

Aplicación práctica del diseño de pruebas de software a nivel de programación

Oscar Hernando Guzmán Cortés

oguzman@icesi.edu.co

Fecha de recepción: 11-12-2003

Fecha de aceptación: 20-4-2004

ABSTRACT

Tests must be present in all software life cycle phases, including requirements, analysis and design, programming, implementation and maintenance.

This article presents the design and execution scheme of software test, specifically centered in programming tests defined to Software Development department of Icesi University. The requirements tests scheme is shown in a basic form. The schemes of analysis and design tests, implementations tests, and maintenance tests, are not shown because they aren't totally defined.

KEYWORDS

Programming test, requirements test, test design.

RESUMEN

Las pruebas deben presentarse a lo largo de todo el ciclo de vida del desarrollo de software, pasando por requerimientos, análisis y diseño, programación, puesta en marcha y mantenimiento.

El siguiente artículo presenta el esquema de diseño y ejecución de pruebas de software, centrándose específicamente en las pruebas de programación, definidas para el departamento de Desarrollo de Sistemas de

la Universidad Icesi. El esquema de pruebas de requerimientos se muestra de manera general, mientras que del esquema de pruebas de análisis y diseño, de puesta en marcha y de mantenimiento no se presentan por no estar todavía totalmente definidas.

PALABRAS CLAVES

Pruebas de programación, pruebas de requerimientos, diseño de pruebas.

Clasificación: B

INTRODUCCIÓN

En el mundo de la computación tan cambiante de hoy en día, y sobre todo de gran evolución tecnológica, y en vista de las exigencias que ha traído la globalización, se ha hecho necesario desarrollar metodologías para asegurar la calidad de los productos de software y obtener un mejoramiento continuo de todos los procesos relacionados con el desarrollo de software. Entre tantas metodologías, se pueden mencionar: STD (Software Technology Diagnostic), CMM (Capability Maturity Model), Bootstrap, Trillium, y HealthCheck. Vale la pena aclarar que CMM es un esquema de diagnóstico y de evaluación de la madurez del proceso de desarrollo de software, más que un esquema de mejoramiento de procesos.

Todos estos mecanismos de evaluación y mejora en el desarrollo de software han permitido que las empresas implementen la metodología que más se ajuste a sus necesidades y forma de trabajar. Desde este enfoque, el equipo de Desarrollo de Sistemas de la Universidad Icesi ha acoplado algunos conceptos relevantes de

CMM y establecido estándares para definir su propio modelo de aseguramiento de la calidad de software; anotando que algunos elementos de dicho modelo están en proceso de definición, otros ya se están implementando, y otros están pendientes de ajustarlos a nuestras necesidades.

A continuación, se presenta el proceso de diseño y ejecución de pruebas de software (básicamente pruebas de programación) que se ha definido para el departamento de Desarrollo de Sistemas de la Universidad Icesi.

1. ¿CÓMO LLEGAR A LA DEFINICIÓN DEL ESQUEMA DE PRUEBAS DE SOFTWARE?

CMM, en términos generales, provee una guía de cómo obtener el control del proceso de desarrollo y mantenimiento de software, de cómo evolucionar hacia una cultura de ingeniería de software. La Figura 1 muestra el esquema general de cinco niveles de madurez del proceso de software propuesto por CMM, y la Figura 2 revela la estructura de cada nivel de madurez.



Figura 1. Niveles de madurez del proceso de software.

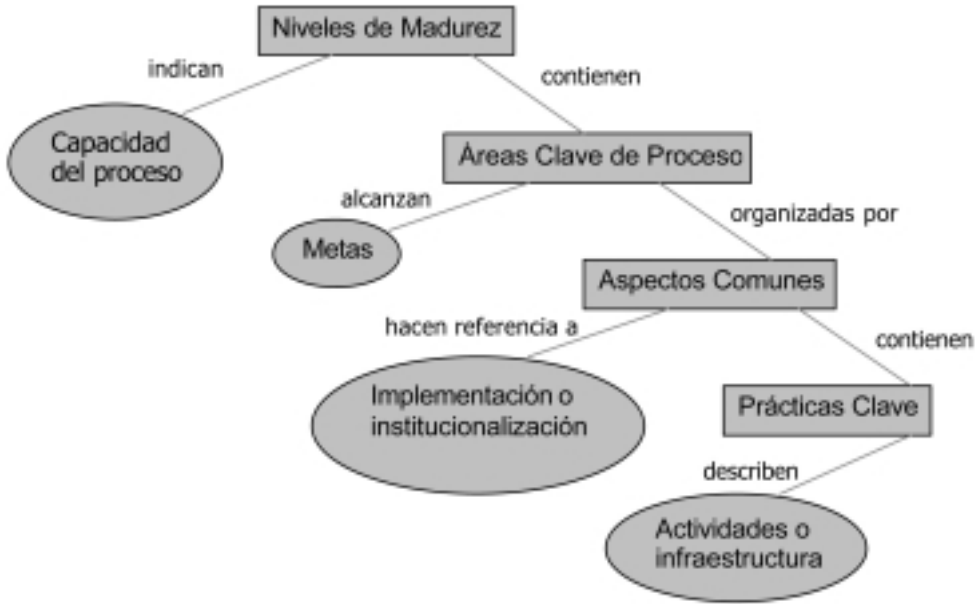


Figura 2. Estructura de los niveles de madurez de CMM.

Como se ha mencionado anteriormente, se ha revisado el esquema propuesto por CMM para determinar el estado actual del proceso de desarrollo de software, y establecer las acciones a tomar en búsqueda de alcanzar un mayor nivel de madurez en dicho proceso.

Y, en lo concerniente específicamente al esquema del plan de aseguramiento de calidad de software, se identificó para los tres primeros niveles lo siguiente:

- Nivel 1. Proceso Inicial:
 - Proceso ad hoc que puede ser caótico.
 - No hay procedimientos formalizados, estimados de costos y planeación de proyectos.

- Las herramientas no están bien integradas con el proceso ni se aplican de manera uniforme.
- El control de cambios no es estricto.
- La instalación y el mantenimiento del software presentan problemas.
- No hay un mecanismo que asegure la utilización de procedimientos formales.
- Se debe establecer un plan formal de aseguramiento de calidad de software.

- Nivel 2. Proceso Repetible:
 - Riesgos:
 - * Nuevas herramientas y métodos podrían afectar el proceso.

- * Nuevo tipo de producto puede cambiar todo el proceso.
 - * Cambios organizacionales grandes.
- Establecer un grupo de procesos: Es el encargado de la definición del proceso de desarrollo, la identificación de necesidades y oportunidades a nivel de tecnología, la asesoría a proyectos, informes gerenciales quincenales sobre el estado y desempeño de los procesos. Si son de menos de cuatro personas, es suficiente con un comité conformado por profesionales con experiencia o consultores.
- El propósito del grupo es el mejoramiento del proceso de software.
- Establecer una arquitectura del proceso de desarrollo de software (ciclo de vida de desarrollo de software): Son las actividades técnicas y administrativas requeridas para la ejecución apropiada del proceso de desarrollo. Como arquitectura se entiende tareas con sus prerrequisitos, descripciones funcionales, procedimientos de verificación y especificaciones que indican si una tarea se ha completado.
- Introducir una familia de métodos de ingeniería de software y tecnologías, tal como inspecciones de código y diseño, métodos formales de diseño, sistemas de control de librerías y métodos de prueba,

- prototipos, lenguajes de implementación sofisticados, etc.
- Se debía establecer un programa de SQA que asegurará que:
- * Se utiliza una metodología de desarrollo de software apropiada.
 - * Los proyectos utilizan estándares y procedimientos.
 - * Se llevan a cabo revisiones y auditorías independientes.
 - * Existe documentación para soportar mantenimiento y mejoras.
 - * La documentación se produce durante y no después del desarrollo de software.
 - * Hay mecanismos para controlar cambios.
 - * Las pruebas hacen énfasis en las áreas de alto riesgo.
 - * Cada tarea de software se completa a satisfacción, antes de continuar con la siguiente.
 - * Las desviaciones con respecto a estándares y procedimientos se detectan lo más pronto posible.
 - * El trabajo de control de calidad de software se trabaja con respecto a unos estándares establecidos.
 - * El plan de aseguramiento de calidad y el plan de desarrollo de software son compatibles.

