

Universidad de las Ciencias Informáticas.

Facultad 3



Título: Guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico.

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

Autora: Yeleny Almora Galvez

*Tutores: Ing. Yanet Edghill Martínez
Ing. Raúl Velázquez Álvarez*

*La Habana
Junio de 2012*



“ La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de las mejores y la mejor posible...”

Ernesto Guevara de la Serna

Dedicatoria

A mima, espero que donde quiera que estés te sientas muy orgullosa de mi, gracias por haber estado siempre ahí para mí.

A mi mamá por haber confiado siempre en mí.

A mis sobrinos, espero servirles de ejemplo.

A todos los que soñaron conmigo y confiaron en que este sueño podía convertirse en realidad.

Agradecimientos

A quien siempre ha confiado en mí y me ha dado su apoyo en todo momento: mi mamá.

A quien supo ser como la hermana que siempre quise tener a mi lado: Nohe.

A mi tía por quererme tanto y apoyarme siempre: Gloria.

A mi papá del corazón: Tío Pelly.

A mi familia de la UCI, los que nunca voy a olvidar, los amigos que todos desearían, y yo tengo la suerte de tener: Yosvany, Aliesky, Dasieki, Yelien, Yadier.

A mi amiga de toda la vida: Saira.

A la persona que me demostró que los buenos amigos no sólo se encuentran en la universidad, sino desde que tienes la oportunidad de que se crucen en tu vida: Anay.

A mi asesora de la tesis, que siempre me brindó su apoyo incondicional: Aliuska.

A mi compañera de equipo con la que compartí maravillosos momentos en el terreno de voleibol: Mónica.

A mis compañeras de cuarto en nuestro último año: Beylen, Daneisy, Anita.

A mi eterna profesora: Teresa.

A mi prima por estar siempre ahí para sentirme como en casa: Lourdes.

A mi compañera de tesis, que siempre lo será a pesar de no haber discutido juntas: Mirbel.

A los que me brindaron su apoyo en la realización de este trabajo de diploma, especialmente a Manuel.

A quien cada día cuidó de mi mamá mientras yo estuve en la universidad: Rey.

A quien siempre estuvo dispuesta a viajar y ayudar en todo lo que hiciera falta: Bertha.

A la persona que me dio todo su apoyo y ayuda, no hubiera sido posible la realización de este trabajo sin ella: mi tutora Yanet.

A mi tutor: Raúl.

A mi papá.

A mi familia por apoyarme siempre.

A todos los que menciono y los que no, pero que de alguna manera contribuyeron a realizar este sueño: ¡GRACIAS!

Declaración de Autoría

Declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste se firma la presente a los ____ días del mes _____ del año 20__.

Yeleny Almora Galvez

Autora

Ing. Yanet Edghill Martínez

Tutor

Ing. Raúl Velázquez Álvarez

Tutor

Resumen

La calidad es un tema que cada día cobra más fuerza en el mundo del software, pues la competencia en el mercado obliga a los productores a desarrollar productos con mayor calidad para brindarles a sus clientes. El aseguramiento de la calidad y específicamente de sus características, donde se encuentra la funcionalidad, es el reto que tiene el Centro de Gobierno Electrónico (CEGEL) de la facultad 3.

En el presente trabajo se propone una guía para el aseguramiento de la funcionalidad como característica de la calidad del software en los sistemas informáticos de Gobierno Electrónico. Para la elaboración de la guía se tuvo en cuenta los resultados arrojados en la entrevista aplicada a miembros de los proyectos de CEGEL que se desempeñan en diferentes roles que intervienen en la funcionalidad del software.

Teniendo como sustento de la guía el estudio de diferentes conceptos como calidad, modelos/estándares de calidad del software respectivamente y la funcionalidad como característica de la calidad, se determinaron los elementos fundamentales a tener en cuenta.

En dicha guía se propone un conjunto de actividades para cada etapa del ciclo de vida del proceso de mejora. La misma ha sido evaluada satisfactoriamente, lo cual permite su aplicación en el CEGEL para contribuir al aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico.

Palabras claves: aseguramiento, calidad, funcionalidad, guía.

Índice

<i>Introducción</i>	1
<i>Capítulo 1. Fundamentación teórica</i>	6
1.1 <i>Introducción al capítulo</i>	6
1.2 <i>Calidad del software a nivel de producto</i>	6
1.3 <i>Funcionalidad en la calidad de software</i>	8
1.4 <i>Aseguramiento de la funcionalidad en la calidad de software</i>	10
1.5 <i>Modelos/Estándares de calidad del software que refieren entre sus atributos a la funcionalidad</i> . 11	
1.5.1 <i>Modelo FURPS (Funcionalidad, Usabilidad, Confiabilidad, Rendimiento, Soporte)</i>	11
1.5.2 <i>Modelo Dromey</i>	12
1.5.3 <i>Estándar ISO 9126</i>	14
1.5.4 <i>Modelo de Calidad WEB (WQM)</i>	16
1.5.5 <i>Calidad en Tecnología de la Información (QUINT2)</i>	17
1.5.6 <i>Estándar ISO/IEC 25010 SQuaRE</i>	19
1.5.7 <i>Análisis Comparativo</i>	21
1.6 <i>Modelo Integrado de Capacidad y Madurez (CMMI)</i>	23
1.7 <i>¿Procedimiento, lista de chequeo, estrategia o guía?</i>	25
1.7.1 <i>Análisis comparativo</i>	26
1.8 <i>Sistemas Informáticos de Gobierno Electrónico</i>	27
1.8.1 <i>Sistemas Informáticos</i>	27
1.8.2 <i>Sistemas Informáticos de Gobierno Electrónico</i>	28
1.9 <i>Conclusiones parciales</i>	29
<i>Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico</i>	30
2.1 <i>Introducción al capítulo</i>	30
2.2 <i>Características de los proyectos del centro CEGEL</i>	30

Índice

2.3	<i>Ciclo de vida del proceso de desarrollo de software basado en el programa de mejora. Elementos definidos por etapas.</i>	30
2.3.1	<i>Estudio Preliminar</i>	31
2.3.3	<i>Requisitos</i>	32
2.3.4	<i>Análisis y Diseño</i>	32
2.3.5	<i>Implementación</i>	32
2.3.6	<i>Pruebas Internas</i>	33
2.3.7	<i>Pruebas de Liberación</i>	33
2.3.8	<i>Despliegue</i>	33
2.3.9	<i>Soporte</i>	33
2.4	<i>Guía de actividades a realizar por etapas.</i>	34
2.5	<i>Conclusiones parciales</i>	47
	<i>Capítulo 3. Validación de la propuesta de solución</i>	48
3.1	<i>Introducción al capítulo</i>	48
3.2	<i>Método Delphi</i>	48
3.3	<i>Conclusiones parciales.</i>	56
	<i>Conclusiones</i>	57
	<i>Recomendaciones</i>	58
	<i>Bibliografía</i>	59
	<i>Anexos</i>	62

Introducción

Introducción

“La calidad de un producto la determina el proceso usado para desarrollarlo”, de esta manera se expresa Watts Humphrey, que es considerado el padre de la calidad de los procesos del software, al referirse a la calidad del producto. Lo cierto es que con el paso acelerado que muestran hoy las industrias se requiere de una estrategia que permita competir con éxito en la industria del software.

Las empresas dedicadas al desarrollo del software tienen como reto crear productos que sean capaces de integrar costo, tiempo y satisfacción del cliente para el buen desarrollo de un proyecto de software. Con el progreso vertiginoso de las tecnologías, los clientes exigen productos que cumplan con sus expectativas, por lo que se debe realizar un producto con la mejor calidad posible.

En el pasado siglo la calidad era vista como un problema que se solucionaba mediante herramientas de inspección, pero el quehacer diario y la experiencia ha demostrado que no es una actividad inspectora, sino preventiva. La calidad del producto se encuentra fuertemente asociada a la portabilidad, confiabilidad, usabilidad, funcionalidad y mantenibilidad, por lo que todas estas características favorecen el buen resultado del producto final.

La funcionalidad como aspecto primordial cuando se quiere realizar un producto de software con calidad, evalúa si el software es capaz de proveer un conjunto de funciones apropiadas según tareas y objetivos específicos del cliente. Para contribuir a este aspecto tan indispensable a la hora de hacer un software a nivel mundial se han desarrollado diferentes investigaciones con vista a contribuir a la funcionalidad del mismo. Normas como la ISO / IEC 25010 SQuaRE, ISO 9126 y la guía aprobada en Perú en el año 2004 “Guía Técnica sobre Evaluación de Software en la Administración Pública”, son algunos ejemplos del trabajo que se lleva a cabo para contribuir a la calidad del producto y principalmente de su funcionalidad.

Cuba sigue este progreso que muestran hoy las tecnologías a nivel mundial, por lo que llevó a cabo el proyecto de la Universidad de las Ciencias Informáticas (UCI). La misma desarrolla productos para el país y el mundo que tengan presente las características de la calidad, fundamentalmente el cumplimiento de la funcionalidad. Para dar respuesta a los clientes se agrupan a estudiantes y profesores en centros de desarrollo que son los encargados mediante soluciones informáticas de satisfacer las necesidades de los mismos.

Introducción

La facultad 3 de la UCI está constituida por dos centros de desarrollo de software: Centro de Gobierno Electrónico (CEGEL) y Centro de Informatización de la Gestión de Entidades (CEIGE) que encaminan sus producciones a satisfacer las necesidades de los clientes en los temas de informática jurídica y gestión gubernamental. Se realizó una entrevista (ver [Anexo 1](#)) a 21 integrantes de CEGEL que se desempeñan en diferentes roles (líder de proyecto, arquitectos, analistas y diseñadores) en los proyectos que pertenecen al centro, la cual arrojó los siguientes resultados donde se aprecia cómo se asegura la funcionalidad en dicho centro. La siguiente gráfica muestra el porcentaje de los entrevistados en cuanto a los aspectos evaluados:

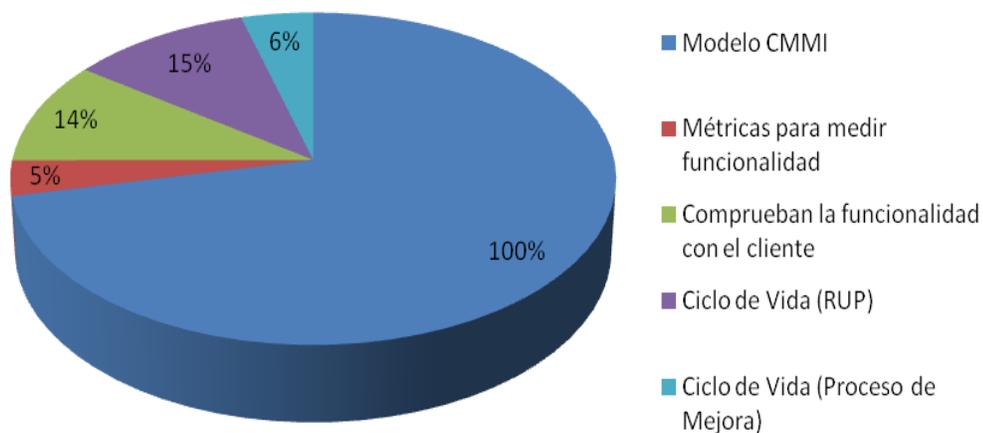


Figura 1. Entrevista de aseguramiento de la funcionalidad.

