/106=0.86
Cierre de Cuentas x Disposición	1- 27/73=0.63	1- 7/106=0.84
Cierre de Cuentas x Mandato	1- 8/73=0.89	1- 9/106=0.83
Cierre de Cuentas x Titular	1- 10/73=0.86	1- 20/106=0.82
Cobrar Cheque Certificado recibido por una gerencia	1- 7/73=0.90	1-21/106=0.81
Cobrar Cheque no Certificado recibido por una gerencia	1- 7/73=0.90	1-21/106=0.81
Cobrar Cheque para aporte al combustible	1- 10/73=0.86	1-21//106=0.81
Cobrar cheque recibido por ventanilla	1- 7/73=0.90	1-19/106=0.83
Control de firmas autorizadas	1- 7/73=0.90	1- 20/106=0.82
Control de firmas autorizadas a CC	1- 10/73=0.86	1-19/106=0.83
Desglose de Pasaje	1- 7/73=0.90	1-19/106=0.83
Encaje Legal	1- 10/73=0.86	1- 22/106=0.80
Enviar Transferencia Internacional	1- 5/62=0.92	1-18/88=0.8
Enviar Transferencia Internacional contra CU	1- 4/55=0.93	1-18/88=0.8
Enviar Transferencia Nacional	1- 4/54=0.93	1-15/88=0.83
Enviar Transferencia Nacional contra CU	1- 4/55=0.89	1-18/88=0.8
Extracción de Efectivo Cliente	1 - 10/73=0.86	1- 21/106=0.81
Pagar Cheque	1- 9/73=0.88	1-22/106=0.80
Procesar Cheque rechazado	1-5/44=0.89	1-12/81=0.86
Recibir Transferencia de Banco a Banco	1-5/70=0.93	1-12/93=0.88
Recibir Transferencia para Aporte al combustible	1- 5/73=0.93	1-12/93=0.88
Recibir Transferencias para Personas Naturales	1- 5/72=0.93	1-11/94=0.89
Recibir Transferencias para el Recargo Comercial 2%	1- 12/73=0.84	1- 25/106=0.77
Recibir Transferencias para Personas Jurídicas	1-6/47=0.87	1-14/80=0.83
Registrar Firma Autorizada	1- 8/73=0.89	1-18/106=0.84
Traspaso	1-2/47=0.96	1-10/83=0.88
Venta de Chequera	1-2/47=0.96	1-9/78=0.89
Inicio del día contable	1- 7/47=0.85	1-10/75=0.87
Cierre del día contable	1- 7/42=0.83	1-14/72=0.81
Contabilización	1- 0/73=1	1-0/106=1

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Actualización de Clasificadores	1- 11/35=0.69	1-15/64=0.77
Actualización de Usuario	1- 0/71=1	1-1/104=0.99
Análisis del Balance General	1- 0/73=1	1-0/106=1
Envío del Modelo Tarea 10	1- 1/38=0.97	1-3/65=0.96
Salvas del DEF_DAFE	1- 0/73=1	1-0/106=1
Reporte de Errores	1- 0/71=1	1-5/104=0.96
Seguimiento y análisis de tablas	1- 5/27=0.81	1- 6/51=0.89
Servicio de asesoramiento	1- 6/38=0.82	1-12/68=0.83
Confirmar operación	1-0/41=1	1-3/67=0.96

Tabla 2: Datos de la aplicación de las métricas, Adecuación Funcional y Conformidad con la Usabilidad

3.3.4 Interpretación

Interpretación: Evaluar los resultados obtenidos de la aplicación de la métrica enfocándose en conseguir una visión interna de la calidad de la representación.

Para analizar los resultados obtenidos durante la aplicación de las métricas *Adecuación funcional* y *Conformidad con la usabilidad* se definió un rango de valores a través del cual se pudo clasificar el proceso modelado según un nivel de aceptación con los atributos medidos. La definición del rango se hizo a partir de la identificación de los resultados predominantes entre el peor, el mejor y una proporción relativa entre ambos.

Los resultados obtenidos para cada proceso han de ser establecidos, según la escala del rango donde pertenezcan, para determinar una clasificación en dependencia del valor correspondiente.

<i>Rango</i>		<i>Clasificación</i>
1	[0.95 - 1]	Alto
2	[0.85- 0.94]	Medio
3	[0 – 0.84]	Bajo

Tabla 3: Rango para la interpretación de las métricas: Adecuación funcional y Conformidad con la usabilidad

- Interpretación del valor obtenido al aplicar la métrica *Adecuación funcional*:
 $0 \leq X \leq 1$; A mayor cercanía al 1 resultará más adecuada.
- Interpretación del valor obtenido al aplicar la métrica *Conformidad con la usabilidad*:
 $0 \leq X \leq 1$; A mayor cercanía al 1 mejor.

3.3.5 Realimentación

Realimentación: Trasmitir al equipo de desarrollo los resultados y las recomendaciones obtenidas de la interpretación de la métrica aplicada.

Con relación a este último paso el Responsable de Calidad está en la obligación de informarle al Jefe de Proyecto los resultados obtenidos una vez realizado el proceso de medición; demostrando con datos reales la veracidad del mismo. A partir de este momento el Jefe de Proyecto debe realizar discusiones dentro de su equipo de desarrollo con relación a las conclusiones obtenidas del proceso de medición, enfocándose en cómo estos resultados reflejan lo que está sucediendo y si alguna nueva cuestión identificada a través del análisis de la métrica es válida o no. El proceso culmina con las conclusiones finales y recomendaciones del caso de estudio.

- Conclusiones de aplicar la métrica: *Conformidad con la Usabilidad*

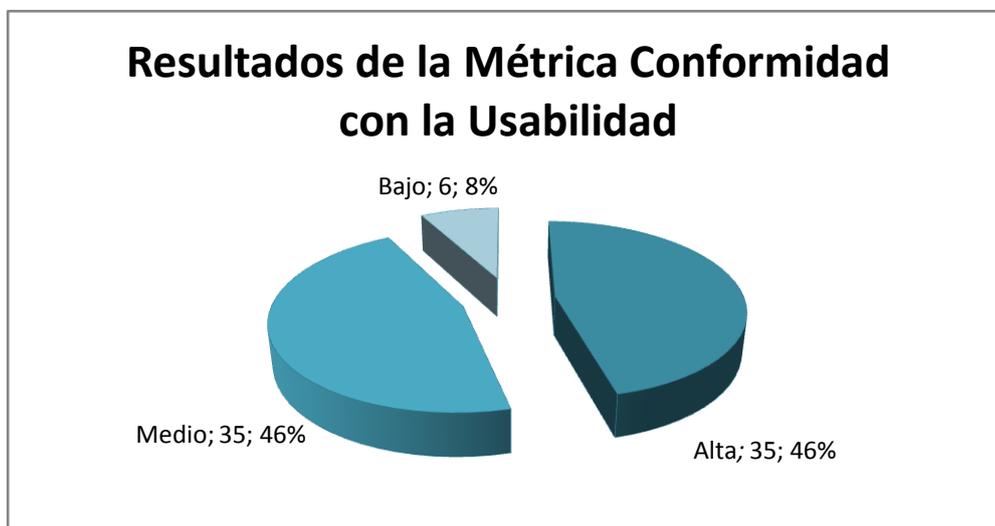


Figura 5: Resultados de aplicar la métrica Conformidad con la usabilidad

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

En el gráfico de la Figura 5 se ilustra los resultados de los 76 procesos revisados, una vez aplicada la métrica y clasificados dichos resultados en el rango establecido. Para dar respuesta a la interrogante que propone medir la métrica sería bueno comenzar partiendo de la especificación de algunas estadísticas. Por ejemplo: Un total de 35 procesos se adhieren con un alto nivel a los criterios evaluados relacionados con la confiabilidad y específicamente al atributo *conformidad con la usabilidad*, de ellos 17 con la máxima evaluación (1), representando un 22,4% respecto a todos los procesos modelados. De igual forma otros 35 procesos se adhieren con un nivel medio y los 6 restantes con un bajo nivel.

Estos valores reflejan que menos de la mitad de los procesos revisados contienen óptimas evaluaciones de los elementos que requieren estar en conformidad, o sea, todos aquellos criterios considerados como los más importantes. Se recomienda al equipo de los analistas que tengan presentes, como primera opción, para dar respuesta a las no conformidades detectadas, todos los procesos revisados que se incluyen en el rango de las clasificaciones medio y bajo. Además, que sean más cuidadosos durante el modelado de los procesos del negocio para lograr obtener un alto grado de conformidad con los elementos esenciales desarrollados.

➤ Conclusiones de aplicar la métrica: *Adecuación Funcional*

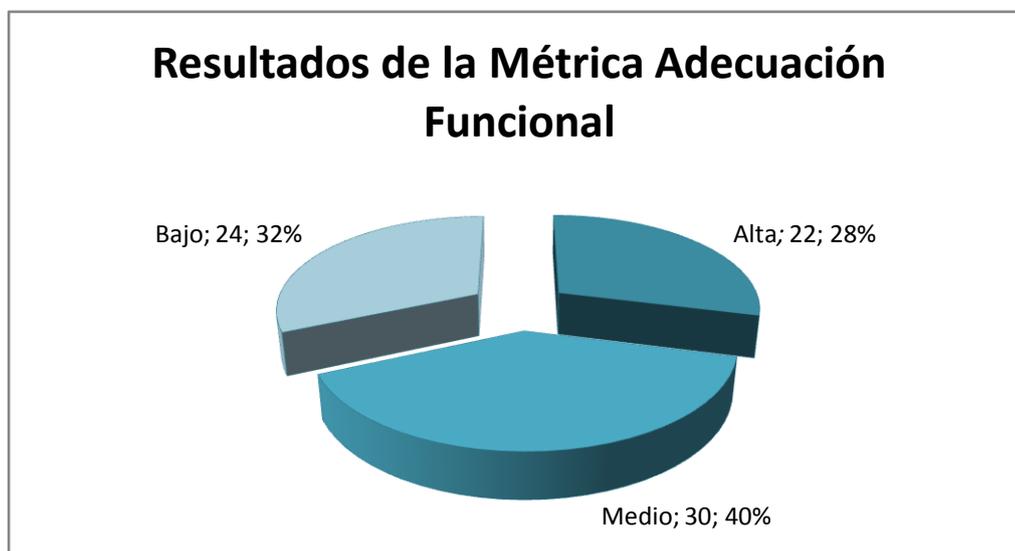


Figura 6: Resultados de aplicar la Métrica Adecuación Funcional

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Con relación a los resultados obtenidos de la aplicación de esta métrica para los 76 procesos revisados, un total de 35 procesos (46%, menos de la mitad con respecto al total) se adecuan con un alto nivel a los criterios evaluados y particularmente con el atributo de idoneidad, aunque sólo 8 de ellos con la máxima evaluación (1), representando un 10,5% del total de procesos modelados. Seguidamente en un nivel medio de adecuación se encuentran 30 procesos (39%) y a los 24 restantes (32%) le corresponden un nivel bajo.

De todas estas estadísticas se puede concluir que: a menos de la mitad de los procesos modelados le corresponde un mayor número de criterios evaluados como idóneos para modelar un proceso del negocio. Teniendo en cuenta este análisis se propone al equipo de analistas del proyecto que prioricen, durante el modelado de los procesos del negocio, aquellos elementos que requieren de un desarrollo más eficiente; para lograr un nivel más elevado de adecuación en los elementos modelados de los procesos donde fueron identificadas las no conformidades y en los futuros procesos que serán modelados.

3.4 Otros resultados

Para determinar el porcentaje de adecuación de un proceso del negocio modelado con los criterios evaluados se desarrolló el cálculo en el cual se obtuvo el valor ideal que se pudiera lograr si la totalidad de los criterios hubiera sido evaluada con la máxima puntuación (5), descartando todos aquellos criterios que se identificaron en el proceso como no proceden y luego se obtuvo el valor real de la sumatoria de cada criterio evaluado según su concordancia con el proceso modelado.

Esta medición fue desarrollada inicialmente para cada lista de chequeo obtenida según el proceso revisado (los datos se pueden encontrar en la tabla incluida en el Anexo 3), y usando la misma estrategia que en el Epígrafe 3.3.4 se definió un rango para clasificar los resultados, el cual se muestra en la siguiente tabla.

<i>Rango</i>		<i>Clasificación</i>
1	[94.5 – 100%]	Alto
2	[84.5 - 94.4%]	Medio
3	[0 – 84.4%]	Bajo

Tabla 4: Rango para la interpretación del porcentaje de adecuación de un proceso del negocio modelado con los criterios evaluados

- Interpretación del valor obtenido de la medición

$0 \leq X \leq 100$; A mayor cercanía al 100 resultará más adecuada.

Seguidamente se procesaron todos los datos, y los resultados obtenidos se representan en el gráfico mostrado a continuación:

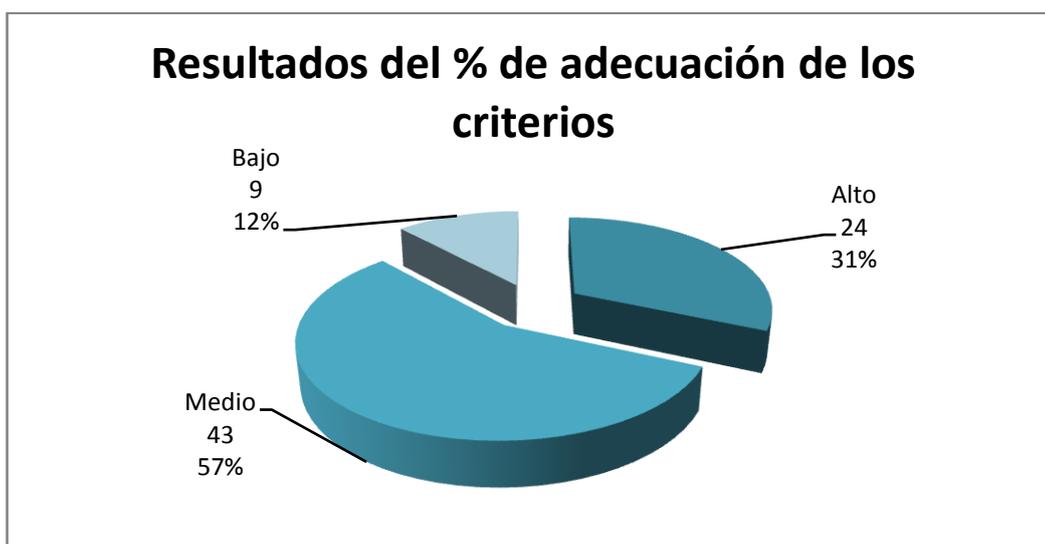


Figura 7: Resultados de la medición del porcentaje de adecuación de un proceso del negocio a los criterios evaluados

También se determinaron estadísticas globales para este tipo de medición. Por ejemplo: la adecuación a los criterios evaluados como promedio fue de un 96,6% para cada proceso del negocio modelado; pero la real, para toda la etapa del modelado de los procesos del negocio fue de un 89,96%.

Posteriormente se presentan otros tipos de resultados referentes a los defectos cometidos por los analistas en la etapa concerniente al modelado de los procesos del negocio:

- *Porcentaje de defectos inyectados para una fase:* Este tipo de medida no posee una métrica directamente asociada; pero sí permite valorar en qué fases del proceso se están inyectando la mayoría de los defectos y cuál requiere una mayor atención y revisión.

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Los defectos cometidos por los analistas que fueron encontrados por los revisores se pueden clasificar en dos grandes grupos, en un primer grupo los problemas por mala redacción de la plantilla que describe el modelado del proceso del negocio y, en un segundo grupo los problemas técnicos que incluyeron aspectos tales como: chequeo en cuanto al uso adecuado de estereotipos mediante el modelado a través de la notación BPMN, definiciones de estados de los artefactos generados, en cuanto al correcto modelado de los diagramas de los procesos y al uso adecuado de conceptos vinculados a principios del negocio.

Para calcular el porcentaje de defectos inyectados durante el modelado del negocio se hizo uso de la lista de chequeo creada como herramienta para obtener los datos necesarios sobre los defectos cometidos. A través de la misma y siguiendo el propósito de conformar una idea más clara de dónde radicaron los mayores problemas, se calcularon las no conformidades que fueron encontradas correspondientes a los 67 criterios de evaluación para los aspectos técnicos y los 39 criterios relativos al correcto llenado de la plantilla en cada proceso del negocio revisado; aclarar que siempre fueron descartados los criterios que no procedían en dependencia del proceso evaluado.