— Este nivel contempla cinco áreas claves: gestión de requerimientos, planeación del proyecto, seguimiento y supervisión del proyecto, aseguramiento de la calidad, gestión de la configuración del software, y gestión de los subcon-

tratos de software. En la Tabla 1 se muestra el puntaje obtenido para las actividades definidas del área clave de aseguramiento de calidad. Y, en la Tabla 2 se muestra la escala de puntajes utilizada.

Tabla 1. Puntajes y prioridades de las actividades del área clave de aseguramiento de calidad.

Actividades	Puntaje	Prioridad
Un plan de SQA es preparado para el proyecto de software de acuerdo con los procedimientos existentes y documentados, y se sigue por parte de los integrantes del Equipo de Desarrollo de Software.	0.0	1
Las actividades del grupo de SQA son realizadas de acuerdo con el plan de SQA definido.	0.0	2
El líder del proyecto revisa las actividades del GIS para verificar el cumplimiento del plan y los estándares.	0.0	3
Se debe presentar el informe de calidad del proyecto, teniendo en cuenta las desviaciones identificadas en las actividades o en los productos de software.	0.0	4
El encargado de investigar SQA debe definir, programar y coordinar revisiones periódicas de las actividades, para garantizar el cumplimiento total del plan SQA.	0.0	5
Puntaje máximo	5.0	
Puntaje total obtenido	0.0	

Tabla 2. Escala de puntajes

Puntaje	Descripción
0.0	No se realiza
0.5	Se realiza parcialmente
1.0	Se realiza totalmente

- Nivel 3. Proceso definido:

Se requiere trabajar en los siguientes aspectos:

- Estándares de software.
- Inspecciones de software.
- Pruebas de software.
- Trabajo adicional en gestión de configuraciones.
- Definición del proceso de software.
- Conformación del grupo del proceso de ingeniería de software.

Debido a este análisis anterior, se decidió crear el Equipo de Aseguramiento de Calidad de Software, el cual se encarga de todo el proceso de sqa. En el Anexo 1 se presenta una lista de chequeo de aseguramiento de calidad. Dentro del proceso de sqa, se muestra en el siguiente punto, lo relacionado con el diseño y ejecución de las pruebas de software.

2. DISEÑO Y EJECUCIÓN DE PRUEBAS DE SOFTWARE

Una definición que se puede dar de *pruebas* es la siguiente: “Una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan y registran, y se realiza una evaluación de algún aspecto”.

La *prueba* es un elemento crítico para la calidad del software. La importancia de los costos asociados a los errores promueven la definición y aplicación de un proceso de pruebas minuciosas y bien planificadas. Las pruebas permiten validar y verificar el software, entendiendo como validación del software el proceso que determina si el software satisface los requisitos, y verificación como el proceso que determina si los productos de una fase satisfacen las condiciones de dicha fase.

Las estrategias de pruebas permiten enfocar el plan de pruebas; éste comprende la visión global del proceso de pruebas, y la definición de actividades y roles involucrados en cada una de ellas.

Las pruebas que se han considerado, dentro del plan de pruebas, son las siguientes:

- Pruebas de requerimientos.
- Pruebas de análisis (*pendiente*).
- Pruebas de diseño (*pendiente*).
- Pruebas de unidad.
- Inspecciones.
- Pruebas de información no periódica.

La Figura 3 muestra, gráficamente, las pruebas de software consideradas.

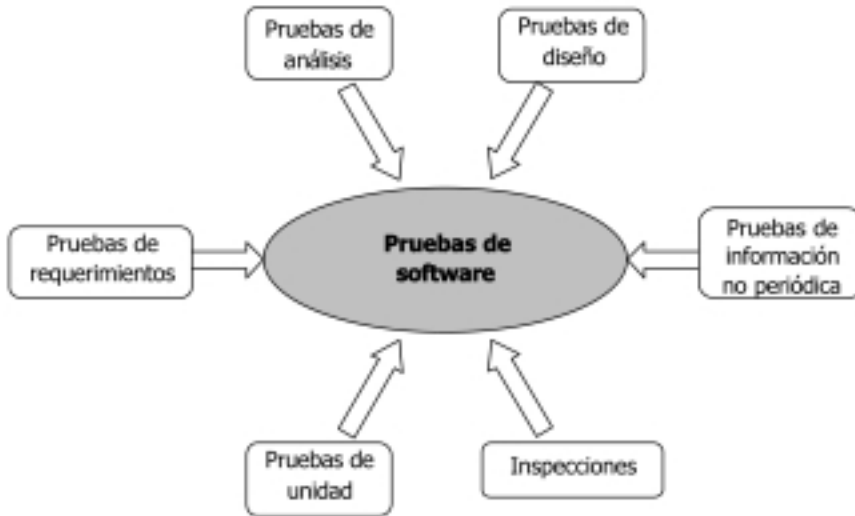


Figura 3. Pruebas de software consideradas.

2.1 Pruebas de requerimientos

Los requerimientos de software deben tener una explicación clara, precisa y completa del problema que facilite el análisis de errores y la generación de casos de prueba. Un asunto de gran importancia es asegurar la corrección, coherencia y exactitud de los requisitos.

Durante el proceso de elicitación de requerimientos, una persona, designada por el Equipo de Aseguramiento de Calidad, revisará el documento de especificación de requerimientos,

con la lista de chequeo general del documento y la lista de chequeo de requerimientos.

La corrección del contenido del documento será responsabilidad del analista y el usuario líder, quienes son los encargados de aprobar los requerimientos definidos en el documento. El diagrama del proceso de elicitación de requerimientos se muestra en la Figura 4, y el detalle del mismo en la Tabla 3, que se presentan a continuación.

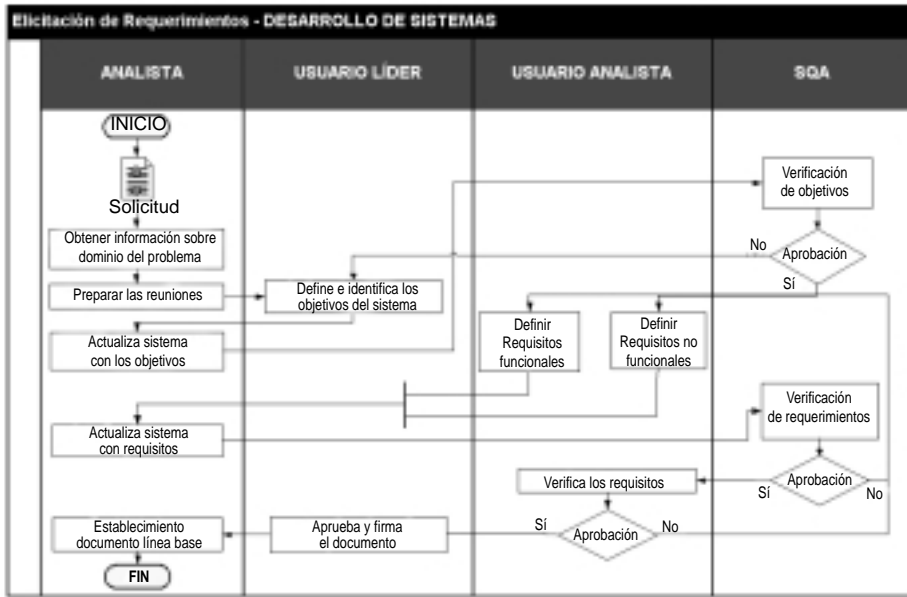


Figura 4. Proceso de elicitación de requerimientos

Tabla 3. Detalle del proceso de elicitación de requerimientos.