Los productos del CEGEL son entregados al grupo de calidad del centro y posteriormente al Centro para la Calidad de Soluciones Informáticas (CALISOFT), los cuales evalúan la funcionalidad a través de pruebas de caja negra. Las no conformidades (NC) detectadas en los 12 proyectos que se llevan a cabo en la facultad 3 permitieron detectar los diferentes problemas que afectan el aseguramiento de la funcionalidad del software en dichos proyectos:

- ✓ No se implementan en su totalidad las funcionalidades requeridas por el cliente.
- ✓ Existen errores en estructuras de datos o en accesos a las bases datos externas.
- ✓ Presentan errores de rendimiento.
- ✓ Errores de interfaz.
- ✓ Se implementan funcionalidades de forma incorrecta.
- ✓ Errores de inicialización y terminación.

Introducción

Como se pudo distinguir en la gráfica anterior y con los problemas descubiertos tras la aplicación de las pruebas, se puede apreciar que CEGEL aún presenta dificultades a la hora de contribuir a la funcionalidad del software que repercuten en las especificaciones del cliente, generando como problema a resolver: ¿Cómo contribuir al aseguramiento de la funcionalidad durante el proceso de desarrollo de los sistemas informáticos de Gobierno Electrónico, de manera que estos provean adecuadas funciones que satisfagan las especificaciones del cliente?

Abarcando como objeto de estudio: Aseguramiento de la calidad del software.

Llevando al objetivo general: Proponer una guía para el aseguramiento de la funcionalidad durante el proceso de desarrollo de los sistemas informáticos de Gobierno Electrónico, de manera que estos provean adecuadas funciones que satisfagan las especificaciones del cliente.

Enmarcado específicamente en el campo de acción: Aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico.

Pretendiendo que con la idea a defender: Realizando una guía para el aseguramiento de la funcionalidad, durante el proceso de desarrollo de los sistemas informáticos de Gobierno Electrónico, se contribuirá a que estos provean adecuadas funciones que satisfagan las especificaciones del cliente.

Para un mejor desarrollo de la investigación se desglosa el objetivo general en los siguientes objetivos específicos:

1. Realizar el estudio del estado del arte de la calidad del software de manera general y específicamente la funcionalidad como una característica primordial.
2. Definir atributos, acciones o actividades a realizar a lo largo del ciclo de desarrollo del software para asegurar la funcionalidad del producto final.
3. Validar la propuesta de solución.

En respuesta a cada uno de los objetivos se determinan las tareas a cumplir:

- ✓ Realización del estudio del estado del arte de la calidad del software de manera general y específicamente la funcionalidad como una característica primordial.

Introducción

- ✓ Análisis del estándar Evaluación de los Requisitos de Calidad de los Productos de Software (SQuaRE), otros estándares o modelos internacionales que traten el tema de la funcionalidad como característica de la calidad del software.
- ✓ Definición de atributos para que los sistemas informáticos de Gobierno Electrónico cumplan con la característica de funcionalidad durante el ciclo de vida que se aplica en la UCI, el mismo está basado en el proceso de mejora que se rige por el Modelo Integrado de Capacidad y Madurez.
- ✓ Definición de acciones o actividades a realizar a lo largo del ciclo de vida del software, para contribuir a la funcionalidad en el producto final, basándose en los elementos anteriormente definidos.
- ✓ Validación de la propuesta de solución.

Los métodos utilizados para cumplir con las tareas a desarrollar son:

Teóricos

- ✓ Analítico-Sintético: se utiliza en la búsqueda exhaustiva de información para extraer los elementos de interés en la investigación.
- ✓ Histórico-Lógico: se utiliza en las revisiones bibliográficas referentes al estado actual de modelos, técnicas y métodos referentes a la funcionalidad.

Empíricos

- ✓ Observación: se utiliza para analizar los procesos desde un marco exterior, o sea, sin profundizar en el problema, permite percibir lo que acontece realmente en el ámbito en que se encuentra enmarcada la investigación.
- ✓ Entrevista: es empleado en la recopilación de información de los procesos que se llevan a cabo en los proyectos del CEGEL referentes a la funcionalidad.
- ✓ Encuesta: se utiliza para conocer los criterios de los especialistas durante el período de validación de la propuesta de la guía.

El presente trabajo de diploma está estructurado de la siguiente manera:

Introducción. Se realiza una introducción al tema del aseguramiento de la calidad del software de manera general, así como la mención de trabajos que tributan a este fin tanto en Cuba como a nivel mundial.

Introducción

También se plantean objetivos a cumplir, tareas y demás elementos imprescindibles a tener en cuenta para la realización del presente trabajo de diploma.

Capítulo 1. Fundamentación teórica. Se realiza el estudio del aseguramiento de la funcionalidad del software además de los modelos o estándares que analizan la funcionalidad como una de las características de la calidad del software. Además se explica por qué una guía para contribuir al aseguramiento de la funcionalidad.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico. Este capítulo consta de dos partes importantes, la primera es la propuesta de atributos primordiales a tener en cuenta durante cada etapa del desarrollo del software según el ciclo de vida propuesto y segundo las actividades recomendadas para contribuir a la funcionalidad en cada etapa de desarrollo del software según los atributos identificados.

Capítulo 3. Validación de la propuesta. Se valida la propuesta de solución a través del método Delphi aplicando una encuesta a especialistas cuyo nivel de competencia en cuanto al nivel de conocimiento de la funcionalidad es alto y están en disposición de conformar el panel de evaluación de la propuesta.

Capítulo 1. Fundamentación teórica.

1.1 Introducción al capítulo.

En el presente capítulo se exponen los principales conceptos que sirven de base para adentrarse en el tema del aseguramiento de la calidad del software, fundamentalmente en la funcionalidad del mismo. El estudio a fondo de todas las temáticas brindará fundamentos más sólidos que validen la propuesta mostrada en el presente trabajo de diploma. Un análisis profundo de definiciones y conceptos permite la comprensión de los elementos indispensables para la realización de una guía que permita asegurar la funcionalidad de los sistemas informáticos.

1.2 Calidad del software a nivel de producto.

La calidad es un término aplicado en múltiples esferas y analizado desde distintos puntos de vista. La Real Academia de la Lengua Española define calidad como *“la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor, así como la condición o requisito que se pone en un contrato”*.

La palabra calidad incluye diferentes elementos en correspondencia con el contexto en el que se trate. A la hora de hacer referencia a la calidad se ajusta a la esfera en cuestión y así su significado es variable. Cuando se trata de definir calidad de software queda evidenciado cómo los avances en la esfera de la informática y las telecomunicaciones integran a cada momento nuevos elementos a tener en cuenta para medir la calidad del producto.

“Concordancia con los requisitos funcionales y de rendimientos explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. **(Pressman, 2005)**

“Los factores de un producto de software que contribuyen a la satisfacción completa y total de las necesidades de un usuario u organización”. **(Vega Lebrún, y otros, 2008)**

La ISO 9000 plantea que es *“el conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas”*.

“La calidad del software no es más que un conjunto de requerimientos que van encaminados a desarrollar un producto funcional, seguro, fiable, interoperable, usable, eficiente, portable y mantenible, un producto que brinde al cliente lo que él en realidad necesita, que satisfaga sus necesidades”. Así define la calidad de software la autora de este trabajo de diploma al hacer un estudio detallado de las definiciones dadas

Capítulo 1. Fundamentación Teórica

sobre este tema por diferentes autores, los cuales coinciden en sus conceptos en algunos puntos para definir la calidad del software.

La calidad es una característica indispensable cuando se trata de la producción de software ya que cada día se exige una mejor calidad en cada producto que se realiza y se plantean nuevos atributos para lograrla. La calidad del software ha evolucionado de manera significativa, eso se puede ver evidenciado en el siguiente gráfico:

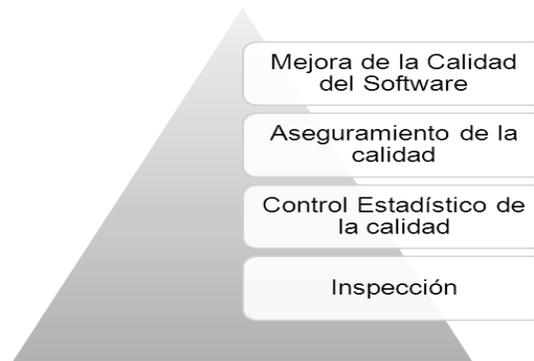


Figura 2. Etapas de evolución de la calidad del software.

Adaptado de (Minguet Melián, y otros, 2003)

La UCI se encuentra actualmente transitando hacia la etapa de mejora de la calidad del software o conocido también como proceso de mejora continua, su aplicación se orienta a perfeccionar los productos, servicios y procesos. Como una manera para aplicar este concepto se puede utilizar el ciclo Planificar, Hacer, Verificar, Actuar o PDCA por su siglas en inglés, que consiste en una secuencia lógica de cuatro pasos repetidos para el mejoramiento y aprendizaje continuo.

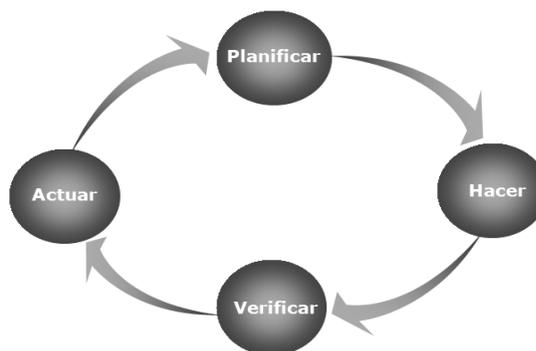


Figura 3. Ciclo de PDCA. Adaptado de (*Camisón Zornoza, y otros, 2009*)

Planificar: Establecer los procesos necesarios para conseguir resultados de acuerdo con los requisitos del cliente y trasladar esos requisitos a especificaciones. Determinar las variables críticas del proceso.

Hacer: Implementar los procesos, y establecer la capacidad de los mismos para satisfacer los requerimientos del cliente.

Verificar: Realizar el seguimiento y la medición de los procesos y los productos respecto a los requisitos para un servicio eficiente y eficaz e informar sobre los resultados.

Actuar: Tomar acciones para mejorar continuamente el desempeño de los procesos e institucionalizar las mejoras alcanzadas.

El proceso de mejora tiene en cuenta las características de la calidad del software, y por tanto la funcionalidad como una de estas características es un elemento a tener en cuenta durante todo el desarrollo del software.

1.3 Funcionalidad en la calidad de software.

El diccionario de la lengua española define funcionalidad como el “conjunto de características que hacen que algo sea práctico y utilitario”, este concepto tiene estrecha relación con el que se define por varios autores cuando abordan la funcionalidad en relación con la calidad del software.

“Grado en que las necesidades asumidas o descritas se satisfacen. Se divide en las subcaracterísticas idoneidad, precisión, interoperabilidad, seguridad” (9126, 2001)

SQuaRE plantea que es “la capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas”.