Como resultado se obtuvieron 196 no conformidades correspondientes a los defectos técnicos, las cuales representan el 23,5% del total de defectos inyectados y 637 correspondientes a las no conformidades vinculadas al mal llenado de la plantilla, las cuales ocupan el 76,5%. Evidentemente los mayores problemas fueron consecuencia del descuido por parte de los analistas a la hora de desarrollar una redacción adecuada de la plantilla.

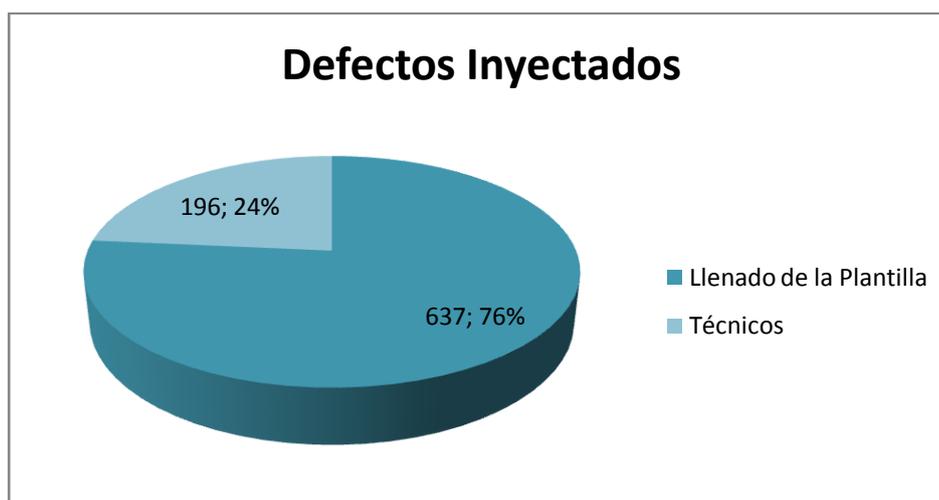


Figura 8: Representación de los datos de defectos inyectados durante el modelado de los procesos del negocio

En el Anexo 4 se representa mediante un gráfico, de forma particular para cada proceso del negocio revisado, la cantidad de defectos inyectados de cada tipo.

Para clasificar la cantidad de defectos inyectados con relación a la aceptabilidad de los mismos por cada proceso modelado, se definió un rango usando los procedimientos antes mencionados.

<i>Rango</i>		<i>Clasificación</i>
1	[0 – 5]	Alto
2	[6 - 15]	Medio
3	[16 o más]	Bajo

Tabla 5: Rango para la interpretación de la aceptabilidad de defectos por procesos

➤ Interpretación del valor obtenido al aplicar la medición:

$0 \leq X \leq 106$; A menor lejanía del 0 resultará más aceptable.

Los resultados obtenidos se muestran en la siguiente gráfica:

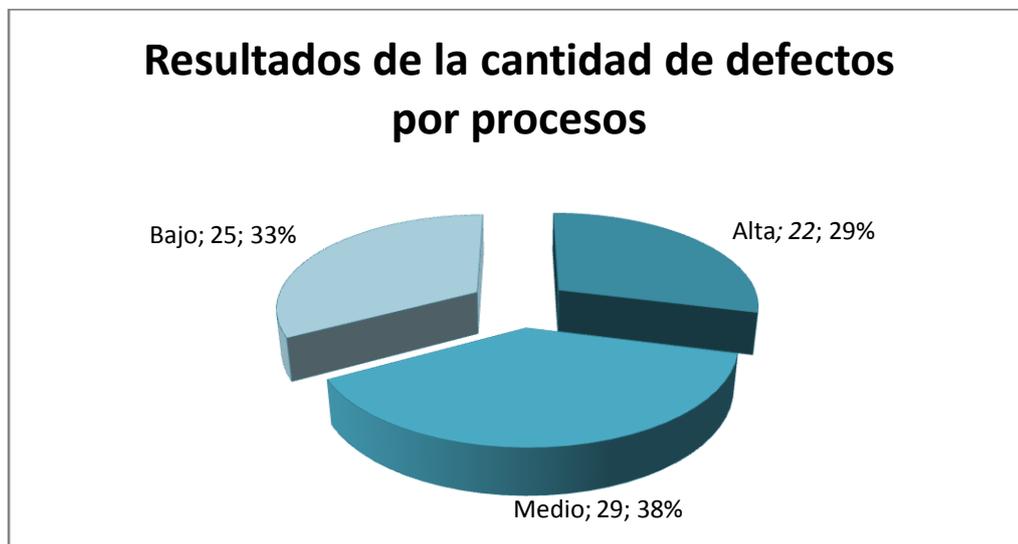


Figura 9: Resultados de la interpretación de la cantidad de defectos por procesos

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Para mayores detalles en las gráficas que se incluyen en el Anexo 5 se representan los resultados de porcentos de defectos obtenidos por cada proceso revisado.

El modelado de un proceso fue considerado como aceptable si no excedía de 5 no conformidades. Teniendo en cuenta los resultados, según el rango establecido, la mayoría de los procesos revisados se encuentran en un nivel medio o bajo de clasificación. Se necesita que, inicialmente, los analistas den respuesta a las no conformidades identificadas en estos procesos para luego, en una segunda revisión, chequear que se logren incluir en un nivel alto de clasificación, mejorado así su evaluación.

Asumiendo que el proyecto sólo ha rebasado la etapa del modelado del negocio no es posible llegar a hacer comparaciones; pero se ha decidido obtener para este trabajo el porcentaje de defectos inyectados durante el modelado del negocio y, se propone que al concluir la etapa de la captura de requisitos (la cual se desarrolla en estos momentos) se obtengan resultados similares, para luego efectuar las comparaciones pertinentes determinando si se logró o no disminuir los defectos inyectados, y por consiguiente elevar también la calidad del trabajo desarrollado.

Se pueden formular un conjunto de estadísticas de los datos obtenidos, como por ejemplo: tan solo a 7 procesos del total de procesos revisados no les fue encontrado ninguna no conformidad, el mayor porcentaje de defectos inyectados a un proceso fue del 25,5%, el promedio de defectos por proceso representa un 12% y el total de defectos reales que fueron inyectados durante este flujo de trabajo también es de un 12%.

Conclusiones generales: Se recomienda a todo el equipo de analistas hacer hincapié en el correcto llenado de la plantilla que describe los procesos del negocio, dado que ahí fueron identificados los mayores problemas, y que conscientemente les den la importancia requerida para lograr que el proyecto tenga una documentación adecuada. Todo esto sin restarle importancia a los defectos técnicos que también fueron identificados aunque a menor escala.

3.5 Uso de Herramientas CASE para incrementar la calidad

Según Pressman las herramientas de Ingeniería del Software Asistida por Computadora (CASE) son más importantes que el hardware para conseguir buena calidad y productividad, aunque la mayoría de los desarrolladores del software todavía no las utilicen eficazmente. Las mismas ayudan al ingeniero

CAPÍTULO III: ANÁLISIS DE LOS RESULTADOS OBTENIDOS

de software en la obtención de resultados de alta calidad y a garantizar que la calidad se diseñe antes de llegar a construir el producto.

En esta última parte del trabajo se pretende realizar un análisis del uso de estas herramientas con el objetivo de profundizar en los beneficios que aportarían las mismas como complemento de las prácticas de ingeniería de software durante el desarrollo del producto, tales como: herramientas automáticas de estimaciones de costo y de esfuerzo, herramientas gráficas para la notación gráfica del diseño, herramientas que prestan su asistencia en la planificación, desarrollo y control de las pruebas, etc.

Es posible decir que el uso de herramientas CASE es visto en el mundo del software como una solución para los mayores problemas relacionados con las formas tradicionales de desarrollo (estas se refieren a: demoras, errores inconsistentes, dificultad para las pruebas e inadecuada documentación); y además como un soporte para el desarrollo de sistemas de calidad a través de la automatización del ciclo de vida de desarrollo del software.

Por ejemplo, una de las causas de errores encontrados al software es debido al Error en la representación de los datos (ERD), para mejorar estos defectos, el equipo de desarrollo del software podría hacer uso de herramientas CASE para su modelización y realizar revisiones del diseño de datos de forma más rigurosa. Existen herramientas CASE para métricas que hacen una auditoría del código fuente para determinar si se ajusta o no a ciertos estándares del lenguaje y de esta forma controlar la calidad con mayor precisión.

Kavi y Nahouraii indican que "la calidad del software puede verse mejorada por una correcta selección de la herramienta que apoye cada fase del proceso de desarrollo, desde el análisis de requerimientos hasta las pruebas finales y la integración. La selección de una herramienta inapropiada, por el contrario, puede impedir el buen desarrollo del software". (MORALES, 2006)

A modo de conclusión se plantea que: en un principio es de vital importancia para un proyecto de software el uso de las herramientas CASE; pero a la par también es significativo realizar una buena elección de las mismas. O sea, el propósito radica en analizar una herramienta con el objetivo de determinar una mejora con respecto a la facilidad de uso y a los resultados obtenidos orientados a incrementar la calidad del software y efectuar su elección después de haber evaluado diferentes criterios.

3.6 Conclusiones

Con el desarrollo del presente capítulo se logró:

- Desarrollar una implementación inicial del plan propuesto una vez efectuada la revisión planificada al modelado de los procesos del negocio por los integrantes del equipo de calidad del Proyecto Banco.
- Haciendo uso de la lista de chequeo creada para esta actividad fue posible llegar a la aplicación de un par de métricas propuestas en el plan, además de otros tipos de mediciones.

De esta forma se obtuvieron resultados referentes al nivel de adecuación del modelado de los procesos del negocio con los atributos de calidad: *idoneidad* y *conformidad con la usabilidad*, incluyendo otras estadísticas de interés relacionadas con las no conformidades para determinar dónde radicaron los mayores problemas cometidos por los analistas y transmitirlos a la vez. Haciendo uso de estos conocimientos ellos están en la obligación de erradicarlos desde un inicio. Como información de apoyo para constatar el uso de las métricas se hizo referencia a los beneficios que aporta su aplicación durante el desarrollo de un producto software y finalmente también se hizo referencia al hecho de que un buen uso y selección de herramientas CASE ayuda a incrementar la calidad del software.

CONCLUSIONES

En el presente trabajo de diploma se realizó un estudio exhaustivo relacionado con la elaboración de un Plan de Aseguramiento de la Calidad formando parte de los procesos de aseguramiento de la calidad desarrollados dentro de un proyecto software. Como resultado se obtuvo un Plan de Aseguramiento de la Calidad para el Proyecto Banco, el mismo servirá de guía para la realización de las actividades necesarias en vistas a lograr un producto con altos índices de calidad como se exige actualmente en el mundo del software.

También se obtuvieron diversas plantillas que servirán para ser utilizadas en un futuro por otros proyectos de la universidad, tales como: el Informe de No Conformidades, la Lista de chequeo general y las demás listas de chequeo elaboradas para la comprobación de las actividades del Modelado del Negocio y la Captura de Requisitos, de las cuales es posible su reutilización.

A medida que se fue desarrollando el plan se requirió incrementar los conocimientos sobre los temas abordados en el mismo para obtener una propuesta lo más acorde posible a las especificidades del Proyecto Banco. Partiendo de este punto y una vez consolidados los conocimientos necesarios, se llegó a la conclusión de que:

- El Plan de Aseguramiento de la Calidad debe contemplarse de una forma muy vinculada al proceso de desarrollo del software que se quiere obtener y no de forma independiente. Desde este punto la planificación de las tareas de aseguramiento de la calidad definidas en el plan están estrechamente vinculadas con el desarrollo y planificación del proyecto en cuestión.
- Con una correcta revisión de los artefactos generados en cada etapa de desarrollo del software, es posible corregir los errores que son introducidos a través de una detección temprana de los mismos, y de esta forma hacer menos costoso el desarrollo del proyecto, además de obtener un producto más eficiente y con mayor calidad.
- La utilización de métricas para saber en qué estado se encuentra los artefactos generados durante el ciclo de vida del software, brinda una visión más amplia para evaluar cómo se va desarrollando el producto, según la calidad requerida, y además dónde se están cometiendo los errores para erradicarlos o prevenirlos.

Finalmente para concluir, es válido aclarar que se logró el objetivo principal de este Trabajo de Diploma consiste en definir los procesos para asegurar la calidad mediante un Plan de Aseguramiento de la Calidad y, se realizaron las tareas correspondientes a la investigación, cumpliendo de esta forma los objetivos específicos.

RECOMENDACIONES

Si se decide llegar a usar la propuesta del Plan de Aseguramiento de la Calidad para el Proyecto Banco obtenida como resultado del presente trabajo de diploma, se recomienda a los miembros del equipo de calidad del proyecto, responsables de su aplicación:

- Seguir todas las indicaciones especificadas en el plan, a la medida como fueron planteadas, y de una forma consciente.
- Desarrollar un Plan de Métricas en el cual se incluyan, además de las que fueron propuestas en el plan, otros tipos de métricas para realizar mediciones tanto al proceso como al producto software y sus recursos, tales como: métricas del proyecto, métricas técnicas, métricas orientadas a función; con el fin de obtener otros tipos de indicadores.
- Desarrollar un Plan de Pruebas utilizando la plantilla propuesta en este trabajo, en el cual se especifiquen los distintos tipos de pruebas propuestas, además de tener en cuenta la estrategia señalada.
- Si finalmente el Proyecto logra diseñar un cronograma, entonces sería más factible elaborar un cronograma real para las actividades de revisiones y auditorías especificadas.
- Analizar, con el equipo de analistas, los resultados de la revisión y medición realizada al modelado de los procesos del negocio, con el objetivo de informarles los resultados obtenidos para que estén conscientes de los problemas encontrados y logren darle una solución inmediata.
- Hacer de conocimiento público el Plan de Aseguramiento de la Calidad propuesto para todos los integrantes del proyecto con el fin de tener en cuenta los estándares y modelos de calidad que se proponen en el mismo a ser aplicados durante el desarrollo del software.
- Ceder a otros proyectos de la universidad la Plantilla de No Conformidades obtenida, la cual hace el trabajo de los miembros de calidad más flexible y ágil a la hora de documentar algún problema encontrado.

REFERENCIAS BIBLIOGRÁFICAS

ALARCÓN, A. S. Modelos de calidad. 2004,

ANTONIO, A. D. Gestión, Control y Garantía de la Calidad del Software. 2002, nº

CAIRO, C. C. Procedimiento para la gestión de los costos de calidad (Cuba) Disponible en:
<http://www.monografias.com/trabajos45/costo-de-calidad/costo-de-calidad.shtml>.

CASAL, S. M. S. ESTÁNDARES. Editado por: Digital, E. Y. F. 2007, Disponible en:
<http://www.cesdonbosco.com/revista/articulos2007/junio07/soniasantove%C3%B1a.pdf>. ISBN
1695.4297.

COMERCIO.ES. ¿Qué es el control de calidad? . 2008, Disponible en:
http://www.comercio.es/NR/exeres/5D43FFE9-1394-4937-957A-8779F1901A61_frameless.htm.

CORONAS, M. J. Ampliación de estándares de desarrollo en Costa Rica. 1997, nº Disponible en:
http://www.micit.go.cr/docs/gal_jenkins.doc, .

ESPAÑOL, E. L. U. E. Lord Kelvin. 2008, Disponible en:
http://enciclopedia.us.es/index.php/Lord_Kelvin.

ESPAÑOLA, R. A. Calidad Disponible en: <http://www.rae.es/rae.html>.

GARCÍA, F. Proceso Software y Gestión del Conocimiento. 2007, nº Disponible en: <http://alarcos.inf-cr.uclm.es/doc/psgc/doc/psgc-4a.pdf>.

GOODMAN, P. Software Metrics-Best Practices for Successful IT Management. 2004.

ISO. Gestión de la calidad y garantía de la calidad. Vocabulario. 1994,

---. ISO 9000. 2000,

---. Software Product Evaluation - Quality Characteristics and Guidelines for their Use. 2001,

JENKINS, M. Enseñando Aseguramiento de la Calidad del Software en un Programa de Posgrado. 2003, Disponible en: <http://lsm.dei.uc.pt/ribie/docfiles/txt20031215113514TCI24.pdf>.

LOUDEN, K. C. Construcción de Compiladores: Principios y Prácticas. 2004.

MANUEL DE LA VILLA, M. R., ISABEL RAMOS. Modelo de Evaluación y Mejora de Procesos: Análisis Comparativo. 2005, nº Disponible en:
<http://www.sc.ehu.es/jiwdocoj/remis/docs/DelaVillaadis2004.doc>.