Detalle	Encargado	Recursos
Se determina si los objetivos son claros, verificables y necesarios (entre otros). El resultado de esta revisión se consigna en la lista de chequeo de objetivos.	Revisor SQA	Lista de chequeo de objetivos
Mediante un proceso iterativo se define la funcionalidad esperada del software, y se consigna usando el documento de requisitos del sistema.	Especificador de requerimientos	Requerimientos del sistema
Debe verificar el documento de requerimientos, usando la lista de chequeo general del documento de especificación de requerimientos LCH lista de chequeo general del documento de elicitación y análisis de requerimientos.	Revisor SQA	Lista de chequeo general del documento de elicitación y análisis de requerimientos
Revisan cada requerimiento (consistencia, ambigüedad, etc.), usando para ello la lista de chequeo de requerimientos.	Revisor SQA	Lista de chequeo de requerimientos.

2.2. Pruebas de unidad

El proceso de pruebas de unidad se describe en el siguiente diagrama de

la Figura 5, así como el detalle del mismo en la Tabla 4.

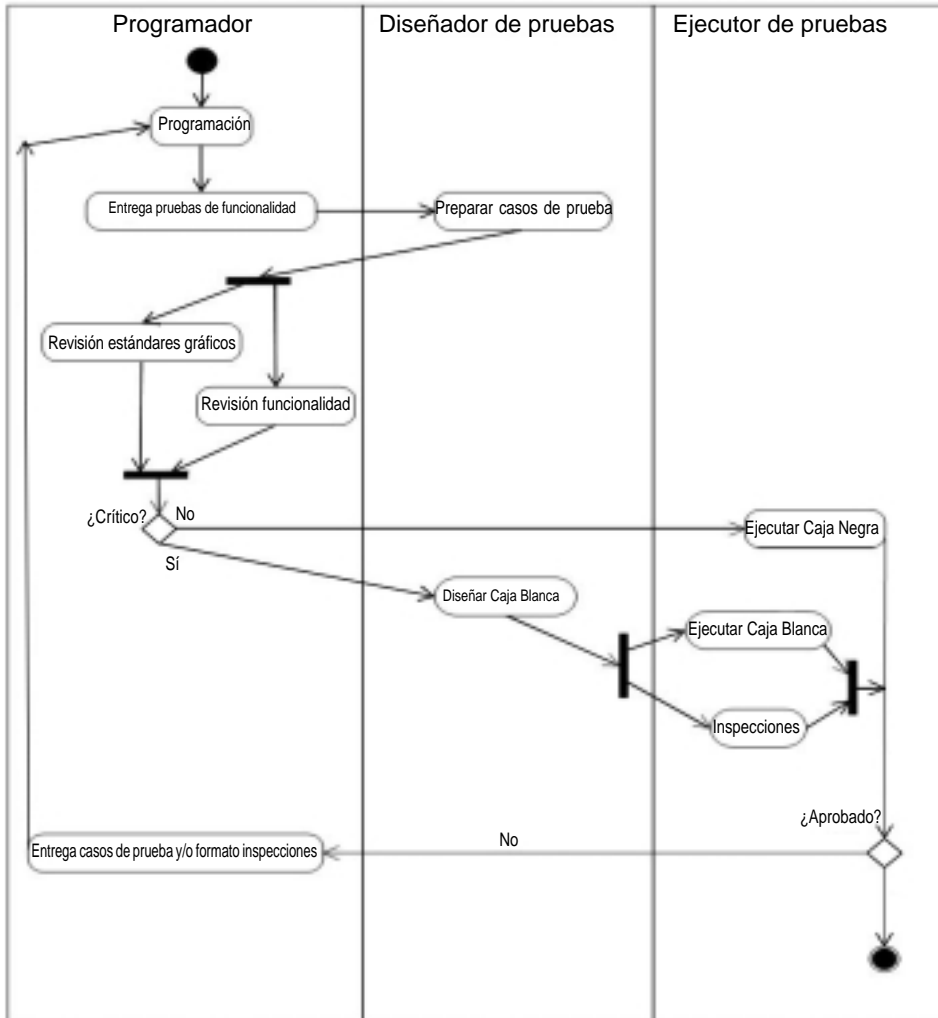


Figura 5. Proceso de pruebas de unidad.

Tabla 4. Detalle del proceso de pruebas de unidad

1. Preparación de casos de prueba			
Detalle	Restricciones	Encargados	Recursos
<p>El encargado de elaborar los casos de prueba recibe del analista el diseño de la forma/reporte y una descripción de la funcionalidad, usando el formato de pruebas de funcionalidad. Así mismo, el diseñador de los casos de prueba debe recibir las listas de chequeo de programación ya revisadas por el analista/programador (Ver Anexos 4, 5, 6, y 8).</p>	<p>Formato de pruebas de funcionalidad (Ver Anexo 13).</p>	<p>Analista diseñador de casos de prueba.</p>	
<p>Si el módulo es nuevo, se elabora el árbol de clases equivalentes, la tabla de particiones y un listado de casos de prueba.</p> <p>Si el módulo no es nuevo, deben revisarse los documentos de caja negra para establecer los cambios, que pueden ser:</p> <ul style="list-style-type: none"> i. Eliminación de entradas, lo cual significa que se deben revisar todos los casos de prueba que incluyan estas entradas para modificarlos apropiadamente. <p>También es posible que se eliminen algunos casos de prueba.</p> <ul style="list-style-type: none"> ii. Inclusión de entradas, lo cual significa revisar los casos de prueba existentes y adicionar nuevos casos. iii. Eliminación/Inclusión de salidas: implica revisar los casos de prueba para establecer los cambios. <p>Cada caso de prueba debe dejarse documentado y se debe ingresar esta información en el sistema.</p>	<p>Formato de pruebas de caja negra (Ver Anexo 10)</p>	<p>Diseñador de casos de prueba.</p>	
	<p>Formato de casos de prueba (Ver Anexo 12)</p>	<p>Diseñador de casos de prueba.</p>	

Tabla 4. Detalle del proceso de pruebas de unidad

2. Revisión de Estándares gráficos			
Detalle	Restricciones	Encargados	Recursos
Realizar la revisión indicada en la lista de chequeo de estándares, y dejar consignados los resultados y la fecha de la revisión en el sistema.	Lista de chequeo de presentación de formas (Ver Anexo 7)	Analista	
3. Revisión de funcionalidad			
Comprobar que se cumpla lo establecido en la lista de chequeo de funcionalidad de formas o reportes, según corresponda. Dejar documentado el resultado y la fecha de realización de esta actividad en el sistema.	Lista de chequeo de funcionalidad de aplicaciones-formas (Ver Anexo 2) Lista de chequeo de funcionalidad de aplicaciones-reportes (Ver Anexo 3)	Analista	
4. Ejecución de pruebas de caja negra			
El ejecutor debe contar con el módulo desarrollado y tener todos los permisos que tendría el usuario final. Además, debe contar con los casos de pruebas diseñados.		Ejecutor de pruebas.	
Se deben ejecutar todos los casos de prueba y consignar los resultados obtenidos en el formato.	Formato de resultados de ejecución de pruebas (Ver Anexo 11).	Ejecutor de pruebas.	