La funcionalidad lleva consigo tipos de pruebas que no deben dejarse de mencionar ya que influyen directamente en el trabajo de esta característica:

- ✓ Función: Consiste en pruebas que fijan su atención en la validación de las funciones, métodos, servicios y caso de uso. El objetivo de esta prueba es asegurar el trabajo apropiado de los requisitos funcionales, entrada de datos, procesamiento, obtención de resultados, etc. Tienen como meta verificar el procesamiento, recuperación e implementación adecuada de las reglas del

Capítulo 1. Fundamentación Teórica

negocio, así como verificar la apropiada aceptación de datos. Esta prueba es implementada mediante la técnica de pruebas de caja negra y caja blanca.

- ✓ Seguridad: En esta prueba se asegura que los datos o el sistema solamente es accedido por los actores deseados. El objetivo de esta prueba es verificar que los actores del sistema solo puedan acceder a las funciones que le competen. La técnica que se emplea para esta prueba es identificar cada tipo de usuario, funciones y datos que los que está autorizado a acceder, de esta manera se contribuye a la seguridad del software. Esta técnica se aplica mediante la creación de pruebas para cada usuario donde se comprueben los permisos del mismo.
- ✓ Volumen: Enfocada en verificar las habilidades de los programas para manejar grandes cantidades de datos, tanto como entrada, salida o residente en la base de datos. Para llevar a cabo esta prueba se emplean por lo general dos técnicas: la primera es con varios usuarios accediendo a la aplicación de manera simultánea, los cuales realizan las mismas consultas a la base de datos, creando el peor caso al que se pueda enfrentar y la segunda es utilizar un tamaño máximo de base de datos y múltiples clientes para correr consultas simultáneamente para períodos extendidos.

En la UCI se llevan a cabo métodos de pruebas y pruebas de función con respecto a la funcionalidad, las que se centran en la verificación de las funciones que debe cumplir el software.

Métodos de prueba de función

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

Prueba de caja negra: Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software donde 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. Se centran principalmente en los requisitos funcionales del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en las técnicas de Caja Blanca. **(Machado Peña, y otros, 2012)**

Con las pruebas de caja negra se pueden detectar errores como: funciones incorrectas o ausentes, errores de interfaz, en estructuras de datos o en accesos a las bases de datos externas, de rendimiento y errores de inicialización y terminación.

Los procesos de pruebas que se ejecutan en el departamento de pruebas de CALISOFT cuentan con tres etapas: planificación, diseño y ejecución.

Planificación: En esta etapa se determinan las personas involucradas en el proceso y todos los recursos que intervienen, teniendo en cuenta el propósito principal de la prueba, además se define la estrategia que se va a seguir, el orden de las actividades necesarias para realizar las pruebas y el tiempo dedicado para su ejecución, así como su cronograma.

Diseño: En esta etapa se diseñan las pruebas que guiarán el proceso planificado, a partir de la revisión a la documentación y del diseño de los casos de pruebas; documentando la secuencia de pasos para ejecutar cada caso de prueba y los resultados esperados.

Ejecución: Etapa en la que se ejecutan las pruebas diseñadas, con el fin de probar los requerimientos y determinar si estos han sido cumplidos.

1.4 Aseguramiento de la funcionalidad en la calidad de software.

El aseguramiento de la funcionalidad requiere que se comience el análisis desde los conceptos básicos que interviene en este aspecto:

El aseguramiento es un “conjunto de acciones planificadas para cumplir un objetivo.”

El modelo SQuaRE define como funcionalidad: “La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas.”

La funcionalidad como parte de las características de la calidad del software requiere un aseguramiento para lograr un producto que cumpla con las expectativas del cliente, pues está bien asociada al usuario final del producto.

Contribuir a que el producto a desarrollar tenga la calidad requerida no es tarea fácil, por lo que el aseguramiento de la funcionalidad del software debe estar presente a lo largo de todo el desarrollo, lo que requiere actividades concretas para este fin:

- ✓ Pruebas basadas en requisitos funcionales.
- ✓ Cumplimiento de las reglas del negocio.
- ✓ Relación de los casos de uso y los requisitos.
- ✓ Técnica de prueba de caja negra.

El aseguramiento de la calidad de software es un diseño planificado de acciones o actividades a realizar a lo largo del ciclo de vida de software que se requieren para contribuir a la funcionalidad en el producto final.

La funcionalidad como característica de la calidad del software ha sido tratada por diferentes modelos/estándares, los cuales plantean conceptos desde el punto de vista del autor o autores. Estos modelos deben tenerse presente a la hora de tratar el tema de la funcionalidad del software por los elementos que aportan al aseguramiento de esta característica.

1.5 Modelos/Estándares de calidad del software que refieren entre sus atributos a la funcionalidad.

La ISO/IEC 9126 define modelos como *“descomposición de los aspectos relativos a la calidad del producto (software) en características y subcaracterísticas organizadas jerárquicamente, las cuales son usadas como una lista de chequeo de calidad.”*

Los modelos de calidad son herramientas que guían a las organizaciones a la mejora continua y la competitividad, dándoles especificaciones de qué tipo de requisitos debe de implementar para poder brindar productos y servicios de alto nivel, los modelos de calidad dicen que hacer, no cómo hacerlo. Son útiles para discutir, planificar y obtener indicadores de calidad.

1.5.1 Modelo FURPS (Funcionalidad, Usabilidad, Confiabilidad, Rendimiento, Soporte)

En este modelo se desarrollan un conjunto de factores de calidad del software, bajo el acrónimo de FURPS: funcionalidad, usabilidad, fiabilidad, rendimiento y soporte. (**Pressman, 2005**)

- ✓ Facilidad de uso (Usabilidad): se valora considerando factores humanos, la estética, la consistencia y la documentación general.

Capítulo 1. Fundamentación Teórica

- ✓ **Confiabilidad:** se evalúa midiendo la frecuencia y gravedad de los fallos, la exactitud de las salidas (resultados), el tiempo medio de fallos, la capacidad de recuperación ante un fallo y la capacidad de predicción del programa.
- ✓ **Rendimiento:** se mide por la velocidad de procesamiento, el tiempo de respuesta, consumo de recursos, rendimiento efectivo total y eficacia.
- ✓ **Capacidad de soporte (Soporte):** combina la capacidad de ampliar el programa (extensibilidad), adaptabilidad y servicios (estos tres atributos representan un término más común: mantenimiento), así como capacidad de hacer pruebas, compatibilidad, capacidad de configuración (la capacidad de organizar y controlar elementos de la configuración del software), la facilidad de instalación de un sistema y la facilidad con que se pueden localizar los problemas. (**Pressman, 2005**)

Factor	Criterio
Funcionalidad	Características y capacidades del programa Generalidad de las funciones Seguridad del sistema
Facilidad de Uso	Factores humanos Factores estéticos Consistencia de la interfaz Documentación
Confiabilidad	Frecuencia y severidad de las fallas Exactitud de las salidas Tiempo medio de fallos Capacidad de recuperación ante fallas Capacidad de predicción

Factor	Criterio
Rendimiento	Velocidad del procesamiento Tiempo de respuesta Consumo de recursos Rendimiento efectivo total Eficacia
Capacidad de Soporte	Extensibilidad Adaptabilidad Capacidad de pruebas Capacidad de configuración Compatibilidad Requisitos de instalación

Figura 4. Modelo de FURPS.

Los factores de calidad FURPS y atributos descritos anteriormente pueden usarse para establecer métricas de la calidad para todas las actividades del proceso del software. (**Pressman, 2005**)

La funcionalidad en este modelo se valora evaluando el conjunto de características y capacidades del programa, la generalidad de las funciones entregadas y la seguridad del sistema global.

1.5.2 Modelo Dromey.

En este modelo se propone un marco de referencia – o metamodelo - para la construcción de modelos de calidad, basado en cómo las propiedades medibles de un producto de software pueden afectar los atributos de calidad generales, como por ejemplo, confiabilidad y mantenibilidad. El problema que se plantea es cómo conectar tales propiedades del producto con los atributos de calidad de alto nivel. Para

Capítulo 1. Fundamentación Teórica

solventar esta situación, Dromey sugiere el uso de cuatro categorías que implican propiedades de calidad, que son: correctitud, internas, contextuales y descriptivas. (**Camacho, y otros, 2004**)

Propiedades del producto	Atributos de Calidad
Correctitud	✓ Funcionalidad ✓ Confiabilidad
Internas	✓ Mantenibilidad ✓ Eficiencia ✓ Confiabilidad
Contextuales	✓ Mantenibilidad ✓ Reusabilidad ✓ Portabilidad ✓ Confiabilidad
Descriptivas	✓ Mantenibilidad ✓ Reusabilidad ✓ Portabilidad ✓ Usabilidad

Figura 5. Modelo de Dromey (Camacho, y otros, 2004)

Dromey sugiere una técnica genérica para construir un modelo de calidad. El mismo resalta el hecho de que la calidad del producto es altamente determinada por los componentes del mismo (incluyendo documentos de requerimientos, guías de usuarios, diseños, y código), las propiedades tangibles de los componentes y las propiedades tangibles de la composición de los componentes.

El proceso de construcción de modelos de calidad propuesto por Dromey consta de 5 pasos:

- ✓ Especificación de los atributos de calidad de alto nivel (por ejemplo, confiabilidad, mantenibilidad).
- ✓ Determinación de los distintos componentes del producto a un apropiado nivel de detalle (por ejemplo, paquetes, subrutinas, declaraciones).
- ✓ Para cada componente, determinación y categorización de sus implicaciones más importantes de calidad.
- ✓ Proposición de enlaces que relacionan las propiedades implícitas a los atributos de calidad, o, alternativamente, uso de enlaces de las cuatro categorías de atributos propuestas.
- ✓ Iteración sobre los pasos anteriores, utilizando un proceso de evaluación y refinamiento.

La funcionalidad en este modelo es controlada como un atributo de la calidad asociado a la propiedad correctitud, es decir, no pertenece por si misma a sus características principales.

1.5.3 Estándar ISO 9126.

Es necesario antes de tratar este estándar definir precisamente qué es un estándar. De acuerdo con la definición de la Real Academia Española, estándar es “*aquello que sirve como tipo, modelo, norma, patrón o referencia*”.

Un estándar de calidad es el que reúne los requisitos mínimos, sirviendo de modelo o referencia para desarrollar el trabajo de una organización. **(9126, 2001)**

El estándar ISO 9126 ha sido desarrollado en un intento de identificar las características claves de calidad para el software.

El estándar identifica las siguientes características de calidad **(Pressman, 2005)**:

- ✓ Funcionalidad. El grado en que el software satisface las necesidades indicadas por las siguientes subatributos: idoneidad, corrección, interoperabilidad, conformidad y seguridad.
- ✓ Confiabilidad. Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes subatributos: madurez, tolerancia a fallos y facilidad de recuperación.
- ✓ Usabilidad. Grado en que el software es fácil de usar. Viene reflejado por los siguientes subatributos: facilidad de comprensión, facilidad de aprendizaje y operatividad.
- ✓ Eficiencia. Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes subatributos: tiempo de uso y recursos utilizados.
- ✓ Facilidad de mantenimiento. La facilidad con que una modificación puede ser realizada. Está indicada por los siguientes subatributos: facilidad de análisis, facilidad de cambio, estabilidad y facilidad de prueba.
- ✓ Portabilidad. La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes subatributos: facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

Este estándar se encuentra dividido en diferentes partes: métricas externas, métricas internas y calidad en las métricas de uso.