MARINESCU, F. Consejos sobre pruebas de rendimientos y optimización Disponible en:
<http://espanol.aurumsol.com/articulos/art4/art4-print.html>.

- MELVIN. ¿Qué es un modelo de calidad? 2008, Disponible en:
http://modelosdegestiondelacalidad.blogspot.com/2008/01/que-es-un-modelo-de-calidad_12.html.
- MENDOZA, L. E. Sistemas de Información III Teoría. 2003, Disponible en:
[http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDa\)/PS6117%20Calidad%20del%20Software.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDa)/PS6117%20Calidad%20del%20Software.pdf).
- MEYERS, G. J. The Art of Software Testing,. 1979.
- MORALES, L. E. M. Estudio del Proceso de Selección de Herramientas Automatizadas para el desarrollo de Sistemas de Información Asistidas por computadora. 2006, nº Disponible en:
<http://www.lisi.usb.ve/reportes/2000-HC-01.pdf>.
- PRESSMAN, R. Ingeniería de Software. 2001. Un Enfoque Práctico.
- QUALITRAIN. Aseguramiento de la calidad del software. 2001, Disponible en:
<http://www.qualitrain.com.mx/index.php/Procesos-de-software/Aseguramiento-de-la-calidad-de-software.html>.
- SANS, M. C. Las normas ISO. Barcelona: Biblio 3W. Revista Bibliográfica de Geografía y Ciencias Sociales 1998, Disponible en: <http://www.ub.es/geocrit/b3w-129.htm>. ISBN 1138-9796.
- SANTOS, J. M. Características de Calidad de Productos de Software y su Evaluación: . Revista Normalización, 2008, ISBN 0138-8118.
- SANZ, L. F. Teoría de la medición. 1998, Disponible en:
<http://www.sc.ehu.es/jiwdocoj/remis/docs/teoriamedicion.html>.
- SCALONE, F. Estudio Comparativo de los Modelos y Estándares de Calidad del Software. Maestría en ingeniería de calidad, Universidad Tecnológica Nacional Facultad Regional Buenos Aires, 2006.
- SOCIETY, S. E. S. C. O. T. I. C. IEEE Computer Dictionary. 1991, ISBN 1559370793
- . IEEE Standard for a software quality metrics methodology. 1998,
- SOLÁNS, M. Métricas. 2004, nº Disponible en:
<http://profesores.sanvalero.net/~w0172/Apuntes%20Gesti%F3n%20Proyectos/Metricas/GP%20Metricas.ppt>.
- VEGA LEBRÚN CARLOS, R. P. L. S., GARCÍA SANTILLÁN ARTURO. Mejores Prácticas para el Establecimiento y Aseguramiento de la Calidad de Software. 2000.

BIBLIOGRAFÍA

- AÑASCO, J. C. Normas ISO. 2008, nº Disponible en:
<http://www.pilar.com.ar/industrias/temasgenerales/normas.htm> .
- ESPAÑOLA, R. A. *Calidad* Disponible en: <http://www.rae.es/rae.html>.
- FRANCISCO J. PINO, F. G., FRANCISCO RUIZ, MARIO PIATTIN. *Adaptación de las normas ISO/IEC 12207:2002 e ISO/IEC 15504:2003 para la evaluación de la madurez de procesos software en países en desarrollo* Disponible en: <http://ieeexplore.ieee.org/iel5/9907/34417/01642455.pdf> .
- INFORMÁTICA, D. D. C. D. C. Y. A. *Control de Calidad de los Sistemas*. 2000, Disponible en:
<http://sistemas.dgsca.unam.mx/publica/pdf/Control%20de%20Calidad.PDF> .
- ISO, I. *Norma Internacional ISO/IEC 12207*. 2003, ISBN 0-7381-0430-2, SS94597.
- JAVIER JESÚS GUTIÉRREZ, M. J. E., MANUEL MEJÍAS, JESÚS TORRES. *Estudio comparativo de propuestas para la generación de casos de prueba a partir de requisitos funcionales* Disponible en: <http://www.lsi.us.es/docs/informes/LSI-2005-01.pdf> .
- MARINESCU, F. *Consejos sobre pruebas de rendimientos y optimización* Disponible en:
<http://espanol.aurumsol.com/articulos/art4/art4-print.html> .
- MEYERS, G. J. *The Art of Software Testing*,. 1979.
- PRESSMAN, R. *Ingeniería de Software*. 2001. Un Enfoque Práctico.
- SOCIETY, S. E. S. C. O. T. I. C. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*. 2000, ISBN 0-7381-2519-9 SS94869.
- . *IEEE Recommended Practice for Software Design Descriptions*. 1998a, ISBN 0-7381-1456-1 SS94688.
- . *IEEE Recommended Practice for Software Requirements Specifications*. 1998b, ISBN 0-7381-0448-5, SS94654.
- . *IEEE Standard for Software Reviews*. 1998c, ISBN 1-55937-987-1.
- . *IEEE Standard for Software Verification and Validation*. 1998d, ISBN 0-7381-0449-3, SS94625.
- TERESITA ROJAS, L. E. M., MARÍA A. PÉREZ. *Indicadores Organizacionales para Comparación de Herramientas CASE en Venezuela* Caracas: Disponible en:
http://www.lisi.usb.ve/publicaciones/05%20herramientas/herramientas_06.pdf .
- TERUEL, E. U. P. D. *Calidad del Software* Disponible en:
http://eupt2.unizar.es/asignaturas/itig/ingenieria_de_software_II/docs/isw2_0708_05_tema5.pdf .

ANEXOS

ANEXO 1: PLAN DE ASEGURAMIENTO DE LA CALIDAD PARA EL PROYECTO BANCO

1 Introducción

1.1.1 Propósito

Este documento pretende entregar la pauta general del proceso que se planifica seguir para asegurar la calidad del producto a desarrollar por los miembros del Proyecto Banco; así como dejar constancia de todo el estilo de trabajo, herramientas, técnicas y metodologías a asumir, flujos de trabajo, artefactos, además de los niveles de responsabilidades dentro de cada rol en el proyecto. Se convierte en una guía de consulta obligatoria para todos los miembros del proyecto que responden por la calidad del producto. Finalmente se propone el uso de un conjunto de documentos que se incluyen como anexos de este trabajo.

1.1.2 Alcance

Este plan ha sido elaborado según la plantilla establecida por la DCS de la universidad, teniendo como punto de referencia el propuesto por el Proceso Unificado Racional (RUP) y pretendiendo llevar a la par conjuntamente las especificaciones requeridas por el estándar IEEE 730 – 1998; como aporte del trabajo se ha considerado otros aspectos de interés particular, vale señalar además, que ha sido adaptado a las condiciones exclusivas del Proyecto Banco. Su alcance radica en que es aplicable al flujo de trabajo planificado para el proyecto mencionado, pero a la vez, pretende entregar pautas generales como futuro modelo inicial para ser usado por cualquier otro proyecto software que desarrolle un producto similar.

1.1.3 Definiciones, Acrónimos y Abreviaturas

Término	Descripción
Auditoría	Evaluación independiente de un producto SW, proceso SW, o conjunto de procesos SW, para evaluar el cumplimiento de especificaciones estándares, acuerdos contractuales y otros criterios.
Revisión	Un proceso o reunión, durante el cual un producto SW se presenta al personal del proyecto, gerentes, usuarios, clientes, representantes de usuarios u otras partes interesadas para comentarios o aprobación.
RTF	Revisión Técnica Formal: Evaluación sistemática de un producto SW por un equipo calificado de personal que examina la adecuación de un producto SW a su uso previsto

	e identifica discrepancias respecto a especificaciones o estándares. También pueden proporcionar recomendaciones de alternativas y examinar varias alternativas.
Prueba	Examen visual de un producto SW para detectar e identificar anomalías de SW, incluyendo errores y desviaciones de estándares y especificaciones.
Lista de Chequeo	Lista de aspectos en forma de preguntas utilizados para revisar o inspeccionar temas de interés.
Métrica	Medidas que se le hacen al software para indicar su calidad, Ej.: productividad de las personas, estimación de tamaño o duración.
SW	Abreviatura de Software.
LCS	Lineamientos de Calidad del Software.
DCS	Dirección de calidad de software.
BPMN	Business Process Modeling Notation.
RUP	Proceso Unificado de Desarrollo del Software.
ISO	International Organization for Standardization.
NC	Normas Cubanas
BD	Base de Datos.

1.1.4 Referencias

En el plan no se hace referencia a ningún documento fuera del ámbito del proyecto.

1.1.5 Resumen

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

1.2 Objetivos de Calidad

Partiendo de los objetivos básicos del proyecto, se han identificado los siguientes objetivos de calidad:

- Garantizar un elevado nivel de idoneidad e interoperabilidad en el producto a desarrollar; asegurando, a través de las revisiones y pruebas, que se logre de una manera flexible y configurable todas las operaciones que se desarrollan en los diferentes bancos del país.
- Verificar que se adquieran altos índices de operabilidad e idoneidad en el sistema, partiendo de la premisa de que, mediante la aplicación de revisiones y pruebas, durante todo el ciclo de vida del software, sea posible alcanzar la parametrización de las funcionalidades, logrando flexibilidad y configurabilidad del sistema.
- Asegurar que el producto final refleje de forma adecuada, según las especificaciones de los clientes, firmes criterios de idoneidad e interoperabilidad y, haciendo uso de las revisiones y pruebas, lograr establecer la estandarización de la información que se maneja y registra entre los diferentes bancos del país.
- Garantizar que el sistema mantenga altos niveles de idoneidad y precisión, asegurando, a través de las revisiones y pruebas, lograr disminuir los tiempos de respuestas en los servicios prestados por los diferentes bancos.
- Velar por que se logre alcanzar satisfactoriamente los niveles de idoneidad deseados, desarrollando para ello revisiones y pruebas; y de esta forma hacer cumplir satisfactoriamente la estrategia de mejorar la centralización de la información, para la toma de decisiones por parte de los directivos del banco.
- Asegurar que el producto a desarrollar gane un alto grado de comprensibilidad y cognoscibilidad, según las especificaciones establecidas; aplicando para ello, revisiones y pruebas durante todo el ciclo de desarrollo del software y pretendiendo de esta forma, contar con una documentación bien detallada de todos los procesos del negocio bancario, además de la documentación propia del sistema.
- Garantizar que el sistema sustente, satisfactoriamente, los términos de seguridad (informática) que fueron establecidos, haciendo uso de revisiones y pruebas durante todo el ciclo de vida del software, para que de esta forma el sistema logre ejecutar todas las operaciones bancarias en dependencia del nivel de acceso especificado para cada usuario.

1.3 Gestión

1.31 Organización

Rol	Descripción	Conocimientos mínimos
-----	-------------	-----------------------

Responsable de calidad	Es una persona bien orientada, su objetivo principal se centra en asegurarse que la aplicación se ajusta a los requerimientos y mantenerse lo más lejano posible de errores. Provee una metodología para realizar las pruebas. Evalúa los resultados que se obtienen al realizar las pruebas de calidad.	Metodología RUP Notación BPMN Lenguaje UML Calidad del software Ingeniería de software
Diseñador de pruebas	Diseña los casos de pruebas. Valora y documenta el efecto de las pruebas realizadas al producto. Define las listas de chequeo.	Metodología RUP Notación BPMN Lenguaje UML Pruebas de software
Revisor técnico	Verifica que los artefactos generados concuerdan con las normas instituidas para su confección.	Metodología RUP Notación BPMN Lenguaje UML
Probador	Ejecuta las pruebas previamente diseñadas. Registra los resultados obtenidos.	Conocimientos del negocio. Habilidades básicas de computación.

Puntualizar que: Las entregas, devoluciones, exigencias y obligaciones serán dirigidas a la persona que en ese instante desempeñe dicha responsabilidad, llevándose un historial de todas las acciones.

1.3.2 Tareas y Responsabilidades

Tarea de Aseguramiento de calidad	Precondición	Poscondición	Responsable	Comentarios
Revisión del modelado de los procesos del negocio.	Que se hayan modelado y documentados todos los procesos del negocio.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad	Verificar el modelado de cada uno de los procesos identificados siguiendo las pautas establecidas por BPMN.
Revisión de la captura de requisitos.	Que se haya efectuado la captura de	Se ha documentado la revisión y en caso de encontrar No	Equipo de calidad.	Verificación del cumplimiento de la lista de

	requisitos.	Conformidades, se habrán solucionado.		chequeo elaborada.
Revisión de los casos de uso del sistema.	Que se haya desarrollado la especificación de casos de usos del sistema.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Verificación del cumplimiento de la lista de chequeo elaborada.
Revisión del diseño.	Luego de finalizar la captura de requisitos se haya terminado el diseño del módulo en cuestión.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Confirmación del diseño de los casos de uso por las pautas establecidas.
Revisión de la arquitectura	Que se haya documentado la arquitectura en el documento Descripción de la Arquitectura.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Verificación del cumplimiento de la lista de chequeo elaborada según las pautas establecidas por los arquitectos.
Revisión del diseño gráfico	Que se hayan establecido las pautas del diseño gráfico para el proyecto.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Confirmación de los modelos de diseños gráficos por las pautas predefinidas.
Revisión del diseño de la base de datos.	Que se haya obtenido un modelo del diseño de la base de datos a través de una herramienta de modelado y establecido un estándar para diseñar la BD.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Verificación de la base de datos por las pautas establecidas.
Revisión de la implementación.	Que se haya establecido el estándar de codificación para	Se ha documentado la revisión y en caso de encontrar No Conformidades, se	Equipo de calidad.	Hacer una revisión del código fuente con el objetivo de

	el proyecto.	habrán solucionado.		encontrar posibles errores de codificación según el estándar establecido.
Revisión de la documentación.	Que todos los documentos establecidos en el expediente de proyecto hayan sido confeccionados.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Verificación del cumplimiento de las listas de chequeo elaboradas.
Revisión de la gestión de la configuración.	Que se haya documentado la Gestión de Configuración, así como que se esté ejecutando.	Se ha documentado la revisión y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Asegurar que se mantiene la calidad durante la realización de los cambios. Los informes de estado proporcionan información sobre cada cambio a aquellos que tienen que estar informados.
Auditoría del diseño.	Que se haya efectuado la anterior revisión planificada para el diseño.	Se ha documentado la auditoría y en caso de encontrarse No Conformidades estas se habrán solucionado.	Equipo de calidad.	Validación de los procesos del diseño.
Auditoría de la arquitectura.	Que se haya efectuado la revisión planificada para la arquitectura.	Se ha documentado la auditoría y en caso de encontrarse No Conformidades estas se habrán solucionado.	Equipo de calidad.	Validación de la arquitectura definida.
Auditoría del diseño gráfico.	Que se haya efectuado la revisión del	Se ha documentado la revisión y en caso de encontrar No	Equipo de calidad.	Validación del diseño gráfico según las pautas

	diseño gráfico planificada.	Conformidades, se habrán solucionado.		establecidas.
Auditoría del diseño de la base de datos.	Que se haya efectuado la revisión del diseño de la base de datos planificada.	Se ha documentado la auditoría y en caso de encontrarse No Conformidades estas se habrán solucionado.	Equipo de calidad.	Validación del diseño de la base de datos según las pautas establecidas.
Auditoría a gestión de configuración.	Que se haya efectuado la revisión planificada para la Gestión de Configuración.	Se ha documentado la auditoría y en caso de encontrarse No Conformidades estas se habrán solucionado.	Equipo de calidad.	Validación del funcionamiento de la gestión de configuración.
Auditoría general al sistema de calidad.	Que se hayan efectuado las revisiones planificadas, documentándose en las Listas de chequeo y el Registro de No Conformidades correspondiente.	Se ha documentado la auditoría y en caso de encontrarse No Conformidades estas se habrán solucionado.	Auditor del DSC	Validación del adecuado funcionamiento del Sistema de Calidad.
Pruebas de Caja Negra a cada módulo concluido.	Que se hayan documentado los requisitos del software.	Se ha documentado los casos de prueba y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Comprobar la adecuada funcionalidad del software.
Pruebas de Integración del Sistema.	Que se encuentren en completo funcionamiento los módulos implicados.	Se ha documentado los casos de prueba y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Comprobar la correcta integración entre los módulos definidos.
Pruebas de Seguridad.	Que se hayan determinado los requisitos de	Se ha documentado los casos de prueba y en caso de	Equipo de calidad.	Comprobar que el producto final mantenga altos

	seguridad del software y que este se encuentre funcionando en su totalidad.	encontrar No Conformidades, se habrán solucionado.		niveles de fiabilidad.
Pruebas de Aceptación del Cliente	Que se haya cumplido con el ciclo de vida del software y que éste se encuentre en condiciones de ser entregado al cliente.	Se ha documentado los casos de prueba y en caso de encontrar No Conformidades, se habrán solucionado.	Equipo de calidad.	Pruebas a través de las cuales el cliente determina si acepta o no el producto entregado.