Tabla 4. Detalle del proceso de pruebas de unidad

5. Diseño y ejecución de pruebas de caja blanca [opcional]			
Detalles	Restricciones	Encargados	Recursos
Preparación de los casos de prueba usando la técnica del camino básico. El analista proporciona el código de la función y el cálculo de la complejidad ciclomática.		Diseñador de pruebas.	
Se establece la lista de casos de prueba y se documenta cada uno.		Diseñador de pruebas.	
Se ejecutan las pruebas para cada caso encontrado y se consigian los resultados.		Ejecutor de pruebas.	
6. Inspecciones [opcional]			
Realizar la inspección siguiendo la lista de chequeo y anotando los errores encontrados en el formato.	FOR Inspecciones-Registro de defectos (Ver Anexo 9).	Revisor Moderador Analista	
Realizar la reunión de inspección, orientada por el moderador. El revisor y el moderador exponen los errores hallados.		Revisor Moderador	
El analista aclara las dudas o indica si hubo un error de apreciación		Analista	
Se consigian los resultados consolidados de la inspección y se entregan al analista.		Moderador	
Excepciones			
Cuando el módulo sea crítico, se ejecutan las pruebas de caja blanca y las inspecciones			

2.3 Inspecciones

El objetivo de las inspecciones es implementar un proceso formal de revisión detallada del producto por parte de pares y un moderador, con el pro-

pósito de encontrar defectos en una etapa muy temprana del desarrollo del producto. El diagrama de la Figura 6 muestra el proceso de inspecciones. El detalle de dicho proceso se encuentra en la Tabla 5.

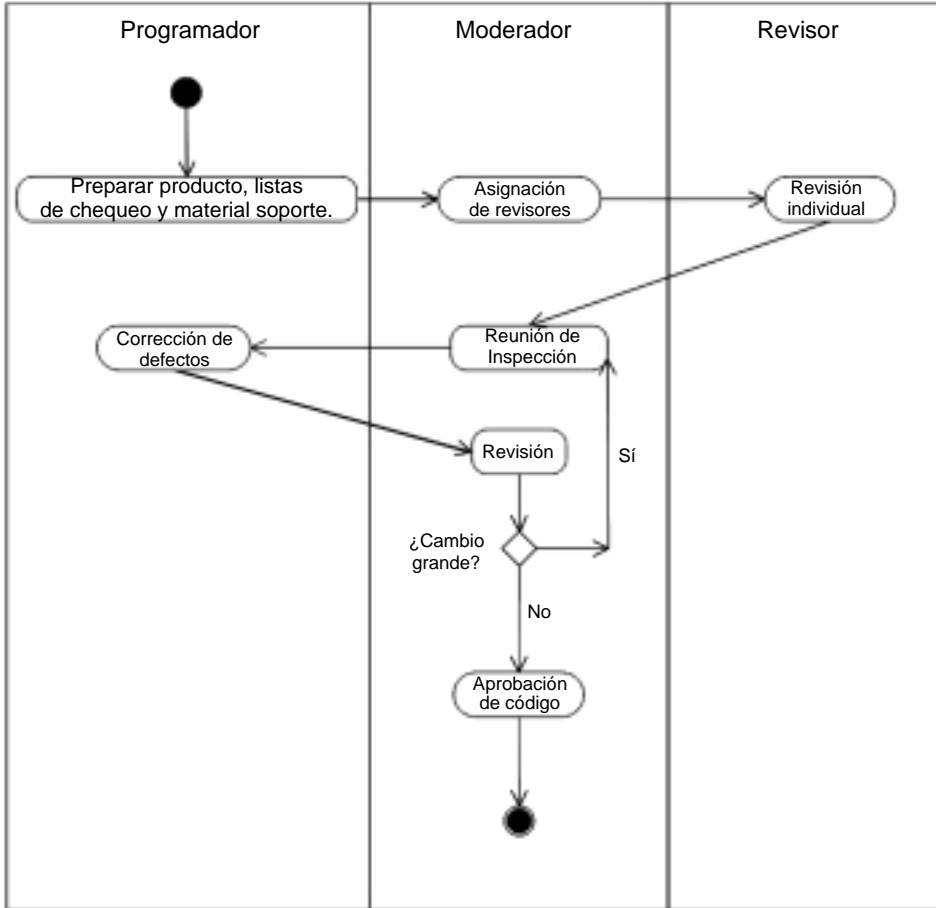


Figura 6. Proceso de inspecciones.

Tabla 5. Detalle del proceso de inspección

1. Elaborar documentos para inspección			
Detalle	Restricciones	Encargados	Recursos
<p>El analista prepara los siguientes elementos para la inspección:</p> <ul style="list-style-type: none"> — código fuente (líneas numeradas o identificación de cada sub-módulo). — material de soporte, como descripción de la funcionalidad. — lista de chequeo para inspecciones. 	<p>Formato de inspecciones- Registro de defectos (Ver Anexo 9).</p>	<p>Analista</p>	
2. Asignación de revisores			
<p>El moderador asigna dos revisores para este módulo, con experiencia en desarrollo de programas en el lenguaje en el cual se elaboró el módulo a evaluar o, si esto no es posible, con experiencia en programación en general.</p>	<p>Uno de los revisores debe de ser de un proyecto diferente al del producto inspeccionado.</p>	<p>Moderador</p>	
<p>Una vez asignados los revisores, se les entrega una copia del material que presentó el analista y el formato para consignar los resultados de la inspección.</p>		<p>Moderador</p>	
<p>Se establece una fecha de reunión y se indica esta fecha a los revisores y al analista.</p>		<p>Moderador</p>	
3. Revisión individual			
<p>Se realiza la inspección siguiendo la lista de chequeo y anotando los errores encontrados en el formato.</p>	<p>Formato de inspecciones. Registro de defectos.</p>	<p>Revisor Moderador</p>	

Tabla 5. Detalle del proceso de inspección

4. Reunión de Inspección			
Detalle	Restricciones	Encargados	Recursos
El moderador orienta esta reunión, haciendo que cada revisor (y él mismo) expongan brevemente los errores encontrados.		Moderador Revisor	
El analista aclara dudas o indica si hay un error de apreciación por parte de los revisores, pero no puede tratar de dar explicaciones sobre algún error o indicar formas de corregirlo.		Analista	
El moderador consigna en el formato los resultados consolidados de la inspección y los pasa al analista para que los corrija.		Moderador	
Excepciones			
Si el moderador es nuevo, se debe explicar en qué consiste el proceso de inspecciones.			

2.4. Pruebas de información no periódica

La Figura 7 muestra el diagrama del

proceso de pruebas de información no periódica. El detalle del proceso se encuentra en la Tabla 6.