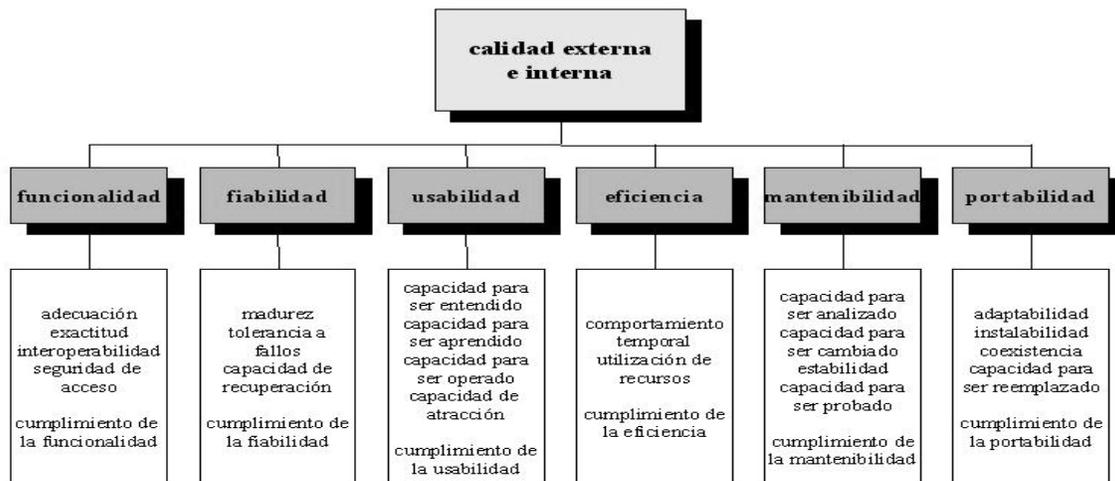


Figura 6. Modelo ISO 9126. Calidad interna y externa. (MG Soluciones Informaticas, 2011)

ISO 9126 propone un modelo de calidad categorizando la calidad de los atributos del software en seis características (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad), las cuales son subdivididas en subcaracterísticas.



Figura 7. Modelo ISO 9126. Calidad de Uso. (Morilla, 2008)

La calidad de uso es definida como “*la capacidad del software que posibilita la obtención de objetivos específicos con efectividad, productividad, satisfacción y seguridad*”.

La funcionalidad es analizada en este modelo en base a sus subcaracterísticas:

Adecuación: Capacidad del producto de software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Exactitud: Capacidad del producto de software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.

Interoperabilidad: Capacidad del producto de software para interactuar con uno o más sistemas especificados.

Seguridad de acceso: Capacidad del producto de software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniega el acceso a las personas o sistemas autorizados.

Cumplimiento funcional: Capacidad del producto software para adherirse a normas, convenciones o regulaciones en leyes y prescripciones similares relacionadas con funcionalidad.

El presente modelo es abarcador en cuanto al tema de la funcionalidad y propone el análisis de 5 elementos fundamentales para contribuir a la funcionalidad del software.

1.5.4 Modelo de Calidad WEB (WQM)

En este modelo se propone una estructura de cubo, la cual muestra los aspectos a evaluar en el desarrollo web:

- ✓ Características web.
- ✓ Procesos del ciclo de vida.
- ✓ Características de calidad.

WQM es un modelo tridimensional, basado en los aspectos anteriores para evaluar la calidad en los sitios web. Se centra principalmente en 3 factores (**Muñoz, 2005**):

- ✓ Contenido.
- ✓ Presentación
- ✓ Navegación.

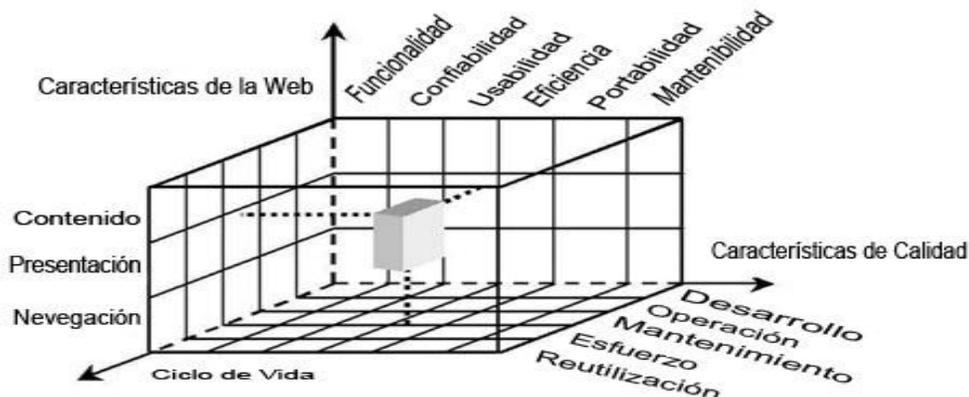


Figura 8. Modelo de Calidad WQM. Adaptado de (**Muñoz, 2005**)

En el caso de WQM la gran presencia de tecnología Web y la gran información asociada a esta tecnología hace imprescindible que los diseños se realicen bajo unos mínimos criterios de calidad, hasta ahora prácticamente inexistentes. Las aplicaciones web desarrolladas sin criterios de calidad tendrán un pobre rendimiento y causarán fallos, por lo que es necesario que los sistemas web sean gestionados y dirigidos de forma rigurosa y cualitativa. (**Scalone, 2006**)

En este modelo la funcionalidad se centra en:

- ✓ Búsqueda y Recuperación.
- ✓ Navegación y Exploración.
- ✓ Funciones y Contenido Específicos del Dominio.

WQM propone se evalúe la funcionalidad a través de la aplicación de las siguientes métricas (**Scalone, 2006**):

- ✓ Deficiencias o cualidades ausentes debido a diferentes navegadores (browsers).
- ✓ Deficiencias o resultados inesperados independientes de browsers (ejemplos: errores de búsqueda imprevistos, deficiencias con marcos, etc.).
- ✓ Testeo de Páginas.
 - ✓ Validadores de HTML (Validator validator.w3.org).
- ✓ Cantidad Total de Applets (distintos).
- ✓ Cantidad de Documentos con JavaScript.
- ✓ Cantidad de Fields en Formularios (promedio).

1.5.5 Calidad en Tecnología de la Información (QUINT2)

Este modelo es una ampliación de la ISO 9126, cubre las perspectivas de calidad del usuario y del desarrollador para una aplicación web. Además se adiciona características a las propuestas por la ISO 9126, amplía subcaracterísticas y atributos de calidad.

Capítulo 1. Fundamentación Teórica

Modelo Calidad en Tecnología de la Información (QUINT2)					
Funcionalidad	Confiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Idoneidad	Madurez	Comprensibilidad	Comportamiento del tiempo	Analizabilidad	Adaptabilidad
Precisión	Tolerancia a fallos	Aprendizaje	Comportamiento de los recursos	Mutabilidad	Facilidad de instalación
Interoperabilidad	Recuperación	Operatividad		Estabilidad	Conformidad
Conformidad	Disponibilidad	Claridad		Capacidad de pruebas	Intercambiabilidad
Seguridad	Degradabilidad	Personalización		Capacidad de Gestión	
Trazabilidad		Atractividad		Reutilización	
		Claridad			
		Utilidad			
		Facilidad de uso			

Figura 9. Calidad en Tecnología de la Información (QUINT2). (Salanova, 2006)

Quint2 permite a los usuarios y desarrolladores hablar de distintos tipos de calidad. La perspectiva del usuario enfoca la calidad externa de un producto de software ejecutable, y la perspectiva del desarrollador se dirige hacia la calidad interna del producto durante su desarrollo y mantenimiento. **(Salanova, 2006)**

Este modelo define como funcionalidad: *“Un conjunto de atributos que tienen que ver con la existencia de un conjunto de funciones y de sus características especificadas. Las funciones son aquellas que satisfacen necesidades indicadas o implicadas.”*

Subcaracterísticas: idoneidad, precisión, interoperabilidad, conformidad, seguridad y trazabilidad.

Idoneidad: Atributo del software que tiene que ver con la presencia y la adecuación de un conjunto de funciones para tareas especificadas. La idoneidad es una subcaracterística de calidad que considera si las funcionalidades deseadas están presentes en el producto de software. Se determina, sobre todo, por la efectividad de las etapas de la ingeniería de requisitos, cuando un usuario y un desarrollador del software tienen que comunicarse para determinar las funcionalidades deseadas de un producto de software. Debe existir la precisión de la comunicación y la comprensión inequívoca del dominio de la aplicación y del problema que se tratará. **(Salanova, 2006)**

Precisión: Atributo del software que tienen que ver con la corrección de los resultados requeridos del producto. La subcaracterística de exactitud estima la corrección de las funcionalidades y otros resultados requeridos (manuales de usuario etc.) que provienen del producto de software. **(Salanova, 2006)**

Interoperabilidad: Atributo del software que tienen que ver con su capacidad de actuar recíprocamente con sistemas especificados. La interoperabilidad caracteriza la capacidad de un producto de software de actuar recíprocamente con sistemas especificados. **(Salanova, 2006)**

Cumplimiento: Atributo del software que indica la facilidad con que un producto de software se adhiere a estándares y normas. **(Salanova, 2006)**

Seguridad: Atributo del software que tiene que ver con su capacidad para prevenir el acceso no autorizado, accidental o deliberado, a programas o datos. **(Salanova, 2006)**

Trazabilidad: Atributo del software que indica la cantidad de esfuerzo necesitado para verificar la corrección del proceso de datos sobre puntos requeridos. **(Salanova, 2006)**

En este modelo la funcionalidad abarca 6 subcaracterísticas que tratan de alcanzar el control de la funcionalidad del software. **(Salanova, 2006)**

1.5.6 Estándar ISO/IEC 25010 SQuaRE

El estándar Evaluación de los Requisitos de Calidad de los Productos de Software por sus siglas en inglés SQuaRE es una revisión de ISO/IEC 9126-1:2001 y se dedica solamente a la calidad del producto de software. Este estándar propone mantener las 6 características que plantea la ISO 9126 (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y además incorporarle 2 características más (interoperabilidad y seguridad).