1.4 Documentación

A continuación se muestra una tabla listando los documentos necesarios durante la elaboración y aplicación del Plan:

Título	Autor	Ubicación
Documento Visión del Proyecto	Jefe Proyecto	Repositorio de información
Plan de Gestión de Configuración	Gestor de Configuración	Repositorio de información
Lista de Riesgos	Jefe Proyecto	Repositorio de información
Listas de Chequeos	Diseñador de Pruebas	Repositorio de información
Registro de No Conformidades	Equipo de calidad	Repositorio de información

1.5 Métricas

Tipo de Métrica	Nombre	Especificaciones
Métricas de calidad	Métricas de idoneidad.	Medir la característica de funcionalidad.
	Métricas de exactitud.	Medir la característica de funcionalidad.
	Métricas de interoperabilidad.	Medir la característica de funcionalidad.
	Métricas de comprensibilidad.	Medir la característica de usabilidad.
	Métricas de conformidad con la usabilidad.	Medir la característica de usabilidad.
Métricas	Eficacia en la eliminación de defectos.	Evaluar objetivamente los productos y los servicios del trabajo.

de procesos	Grado de efectividad en identificar las no-conformidades.	Evaluar objetivamente los productos y los servicios del trabajo.
	Por ciento de efectividad para resolver las no-conformidades.	Asegurar la comunicación y resolución de las no-conformidades.
	Capacidad del procedimiento especificado para evaluar los procesos.	Evaluar objetivamente los procesos.

1.6 Estándares y Guías

Se plantean el siguiente conjunto de estándares a ser aplicados durante las diferentes etapas de desarrollo del software; aunque siempre aclarando que el equipo de desarrollo puede llegar a un consenso y hacer modificaciones a esta propuesta.

Estándar	Etapas de aplicación	Comentarios
IEEE 830 ISO/IEC 12 207 ISO/IEC 9126 – 1	Captura de Requisitos	La ISO 12 207 abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso. La ISO/IEC 9126-1 para definir objetivos de calidad.
IEEE 1016	Diseño Gráfico	Aborda aspectos referentes al diseño de entidades, diseño de vistas, a la descripción del diseño del software. Presenta una tabla de anexo donde se propone un formato como una posible forma de organización en la descripción del diseño.
IEEE 1471	Arquitectura	Establece un marco conceptual para pensar la arquitectura, la codificación de las mejores prácticas actuales y la ingeniería de software.
Codificación en J2EE	Implementación	Estándar de código que deberán definir los programadores del proyecto.
ISO/IEC 9126 – 2 ISO/IEC 9126 – 3 ISO/IEC 9126 – 4	Pruebas y Métricas	Parte 2: Métricas externas. Parte 3: Métricas internas. Parte 4: Métricas de la calidad en uso.
ISO 12 119	Documentación	Norma cubana aplicable a los paquetes de software.
IEEE 828	Durante todo el ciclo de vida del desarrollo del SW	Guía para la elaboración de un Plan de Gestión de Configuración del Software.
IEEE 1028	Durante todo el ciclo de vida	Requerimientos para Procesos de Revisiones

	del desarrollo del SW	y Auditorías.
RUP	Durante todo el ciclo de vida del desarrollo del SW	Guía para la elaboración del Plan de Aseguramiento de la Calidad.
IEEE 730	Durante todo el ciclo de vida del desarrollo del SW	Guía para la elaboración del Plan de Aseguramiento de la Calidad.
LCS	Durante todo el proceso de desarrollo del SW	Políticas emitidas por la DCS de la UCI.
ISO/IEC 12 207	Durante todo el proceso de desarrollo del SW	Guía para la confección de metodologías del ciclo de vida.

1.7 Plan de Revisiones y Auditorías

1.7.1 Tareas generales de Revisiones y Auditorías

En la siguiente tabla se especifican todas las tareas de revisiones y auditorías que han sido previstas para intentar asegurar la calidad del software durante su ciclo de vida, dando una breve descripción y mencionando además, los artefactos involucrados.

Tarea	Descripción	Artefacto involucrado
Revisión de los procesos del negocio.	Aplicar la lista de chequeo para el modelado del negocio.	<ul style="list-style-type: none"> - Plantilla DCS Diagrama de Proceso-Nombre del Proceso. - Plantilla DCS Modelo del Negocio. - Plantilla DCS Modelo del Dominio. - Informe de No Conformidades correspondiente.
Revisión de la captura de requisitos.	Aplicar la Lista de chequeo para la especificación de requisitos.	<ul style="list-style-type: none"> - Plantilla DCS Especificación de requisitos. - Plantilla DCS Plan de gestión de requisitos. - Informe de No Conformidades correspondiente.
Revisión de la arquitectura.	Aplicar lista de chequeo para la arquitectura.	<ul style="list-style-type: none"> - Plantilla DCS Arquitectura de Información. - Plantilla DCS Informe del Levantamiento de Información para la Arquitectura de Información. - Plantilla DCS Documento de Arquitectura de Software. - Informe de No Conformidades correspondiente.
Revisión de los casos de uso del sistema.	Aplicar lista de chequeo para especificación de casos de uso.	<ul style="list-style-type: none"> - Plantilla DSC Modelo de Casos de uso del sistema. - Informe de No Conformidades

		correspondiente.
Revisión del diseño.	Aplicar lista de chequeo para el diseño.	-Plantilla DCS Modelo de Diseño. - Informe de No Conformidades correspondiente.
Revisión del diseño gráfico.	Aplicar lista de chequeo para el diseño gráfico.	-Documento de pautas establecidas para el diseño gráfico. - Informe de No Conformidades correspondiente.
Revisión de la base de datos.	Aplicar lista de chequeo para chequear el diseño de la base de datos.	- Script de la base de datos. - Modelo de datos. - Informe de No Conformidades correspondiente.
Revisión de la implementación.	Aplicar lista de chequeo para la implementación.	- Código fuente. - Informe de No Conformidades correspondiente.
	Pruebas de Caja Blanca a cada caso de uso implementado.	- Plantilla DCS Diseño casos de prueba. - Plantilla DCS Plan de pruebas. -Informe de No Conformidades correspondiente.
	Pruebas de Caja Negra a cada módulo concluido.	
	Pruebas de integración del sistema.	
	Pruebas de seguridad.	
Revisión de la documentación.	Aplicar la lista de chequeo para la revisión de la documentación de las plantillas definidas en el expediente de proyecto.	- Manual de usuario. - Plantilla DCS Glosario de términos. -Todas las plantillas del expediente de proyecto. -Informe de No Conformidades correspondiente.
Revisión de la gestión de configuración.	Aplicar lista de chequeo para la gestión de configuración.	- Plantilla DCS Plan Gestión de Configuración. - Plantilla DCS Pedido de cambio. - Plantilla DCS Solicitud de cambio. - Informe de No Conformidades correspondiente.
Auditoría del diseño.	Aplicar la lista de chequeo desarrollada para esta actividad. Aplicar la Guía	- Plantilla DCS Modelo de Diseño. - Informe de No Conformidades. - Plantilla Reunión de Apertura. - Plantilla de Acciones Correctivas o de

	Exploratoria establecida por la DSC para la realización de las auditorías.	mejoras. - Plantilla Reunión de Cierre. - Plantilla Informe Final de Auditoría.
Auditoría de la arquitectura.	Aplicar la lista de chequeo desarrollada para esta actividad. Aplicar la Guía Exploratoria establecida por la DSC para la realización de las auditorías.	- Plantilla DCS Arquitectura de Información. - Plantilla DCS Informe del Levantamiento de Información para la Arquitectura de Información. - Plantilla DCS Documento de Arquitectura de Software. - Informe de No Conformidades. - Plantilla Reunión de Apertura. - Plantilla de Acciones Correctivas o de mejoras. - Plantilla Reunión de Cierre. - Plantilla Informe Final de Auditoría.
Auditoría del diseño gráfico.	Aplicar la lista de chequeo desarrollada para esta actividad. Aplicar la Guía Exploratoria establecida por la DSC para la realización de las auditorías.	- Documento de pautas establecidas para el diseño gráfico. - Informe de No Conformidades. - Plantilla Reunión de Apertura. - Plantilla de Acciones Correctivas o de mejoras. - Plantilla Reunión de Cierre. - Plantilla Informe Final de Auditoría.
Auditoría de las bases de datos.	Aplicar la lista de chequeo desarrollada para esta actividad. Aplicar la Guía Exploratoria establecida por la DSC para la realización de las auditorías.	- Script de la base de datos. - Modelo de datos. - Informe de No Conformidades. - Plantilla Reunión de Apertura. - Plantilla de Acciones Correctivas o de mejoras. - Plantilla Reunión de Cierre. - Plantilla Informe Final de Auditoría.
Auditoría a la gestión de configuración.	Aplicar la lista de chequeo desarrollada para esta actividad. Aplicar la Guía Exploratoria establecida por la DSC para la realización de las auditorías.	- Plantilla DCS Plan Gestión de Configuración. - Plantilla DCS Pedido de cambio. - Plantilla DCS Solicitud de cambio. - Informe de No Conformidades. - Plantilla Reunión de Apertura. - Plantilla de Acciones Correctivas o de mejoras. - Plantilla Reunión de Cierre. - Plantilla Informe Final de Auditoría.

Auditoría general al sistema de calidad.	Aplicar la Guía Exploratoria establecida por la DSC para la realización de las auditorías.	<ul style="list-style-type: none"> - Listas de chequeo generadas para cada revisión. - Informe de No Conformidades correspondiente a cada revisión efectuada. - Plantillas del Expediente de Proyecto. - Informe de No Conformidades correspondiente a la auditoría. - Plantilla Reunión de Apertura. - Plantilla de Acciones Correctivas o de mejoras. - Plantilla Reunión de Cierre. - Plantilla Informe Final de Auditoría.
--	--	--

1.7.2 Cronograma

Las actividades a desarrollar se llevarán a cabo bajo las tareas generales antes expuestas, con las herramientas, técnicas y metodologías ya definidas, y responsabilizadas a las personas que en ese instante cumplen con el rol designado.

Revisiones y Auditorías	Fecha según el avance del proyecto	
	Inicio	Duración
Revisión de los procesos del negocio.	1ra Revisión: 15 días después de haber iniciado el modelado del negocio. 2da Revisión: 7 días antes de concluir el modelado del negocio.	2 semanas
Revisión de la captura de requisitos.	1ra Revisión: 8 días después de haber iniciado la captura de requisitos. 2da Revisión: 7 días antes de concluir la captura de requisitos.	1 semana
Revisión de los casos de usos del sistema.	1ra Revisión: 14 días después de haber iniciado el diseño casos de usos del sistema. 2da Revisión: 7 días antes de concluir el diseño casos de usos del sistema.	2 semanas
Revisión de la arquitectura.	1ra Revisión: 10 días después de haber presentado la propuesta inicial de la arquitectura del sistema. 2da Revisión: Al concluir la propuesta final de la arquitectura.	1 semana

Revisión del diseño gráfico.	1ra Revisión: 7 días después de haber iniciado el diseño gráfico. 2da Revisión: 3 días antes de concluir el diseño gráfico.	1 semana
Revisión del diseño.	1ra Revisión: 15 días después de haber iniciado el diseño. 2da Revisión: 7 días antes de concluir el diseño.	2 semanas
Revisión de la base de datos.	1ra Revisión: 12 días después de haber iniciado el diseño de la base de datos. 2da Revisión: 7 días antes de concluir el diseño de la base de datos.	1 semana
Revisión de la implementación.	1ra Revisión: 20 días después de haber iniciado la implementación del módulo. 2da Revisión: 7 días antes de concluir la implementación del módulo.	2 semanas
Revisión de la documentación.	1ra Revisión: Al concluir la implementación del módulo desarrollado. 2da Revisión: 7 días después de haber concluido la primera revisión a la documentación.	1 semana
Revisión de la gestión de configuración.	1ra Revisión: 8 días después de haber iniciado la captura de requisitos. 2da Revisión: 7 días antes de concluir la captura de requisitos.	1 semana
Auditoría general a los procesos.	3 días después de haber concluido la revisión a la documentación para el módulo desarrollado.	4 semanas
Auditoría general al sistema de calidad.	7 días después de concluida la Auditoría general a los procesos para el módulo desarrollado.	2 semanas

1.7.3 Organización y Responsabilidades

Revisiones y Auditorías	Rol de calidad
Revisión de los procesos del negocio.	- Revisor Técnico
Revisión de la captura de requisitos.	- Revisor Técnico
Revisión de la arquitectura.	- Revisor Técnico
Revisión de los casos de usos del sistema.	- Revisor Técnico
Revisión del diseño.	- Revisor Técnico
Revisión del diseño gráfico.	- Revisor Técnico
Revisión de la implementación.	- Revisor Técnico, Diseñador de Prueba, Probador

Revisión de la documentación.	- Revisor Técnico
Auditoría del diseño.	- Revisor Técnico
Auditoría de la arquitectura.	- Revisor Técnico
Auditoría del diseño gráfico.	- Revisor Técnico
Auditoría de las bases de datos.	- Revisor Técnico
Auditoría a la Gestión de Configuración.	- Revisor Técnico
Auditoría general al Sistema de Calidad.	- Auditor de la DCS

1.7.4 Resolución de problemas y actividades de corrección

Luego de cada proceso de auditoría o revisión, se deberán registrar los errores o problemas encontrados tanto en la Lista de Chequeo como en la Plantilla de No Conformidades correspondiente. Dichos documentos se harán llegar al responsable de atenuar o eliminar los problemas, una copia se le pasará al líder del proyecto y otra al historial manejado dentro del grupo de Calidad. En el Anexo 7 se propone una plantilla para elaborar la Lista de Chequeo de forma tal que se refiera a aspectos de interés según el artefacto a evaluar, con el fin de encontrar errores que luego serán documentados de forma más específica en la Plantilla de No Conformidades.

1.7.5 Herramientas, Técnicas y Metodologías

Para el desarrollo de las actividades de revisiones y auditorías protagonizadas por el equipo de calidad se definieron los siguientes patrones:

- Aplicación de las normas internacionales IEEE, ISO, NC.
- Manejo de los conceptos de RUP y utilización de varias plantillas.
- Aplicación de los LCS establecidos por la dirección de calidad de la universidad.
- Aplicación de las listas de chequeo creadas a partir de las normas internacionales.

En el caso de esta última herramienta definida, mencionar que se presentan dos listas de chequeo las cuales fueron desarrolladas específicamente para ser usadas en el proyecto Banco, de forma más detallada la propuesta consiste en:

Aplicación de la Lista de chequeo para el modelo del negocio; incluye 5 criterios diferentes a evaluar: Uno en cuanto al uso adecuado de estereotipos mediante el modelado a través de la notación BPMN; otro en cuanto a la definición de estados de los artefactos generados, según dicho lenguaje; el próximo incluye aspectos para la correcta documentación de la Plantilla de Descripción de los Procesos del Negocio; el siguiente se refiere al modelado de los diagramas de los procesos definidos; y finalmente

el último criterio consiste en la evaluación adecuada de la aplicación de conceptos vinculados al ambiente del negocio.

Aplicación de la Lista de chequeo para la especificación de requisitos, en la misma se tienen en cuenta diferentes aspectos a chequear según un conjunto de criterios establecidos para evaluar cada uno de los requisitos que fueron identificados por los analistas del proyecto (claridad, completitud, consistencia, confiabilidad, etc.), además de incluye una serie de criterios a tener en cuenta para la correcta documentación de las plantillas que han de ser generadas durante esta etapa del proyecto, las mismas son: Plan de Gestión de Requisitos, Especificación de Requisitos, Modelo de Casos de Usos del Sistema.

Propuesta de pasos a seguir para el proceso de revisión

- 1ro: Fijar un cronograma.
- 2do: Delimitar el centro de atención de la revisión.
- 3ro: Desarrollar una lista de chequeo para la revisión planificada.
- 4to: Buscar no conformidades y documentarlas.
- 5to: Proponer soluciones (en caso de que le sea posible al revisor).
- 6to: Aplicar Métricas.