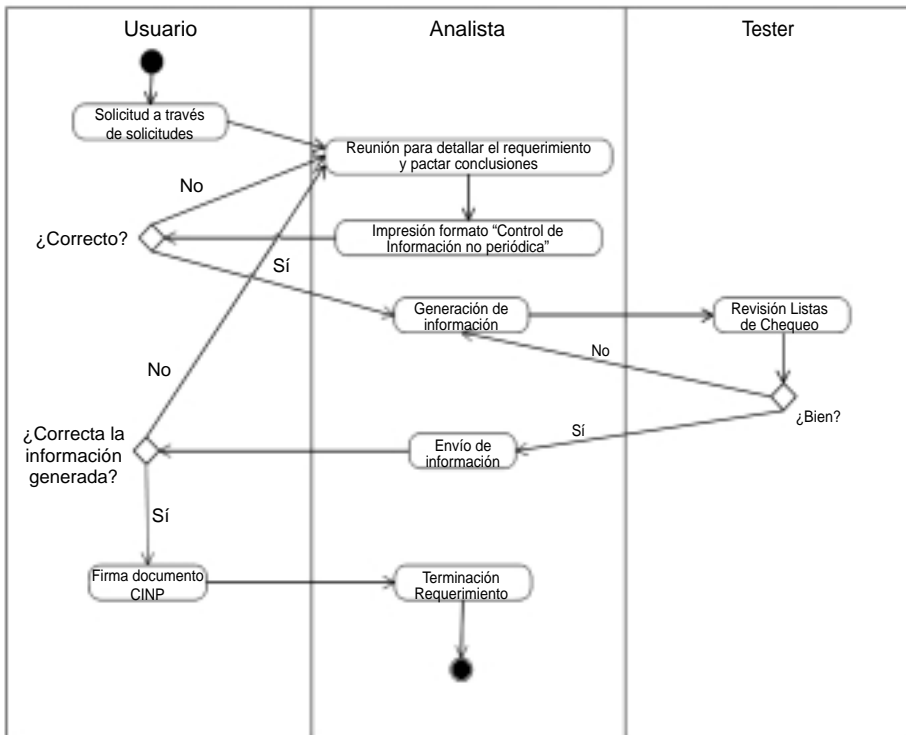


Figura 7. Proceso de pruebas de información no periódica.

Tabla 6. Detalle del proceso de pruebas para información no periódica

1. Pruebas e inspecciones para información no periódica			
Detalle	Restricciones	Encargados	Recursos
Ingreso de la solicitud a través del sistema de solicitudes.		Usuario	
Reunión para detallar el requerimiento y pactar conclusiones.		Analista Usuario	
Se imprime el formato de control de información no periódica: (*1).		Analista	
El usuario verifica la información en el formato.		Usuario	
Comprobado el formato, se genera la información pedida.		Analista	
Revisión de las listas de chequeo.(*2)		SQA	
Aprobada la información, se envía al usuario.		Analista	
El usuario corrobora la información generada.(*3)		Usuario	
Si la información es correcta, firma el documento CINP.		Usuario	
Finaliza el requerimiento.		Analista	

Excepciones (*)
1. Si el formato de control de información no periódica no es correcto, se vuelve a realizar la reunión del analista y el usuario para detallar el requerimiento.
2. Si la revisión de la lista de chequeo no es aprobada, el analista vuelve a generar la información.
3. Si el usuario no aprueba la información generada, se vuelve a realizar la reunión del analista y el usuario para detallar el requerimiento y el proceso continúa desde esa etapa.

3. AUTOEVALUACIÓN SOBRE EL PROCESO DE PRUEBAS

Las encuestas y evaluaciones son una herramienta de gran valor para medir la percepción y el conocimiento de las personas con respecto a algún tema en particular. Como parte de un proceso de concientización, evaluación y aprendizaje por parte de los programadores, se ha diseñado un

formato de autoevaluación que permite determinar, en alguna medida, el conocimiento que se tiene del proceso de pruebas. Los resultados de la autoevaluación sirven de retroalimentación dentro del proceso de pruebas de software, y son utilizados para direccionar las medidas que se tengan que tomar para solucionar los problemas identificados (Ver Tabla 7).

Tabla 7. Formato de autoevaluación sobre el proceso de pruebas

<i>Determine si las siguientes afirmaciones son verdaderas o falsas. En el caso de las falsas explique la razón</i>	
1.	Deben realizarse inspecciones para todos los módulos nuevos.
2.	Antes de codificar un módulo nuevo el usuario debe haber aprobado el diseño gráfico (I/O) del mismo.
3.	Al terminar un módulo, el analista debe revisar los estándares gráficos y consignar los resultados de esta revisión en el sistema.
4.	El analista, además de programar, está involucrado en las siguientes actividades de pruebas: revisiones de funcionalidad, revisiones de estándares gráficos y aprobación del diseño gráfico por parte del usuario.
5.	Cuando se modifica un módulo debe realizarse de nuevo la prueba de caja negra completa (incluir todos los casos de prueba).
<i>Complete las siguientes frases</i>	
1.	En un proceso de inspecciones, el papel de un analista (diferente a quien realizó el módulo a inspeccionar), es:
2.	Si otro analista le envía un nuevo módulo para que lo revise, usted debe realizar las siguientes actividades:
3.	Si al realizar pruebas de caja negra se encuentran errores, el analista debe recibir los siguientes documentos: Y los pasos a seguir son:
4.	Las técnicas de caja blanca que se usarán, de acuerdo con el proceso de pruebas del equipo, son:
5.	Para que la persona encargada de las pruebas pueda elaborar los casos de prueba usando la caja negra, el analista debe proporcionarle: Y usando caja blanca:

Table 7. Formato de autoevaluación sobre el proceso de pruebas

<i>Preguntas</i>	
1.	Explique algunas de las actividades que realiza una persona de SQA.
2.	Si usted es nombrado revisor en un proceso de inspección, ¿dónde debe buscar información sobre este proceso? Si se requiere llenar un formato, ¿dónde debe buscarlo?
3.	¿Quién define cuándo un módulo es crítico? ¿Por qué es necesario saber si un módulo es crítico?
4.	¿Cómo puede usted contribuir a mejorar este proceso?
<i>Repaso final</i>	
1.	Exponga, con un ejemplo, un caso donde no es necesario seguir todo el proceso de pruebas y explique por qué no lo considera conveniente.
2.	Exponga, con un ejemplo, un caso de un módulo que se deba considerar como crítico.
3.	Explique, con sus propias palabras, todo lo que tiene que hacer (relacionado con las pruebas) cuando se va a desarrollar un módulo nuevo.
4.	Explique qué cambia con respecto al punto anterior cuando se va a realizar una modificación a un módulo existente.

ANEXOS

Anexo 1

Lista de chequeo de aseguramiento de calidad

Fecha:

Analista:

Revisión de aseguramiento de calidad

Actividad	Sí	No	No aplica	Información adicional
¿Existe alguien en su organización responsable por los procesos de pruebas?				
¿Tiene y usa un estándar para el plan de pruebas?				
¿Tiene y usa un estándar para las pruebas de unidad?				
¿Tiene y usa un estándar para el reporte de la ejecución de las pruebas?				
¿La planeación y ejecución de pruebas se realiza en paralelo con el proceso de desarrollo de software?				
¿Se verifica que las especificaciones estén correctamente implementadas?				
¿Se verifica que las expectativas del cliente sean satisfechas?				
¿Los probadores verifican la precisión y completitud de productos internos tales como el documento de requerimientos o los diseños?				
¿Los probadores reportan los defectos al equipo de desarrollo de software para corrección?				
¿Los probadores identifican la prioridad de los riesgos del negocio para el desarrollo del plan de pruebas?				
¿Existen objetivos de pruebas medibles para cada sistema de software que está siendo probado?				
¿Los objetivos están alineados con los riesgos del negocio?				

Revisión de aseguramiento de calidad				
Actividad	Sí	No	No aplica	Información adicional
¿Se usan métricas para mejorar el proceso de aseguramiento de la calidad?				
¿Los probadores han definido pronósticos de defectos basándose en datos y experiencias previos?				
¿Existe un proceso de mejoramiento continuo para su proceso de pruebas?				
¿Los tipos de defectos están identificados?				
¿Se registra, acumula y se usan los datos de fallas para evaluar la efectividad del proceso de pruebas y producir un software libre de defectos?				
¿Se usan métricas para planear y evaluar el proceso de pruebas?				
¿Tiene un proceso de entrenamiento de probadores?				
¿El uso de una herramienta automatizada de pruebas es parte significativa de su proceso?				