- ✓ Funcionalidad: La capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas. **(Departement of Software and IT Engineering, 2007)**
- ✓ Seguridad: La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos, y a las personas o sistemas autorizados no se les niegue el acceso a ellos. **(Departement of Software and IT Engineering, 2007)**
- ✓ Interoperabilidad: La capacidad del producto de software de interactuar con uno o más sistemas especificados. La interoperabilidad se utiliza en lugar de compatibilidad para evitar una posible ambigüedad con la reemplazabilidad. **(Departement of Software and IT Engineering, 2007)**
- ✓ Fiabilidad: La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo condiciones especificadas. **(Departement of Software and IT Engineering, 2007)**

Capítulo 1. Fundamentación Teórica

- ✓ **Usabilidad:** La capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas. (*Departement of Software and IT Engineering, 2007*)
- ✓ **Eficiencia:** La capacidad del producto de software para proveer un desempeño adecuado, de acuerdo con la cantidad de recursos utilizados y bajo las condiciones planteadas. (*Departement of Software and IT Engineering, 2007*)
- ✓ **Mantenibilidad:** Capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, especificaciones de requerimientos funcionales. (*Departement of Software and IT Engineering, 2007*)
- ✓ **Portabilidad:** La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software. (*Departement of Software and IT Engineering, 2007*)

SQuaRE nace con el objetivo de responder a las necesidades de los usuarios a través de un conjunto de documentos unificados cubriendo tres procesos de calidad complementarios: especificación de requisitos, medidas y evaluación. Por lo tanto, SQuaRE se creó para satisfacer una serie de necesidades que existían con la ISO 9126 y la ISO/IEC 14598, donde ellos (ISO 9126 y ISO/IEC 14598) pertenecen a la primera generación de estándares de calidad de un producto software. Por consiguiente, SQuaRE pertenece a la segunda generación de calidad de un producto software.

Este estándar cuenta con las siguientes divisiones:

- ✓ ISO/IEC 2500n. División de dirección de calidad.
- ✓ ISO/IEC 2501n. División del modelo de calidad.
- ✓ ISO/IEC 2502n. División de medida de calidad.
- ✓ ISO/IEC 2503n. División de requisitos de calidad.
- ✓ ISO/IEC 2504n. División de evaluación de calidad.

Capítulo 1. Fundamentación Teórica

Calidad del software (interna y externa)							
Funcionalidad	Seguridad	Interoperabilidad	Fiabilidad	Usabilidad	Eficiencia	Mantenibilidad	Portabilidad
Adecuación	Adherencia a normas	Adherencia a normas	Madurez	Compresibilidad	Tiempo de respuesta	Capacidad de análisis	Adaptabilidad
Exactitud			Tolerancia a fallos	Capacidad de aprendizaje	Utilización de recursos	Capacidad de cambios	Capacidad a instalación
Adherencia a normas			Recuperabilidad	Operabilidad	Adherencia a normas	Estabilidad	Capacidad a coexistencia
			Adherencia a normas	Atractivo		Capacidad a testing	Adherencia a normas
		Adherencia a normas	Adherencia a normas				

Figura 10. Estándar de calidad ISO/IEC 25010 para la calidad externa e interna.

Este estándar hace un análisis más completo sobre la funcionalidad del software, propone 3 subcaracterísticas que integran el funcionamiento del producto que se desea desarrollar.

Adecuación: La capacidad del producto de software para proporcionar un conjunto apropiado de funciones para tareas específicas y los objetivos del usuario.

Exactitud: La capacidad del producto de software para proporcionar los resultados deseados o efectos con el necesario grado de precisión.

Adherencia a normas: La capacidad del producto de software para adherirse a normas, convenciones o regulaciones, leyes y prescripciones similares relacionadas con funcionalidad.

1.5.7 Análisis Comparativo

Realizado el estudio de los diferentes modelos/estándares de calidad y la funcionalidad como una característica indispensable a la hora de analizar la calidad del software se determinan los elementos que posibilitan contribuir a la calidad del producto final. Es importante recordar que los modelos de calidad dicen qué hacer, no cómo hacerlo, por lo que a partir de los conceptos planteados en los modelos, se determinan las pruebas, y técnicas para contribuir a la calidad.

A continuación se muestra una tabla comparativa donde se puede apreciar los criterios que plantean sean evaluados en cada uno de los modelos/estándares respecto a la funcionalidad:

Modelos/Estándares	Criterios a evaluar de funcionalidad
FURPS	<ul style="list-style-type: none"> ✓ Características y capacidades del programa. ✓ Generalidad de las funciones. ✓ Seguridad del sistema.
DROMEY	<ul style="list-style-type: none"> ✓ Correctitud. <ul style="list-style-type: none"> • Funcionalidad.

Capítulo 1. Fundamentación Teórica

	<ul style="list-style-type: none">• Confiabilidad.
Estándar ISO 9126	<ul style="list-style-type: none">✓ Adecuación.✓ Exactitud.✓ Interoperabilidad.✓ Seguridad de acceso.✓ Cumplimiento funcional.
WQM	<ul style="list-style-type: none">✓ Búsqueda y recuperación.✓ Navegación y exploración.✓ Funciones y contenido específicos del dominio.
QUINT2	<ul style="list-style-type: none">✓ Idoneidad.✓ Precisión.✓ Interoperabilidad.✓ Conformidad.✓ Seguridad.✓ Trazabilidad.
SQuaRE	<ul style="list-style-type: none">✓ Adecuación.✓ Exactitud.✓ Adherencia a normas.

El modelo FURPS plantea 3 subcaracterísticas de la funcionalidad, dentro de las cuales es importante resaltar la seguridad, que es tratada también en el modelo QUINT2 y el estándar ISO 9126. Dicha subcaracterística hay que tenerla en cuenta para contribuir a la funcionalidad del software, pero ya en el estándar SQuaRE no se considera una subcaracterística, sino una característica brindándole más peso para su análisis y aseguramiento; lo mismo sucede con interoperabilidad, subcaracterística de estos mismos modelos.

DROMEY por su parte ve la funcionalidad no como subcaracterística, sino como un elemento a tener en cuenta dentro de la correctitud, por lo que no le da el peso requerido a la funcionalidad dentro del proceso de desarrollo de software.

El modelo WQM es básicamente un modelo web, pero la funcionalidad no debe verse por separado en los sistemas web o de escritorio, pues la funcionalidad son las necesidades del cliente y su concepto no sufre variaciones por tratarse de un tipo de sistema u otro, por lo que este modelo no es el idóneo a utilizar ya que se requiere de un modelo que sea capaz de interactuar con los dos tipos de sistemas (web y escritorio).

Capítulo 1. Fundamentación Teórica

De los modelos/estándares estudiados el ISO/IEC 25010 SQuaRE introduce nuevas características a la calidad que antes eran tratadas como subcaracterísticas, de esta manera se puede centrar a la funcionalidad en factores más específicos. ISO/IEC 25010 SQuaRE propone 3 subcaracterísticas para ser analizadas en el proceso de desarrollo del software que están más centradas en la funcionalidad. Dicho estándar brinda información actualizada sobre las características y subcaracterísticas de la calidad, pertenece a la segunda generación de estándares de calidad y es una mejora de la ISO 9126 por la cual se realizan las pruebas de funcionalidad del software en la Universidad. Por las razones expresadas con anterioridad se decide emplear como estándar guía para la investigación el ISO/IEC 25010 SQuaRE.

1.6 Modelo Integrado de Capacidad y Madurez (CMMI).

CMMI es un enfoque de mejora de procesos que proporciona a las organizaciones los elementos esenciales de procesos efectivos, lo que mejorará su rendimiento. CMMI basado en la mejora del proceso incluye la identificación de las fortalezas de su organización, procesos y debilidades y hacer cambios en el proceso de convertir debilidades en fortalezas. (SEI, 2011)

Representación continua	Representación escalonada
Las áreas de proceso se organizan por categorías de áreas de proceso.	Las áreas de proceso se organizan por niveles de madurez.
La mejora se mide en niveles de capacidad que reflejan la implantación incremental de un área de proceso particular.	La mejora se mide utilizando niveles de madurez que reflejan la implementación concurrente de múltiples áreas de proceso.
Hay seis niveles de capacidad (0-6).	Hay cinco niveles de madurez (1-5).
Hay dos tipos de prácticas: básicas y avanzadas.	Hay sólo un tipo de prácticas. El concepto de práctica avanzada se consigue por otros medios.
Los niveles de capacidad se usan para organizar las prácticas genéricas.	Las prácticas genéricas se usan según características comunes.
Todas las prácticas genéricas se usan en todas las áreas de proceso.	Sólo se usan en un área de proceso las prácticas aplicables al nivel de madurez.
Existen prácticas genéricas para los niveles de capacidad del 1 al 5.	Existen prácticas genéricas para los niveles de madurez del 2 al 5. Algunas de las prácticas utilizadas en la representación continua se aplican en algunas áreas de proceso.
Existe la posibilidad de obtener el nivel de madurez equivalente al perfil obtenido.	No es posible determinar con qué perfil de la representación continua se corresponde un determinado nivel.

Figura 11. Representaciones de CMMI. Adaptado de (Pérez, 2004)

Este modelo tiene dos representaciones (escalonada y continua), en la Universidad se decidió realizar el programa de mejora por la versión escalonada, que establece 5 niveles de madurez en función de las

áreas de proceso que se asocian a uno de estos niveles y a medida que la organización cumple con los procesos definidos para cada uno, alcanza el nivel de madurez de referencia. Una organización alcanza un nivel de madurez determinado cuando ha puesto en práctica todas y cada una de las áreas de proceso aplicables a ese nivel y a los niveles inferiores.

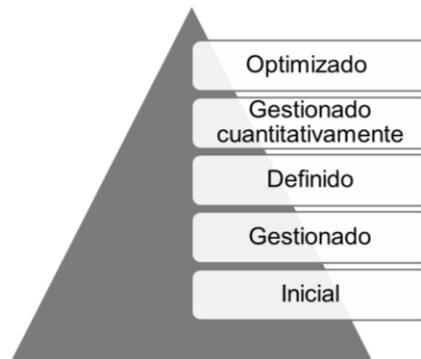


Figura 12. Representación Escalonada de CMMI. Adaptado de **(Axentia, 2006)**

En la actualidad la Universidad está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI (Capability Maturity Model Integration) y con la contratación de los servicios de consultoría del SIE Center (Software Industry Excellence Center) del Tecnológico de Monterrey. **(CALISOFT, 2012)**

El nivel 2 cuenta con 7 áreas de procesos, que son: Administración de Acuerdo con Proveedores (SAM), Aseguramiento de la Calidad de Procesos y Productos (PPQA), Administración de la Configuración (CM), Planeación del Proyecto (PP), Monitoreo y Control de Proyecto (PMC), Administración de Requisitos (REQM) y Medición y Análisis (MA). **(Calisoft, 2012)**



Figura 13. Áreas de proceso del nivel 2 de CMMI

Basado en este modelo la Universidad encamina su proceso de desarrollo por el ciclo de vida definido, el cual consta de 9 etapas por las cuales se rige esta investigación.

El establecimiento de este modelo ha ayudado a establecer las bases y fundamentos para seguir mejorando los procesos y fortalecer la cultura de calidad en el desarrollo de software. **(Calisoft, 2012)**

La mejora de procesos se ha expandido a todo tipo de proyectos que se desarrollan en la Universidad, entre ellos los proyectos que encaminan sus producciones a los sistemas informáticos de Gobierno Electrónico.

1.7 ¿Procedimiento, lista de chequeo, estrategia o guía?

El estudio de diferentes instrumentos que permitan definir una línea de trabajo se hace necesario para determinar cuál es el idóneo para cumplir con los objetivos trazados.

Procedimiento

La Real academia de la lengua española define procedimiento como un “*método de ejecutar algunas cosas*”.