Propuesta de pasos a seguir para el proceso de auditoría

- 1ro: Planificación.
- 2do: Efectuar la investigación pertinente.
- 3ro: Desarrollar una lista de chequeo para la auditoría planificada.
- 4to: Analizar datos obtenidos.
- 5to: Proponer solución a los problemas encontrados y posibles mejoras.
- 6to: Elaborar y presentar un informe de resultados.

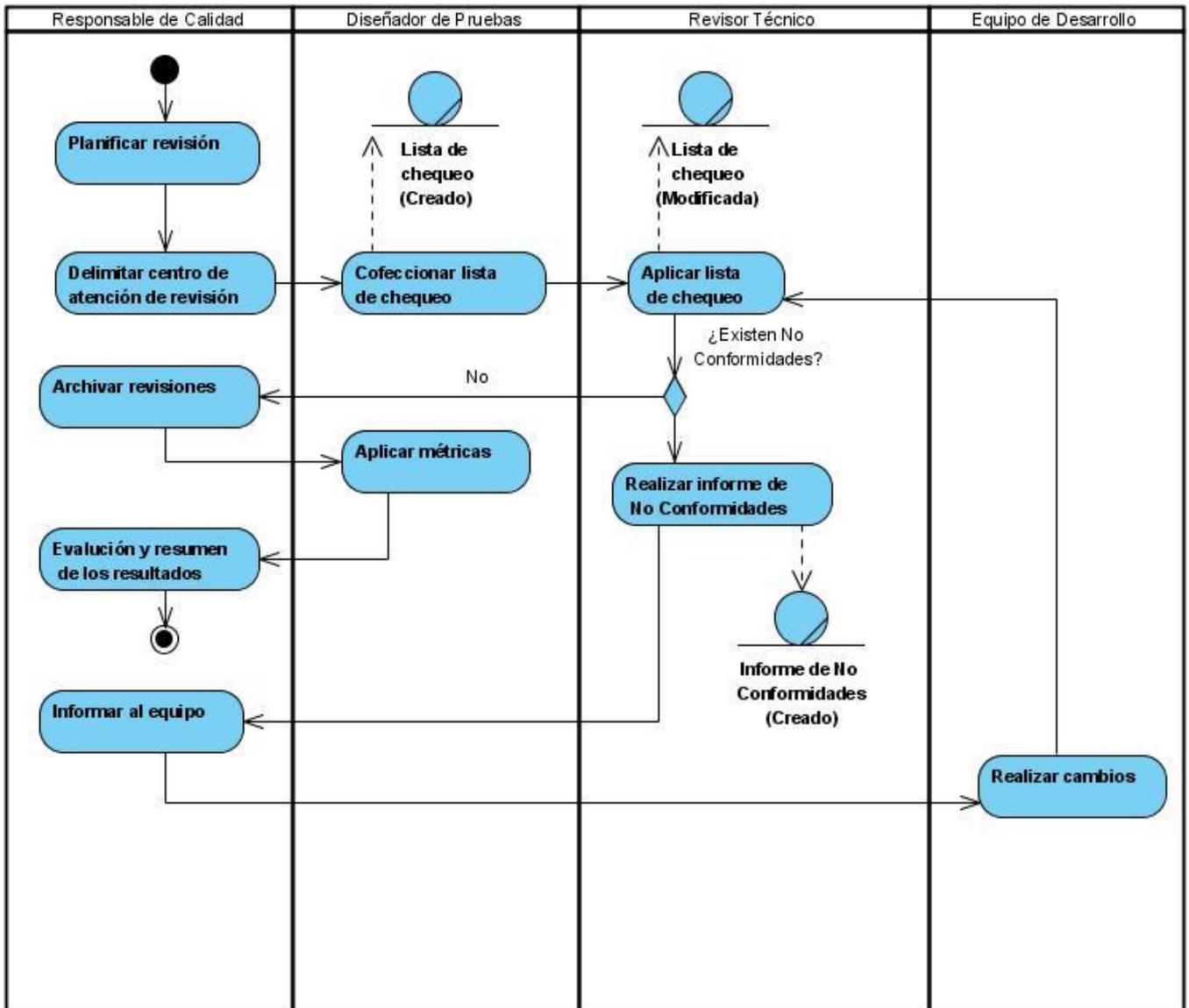
1.8 Pruebas y Evaluación

Para la realización de las pruebas se propone realizar un Plan de Pruebas. Este documento debe reflejar cómo el equipo de calidad ejecutará todas las pruebas de forma específica según las condiciones identificadas. En el Anexo 8 se ha incorporado una plantilla correspondiente a un Plan de Pruebas, la cual se propone como guía para la ejecución de los diferentes tipos de pruebas que se establecieron cuando se definieron las tareas a realizar por el equipo de calidad. Fue elaborada a partir de la plantilla especificada por RUP y además, revisada y aprobada por el departamento de calidad de

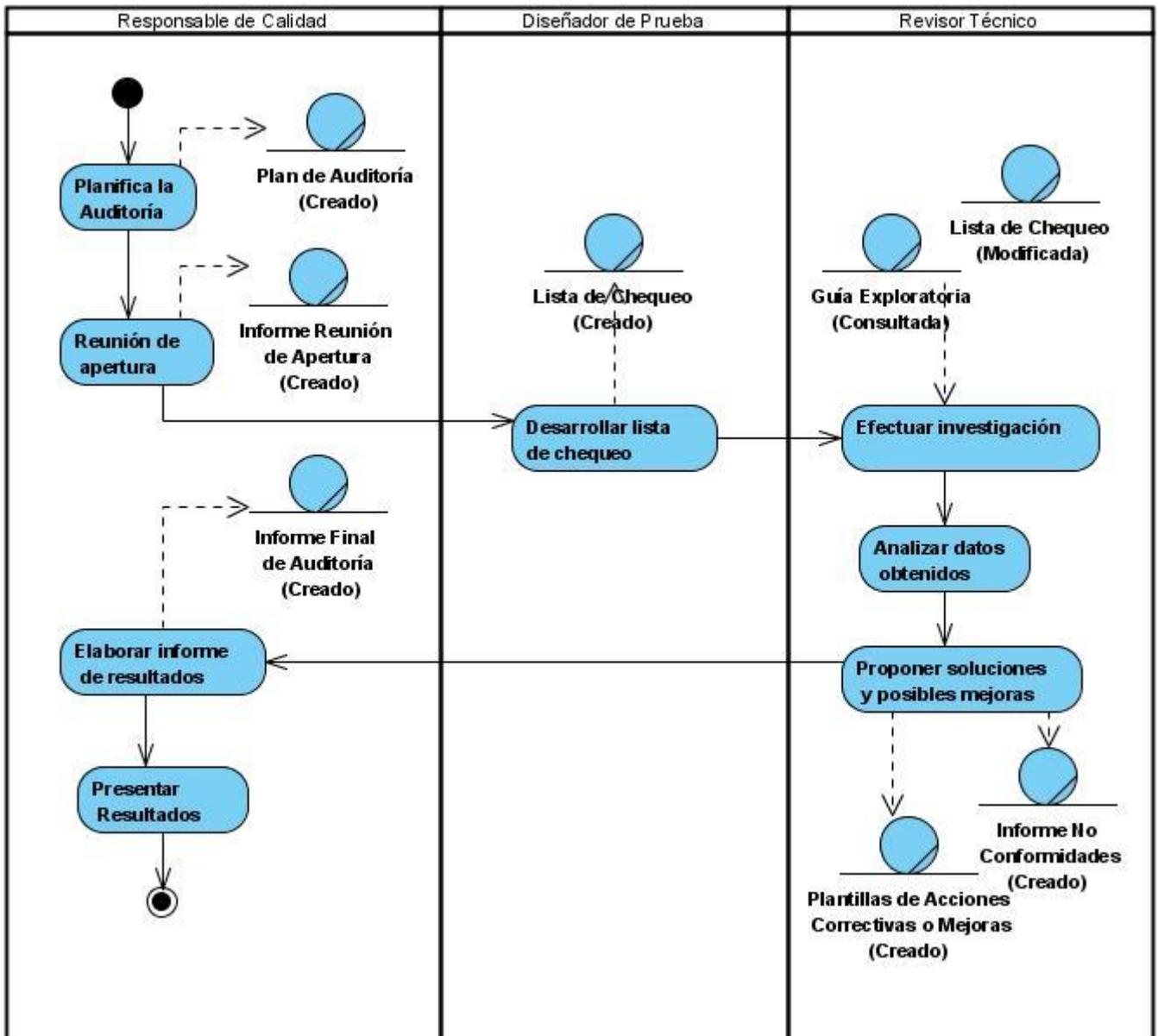
la universidad; pero ha sido adaptada considerando las particularidades de interés para el proyecto, por tal razón, su uso será exclusivo del Proyecto Banco.

1.9 Flujo de trabajo

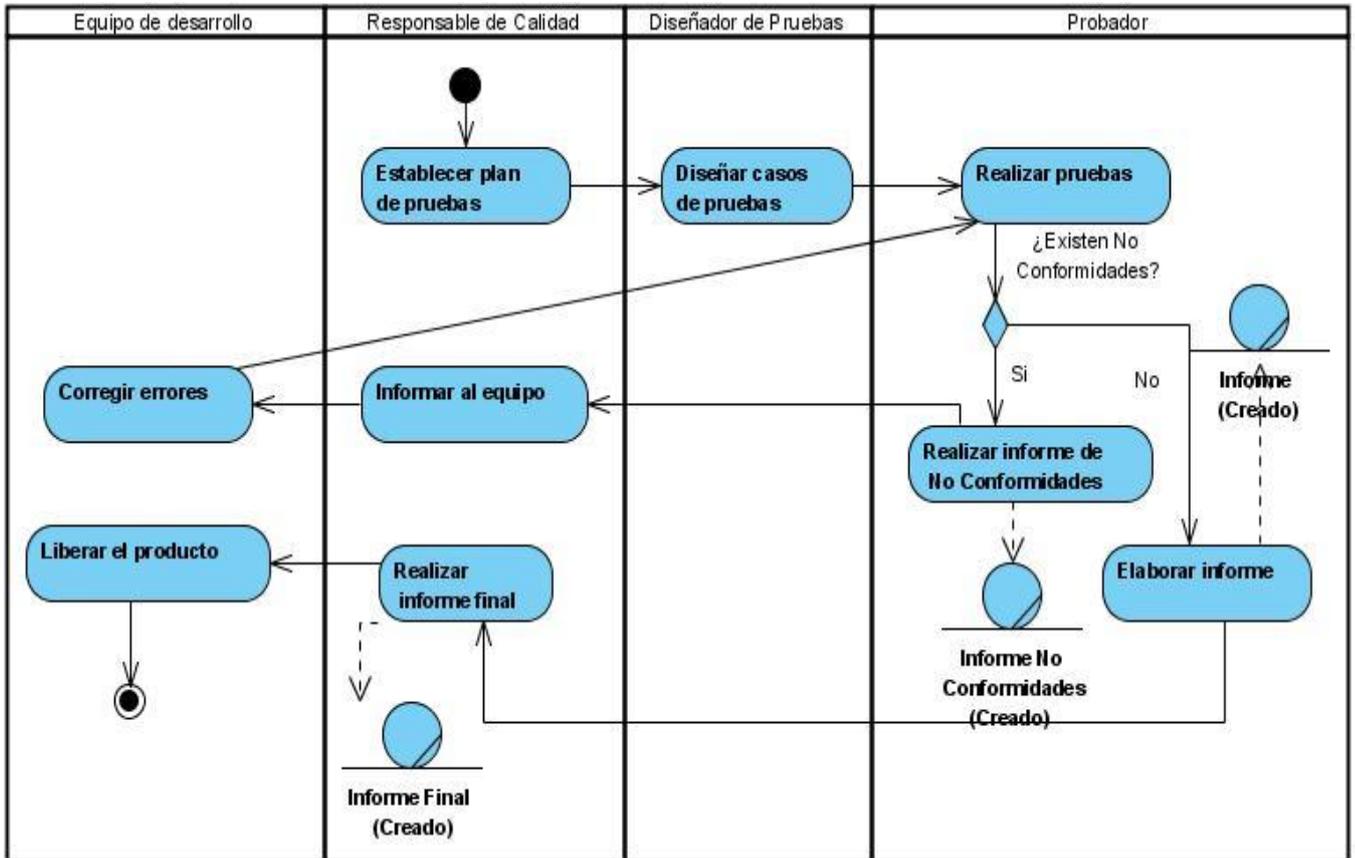
Flujo de trabajo para las actividades de revisión:



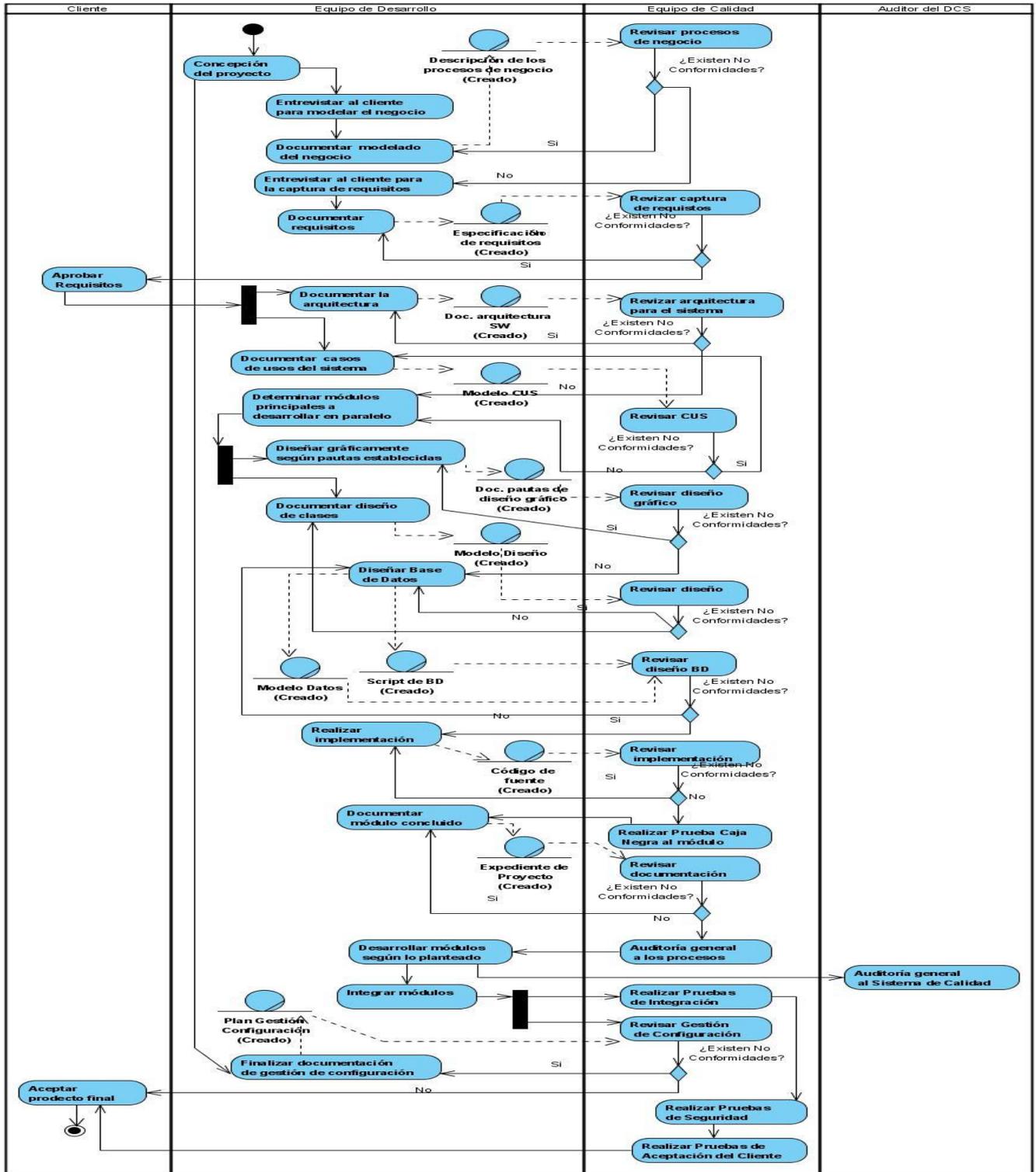
Flujo de trabajo para las actividades de auditoría:



Flujo de trabajo para las actividades de prueba:



Flujo de trabajo general para las actividades de aseguramiento de la calidad como complemento de las actividades del equipo de desarrollo del proyecto



1.10 Herramientas, Técnicas y Metodologías

De forma general, a continuación se propone una selección de algunos de estos elementos de apoyos para ser usados durante la realización de diferentes secciones del plan:

- Para el caso de las revisiones del código fuente se propone auxiliarse mediante el uso de herramientas, tales como: Depuradores y Perfiladores. Siempre aclarando que se tienen que cumplir las políticas establecidas en el estándar de codificación.
- De forma general se propone el uso de Listas de Chequeo para ser usadas durante las actividades de revisiones y auditorías.
- Se utilizará la Metodología RUP como apoyo para el establecimiento de la estrategia de prueba, el plan de prueba, el diseño de los casos de prueba.
- Generación de casos de prueba a partir de casos de uso.
- Plantillas del expediente de proyecto establecidas por la DCS para registrar la documentación del proyecto.
- Uso de modelos y estándares de la IEEE, NC, ISO.
- Regulaciones y normas establecidas por la dirección de informatización, dirección de diseño y la dirección técnica de la IP. Aplicación de los LCS establecidos por la dirección de calidad de la universidad.
- Uso del Trac para la gestión de los registros de calidad.