Anexo 2

Lista de chequeo de estándares de presentación y funcionalidad de la aplicación para formas

Fecha: _____

Forma: _____ **Descripción:** _____

Analista: _____ **Revisor:** _____

Revisión de estándares de presentación

Actividad	Sí	No	No aplica	Información adicional
¿Están claramente definidos los bloques de información (Frames)?				
¿Tiene los encabezados de título y nombre de aplicación correctos?				
¿Las etiquetas de los campos son claras y representativas?				
¿Los campos de despliegue están completamente inhabilitados y del color respectivo?				
¿Los campos de solamente despliegue están claramente identificados?				
¿Tiene los colores estándar?				
¿Los campos fecha tienen el formato DD-MON-RRRR y se puede ingresar los datos como Ej: 12ago2001?				
Cuando se tiene una forma con múltiples tabs, ¿se conoce cuál es el registro padre de los tabs?				
¿La forma tiene la dimensión correcta?				
¿Los Radio Groups tienen un frame que los abarca?				
¿Los campos están alineados en forma correcta?				
¿Los campos requieren y tienen Tooltip?				
¿Los LOVs tienen el tamaño y la posición adecuados (que no requieran ser movidos)?				
¿Los LOV's están heredados?				

Revisión de aseguramiento de calidad				
Actividad	Sí	No	No aplica	Información adicional
¿Los barras de Scroll son blancas y de ancho 15?				
¿Están los RadioButtons azules y heredan de la propiedad VA_RadioButtons?				
¿Están habilitados los botones del toolbar de manera adecuada y corresponden con las teclas de función?				

Revisión de funcionalidad				
Actividad	Sí	No	No aplica	Información adicional
¿La forma realiza la función que se necesita?				
¿La forma ha sido ingresada a SIABD con todas las funciones, tablas y roles asociados?				
¿Los datos de la forma cambian en forma sincronizada?				
¿Es rápido y fácil el manejo de la forma?				
Cuando se cambia el valor de un campo de entrada, ¿se modifica también el campo de despliegue?				
¿Los bloques hijos están coordinados con el bloque padre en consulta, borrado y cuando se limpia la forma?				
Los campos que hacen referencia a datos de otras tablas ¿tienen cada uno su lista de valores?				
¿Las listas de valores son lentas para recuperar la información?				
¿El tiempo de respuesta es adecuado?				

Revisión de funcionalidad				
Actividad	Sí	No	No aplica	Información adicional
¿El orden de navegación de los campos es el correcto?				
¿Los mensajes graves son manejados adecuadamente?				
¿Los campos Validate from LOV funcionan adecuadamente?				
¿Si el reporte requiere mucho tiempo, esto le es notificado al usuario?				
¿Está la forma documentada?				
¿Si llama reportes, la extensión de los reportes es la correcta? (NO rdf, debe estar sin extensión).				

Revisión del código y los datos que retorna				
Actividad	Sí	No	No aplica	Información adicional
¿Se ha realizado el proceso de prueba formal?				
¿Está la mayor cantidad de código en la base de datos?				
¿Se ha realizado el proceso de afinamiento Sql?				

Comentarios adicionales

Anexo 3

Lista de chequeo de estándares de presentación y funcionalidad de la aplicación para reportes

Fecha: _____

Reporte: _____ **Descripción:** _____

Analista: _____ **Revisor:** _____

Revisión de estándares de presentación

Actividad	Sí	No	No aplica	Información adicional
¿El reporte tiene el nombre del sistema correcto?				
¿El reporte tiene los encabezados de título y nombre de aplicación correctos?				
¿El reporte tiene la fecha de generación?				
¿El reporte tiene el número de página y el total de páginas?				
¿Están claramente definidos los bloques de información?				
¿Las etiquetas de los campos son claras y representativas?				
¿El reporte tiene los colores estándares? Negro y tonos de grises.				
¿Los campos fecha tienen el formato DD-MON-YYYY?				
¿Los campos están alineados en forma correcta?				
¿Se ha utilizado la indentación para mejorar la legibilidad del reporte?				
¿El reporte tiene enumeradas las filas?				
¿El reporte tiene subtotales y totales de control?				
¿El reporte tiene, en la parte superior, las condiciones de generación del listado?				
¿El reporte tiene el visto bueno del usuario?				

Revisión del Código				
Actividad	Sí	No	No aplica	Información adicional
¿Se ha hecho revisión por pares?				
¿Se ha realizado el proceso de afinamiento sql?				
¿Está la mayor cantidad de código en la base de datos?				
¿El código cumple con los estándares?				
¿Está el reporte registrado en SIABD?				

Anexo 4

Lista de chequeo de estándares de tablas

Fecha: _____

Tabla: _____ **Descripción:** _____

Analista: _____

Estándares de las tablas

Actividad	Sí	No	No aplica	Información adicional
¿El nombre de la tabla es correcto según los estándares?				
¿Tiene las descripciones de la columna en la base de datos?				
¿Tiene las llaves e índices adecuados?				
¿La tabla ha sido recreada teniendo en cuenta su uso?				

Anexo 5

Lista de chequeo de estándares de funciones y procedimientos

Fecha: _____

Func/Proc: _____ **Descripción:** _____

Analista: _____ **Revisor:** _____

Revisión de estándares

Actividad	Sí	No	No aplica	Información adicional
<p>¿El nombre cumple con los estándares?</p> <p>¿El código cumple con los estándares?</p> <p>¿Está la función/procedimiento documentado?</p> <p>¿Se ha realizado el proceso de afinamiento sql?</p> <p>¿Se ha registrado en SIABD?</p> <p>¿Se usan todas las variables, constantes y parámetros?</p> <p>¿La asignación de valores a las variables, constantes y parámetros tiene un propósito?</p> <p>¿Son correctas las validaciones de condiciones?</p> <p>Por ejemplo: código no alcanzable, ciclos infinitos, división por cero, verificación de rangos, redondeos.</p> <p>¿Faltan validaciones?</p> <p>¿Se manejan todas las posibles excepciones?</p> <p>¿Las variables que guardan datos de columnas de tablas se han definido de acuerdo con esto?</p> <p>Tabla.columna%type</p> <p>Si se llaman otras funciones y/o procedimientos, ¿tienen el número de parámetros y el tipo de datos adecuado?</p>				

Anexo 6

Lista de chequeo de estándares de programación-Código

Objeto			Sí	No
Fecha de revisión				
Revisado por		Aprobado		

Elemento a revisar	Sí	No	No aplica	Información adicional
Código en general				
¿Está el código indentado a, por lo menos dos espacios?				
¿Están ordenadas alfabéticamente las constantes, variables y cursores?				
¿Están alineados a la izquierda las constantes, variables y cursores?				
¿Están alineados a la izquierda la definición del tipo de dato de las constantes, variables y cursores?				
¿Está definida sola una constante, variable o cursor por línea?				
Documentación				
¿Está toda la documentación en una línea diferente al código que se está documentando?				
¿Comprende la documentación de funciones/procedimiento tres partes: una descripción general de lo que hace la función o procedimiento, la descripción de los parámetros de entrada y la descripción de los posibles valores y/o parámetros de salida?				
Parámetros				
¿El nombre de los parámetros empieza con la letra <i>p</i> minúscula y es significativo?				
Constantes				
¿El nombre de las constantes empieza con la letra <i>c</i> minúscula y es significativo?				