El procedimiento es una secuencia de operaciones que se ejecutan para llevar a cabo una actividad o tarea. El mismo implica actividades o tareas por parte del ejecutor para el desarrollo oportuno de la labor.

El procedimiento de calidad se refiere al enfoque y a la organización operativa usada para alcanzar los objetivos establecidos por la política de calidad.

Lista de chequeo

“*Se entiende por lista de chequeo o check-list, en inglés, a un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado*”. **(Bichachi, 2004)**

De manera general las listas de chequeo permiten evaluar una actividad o acción.

Estrategia

Una estrategia es un conjunto de acciones planificadas sistemáticamente en el tiempo que se llevan a cabo para lograr un determinado fin.

Pasos requeridos para realizar una estrategia:

- ✓ ¿Qué se hace?
- ✓ ¿Cuándo se hace?
- ✓ ¿Cómo se hace?
- ✓ ¿Quién lo hace?
- ✓ ¿Quién lo financia?

- ✓ ¿Quién lo gestiona?

Definir una buena estrategia requiere de la participación tanto de los ejecutores como de los beneficiados, ya que se asegura la sostenibilidad de las acciones que se van a realizar.

Guía

“*Aquello que dirige o encamina*”. (RAE, 2012)

Se puede decir que una guía en la calidad del software es una secuencia de actividades planificadas basadas en los principios de la misma, de manera que:

- ✓ Guíe el proceso.
- ✓ Logre transparencia en la información.
- ✓ Facilite el trabajo.
- ✓ Mejore la calidad del proceso.

1.7.1 Análisis comparativo

Realizado el estudio de los instrumentos se pudo determinar los conceptos principales que lo definen así como las principales características de cada uno de ellos.

A continuación se muestra una tabla comparativa donde se puede apreciar los objetivos/principios de estos instrumentos:

Instrumentos	Objetivos/Principios
Procedimiento	Alcanzar los objetivos establecidos por la política de calidad.
Lista de chequeo	Evaluar una actividad o acción.
Estrategia	<ul style="list-style-type: none">✓ ¿Qué se hace?✓ ¿Cuándo se hace?✓ ¿Cómo se hace?✓ ¿Quién lo hace?✓ ¿Quién lo financia?✓ ¿Quién lo gestiona?
Guía	<ul style="list-style-type: none">✓ Guíe el proceso.✓ Logra transparencia en la información.

	<ul style="list-style-type: none">✓ Facilita el trabajo.✓ Mejora la calidad del proceso.
--	---

El objetivo del presente trabajo de diploma es proponer una secuencia de actividades que deben contribuir al aseguramiento de la calidad del software. El instrumento más acorde a este objetivo es la guía, a través de la cual se confeccionará una secuencia de actividades planificadas para el proceso de aseguramiento de la funcionalidad del software en CEGEL, teniendo en cuenta el modelo CMMI que rige el proceso de desarrollo de software en la universidad. La guía permite evaluar el proceso de desarrollo de manera permanente, detectándose oportunamente los resultados en el cumplimiento de logros previstos. También teniendo en cuenta las dificultades detectadas, la guía facilita realizar el proceso de retroalimentación.

1.8 Sistemas Informáticos de Gobierno Electrónico.

1.8.1 Sistemas Informáticos.

Un sistema informático es un sistema de información que integra hardware, software y recursos humanos, utiliza ordenadores para almacenar datos, procesarlos y ponerlos a disposición del usuario que así lo requiera.

Componentes de un sistema informático

- ✓ Personas.
- ✓ Equipos de procesamiento de información.
- ✓ Equipos de apoyo a las transmisiones.
- ✓ Programas de computadora.
- ✓ Manuales de usuario para orientar a los usuarios-operadores sobre su trabajo con el sistema de información.

Beneficios de los sistemas informáticos

- ✓ Integridad de la información.
- ✓ Almacenamiento ordenado de información.
- ✓ Centralización de la información.
- ✓ Claridad en los procesos.
- ✓ Confidencialidad de la información y seguridad.
- ✓ Rapidez en la generación de reportes.

- ✓ Disponibilidad de datos estadísticos.
- ✓ Seguimiento de indicadores de gestión y calidad.

1.8.2 Sistemas Informáticos de Gobierno Electrónico.

Es el uso de las Tecnologías de Información y Comunicación en el marco de la modernización del Estado, que posibilita el acceso y la entrega de servicios gubernamentales y promueve la participación y transparencia de la gestión pública en beneficio de la ciudadanía en general.

Beneficios de los sistemas de Gobierno Electrónico

- ✓ Mejor acceso, con servicios que son entregados dónde y cuándo se necesitan.
- ✓ Mayor variedad de medios para la distribución del servicio.
- ✓ Segmentación del mercado, con servicios enfocados a las necesidades del ciudadano individual.
- ✓ Respuesta sobre la satisfacción del ciudadano con el servicio entregado.
- ✓ Ganancias en eficiencia y efectividad por la mejor utilización de la información y el mejor manejo de programas de trabajo.

En el CEGEL los proyectos que se desarrollan tienen como misión satisfacer necesidades de clientes gubernamentales mediante el desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia, a partir de un personal altamente calificado.

Los proyectos pertenecientes a este centro abarcan temáticas como:

- ✓ Interoperabilidad entre Sistemas Judiciales (Interoperabilidad)
- ✓ Grupo Legal (LEGAL)
- ✓ Grupo de Arquitectura, Tecnología y Seguridad (GATS)
- ✓ Estadística Judicial y Tecnologías de Apoyo a la Decisión.

Los sistemas de Gobierno Electrónico permiten un ahorro en cuanto a tiempo y costo, además proveen rápidas repuestas que facilitan el trabajo del gobierno hacia los ciudadanos.

1.9 Conclusiones parciales.

El aseguramiento de la funcionalidad como característica de la calidad del software requiere un conjunto de actividades planificadas a ejecutar, el estudio de modelos/estándares, actividades específicas para la funcionalidad y otros elementos asociados a este tema brindó fundamentos que se deben tener en cuenta en el desarrollo del software que respondan a esta característica.

Se propone el uso del estándar ISO/IEC 25010 SQuaRE teniendo en cuenta las subcaracterísticas de la funcionalidad de forma que respondan a las características particulares de los sistemas informáticos de Gobierno Electrónico. Este estándar es una mejora de la ISO 9126 por la que se realizan las pruebas de funcionalidad al software en la universidad. También se tiene en cuenta el modelo CMMI que es el que se utiliza en la universidad y al cual responde el ciclo de vida del proceso de mejora por el que los proyectos deben desarrollar sus soluciones informáticas.

Además se tiene presente la utilización de una guía para mostrar los resultados de la investigación de forma que pueda ser aplicada en el centro, para contribuir al aseguramiento de la funcionalidad de los sistemas informáticos de Gobierno Electrónico.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico.

2.1 Introducción al capítulo.

En el presente capítulo se propone una guía para asegurar la funcionalidad en los sistemas informáticos de Gobierno Electrónico, teniendo como base de trabajo el nivel 2 de CMMI por el cual se rige el desarrollo de software en la Universidad, así como el ciclo de vida del proceso de mejora. Además se tiene en cuenta el estándar ISO/IEC 25010 SQuaRE que fue seleccionado para desarrollar la propuesta.

2.2 Características de los proyectos del centro CEGEL.

El centro CEGEL consta en la actualidad con 12 proyectos, los que se agrupan en los departamentos de Informática Jurídica (IJ) y Gestión Gubernamental (GG). Los mismos elaboran los productos en diferentes plataformas, convirtiendo en funcionalidades las necesidades que el cliente espera cumpla el software que requiere.

Actualmente se sigue el ciclo de vida que propone el proceso de mejora que lleva a cabo la universidad y a partir de las etapas definidas en el mismo se elaboran las actividades que se llevarán a cabo para contribuir a la funcionalidad del software. Cada proyecto ajusta las actividades a sus necesidades específicas y determina qué importancia le brinda a la funcionalidad dentro del aseguramiento de la calidad del software que desarrolla.

2.3 Ciclo de vida del proceso de desarrollo de software basado en el programa de mejora.

Elementos definidos por etapas.

La funcionalidad es un elemento que se debe tener en cuenta en cada una de las etapas del ciclo de vida del desarrollo del software, por lo que se debe analizar los elementos críticos en cada una de ellas para proponer las actividades que contribuyan a la funcionalidad en todo momento del desarrollo. El ciclo de vida del proceso de mejora define 9 etapas:

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

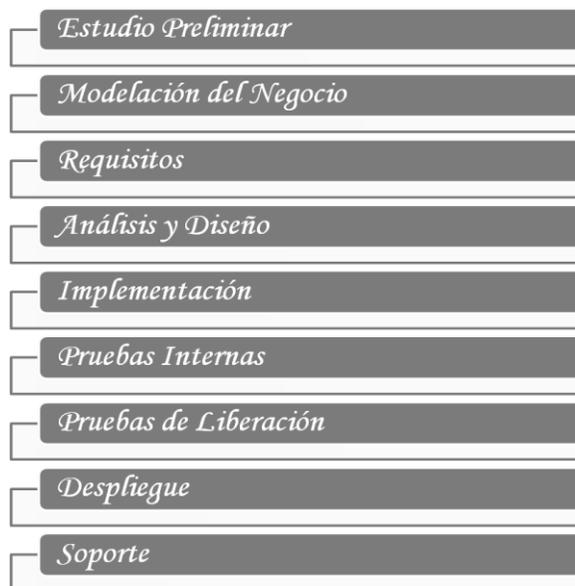


Figura 14. Ciclo de Vida

2.3.1 Estudio Preliminar

Esta es la primera etapa, donde a través de entrevistas con los clientes se obtiene la información necesaria para la realización del software. Se realiza la planeación del proyecto, así como las estimaciones de tiempo, esfuerzo y costo

Elementos críticos de la etapa:

- ✓ Necesidades del cliente.
- ✓ Planificación del proyecto.

2.3.2 Modelación del Negocio

En esta etapa se comprende cómo funciona el negocio, los procesos que necesita para que el software cumpla con su propósito. Se describe y se modela el negocio utilizando diferentes técnicas de modelación, como el Modelado de Casos de Uso del Negocio y como lenguaje de modelado se utiliza Business Process Modeling Notation (BPMN).

Elementos críticos de la etapa:

- ✓ Reglas del negocio.
- ✓ Modelación del negocio.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

2.3.3 Requisitos

En esta etapa se describen los casos de usos que responden a las funcionalidades que debe cumplir el sistema. El objetivo principal de esta etapa es desarrollar el modelo del sistema de software, donde se incluyen los requisitos funcionales y no funcionales.

Elementos críticos de la etapa:

- ✓ Requisitos funcionales.
- ✓ Planificación de la arquitectura.
- ✓ Requisitos no funcionales.
- ✓ Validación de artefactos.

2.3.4 Análisis y Diseño

En esta etapa se transforman los requisitos en un diseño del sistema, el cual se adapta para que se ajuste al entorno de implementación. En caso de ser necesario se pueden realizar cambios en los requisitos. Además se modela el sistema acorde a los requisitos recopilados en las entrevistas realizadas a los clientes. Se desarrollan diagramas de entidad-relación, diagramas de clases, diagrama de despliegue, se diseña la base de datos, entre otros elementos esenciales para lograr el producto deseado. Los modelos desarrollados en esta etapa son más formales y específicos de una implementación.