1.11 Resolución de Problemas y Acción Correctiva

Procedimiento a seguir para la resolución de problemas y acción correctiva

1ero: Establecer el equipo.

2do: Describir el problema.

3ro: Proponer la acción correctiva.

4to: Informar al equipo.

El procedimiento que se propone en esta sección pretende garantizar el establecimiento de acciones para eliminar las causas de las no conformidades (incumplimiento de requisitos), además de evaluar la necesidad de adoptar acciones para asegurar la imposibilidad de que las no conformidades vuelvan a ocurrir. No se admite bajo ningún concepto que algún problema se quede sin solución, para ello en el Anexo 6 se adjunta una Plantilla de No Conformidades en la cual los miembros del equipo de calidad reflejarán de forma detallada los problemas encontrados, para que luego, al equipo de desarrolladores, le sea más cómodo darle una solución factible y de forma inmediata.

1.12 Registros de Calidad

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

Registros que se guardarán:

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

1.13 Entrenamiento

Seguidamente se proponen un conjunto de actividades para capacitar al equipo de calidad las cuales deberán ser cumplidas de forma inmediata:

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

ANEXO 2: Encuesta

Encuesta relaciona con el aseguramiento de la calidad en los proyectos de la UCI.

Marque con una X según corresponda.

#	Preguntas	S	N	NS
1	¿Sabe lo que es aseguramiento de la calidad?			
2	¿Conoce lo que es un Plan de Aseguramiento de la Calidad?			
3	¿En su proyecto se tiene elaborado un Plan de Aseguramiento de la Calidad?			
4	¿En caso de tenerlo, se rigen por él?			

Criterios: **S:** sí, **N:** no, **NS:** no lo suficiente.

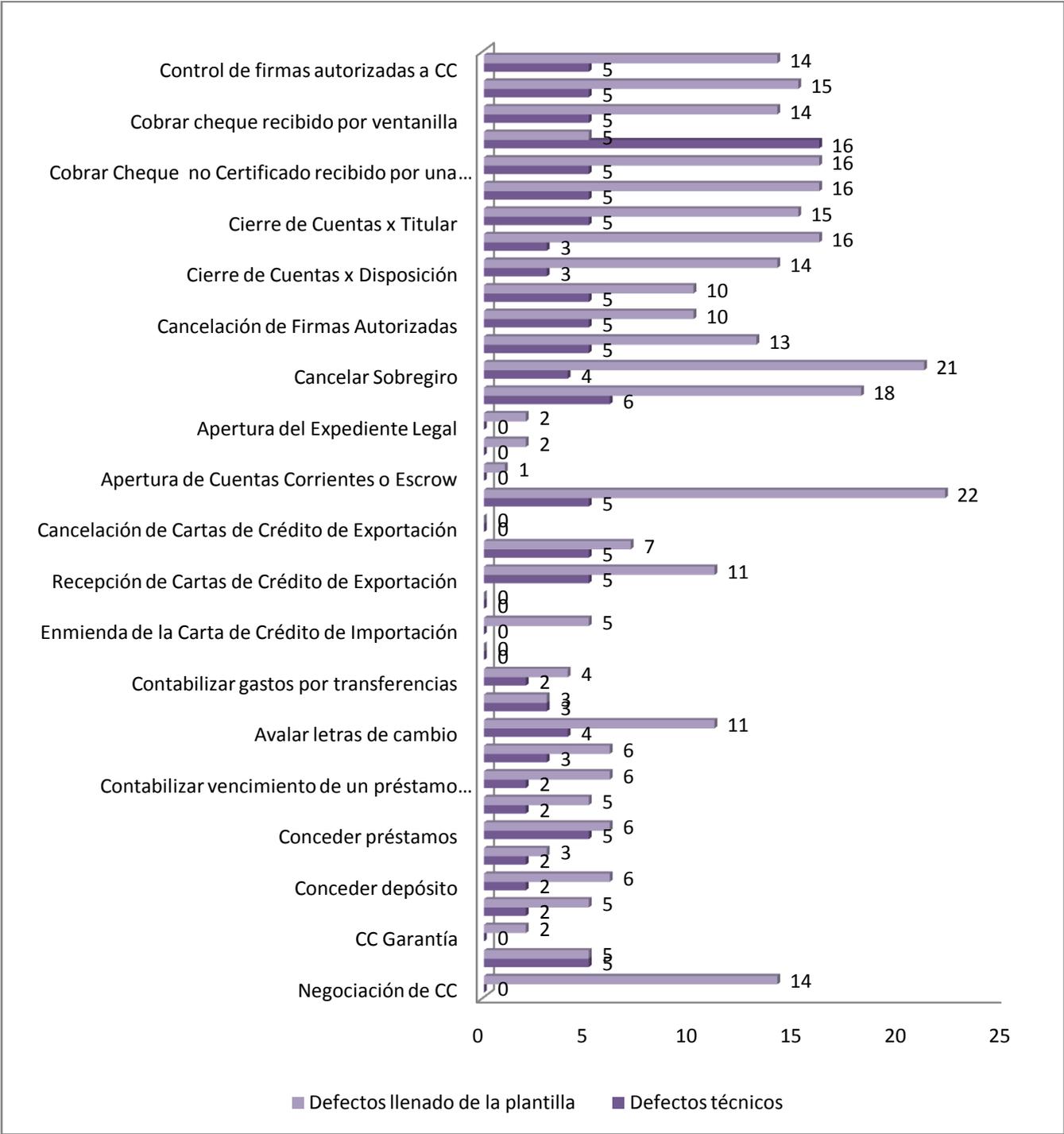
ANEXO 3: Datos de la medición del porcentaje de adecuación de un proceso modelado con los criterios evaluados

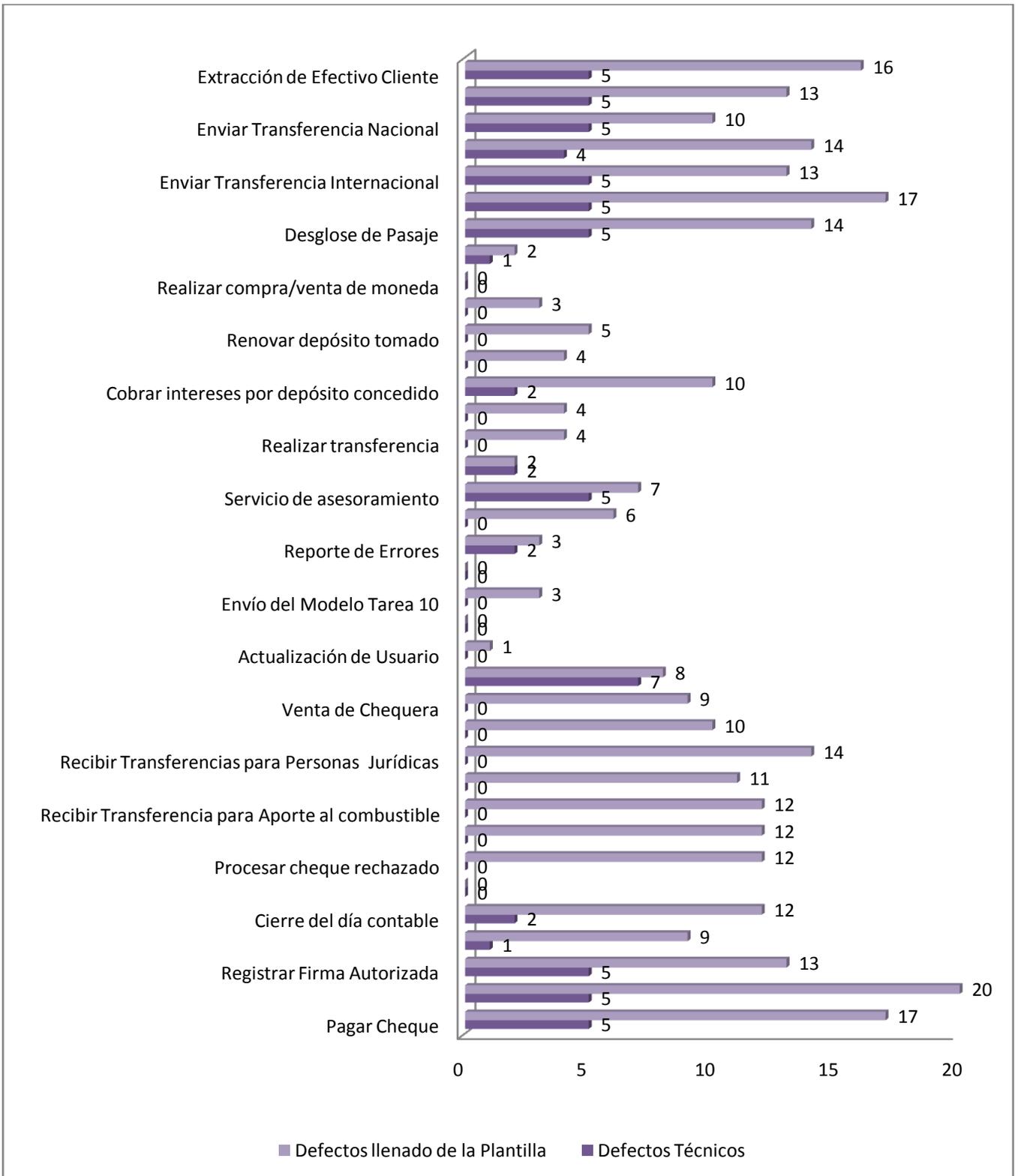
<i>Nombre del proceso revisado</i>	<i>Adecuación de los criterios</i>	<i>% de adecuación</i>
Negociación de CC	43400/530	81.88
Negociación de CC Importación	47000/520	90.38
CC Garantía	51300/520	98.65
Realizar informe de liquidez	32500/325	100
Confeccionar posición del banco por cuenta	37700/385	97.92
Cargar tipo de cambio	35900/380	94.97
Realizar compra/venta de moneda	35500/355	100
Realizar transferencia	37800/395	95.69
Conceder depósito	37700/410	91.95
Renovar depósito concedido	36000/385	93.5
Cobrar intereses por depósito concedido	33800/385	87.79
Tomar depósito	38000/400	95
Renovar depósito tomado	35500/380	93.42

Chequear préstamos/depositos	33000/345	95.65
Conceder préstamos	35800/390	91.79
Cobrar por préstamo concedido	37100/395	93.92
Contabilizar vencimiento de un préstamo recibido/concedido	31300/335	93.43
Recibir préstamos	33700/375	89.86
Renovar préstamo recibido/concedido	34000/350	97.14
Avalar letras de cambio	37600/395	95.18
Conceder financiamientos contra cuenta única	47100/495	95.15
Contabilizar gastos por transferencias	37100/385	96.36
Apertura de la Carta de Crédito de Importación	51000/530	96.22
Enmienda de la Carta de Crédito de Importación	36500/390	93.58
Vencimiento de la Carta de Crédito de Importación	34000/340	100
Recepción de Cartas de Crédito de Exportación	45800/520	88.07
Enmiendas de Cartas de Crédito de Exportación	46800/520	90
Cancelación de Cartas de Crédito de Exportación	36000/360	100
Ampliación de la Licencia	40600/490	82.85
Apertura de Cuentas Corrientes o Escrow	33400/335	99.7
Apertura de Cuentas Ingreso o Financiamiento	33300/335	99.4
Apertura del Expediente Legal	31800/320	99.37
Autorizar Sobregiro	46100/530	86.98
Cancelar Sobregiro	45400/530	85.66
Bloquear Cheque	47600/530	89.81
Cancelación de Firmas Autorizadas	48500/530	91.5
Certificar Cheque	48500/530	91.5
Cierre de Cuentas x Disposición	38000/530	71.69
Cierre de Cuentas x Mandato	46700/530	88.11
Cierre de Cuentas x Titular	47000/530	88.67
Cobrar Cheque Certificado recibido por una gerencia	46700/530	88.11
Cobrar Cheque no Certificado recibido por una gerencia	46700/530	88.11
Cobrar Cheque para aporte al combustible	46700/530	88.11
Cobrar cheque recibido por ventanilla	47300/530	89.24
Control de firmas autorizadas	47000/530	88.67
Control de firmas autorizadas a CC	47300/530	89.24
Desglose de Pasaje	47300/530	89.24
Encaje Legal	46400/530	87.54
Enviar Transferencia Internacional	38900/440	81.88
Enviar Transferencia Internacional contra CU	38900/440	95.65
Enviar Transferencia Nacional	39500/440	89.77
Enviar Transferencia Nacional contra CU	38900/440	88.4

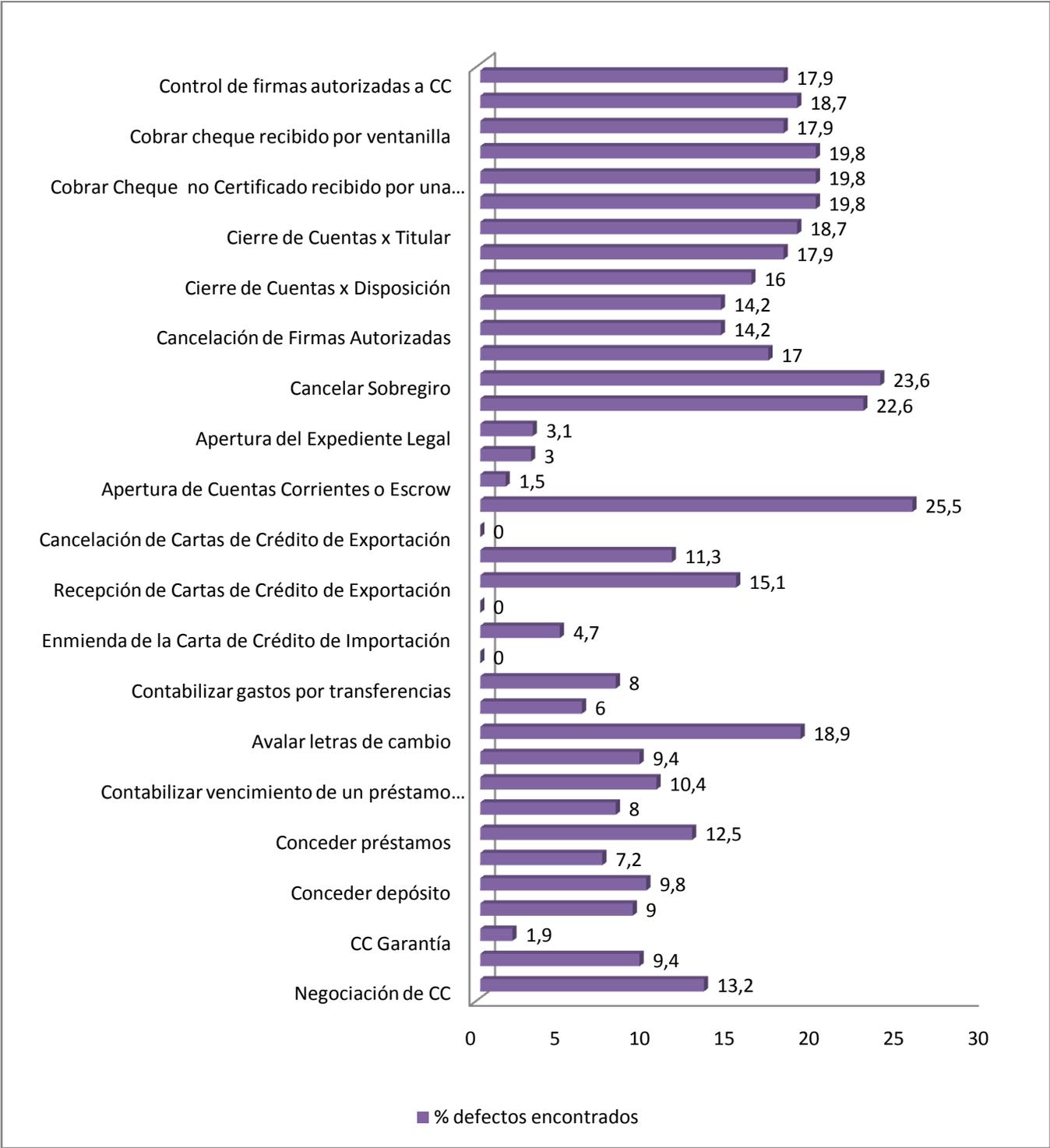
Extracción de Efectivo Cliente	46700/530	88.11
Pagar Cheque	46400/530	87.54
Procesar Cheque rechazado	34500/405	85.18
Recibir Transferencia de Banco a Banco	41300/465	88.81
Recibir Transferencia para Aporte al combustible	41300/465	88.81
Recibir Transferencias para Personas Naturales	41300/470	87.87
Recibir Transferencias para el Recargo Comercial 2%	45700/530	86.22
Recibir Transferencias para Personas Jurídicas	33000/400	82
Registrar Firma Autorizada	47600/530	89.81
Traspaso	36500/415	87.95
Venta de Chequera	34500/390	88.46
Inicio del día contable	34000/375	90.66
Cierre del día contable	30200/375	80.53
Contabilización	53000/530	100
Actualización de Clasificadores	24700/320	77.18
Actualización de Usuario	51700/520	99.42
Análisis del Balance General	53000/530	100
Envío del Modelo Tarea 10	31200/325	96
Salvas del DEF_DAFE	53000/530	100
Reporte de Errores	49700/520	95.57
Saneamiento de cuentas	8700/425	20.47
Seguimiento y análisis de tablas	23700/255	92.94
Servicio de asesoramiento	28400/340	83.53
Confirmar operación	33100/335	98.8