Elemento a revisar	Sí	No	No aplica	Información adicional
<p>Variables</p> <p>¿El nombre de las variables empieza con la letra <i>v</i> minúscula y es significativo?</p> <p>Cursores</p> <p>¿El nombre de los cursores empieza con las letras <i>cur</i> minúsculas y es significativo?</p> <p>¿Están los nombres de los cursores alineados a la izquierda junto con la definición del tipo de dato de las constantes y variables?</p> <p>Instrucciones Select, Insert, Update y Delete</p> <p>¿Están todas las instrucciones Select, Insert, Update y Delete escritas en minúsculas, a excepción de variables que hagan referencia a campos de las formas?</p> <p>Instrucciones Select</p> <p>¿Están las cláusulas Select, Into, From, Where, Order BY, Group BY y Having escritas en líneas diferentes?</p> <p>Instrucciones Insert</p> <p>¿Están las cláusulas Insert Into y Values escritas en líneas diferentes?</p> <p>Instrucciones Update</p> <p>¿Están las cláusulas Update, SET y Where escritas en líneas diferentes?</p> <p>¿Está cada columna que se actualice en una línea diferente?</p> <p>¿Están todas las columnas que se actualicen alineadas a la izquierda?</p> <p>Instrucciones Delete</p> <p>¿Están las cláusulas Delete y Where escritas en líneas diferentes?</p>				

Anexo 7

Lista de chequeo de estándares de presentación - Formas

Forma			Sí	No
Fecha de revisión				
Revisado por			Aprobado	

Elemento a revisar	Sí	No	No aplica	Información adicional
<p>Forma</p> <p>¿Tiene la forma la descripción y su título de acuerdo con los estándares?</p> <p>¿Tiene la forma la dimensión correcta?</p> <p>Cuando se tiene una forma con múltiples tabs, ¿se conoce cuál es el registro padre de los tabs?</p> <p>Título del frame</p> <p>¿Está el título en mayúscula inicial?</p> <p>Si el frame es de un solo registro, ¿está su título en singular?</p> <p>Si el frame es multirregistro, ¿está su título en plural?</p> <p>¿Está localizado el título en la parte superior izquierda del frame?</p> <p>Campos</p> <p>¿Tiene el contenido del campo la alineación adecuada, de acuerdo con su tipo de dato?</p> <p>Si existen varios campos organizados verticalmente, ¿están alineados todos a la izquierda?</p> <p>Etiquetas</p> <p>Si la organización NO es tabular, ¿están situadas las etiquetas a la izquierda del campo al que pertenecen?</p> <p>Si la organización SÍ es tabular, ¿están situadas las etiquetas en la parte superior del campo al que pertenecen?</p>				

Elemento a revisar	Sí	No	No aplica	Información adicional
<p>Etiquetas</p> <p>Si la organización SÍ es tabular, ¿están las etiquetas centradas?</p> <p>¿Están las etiquetas en mayúscula inicial?</p> <p>¿Están las etiquetas sin los dos puntos al final?</p> <p>Si existen varios campos organizados verticalmente, ¿están alineadas todas las etiquetas a la derecha?</p> <p>¿Están las etiquetas formadas de manera que no utilicen abreviaturas ni expresiones de solicitud?</p> <p>Radio Buttons y Check Box</p> <p>¿Emplean mayúscula inicial?</p> <p>En cuanto a su estructura, ¿emplean orientación en forma de columna?</p> <p>En cuanto a su estructura, ¿emplean alineamiento a la izquierda?</p> <p>¿Están organizadas las opciones en el orden esperado, de mayor a menor frecuencia de ocurrencia?</p> <p>¿Están enmarcados dentro de un frame?</p> <p>Listas de valores</p> <p>¿Están organizados los descriptores alineados a la izquierda en forma de columna?</p> <p>¿Están organizados los descriptores en orden alfabético o numérico, según sea el caso?</p> <p>¿Están los descriptores en mayúscula inicial?</p> <p>¿El ancho es suficiente para evitar el uso de scroll horizontal?</p> <p>¿Tienen las listas de valores la posición adecuada, de forma que no requieran ser movidas?</p>				

Elemento a revisar	Sí	No	No aplica	Información adicional
<p>Scroll</p> <p>¿Están las barras de scroll vertical ubicadas a la derecha?</p> <p>¿Están las barras de scroll vertical iguales a la altura de sus campos asociados?</p> <p>¿Están las barras de scroll horizontal ubicadas en la parte inferior?</p> <p>¿Están las barras de scroll horizontal iguales al ancho de sus campos asociados?</p> <p>¿Son las barras de scroll blancas?</p> <p>Botones</p> <p>Si están ubicados horizontalmente, ¿están en la parte inferior de la pantalla?</p> <p>Si están ubicados verticalmente, ¿están a la derecha de la pantalla?</p> <p>Los botones organizados horizontalmente, ¿tienen la misma altura?</p> <p>Los botones organizados verticalmente, ¿tienen el mismo ancho?</p> <p>¿Está colocada la opción más frecuente a la izquierda o en el tope, según corresponda?</p> <p>¿Usan los botones mayúscula inicial?</p> <p>¿Incluye puntos suspensivos (...) si la acción despliega otra ventana?</p>				

Anexo 8

Lista de chequeo de programación - Formas

Forma			Sí	No
Fecha de revisión				
Revisado por		Aprobado		

Elemento a revisar	Sí	No	No aplica	Información adicional
Nombres de los objetos				
¿Cumplen los siguientes objetos con los estándares?				
Alert				
Bloques				
Canvas				
Forma o reporte				
Funciones y/o procedimientos				
Gráficos				
Librerías				
Listas de valores				
Object Group				
Parámetros				
Push Button				
Radio Group				
RecordGroup				
Relación				
Variables				
Atributos visuales				
Ventanas				
Título del frame				
¿Hereda el título del frame el atributo visual VA_TITULO?				

Elemento a revisar	Sí	No	No aplica	Información adicional
<p>Campos</p> <p>¿Hereda el campo el atributo visual correspondiente?</p> <p>¿Hereda el prompt del campo el atributo visual VA_ETIQUETA?</p> <p>Si el campo pertenece a un bloque multirregistros, ¿hereda el atributo visual VA_CURRENTRECORD?</p> <p>Si el campo es tipo date, ¿tiene el formato DD-MON-RRRR?</p> <p>Si el campo es numérico y representa dinero, ¿lleva delante de él el signo de pesos (\$)?</p> <p>Si el campo indica hora, ¿tiene el formato HH24:MI (hora militar)?</p> <p>Si el campo es un porcentaje, ¿está ubicado el símbolo “%” después del número?</p> <p>¿Heredan los radio_buttons el atributo Visual VA_RADIO_BUTTON?</p> <p>¿Heredan las listas de valores el atributo visual VA_LOV?</p> <p>Scroll</p> <p>¿Tienen las barras de scroll un ancho de 15 puntos?</p> <p>Canvas</p> <p>¿Heredan los canvas el atributo visual VA_CANVAS?</p>				

Anexo 9

Formato de registro de defectos - Inspecciones

Objeto: _____ **Revisor:** _____

Fecha: _____ **Inspección No.** _____

Tabla de tipos estándar de defectos

Código	Nombre	Descripción
10	Documentación	Comentarios, mensajes
20	Sintaxis	Ortografía de los comandos, puntuación, errores de tecleo, formato de las instrucciones
30	Manejo de versiones	Manejo de cambios, librerías, control de versiones
40	Asignación	Declaraciones, identificadores duplicados, alcance y límites de los mismos
50	Interfaces	Llamadas y referencias a procedimientos, I/O, interfaz de usuario
60	Validación	Mensajes de error, validaciones incorrectas
70	Datos	Estructuras, contenidos, inicializaciones
80	Funciones	Defectos de lógica, puntero, ciclos, recursividad, cálculo y funcionamiento
90	Sistema	Configuración, memoria, tiempo de respuesta
100	Entorno	Problemas de diseño, compilación, pruebas del ambiente de desarrollo

Listado de defectos encontrados

Código Defecto	Localización	Descripción del defecto encontrado

Comentarios generales

Anexo 10

Formato de pruebas de caja negra - Partición equivalente

< > - Tabla de particiones		
Campos	Clases válidas	Clases no válidas

< > - Casos de Prueba

1. .	23. .
2. .	24. .
3. .	25. .
4. .	26. .
5. .	27. .
6. .	28. .
7. .	29. .
8. .	30. .
9. .	31. .
10. .	32. .
11. .	33. .
12. .	34. .
13. .	35. .
14. .	36. .
15. .	37. .
16. .	38. .
17. .	39. .
18. .	40. .
19. .	41. .
20. .	42. .
21. .	43. .
22. .	44. .