Elementos críticos de la etapa:

- ✓ Patrones de diseño para la implementación.
- ✓ Diseño de la base de datos.

2.3.5 Implementación

En esta etapa se implementan los artefactos generados en la etapa de análisis y diseño, creando scripts, ejecutables, ficheros de código y similares, acorde al lenguaje de programación que se utilice.

Elementos críticos de la etapa:

- ✓ Implementación de las funcionalidades.
- ✓ Comprobación del estado de la implementación.
- ✓ Acceso a datos.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

2.3.6 Pruebas Internas

En esta etapa cada proyecto evalúa los resultados de la implementación, comprobando cada componente a través de casos de prueba, listas de chequeo y demás recursos que el proyecto considere necesario para evaluar el software.

Elementos críticos de la etapa:

- ✓ Métodos de evaluación.

2.3.7 Pruebas de Liberación

Pruebas diseñadas e implementada por el Laboratorio Industrial de Pruebas de Software a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.

2.3.8 Despliegue

Durante esta etapa se procede a la entrega de la solución, así como a la instalación, configuración, prueba y puesta en marcha del software en el entorno real del cliente. Las pruebas de esta etapa incluyen pruebas de aceptación y pruebas pilotos. También deben realizarse en este período la capacitación y acompañamiento a clientes para asegurar que adquieran los conocimientos necesarios en la manipulación del software.

2.3.9 Soporte

Durante esta etapa y por un tiempo limitado el proyecto ofrecerá un servicio para resolver conflictos y problemas de usabilidad y rendimiento del software entregado al cliente, suministrándole actualizaciones y parches a errores.

Para las etapas de Despliegue y Soporte no se determinaron elementos críticos ni actividades en respuesta a los mismos. En estas etapas el personal que interviene en su mayoría es ajeno al equipo de desarrollo por lo que el mismo no interviene directamente en las variaciones que pueda sufrir el software durante estas etapas.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

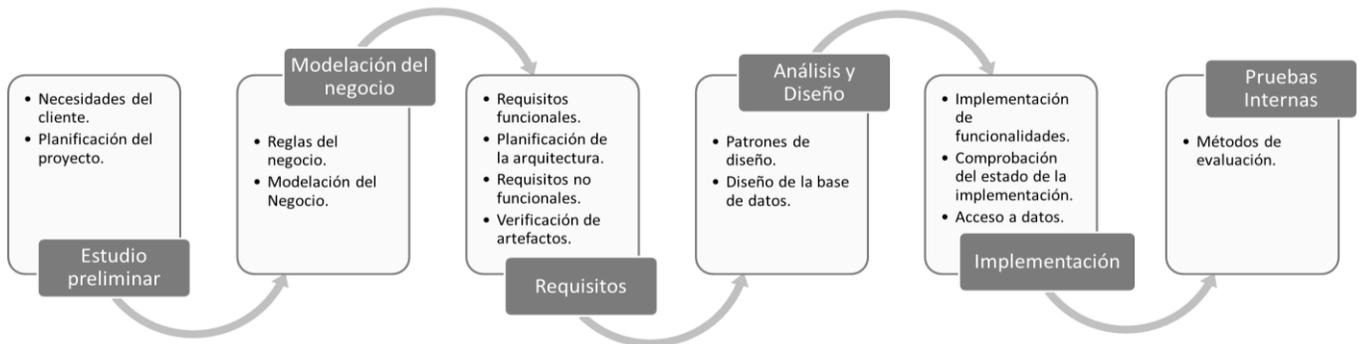


Figura 15. Elementos por etapas

2.4 Guía de actividades a realizar por etapas.

Se considera importante resaltar que el programa de mejora plantea actividades a cumplir durante las etapas del ciclo de vida del software, las que deberán ser cumplidas por el equipo de desarrollo del software. Las actividades que se plantean en la guía, que se presenta a continuación, son de apoyo a las actividades ya definidas en el programa de mejora, que van encaminadas específicamente a la funcionalidad del software.

La guía se encuentra representada a través de una tabla, para su análisis y mejor comprensión se explica su contenido a continuación: consta con 4 partes fundamentales, la primera parte es la relación de las etapas del ciclo de vida del proceso de mejora, la segunda parte son las actividades propuestas para cada etapa del ciclo de vida, la tercera parte es la descripción de la actividad y la última parte son las subcaracterísticas del estándar ISO/IEC 25010 SQuaRE a la que responde la actividad. Las tres últimas columnas son encabezadas por el elemento crítico que se definió para la etapa, y en respuesta al mismo son las actividades planificadas.

Etapas	Actividades por etapa	Descripción de las actividades	Subcaracterística de la ISO/IEC 25010 SQuaRE
	Necesidades del cliente.		
	Definir entrevista entre el analista y el cliente.	En esta actividad el analista debe definir con anterioridad las preguntas fundamentales para realizarles a los clientes. En la misma se debe detectar con	Adecuación.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

Estudio Preliminar		claridad las necesidades del cliente, y a partir de esto poder elaborar la planificación del proyecto en correspondencia al estudio realizado.	
	Planificación del proyecto.		
	Planificar en qué momento se realizarán las pruebas y verificaciones.	Esta actividad propone la planificación para comprobar el estado del proyecto según el hito a evaluar.	Exactitud.
Concebir estrategia para el levantamiento de los requisitos.	En esta actividad se planifica la estrategia que debe definirse para el levantamiento de los requisitos de manera que el analista logre captar la mayor cantidad de necesidades del cliente.	Exactitud.	
Modelación del Negocio	Reglas del negocio.		
	Establecer estrategia para redactar reglas del negocio.	En esta actividad se define la estrategia ¹ para lograr que exista una buena comunicación entre el analista y el cliente, y así redactar las reglas del negocio de forma clara y precisa para su entendimiento, cumpliendo con los estatutos y regulaciones	Adherencia a normas. Exactitud.

¹ En esta estrategia se pueden planificar diferentes actividades, ejemplo de ello sería: definir con anterioridad entrevista que contribuya a tener identificados los elementos que no deben faltar en las reglas del negocio. Identificar que los actores que intervengan en las reglas del negocio sean los adecuados (analista, proveedor de requisitos, arquitecto de información).

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	vigentes del gobierno.	
Modelación del Negocio.		
Describir mínimos detalles en la modelación.	<p>Esta actividad está encaminada al modelo del negocio, pues en el mismo deben estar reflejados los más mínimos detalles que influyan en el desarrollo del software.</p> <p>* Esta actividad está encaminada a la descripción de procesos del negocio, cumpliendo con las mismas especificaciones que para el modelo de negocio.²</p>	Adecuación.
Verificar la calidad de los artefactos.	<p>Esta actividad puede realizarse mediante la aplicación de listas de chequeo, con el objetivo de que el modelo del negocio tenga calidad antes de su firma y aprobación. Se pueden utilizar las siguientes listas de chequeo definidas en CALISOFT:</p> <ul style="list-style-type: none"> ✓ Lista de Chequeo Especificación de CU³. ✓ 5205_Lista de Verificación (Detallada) de PPQA a 	Adecuación.

² Esta actividad se realiza en el caso de que no se realice la etapa de modelación del negocio, sino modelación de procesos.

³ CU: casos de uso.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	<p>Proyecto v3.2 (en este caso se recomienda el uso del acápito Administración de Requisitos (REQM))</p> <p>* Esta actividad puede realizarse mediante la aplicación de listas de chequeo, con el objetivo de que la descripción de procesos del negocio tenga calidad antes de su firma y aprobación.⁴</p>	
Firmar el modelo del negocio.	<p>En esta actividad el modelo del negocio debe firmarse por las personas correctas (proveedor de requisitos, jefe de proyecto, analista) y en el tiempo establecido para que no se vea afectado en un futuro el proceso de desarrollo del software.</p>	Adecuación.
Requisitos funcionales.		
Validar los requisitos.	<p>En esta actividad los requisitos deben ser validados por los clientes representativos de la organización que se va a informatizar, para que el impacto de los cambios sea menor.</p>	Adecuación.

⁴ Esta actividad se realiza en el caso de que no se realice la etapa de modelación del negocio, sino modelación de procesos.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

Requisitos	Verificar correspondencia de los requisitos con la interfaz de usuario.	Esta actividad consiste en la comprobación manual de la presencia de cada uno de los requisitos en el prototipo de interfaz de usuario que es presentado al cliente.	Exactitud. Adecuación.
	Planificación de la arquitectura.		
	Definir el tipo de sistema que se va a desarrollar.	En esta actividad se determina el tipo de sistema que se va a desarrollar (sistema web distribuido o centralizado, sistema de escritorio distribuido o centralizado).	Adecuación.
	Establecer entorno de desarrollo.	En esta actividad se planifican los elementos de la arquitectura que se deben tener en cuenta para que respondan a la modelación del negocio y necesidades del cliente.	Adecuación
	Requisitos no funcionales		
	Diagnosticar condiciones técnicas.	En esta actividad se comprueba las condiciones técnicas con que cuenta el cliente para que el software funcione (hardware: RAM, disco duro, microprocesador, tarjeta gráfica en caso de ser necesaria, periféricos, conectividad).	Exactitud.
Verificación de artefactos.			

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	<p>Verificar los artefactos de la etapa.</p>	<p>En esta actividad se emplean listas de chequeo para validar los artefactos de la etapa, los mismos son de gran importancia para el posterior desarrollo del software. Se pueden utilizar las siguientes listas de chequeo:</p> <ul style="list-style-type: none"> ✓ 083_Listas de Chequeo Especificación de Requisitos. ✓ 5205_Lista de Verificación (Detallada) de PPQA a Proyecto v3.2 (en este caso se recomienda el uso del acápite Administración de Requisitos (REQM)). 	<p>Exactitud.</p>
<p>Análisis y Diseño</p>	<p>Patrones de diseño.</p>		
	<p>Definir patrones de diseño a emplear en la implementación.</p>	<p>En esta actividad se proponen patrones que contribuyen a la funcionalidad del software y se identifica la utilización de los mismos. Estos patrones deben reflejarse en el artefacto 0120_6 <i>Arquitectura Vista de Entorno de Desarrollo Tecnológico</i>.</p>	<p>Adecuación.</p>
	<p>Diseño de la base de datos.</p>		
<p>Documentar el modelo entidad-relación.</p>	<p>Esta actividad propone la documentación del modelo entidad-relación, con el fin de una posterior consulta del mismo,</p>	<p>Adecuación.</p>	