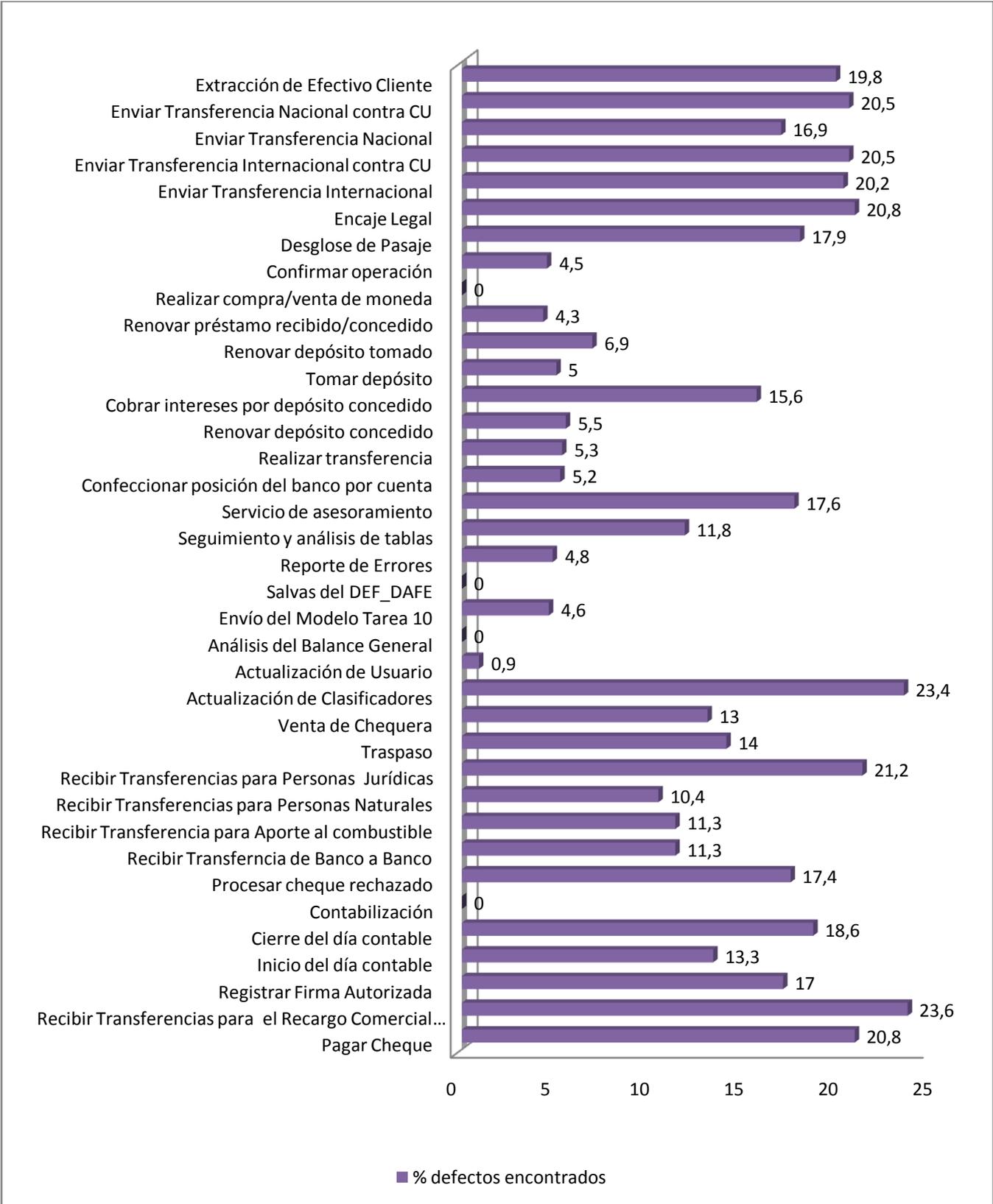
ANEXO 4: Cantidad de defectos inyectados de cada tipo por cada proceso modelado





ANEXO 5: Por ciento de defectos obtenidos por cada proceso modelado





ANEXO 6: Plantilla de No Conformidades



No Conformidades

Nombre del proyecto

Versión x.x

Plantilla de No Conformidades

Elemento a probar <Nombre del artefacto a probar>

Versión x.x

Control de versiones:

Fecha	Versión	Descripción	Autor
<00/00/0000>	<1.0>	<Descripción>	<Autor>

Descripción General

[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]

Elementos probados

[Descripción general o lista de los Elementos Probados, y otros aspectos importantes a tener en cuenta a la hora analizar las No Conformidades Detectadas.]

Elementos no probados y causas

[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]

Etapa de detección: *[Etapa en el que se encuentra el defecto]*

➤ Registro de defectos y dificultades detectadas

Elemento	No	No conformidad	Clasificación	Estado NC	Respuesta del Equipo Desarrollo
<Aplicación, Documento, Artefacto determinado >	< 1>	<Descripción de la No Conformidad>	< Significativa, No Significativa o Recomendación >	[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.] RA: Resuelta, PD: Pendiente NP: No Procede	[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]

[En la columna **Clasificación** se ubica el tipo de NC dado el grado de importancia: Significativa, No Significativa o Recomendación]

2. Anexos

Anexo <1>

ANEXO 7: Plantilla general para la elaboración de las Listas de Chequeo

Lista de Chequeo General

<Nombre del Proyecto >

<Responsable>

Versión <0.0>

Tabla de Contenidos:

LISTA DE CHEQUEO GENERAL125

1.INTRODUCCIÓN.....126

- 1.1. Propósito y objetivos.....126
- 1.2. Alcance.....126
- 1.3. Definiciones, Acrónimos y Abreviaturas.....126
- 1.4. Referencias.....127
- 1.5. Resumen.....127

2.LISTA DE CHEQUEO GENERAL.....127

- 2.1. Chequeo en cuanto al uso adecuado de conceptos.....127
- 2.2. Chequeo en cuanto a diversos temas de interés.....127
- 2.3. Chequeo en cuanto a la correcta redacción de la plantilla por revisar.....128

1. Introducción

[

Se da una breve panorámica de lo que será tratado en la lista de chequeo.

]

1.1 Propósito y objetivos

[

Se debe especificar el propósito de la aplicación de este documento y los objetivos que se quieren lograr al usarlo.

]

1.2 Alcance

[

Describir los procesos o proyectos con los que está involucrado este documento.

]

1.3 Definiciones, Acrónimos y Abreviaturas

[

Definiciones, acrónimos y abreviaturas utilizadas en el documento.

Por Ej.

N: (Nivel) Importancia del aspecto a evaluar

E: Evaluación

NP: No Procede

Comentario: Es obligatorio en las respuestas negativas.

Las evaluaciones serán:

Mal, no es entendible. **(0)**

Regular, no es claro. **(2)**

Satisfactorio, se entiende lo que quiere decir. **(4)**

Excelente, está correctamente. **(5)**

[El peso de la pregunta se refiere a la evaluación en cuanto al nivel]

El peso de la pregunta será:

[

Muy importante **(5)**

Menos importante **(3)**

]

1.4 Referencias.

[

Lista de documentos a los que se hace referencia en la presente lista de chequeo.

]

1.5 Resumen.

[

Resumen de los aspectos de la lista de chequeo.

]

2. Lista de Chequeo General

2.1 Chequeo en cuanto al uso adecuado de conceptos.

[

Se deben evaluar los conceptos relacionados con el proceso a revisar. Se deberá confeccionar la siguiente tabla:

]

Evaluación	N	E	NP	Comentario
Se debe redactar una pregunta con relación al tema a tratar.				

2.2 Chequeo en cuanto a diversos temas de interés.

[

Se deberán realizar diversos chequeos según el proceso o etapa a la que se le esté aplicando la lista de chequeo. Se pondrá una tabla igual a la del sub-tópico anterior pero la pregunta será relacionada con el tema a tratar aquí.

]

2.3 Chequeo en cuanto a la correcta redacción de la plantilla por revisar.

[

Se revisará la correcta redacción del documento a revisar. Se pondrá una tabla igual a la mostrada anteriormente; pero la pregunta será relacionada con el tema tratado en este sub-tópico.

]

ANEXO 8: Plantilla para la elaboración del Plan de Pruebas



Plan de Pruebas del Sistema

Plan de Pruebas del Sistema

Interno

<Proyecto Banco>

<Versión>

Control de versiones

Fecha	Versión	Descripción	Autor
<00/00/00>	<1.0>	<detalles>	<nombre>

Tabla de contenidos

1.INTRODUCCIÓN.....130

- 1.1.2. Alcance 131
- 1.1.3. Definiciones, Acrónimos y Abreviaturas..... 131
- 1.1.4. Referencias..... 131
- .2. Organización del Equipo de Pruebas..... 131
- .3. Arquitectura técnica..... 131

.4. Especificaciones del Software y Hardware	131
1.5.DESCRIPCIÓN DEL PLAN DE PRUEBAS.....	132
1.5.1. Descripción de los requerimientos.....	132
1.5.2. Requerimientos Funcionales.....	132
1.5.3. Casos de Uso: <Caso de Uso1>	132
1.5.4. Requerimientos de Diseño	134
1.5.5. Requerimientos de Integración	134
1.5.6. Otros Requerimientos.....	134
2.ESTRATEGIA DE PRUEBA.....	134
2.1.1. Objetivo	134
2.1.2. Técnica	134
2.1.3. Entorno de Prueba	135
2.1.4. Proceso	135
2.1.5. Casos de Prueba.....	135
2.1.6. Criterios de Término	135
2.1.7. Herramientas.....	135
3.RECURSOS REQUERIDOS	135
4.PLAN DE PROYECTO	135
5.CALENDARIO Y PLAZOS.....	135
6.DEFINICIÓN DE LOS ENTREGABLES.	136
7.SEGUIMIENTO Y REPORTE DE DEFECTOS.....	136
8.APROBACIÓN DEL PLAN.	137
9.DOCUMENTACIÓN DE LOS RESULTADOS.....	137

1. Introducción

El presente documento se confeccionó con el objetivo de definir el plan de pruebas al software desarrollado por el Proyecto Banco.

1.1.1. Objetivos

Los objetivos de este Plan de Pruebas consisten en:

- Identificar los elementos de pruebas.
- Identificar los recursos y configuraciones necesarias para llevar a cabo las pruebas.

- Describir y recomendar las estrategias de las pruebas a ser empleadas.
- Definir el cronograma de las pruebas.

1.1.2. Alcance

[Proyectos con los que se involucra el Plan]

El siguiente Plan de Pruebas debe ser confeccionado para el uso exclusivo del Proyecto Banco.

1.1.3. Definiciones, Acrónimos y Abreviaturas

[Listado del significado de las definiciones, acrónimos y abreviaturas que hace referencia el Plan]

1.1.4. Referencias

[Lista de documentos a los que se hace referencia en el Plan]

Código	Título
[1]	Documento 1
[2]	Documento 2
[3]	Modelo de Diseño - Módulo de Administración v0.0

2. Organización del Equipo de Pruebas

[Descripción del equipo de probadores, por quienes está compuesto, responsabilidad de cada miembro. Ejemplo:

Diseñador de prueba: Planifica, diseña y evalúa los resultados de la prueba.

Probadores: Realizan las pruebas unitarias, de integración y sistema.]

3. Arquitectura técnica.

[Descripción mediante diagramas (Despliegue, componentes) las partes que componen el sistema bajo prueba. Incluye el almacenamiento de datos y las conexiones para su transferencia y describe el objetivo de cada componente, inclusive la forma de su actualización. Se debe documentar tanto las capas, como la presentación / interfaz del usuario, la base de datos, los emisores de informes, etc. Un diagrama de alto nivel que muestre como el sistema en prueba se inserta en un contexto de automatización mayor también puede ser agregado, si el mismo está disponible.]

4. Especificaciones del Software y Hardware

[Corresponde a una lista individualizada de todo el hardware y el software que utiliza la aplicación, incluyendo proveedores y versiones].

1.5. Descripción del Plan de Pruebas

1.5.1. Descripción de los requerimientos

[Esta sección del Plan de Pruebas contiene una lista de todos los requerimientos. Cualquier requerimiento no incluido en esta lista estará fuera del alcance de las pruebas. Esto libera de responsabilidad en caso de que existan problemas con la funcionalidad del sistema que se relaciona con un requisito no incluido en este listado.]

1.5.2. Requerimientos Funcionales

[Todas las funciones que deben ser probadas, como por ejemplo la creación, la corrección y supresión de registros, son puestas en esta lista. Puede incluirse la lista completa en esta sección o bien hacerse referencia a otro documento que contenga la información.]

1.5.3. Casos de Uso: <Caso de Uso1>

[Aquí se debe indicar brevemente en que consiste el CU, y en las secciones posteriores incluir la información necesaria para verificar su funcionamiento]

Escenarios

[Aquí se plantea la matriz de escenarios correspondiente al Caso de Uso 1, esto es todas las posibles combinaciones del flujo normal de los eventos y los flujos alternativos, caminos que se derivan del flujo de eventos del Caso de Uso]

Nombre de escenario	Flujo inicial	Flujo alternativo	Flujo alternativo

Plantilla de condiciones

[Aquí se plantea la planilla de Condiciones, esto es las condiciones que causan que se ejecute un escenario específico dentro de los posibles escenarios identificados para el Caso de Uso1, indicando

el resultado esperado de ejercitar ese Escenario-Condición y la referencia a los casos de prueba de la planilla CasosPruebaConDatos.xls para ese CU que se corresponden con ese Escenario-Condición]

Escenario	Descripción	Condiciones o Elementos			Resultado Esperado
		<Las condiciones o elementos requeridos para ejecutar los distintos escenarios>			
		Condición 1	Condición 2	...	
Esc 1					
...					
Esc 2					

Diagrama de Entidad Relación

Entidades que intervienen en el caso de uso en cada uno de los estados por los que transitan.

CASOS DE PRUEBA

Aquí se plantean las pruebas para ese ciclo, dado que una prueba encadena un escenario de cada caso de uso, se debe dar la referencia al Escenario del Caso de Uso, la referencia a los datos que prueban ese escenario y las entidades involucradas en cada prueba y sus estados, para el caso de Prueba del ciclo debe darse el resultado esperado.