Árbol de clases equivalentes

Anexo 11

Formato de resultados, ejecución de pruebas

Forma:

Fecha:

No. de ejecución:

Aprobado:

Sí

No

Ejecutor:

Casos:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

Errores:

Caso:		
Observaciones:	Entradas:	Salidas:

Caso:		
Observaciones:	Entradas:	Salidas:

Caso:		
Observaciones:	Entradas:	Salidas:

Caso:		
Observaciones:	Entradas:	Salidas:

Anexo 12

Formato de casos de pruebas

Tipo de prueba:

Objeto:

Complejidad:

Descripción:

Caso No. 1

Descripción

Entradas

Salidas esperadas

Caso No. 2

Descripción

Entradas

Salidas esperadas

Anexo 13

Formato de pruebas de funcionalidad

Forma:

Analista:

Sistema:

Ejecución:	1	2	3	4	5
Aprobado:					
Tiene scripts:	Sí		No		

Diseño:

Ejecución:

Descripción/Observaciones/Reportes

Pantallas:

CONCLUSIONES

- Las acciones correspondientes a la calidad de software que se estaban llevando a cabo eran acciones aisladas no contempladas en un plan formal. Esto se daba, principalmente, por el tamaño del departamento de Desarrollo de Sistemas, pues no se trata de una compañía desarrolladora de software.
- El aseguramiento de calidad de software se podía trabajar, dentro de cada una de las actividades de desarrollo, de la siguiente manera:
 - Realizando una revisión por pares para cada proyecto.
 - Haciendo seguimiento formal a los proyectos, no sólo en tiempos y costos, sino en cumplimiento con estándares y metodología de desarrollo de software.
 - Consolidando la metodología de desarrollo de software en busca de alcanzar niveles superiores de madurez.
 - Involucrando puntos de chequeo y pruebas de software.
 - Desarrollando estándares muy concretos para programación e interfase de usuario.
- Los objetivos básicos de las inspecciones son:
 - Encontrar errores lo más temprano posible en el ciclo de desarrollo.
 - Asegurar que todos los participantes están de acuerdo en la parte técnica del trabajo.
 - Verificar que el trabajo cumple con los criterios preestablecidos.
 - Completar formalmente una tarea técnica.
 - Suministrar información sobre el producto y el proceso de inspección.
- Como beneficios secundarios de las inspecciones se destacan:
 - Asegurar que las personas asociadas estén familiarizadas técnicamente con el producto.
 - Ayudar a crear un equipo técnico efectivo.
 - Ayudar a utilizar los mejores talentos de la organización.
 - Ayudar a los participantes a desarrollar sus habilidades como revisores.
- Vale la pena hacer una distinción entre los diferentes tipos de revisiones que se pueden llevar a cabo durante el proceso de desarrollo, teniendo en cuenta que las inspecciones son tan sólo un tipo:
 - Revisiones gerenciales: Buscan asegurar el progreso, recomendar acciones correctivas y asegurar el suministro adecuado de recursos.
 - Revisiones técnicas: Evalúan el cumplimiento de especificaciones y planes y aseguran la integridad de los cambios.
 - Inspecciones de software: Buscan detectar e identificar defectos y verificar resoluciones.
 - “Walkthrough”: Buscan detectar defectos, examinar alternativas y ser un foro para el aprendizaje.

Todas las revisiones anteriores se pueden combinar durante las diversas etapas del proyecto. La Tabla 8 muestra un ejemplo de ello.

Tabla 8. Ejemplo de revisiones durante las etapas del proyecto

Etapa	Inspecciones	Walkthroughs	Revisiones técnicas
Requerimientos		Requerimientos detallados	Requerimientos iniciales
Planeación			Planes de desarrollo
Desarrollo	Diseño detallado Codificación	Diseño del sistema Diseño de alto nivel	
Publicaciones		Publicaciones en borrador Publicaciones finales	
Pruebas		Implementación de pruebas	

- Para toda prueba debe haber un plan que incluye:
 - Objetivos para cada fase de prueba.
 - Cronograma y responsabilidades para cada actividad de prueba.
 - Disponibilidad de herramientas, documentación y librerías de prueba.
 - Procedimientos y estándares a ser utilizados para planear y llevar a cabo las pruebas y reportar los resultados.
 - Criterios para determinar si la prueba está completa, como también el éxito de cada prueba.
- Después de tener el plan de pruebas, se deben generar los casos de prueba, siguiendo cualquier técnica de prueba.
- Cada caso de prueba se debe ejecutar y al final debe quedar un reporte de las pruebas con la siguiente información:
 - Proyecto y programas que se están probando, objetivo de la prueba y el plan de pruebas.
 - Responsables y participantes de las pruebas.
 - Casos de prueba.
 - Herramientas especiales utilizadas.
 - Configuraciones de hardware y software utilizadas.
 - Resultados de las pruebas.
 - Identificación de los elementos que quedan en la librería de pruebas.
 - Firma de los responsables de las pruebas y certificación de

que se siguió el procedimiento apropiado.

- Como parte del reporte del proceso de pruebas, es deseable clasificar los errores encontrados, con el fin de tomar correctivos en el proceso de prueba o retroalimentación para los desarrolladores.
- Las pruebas deben ser analizadas teniendo en cuenta:
 - Los errores graves encontrados deben ser analizados en grupo, para hallar soluciones y maneras de que no vuelvan a ocurrir.
 - Analizar el tipo de error más frecuente y revisar qué ocurren y cómo se pueden mejorar las inspecciones.
 - Revisar la efectividad de las pruebas y reforzar aquellas que más errores detectan.
- Los métodos y herramientas de prueba a emplear pueden ser cualquiera, siempre y cuando se utilicen dentro de un plan de pruebas.
- Uno de los beneficios de las autoevaluaciones es que los analis-

tas/programadores pueden sugerir mejoras dentro de todo el proceso de desarrollo de software, para así incrementar su calidad y productividad en el trabajo.

BIBLIOGRAFÍA

- *The Capability Maturity Model*. Carnegie Mellon University - Software Engineering Institute. Addison-Wesley. 1994
- <http://www.sei.cmu.edu/cmm/cmms/cmms.html>
- <http://www.sei.cmu.edu/cmml/>

CURRÍCULO

Oscar Hernando Guzmán Cortés.

Ingeniero de Sistemas de la Universidad Icesi, grado *cum laude*. Especialista en gerencia de Informática Organizacional de la Universidad Icesi. Analista senior del departamento de Desarrollo de Sistemas de la Universidad Icesi. Profesor de la Universidad Icesi, en los cursos de Estructuras de datos y Bases de datos. ☀