Caso del Ciclo	Datos de la Prueba				Resultado Esperado
	Caso de Uso	Escenario-Condición	Caso de Prueba	Entidades/Estados	
1					
2	Caso de	Escenario-	Caso de Prueba	Entidades/Estados	

	Uso	Condición			

1.5.4. Requerimientos de Diseño

Las pruebas de la interfaz de usuario, las estructuras de menú u otros elementos de diseño también deberían ser puestas en una lista o referenciados hacia otro documento.

1.5.5. Requerimientos de Integración

Los requerimientos para probar el flujo de datos desde un componente a otro deben ser incluidos si ellos harán parte del Plan de Pruebas.

1.5.6. Otros Requerimientos

Cualesquiera otras exigencias que tenga la aplicación y que necesiten ser probadas.

2. Estrategia de Prueba

Use esta sección para describir como los objetivos de la prueba serán alcanzados para cada uno de los tipos de pruebas que hacen parte del plan, ejemplo:

- *Pruebas de Caja Blanca a cada caso de uso implementado.*
- *Pruebas de Caja Negra a cada módulo concluido.*
- *Pruebas de integración del sistema.*
- *Pruebas de seguridad.*
- *Pruebas de aceptación del cliente.*

Para cada subconjunto requerido o definido como necesario, debe detallarse lo siguiente:

2.1.1. Objetivo

El objetivo global de esta estrategia debe alcanzarse. Por ejemplo, para una prueba de sistema, este objetivo puede ser una declaración de que todos los requerimientos funcionales deben comportarse de acuerdo a lo esperado, o como quedó documentado.

2.1.2. Técnica

Especifica como los casos de prueba serán desarrollados, el instrumento o herramienta usado para almacenarlos y donde pueden ser encontrados; como ellos serán ejecutados y los datos que serán usados. Declare aquí si las pruebas deben ser realizadas en ciclos, o de común acuerdo con los otros esfuerzos de pruebas.

2.1.3. Entorno de Prueba

Especificar las condiciones de hardware y configuración bajo las cuales se deben realizar las pruebas.

2.1.4. Proceso

Breve descripción del proceso que se realiza.

2.1.5. Casos de Prueba

Hacer una lista detallada o una referencia a los casos reales de prueba que serán utilizados para poner en práctica el plan.

2.1.6. Criterios de Término

Registrar los criterios que serán usados para determinar la aprobación o rechazo de pruebas y la acción que debe ser tomada con base en los resultados de la prueba.

2.1.7. Herramientas

Documentar los instrumentos o herramientas que serán empleados para las pruebas. Citar al proveedor, la versión y el número de la Mesa de Ayuda para pedir el apoyo, si fuera necesario.

3. Recursos Requeridos

Identificar los roles y las responsabilidades que serán requeridas para la ejecución del Plan de Pruebas.

Rol	Responsabilidad

4. Plan de Proyecto

Parte del cronograma del proyecto que abarca la etapa de pruebas.

5. Calendario y Plazos.

Documentar el plazo en el cual la aplicación a probar estará disponible para pruebas y el tiempo estimado para ejecutar los casos de prueba. Especifique si se proporcionará partes construidas, sobre una base regular durante el ciclo de prueba, o cuando se espera que los componentes del sistema estén listos para pruebas.

6. Definición de los Entregables.

Ponga en una lista cualquier entregable asociado con el esfuerzo de pruebas y donde las copias de estos entregables o documentos pueden ser localizados. Esto incluye el Plan de Pruebas en sí mismo, escenarios para prueba, casos de prueba y el plan de proyecto.

Los entregables asociados a las pruebas son:

- *Plan de pruebas*
- *Casos de pruebas de caja blanca.*
- *Casos de pruebas de caja negra.*
- *Casos de prueba para cada prueba diseñada.*
- *Informe de verificación de pruebas código.*
- *Informe de verificación de pruebas de integración.*
- *Informe de verificación de pruebas de sistema.*
- *Informe de verificación de pruebas de aceptación del cliente.*
- *Plan de proyecto.*

Estos pueden ser localizados en el repositorio del proyecto.

7. Seguimiento y Reporte de Defectos.

Para el seguimiento de los defectos encontrados y sus reportes se identificarán los entregables tales como: Informe de No Conformidades.

Documente el instrumento y el proceso usado para registrar y rastrear los defectos. Ponga en una lista todos los informes que serán generados incluyendo repositorios, frecuencias, mecanismos de entrega y ejemplos. Identifique los recursos involucrados en el proceso de seguimiento.

Describa cualesquiera calificación, categoría o clasificación que se usará para identificar o priorizar defectos. Las siguientes son categorías, de ejemplo, 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 esperado, o no se ajusta a las normas y convenciones.
- **Cosmético:** No crítico para el funcionamiento de sistema: palabras con mala ortografía, formateo incorrecto, mensajes de error vagos o confusos o advertencias.

8. Aprobación del Plan.

El siguiente Plan de Pruebas deberá ser revisado y aprobado por el jefe del proyecto y el equipo de prueba. Para verificar que todos los involucrados aprueben el plan se deberá hacer una reunión final después de su revisión y dejar constancia de lo acordado.

Si se aprueba el Plan de Pruebas para el proyecto Banco se llenará la siguiente tabla:

Versión:		Nombre:	Fecha:	Firma:
	Elaborado por:			
	Aprobado por:			

9. 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.

Caso de Prueba	
Fecha	
Resultado	
Observaciones	

Responsables de las Pruebas	

ANEXO 9: Fórmula para las métricas

TABLA CONTENTIVA DE LAS MÉTRICAS DE CALIDAD PROPUESTAS
(Tomado de la ISO/IEC 9126)

1.1 Tablas contentivas de las métricas de funcionalidad

Tabla 1.1.1 Métricas de idoneidad

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Escala	Tipo de medida
a) Adecuación funcional	¿Cuán adecuada es la función evaluada?	Número de funciones idóneas para ejecutar funciones específicas en comparación con el número de funciones evaluadas	$X = 1 - A/B$ A - Número de funciones en las cuales se detectaron problemas en la evaluación B - Número de funciones evaluadas	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada	Absoluta	X = contab/ Contable A = Contab B = Contab
b)	¿Cuán	Ejecutar las pruebas (de	$X = 1 - A/B$	$0 \leq X \leq 1$		X =

Completitud de la implementación funcional	completa ha sido la implementación y su conformidad con la Especificación de requisitos?	caja negra) funcionales de acuerdo con la especificación de requisitos. Cuenta el número de funciones perdidas detectadas y compare el resultado con el número de funciones descritas en la especificación de requisitos.	A - Número de funciones perdidas detectadas en la evaluación B - Número de funciones descritas en especificación de requisitos.	A mayor cercanía al 1 resultará mejor	Absoluta	contab/ Contable $A = \text{contabl}$ $B = \text{contabl}$
c) Cobertura de la implementación funcional	¿Cuán correcta ha sido la implementación funcional?	Ejecutar las pruebas funcionales (de caja negra) de acuerdo con la especificación de requisitos. Cuenta el número de funciones incorrectamente implementadas o funciones perdidas detectadas y compare el resultado con el número total de funciones descritas en la especificación de requisitos. Cuenta el número de funciones que están completas en relación con las que no lo están..	$X = 1 - A/B$ A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas. B - Número de funciones descritas en la especificación de requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	Absoluta	$X = \text{contab}/\text{Contable}$ $A = \text{contabl}$ $B = \text{contabl}$

Tabla 1.1.2 Métricas de exactitud

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Escala	Tipo de medida
a) Exactitud esperada	¿Existen diferencias entre los resultados actuales y los razonablemente esperados?	Ejecutar los casos de pruebas de entrada versus salida y comparar los resultados actuales y los razonablemente esperados. Cuenta el número de casos encontrados con diferencias inaceptables en relación con los resultados razonablemente esperados. .	$X = A/T$ A - Número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible. T - Tiempo de operación	$0 \leq X$ A mayor cercanía al 0 resultará mejor	Valorativa	X = contab/ tiempo A = contabl T = tiempo

Tabla 1.1.3 Métricas de interoperabilidad

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Escala	Tipo de medida
----------------------	-----------------------------	----------------------	--------------------	-----------------------------------	--------	----------------

<p>a) Intercambiabilidad de datos, en base su formato</p>	<p>¿Cuán correctamente ha sido implementado el intercambio de funciones de interfaces para una transferencia de datos específica?</p>	<p>Número Ejecutar las pruebas a cada registro de salida de las funciones de interfaces de acuerdo con la especificación de los campos de datos. Cuente el número de formatos de datos que deben ser intercambiados con otros software o sistemas durante las pruebas en comparación con el número total</p>	<p>$X = A / B$</p> <p>A- Número de formatos de datos intercambiados exitosamente con otro software o sistemas durante las pruebas del intercambio de datos.</p> <p>B - Número total de formatos de datos a intercambiar</p>	<p>$0 \leq X \leq 1$</p> <p>A mayor cercanía al 1 resultará mayor intercambiabilidad</p>	<p>Aboluta</p>	<p>$X = \frac{\text{contab/}}{\text{Contable}}$</p> <p>$A = \text{Contab}$</p> <p>$B = \text{Contab}$</p>
<p>b) Intercambiabilidad de datos, en base éxito del intento</p>	<p>¿Cuán frecuentemente falló el intento de intercambio de datos entre el software objeto de la prueba y otro software?</p> <p>¿Cuán</p>	<p>Ejecutar las pruebas. Cuente el número de casos en que las funciones de interfaces fueron usadas y fallaron.</p>	<p>$1) X = 1 - A/B$</p> <p>A- Número de casos en que se falló al proceder a un intercambio de datos con otro software o sistemas.</p> <p>B - Número</p>	<p>$0 \leq X \leq 1$</p> <p>A mayor cercanía al 1 resultará mejor</p>	<p>1) Aboluta</p>	<p>1) $X = \frac{\text{cont/}}{\text{cont}}$</p> <p>$A = \text{contabl}$</p> <p>$B = \text{contabl}$</p> <p>2) $Y =$</p>

	frecuentemente ES SATISFACTORIA la transferencia de datos entre el software objeto de la prueba y otro?		de casos en que se intentó proceder a un intercambio de datos 2) $Y = A / T$ T - Período de tiempo de operación	$0 \leq Y$ A mayor cercanía al 0 resultará mejor		cont/ tiempo T - tiempo 2) Valorativa
--	---	--	---	---	--	--

1.2 Tabla contentiva de las métricas de usabilidad

Tabla 1.2.1 Métricas de comprensibilidad

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Escala	Tipo de medida
a) Integridad de la descripción de producto	¿Qué proporción de funciones (o tipos de funciones) son comprendidas después de leer la	Conduzca las pruebas de usuario. Realice entrevistas a usuarios con cuestionarios preparados al efecto. Observe el comportamiento del usuario. Cuente el número de funciones	1) $X = A / B$ A - número de funciones comprendida B - número total de funciones	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	1), 2) Absolutas	X = cont/cont A = Contable B = Contable

	descripción del producto?	que fueron adecuadamente comprendidas en comparación con el número total de funciones en el producto.				
b)	Accesibilidad a demos	¿A qué proporción de demos/tutoriales pueden acceder los usuarios?	Conduzca las pruebas de usuario. Observe el comportamiento del usuario.	$X = A / B$ A - número de demos/tutoriales a los que pueden acceder los usuarios exitosamente. B - número total de demos/tutoriales a los que se puede acceder	$0 < X \leq 1$ Mientras cercano al 1, mejor	Absoluta $X = \frac{\text{Contable}}{\text{contable}}$ $A = \text{Contable}$ $B = \text{Contable}$
c)	Comprensibilidad de entradas y salidas	¿Pueden los usuarios comprender lo que se requiere como entrada y lo que suministra el sistema de	Conduzca las pruebas de usuario. Realice entrevistas a usuarios con cuestionarios preparados al efecto. Observe el comportamiento del usuario. Cuente el número de elementos de entrada.	$X = A / B$ A - número de elementos de entrada y que suministra el sistema de software como salida	$0 \leq X \leq 1$ Mientras cercano al 1, mejor	Absoluta $X = \frac{\text{Cont}}{\text{cont}}$ $A = \text{Contable}$ $B = \text{Contable}$

	software como salida?		comprendidas correctamente B - número total de elementos de entrada y que suministra el sistema de software como salida proporcionados por el interfaz.			
--	-----------------------	--	--	--	--	--

Tabla 1.2.2 Métricas de conformidad con la usabilidad

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Escala	Tipo de medida
a) Conformidad con la usabilidad	¿Cuánto se adhiere el producto de software a las regulaciones aplicables, las normas, convenciones, estilos y	Especifique los elementos que requieren estar en conformidad, diseñe los casos de prueba de esos elementos, conduzca las pruebas funcionales para esos elementos que requieren estar en conformidad.	$X = 1 - A / B$ A- número de elementos especificados que requiriendo estar en conformidad no han sido implementado	$0 \leq X \leq 1$ A mayor cercanía al 1 mejor	Absoluta	X = Contable/ Contable A = Contable B = Contable

	guías relacionadas con la usabilidad?		s durante las pruebas B - Número total de elementos que requieren estar en conformidad especificados.			
--	---------------------------------------	--	--	--	--	--

ESPECIFICACIÓN DE LAS MÉTRICAS PARA EL PROCESO PROPUESTAS EN EL PLAN

- Eficacia en la eliminación de defectos (**EED**)

Fórmula:

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

E: es el número de errores encontrados antes de la entrega final.

D: número de defectos encontrados después de la entrega.

El objetivo es que EED adquiera el valor de 1, si esto sucede es que no se han encontrado defectos en el software. Entre más errores encontremos antes de la entrega mejor funcionarán las técnicas de garantía de calidad.

La EED también puede utilizarse para medir la habilidad de un equipo para encontrar errores antes de pasar a la siguiente actividad, entonces definiríamos la siguiente Fórmula:

$$EED (i) = E (i) / (E (i) + E (i + 1))$$

Donde:

E_i: errores detectados en la actividad i,

E (i+1): errores detectados en la actividad i+1 que no se detectaron y provienen de la actividad i, y el objetivo es obtener un valor de 1.

- Grado de efectividad en identificar las no-conformidades **(ENC)**

Fórmula:

$$ENC = NCE/NCR$$

ENC: Efectividad en encontrar No Conformidades.

NCE: No Conformidades Encontradas.

NCR: No Conformidades Reales.

- Por ciento de efectividad para resolver las no-conformidades **(PER)**

Fórmula:

$$PER \% = NCR *100/ TNCI$$

PER %= Por ciento de Efectividad en Resolver las no conformidades.

NCR: No Conformidades Resueltas.

TNCI: Total de No Conformidades Identificadas.

- Capacidad del procedimiento especificado para evaluar los procesos **(C)**

Fórmula:

$$C = PEC/TPE$$

C: Capacidad del procedimiento definido.

PEC: Procesos Evaluados Correctamente.

TPE: Total de Procesos Evaluados.

GLOSARIO

ANSI: American National Standards Institute.

Caso de Prueba: Especificación de cómo probar el sistema, incluyendo la entrada, salida y condiciones bajo las cuales de prueba.

Cliente: Persona, organización o grupo de personas que encargan la construcción de un producto software.

EEUU: Estados Unidos.

IEC: Comisión Electrotécnica Internacional

Indicadores: Métricas o combinación de ellas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en sí.

Inspecciones: Repaso detallado y formal del trabajo en proceso.

Lista de Chequeo: Lista de aspectos en forma de preguntas utilizados para revisar o inspeccionar temas de interés de un producto o proceso.

MS-DOS: Siglas de **Micro**Soft **Disk** **O**perating **S**ystem, Sistema operativo de disco de Microsoft.

No Conformidad: Un error o falla detectado en la funcionalidad que está pactado con el cliente.

Proceso de Software: (ISO-15504) Es el proceso (o procesos) usado por una organización (o proyecto) para planificar, administrar, ejecutar, monitorear, controlar y mejorar sus actividades relacionadas con el software.

Producto software: (IEEE-12207) Es el conjunto de programas de computadora, procedimientos, documentación y datos asociados.

Rol: Papel, cometido o función que tiene o desempeña un actor.

SEI: Instituto de Ingeniería de Software

SQA: Software Quality Assurance (Aseguramiento de la calidad del software).

SQL: Structured Query Language (Lenguaje de consulta estructurado). Es un lenguaje declarativo de acceso a bases de datos relacionales.

SW: Abreviatura de Software

UML: Unified Modeling Language. Lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

Usuario: Persona encargada de utilizar el sistema, obteniendo algún beneficio.

Visual FoxPro: Lenguaje de programación orientado a objetos y procedural que ofrece a los desarrolladores un conjunto de herramientas para crear aplicaciones de bases de datos para el escritorio.