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	además de tener registrado para caso de aclaración a las entidades relevantes, así como sus interrelaciones y propiedades.	
Utilizar patrones de diseño.	Esta actividad está encaminada para establecer un correcto funcionamiento del negocio del sistema, y así lograr una estructura estándar dentro del modelo de la base de datos.	Exactitud.
Aplicar buenas prácticas en conjunto con los patrones de diseño.	<p>En esta actividad se pueden llevar a cabo buenas prácticas como:</p> <ul style="list-style-type: none"> ✓ Vistas para las consultas contra las tablas, para que los usuarios no accedan directamente a las tablas. ✓ No utilizar prefijos para las tablas, usarlos para las vistas y procedimientos. ✓ Utilizar sufijos para las columnas. <p><i>Nota: el proyecto debe sentirse en la libertad de utilizar estas prácticas acorde a sus características específicas.</i></p>	Exactitud.
Establecer nomenclatura en las	En esta actividad las nomenclaturas facilitan el trabajo con las tablas, pues su objetivo es nombrar las tablas de forma más	Adecuación.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	tablas.	reducida y clara. Además se deben definir las pautas que deben cumplir estos nombres para una mejor estandarización.	
	Establecer correlación entre la normalización y desnormalización de la base de datos.	Esta actividad propone la adecuada correlación entre la normalización y desnormalización, pues si la base de datos se encuentra estrictamente en una de estas formas puede atentar contra el buen funcionamiento y cumplimiento de los objetivos de dicha base de datos.	Exactitud.
	Utilizar índices en las tablas.	En esta actividad se propone la utilización de índices en las tablas, (además de las llaves primarias y llaves foráneas), pues los mismos agilizan la base de datos.	Exactitud.
	Validar la calidad del diseño de la base de datos.	En esta actividad se debe utilizar una lista de chequeo (deberá ser creada por el proyecto) para la verificación de la calidad.	Exactitud.
	Implementación de funcionalidades.		
Implementación	Utilizar patrones definidos en el diseño para la implementación.	En esta actividad se aplican los patrones definidos en el diseño, de manera que se establezca un correcto funcionamiento a partir	Adecuación.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	de los requisitos, de forma que exista la menor dependencia entre un caso de uso y otro para contribuir a la funcionalidad.	
Implementar funcionalidades con exactitud.	Esta actividad está encaminada a que la implementación de las funcionalidades se haga basada en las necesidades del cliente, no querer hacer más de lo que está especificado en los requisitos que ya han sido aprobados.	Exactitud.
Comprobación del estado de la implementación.		
Verificar la trazabilidad de los casos de uso implementados.	En esta actividad se comprueba que en los casos de uso implementados estén reflejados todos los requisitos definidos.	Adecuación.
Acceso a datos.		
Controlar permisos desde la programación.	En la implementación el programador define los permisos que van a tener los usuarios y de esta manera los mismos solo podrán acceder a lo que realmente les compete acorde al tipo de usuario.	Exactitud. Adherencia a normas.
Métodos de evaluación.		
Verificar el procesamiento e implementación adecuada de las reglas	En esta actividad se comprueba que en la implementación hayan sido cumplidas las reglas del negocio, haciendo énfasis en las	Adecuación. Adherencia a normas.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

Pruebas Internas	del negocio.	regulaciones vigentes del gobierno en aquellas reglas que así lo requieran, pues así lo solicita cualquier software que implemente procesos gubernamentales.	
	Desarrollar capacitación a probadores.	En esta actividad se debe capacitar a los probadores. De esta forma pueden tener una idea general de lo que permite el sistema.	Exactitud.
	Definir escenario para aplicar las pruebas.	En esta actividad se aseguran los requerimientos para aplicar las pruebas de manera que los resultados sean los más reales posibles y estén en correspondencia con las condiciones técnicas que posee el usuario final.	Adecuación. Exactitud.
	Comprobar implementación de casos de uso.	En esta actividad se debe comprobar que en la implementación estén reflejados todos los casos de uso que fueron identificados. Esta actividad puede ser cumplida mediante la aplicación de listas de chequeos creadas en el propio proyecto.	Exactitud.
	Aplicar pruebas al software.	En esta actividad las pruebas serán aplicadas para comprobar	Exactitud. Adecuación.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	<p>las funcionalidades. Pueden llevarse a cabo pruebas de:</p> <ul style="list-style-type: none"> ✓ Caja blanca. ✓ Pruebas de rendimiento. ✓ Pruebas funcionales. ✓ Pruebas de carga y stress. <p><i>Nota:</i> las pruebas deben realizarse tanto a los módulos como cuando se haga la integración del sistema.</p>	
Utilizar herramientas de prueba.	<p>En esta actividad las herramientas deben permitir evaluar el software en correspondencia con el tipo de aplicación, las mismas pueden ser manuales o automáticas. Se mencionan a continuación herramientas que pueden ser utilizadas:</p> <ul style="list-style-type: none"> ✓ JMeter ✓ Selenium IDE ✓ Diseño de casos de prueba. ✓ Openload. ✓ PHPUnit. 	Adecuación.
Confirmar verificación de los datos.	<p>En esta actividad mediante la aplicación de los casos de pruebas se realizan pruebas simulando el trabajo del usuario final, verificando que los campos</p>	<p>Exactitud. Adecuación.</p>

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

	solo acepten los datos que le fueron definidos. Se recomienda que los proyectos definan el diccionario de datos y de esta manera es mucho más fácil para los diseñadores de casos de prueba definirlos.	
Verificar permisos de acceso.	En esta actividad mediante la creación de transacciones específicas para cada tipo de usuario se verifica los permisos que tiene en la aplicación.	Exactitud. Adherencia a normas.
Realizar pruebas que permitan controlar el acceso.	Esta actividad es la encargada de que un usuario solo pueda acceder a las funciones y datos que tenga permitido y que solo los usuarios con acceso al sistema y a la aplicación están habilitados para accederla.	Exactitud. Adherencia a normas.
Verificar la cantidad de peticiones por usuarios concurrentemente contra el tiempo de respuesta de la aplicación.	En esta actividad utilizando las pruebas de volumen se comprueba la cantidad de usuarios con que el sistema es capaz de interactuar siempre que cumpla con el tiempo definido en los requisitos del sistema.	Adecuación. Exactitud.
Aplicar métricas para evaluar la funcionalidad.	En esta actividad se propone la utilización de las métricas (ver Anexo 2 y Anexo 3) para evaluar	Exactitud.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

		la funcionalidad, en las mismas se tuvieron en cuenta elementos a los cuales se le planificaron actividades para contribuir a su aseguramiento desde etapas anteriores.	Exactitud. Adherencia a normas.
Pruebas de Liberación.	Solicitar a CALISOFT la realización de las actividades que fueron definidas en Pruebas Internas.	En esta actividad el centro CEGEL puede proponerle a CALISOFT la aplicación de las actividades planificadas en la etapa de Pruebas Internas para aquellos proyectos que así lo considere necesario el centro.	
	Realizar laboratorio de aceptación.	Esta actividad propone la realización de un laboratorio de aceptación con el cliente de manera que queden claras las funcionalidades implementadas.	Adecuación. Exactitud. Adherencia a normas.

Luego de concluida la etapa de Pruebas de Liberación se propone la actividad: realizar laboratorio de aceptación, donde el máximo encargado de esta actividad será el centro. Si el cliente acepta el software se entiende que en el mismo se encuentran las funcionalidades requeridas, contribuyendo a que durante el despliegue no existan grandes cambios en la aplicación que influyan en las funcionalidades del software.

En la etapa de Soporte, interviene el personal preparado para brindar el mismo a la aplicación y no influye directamente el equipo de desarrollo del proyecto, en esta etapa se brindan soluciones al software en cuanto a usabilidad y rendimiento.

Capítulo 2. Propuesta de una guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico

En la etapa de pruebas internas se planificaron actividades en respuesta a las pruebas de función, volumen y seguridad, lo cual es importante resaltar, pues son pruebas de gran envergadura que se llevan a cabo en la Universidad para contribuir a la funcionalidad del software:

- ✓ Confirmar verificación de los datos.
- ✓ Realizar pruebas que permitan controlar el acceso.
- ✓ Verificar permisos de acceso.
- ✓ Cantidad de peticiones por usuarios concurrentemente contra el tiempo de respuesta de la aplicación.

Es primordial destacar que el hecho de que la guía esté encaminada al centro CEGEL, requiere un mayor énfasis en la seguridad, pues la información relacionada con el gobierno debe ser tratada con la mayor discreción posible, por los resultados negativos que pudiera traer consigo el mal manejo de la misma.

2.5 Conclusiones parciales

La guía es un instrumento que permite la planificación de actividades a cumplir, en este caso encaminadas al aseguramiento de la funcionalidad como característica de la calidad del software. El estudio detallado del proceso de mejora, así como de los libros de cada una de las áreas de procesos facilitó la definición de elementos críticos a tener en cuenta para el aseguramiento de la funcionalidad en cada una de las etapas del proceso de mejora.

En respuesta a cada uno de los elementos críticos se planificaron actividades a cumplir por los miembros del proyecto de desarrollo, resaltando siempre que la guía es un instrumento de apoyo, pues el proceso de mejora en sí planifica actividades para cada una de sus etapas.

Para la planificación de las actividades se tuvo en cuenta las subcaracterísticas de la funcionalidad según el estándar seleccionado, por lo que se establece la relación entre actividades y subcaracterísticas de manera que se pueda apreciar cómo responden las actividades a la funcionalidad a: exactitud, adecuación y adherencia a normas.

Capítulo 3. Validación de la propuesta de solución

3.1 Introducción al capítulo

En el presente capítulo se hace la validación de la guía propuesta teniendo en cuenta las valoraciones y criterios de un panel de especialistas en el tema y el empleo de una representación gráfica de los datos recogidos en la encuesta realizada a los mismos. Se utilizaron técnicas propuestas por el método Delphi, basado en el criterio de especialistas para la obtención y modelado de estos criterios.

3.2 Método Delphi

El método Delphi, deriva su nombre del antiguo oráculo de Delphos. Consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a un acontecimiento dado. Las estimaciones de los expertos se realizan por encuesta en sucesivas rondas, anónimas, con el objetivo de lograr un consenso entre los mismos. Este método presenta varias características fórmula **(Rodríguez Perón, y otros, 2010)**:

- ✓ Anonimato: Durante el Delphi ningún especialista conoce la identidad de los otros que componen el grupo de debate.
- ✓ Iteración y realimentación controlada: La iteración se consigue al presentar varias veces el mismo cuestionario, para que los especialistas vayan conociendo los diferentes puntos y puedan ir modificando su opinión.
- ✓ Respuesta del grupo en forma estadística: La información que se presenta a los especialistas no es solo el punto de vista de la mayoría sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

Dicho método debe llevarse a cabo siguiendo los siguientes pasos:

- ✓ Selección de los expertos.
- ✓ Elaboración del cuestionario para la validación de la propuesta.
- ✓ Cálculo de concordancia entre los expertos.

En la presente investigación no se cuenta con especialistas con un máximo de competencia, que puedan ser catalogados como expertos. Por tal motivo la validación se realizará mediante especialistas, en este término se clasifican a las personas cuyas opiniones son válidas para fines investigativos.

Selección de especialistas.

Este método no tiene definido un número exacto de especialistas, sin embargo un grupo de investigadores del tema indican que este número debería estar comprendido entre un mínimo de cinco y un máximo de treinta especialistas.

Para la selección se aplicó una encuesta (ver [Anexo 4](#)) para establecer los cinco especialistas más idóneos que están en disposición de conformar el panel. Con la encuesta se determinó el grado de competencia de los mismos, mediante la aplicación de la siguiente fórmula (**Rodríguez Perón, y otros, 2010**):

$$k = \frac{1}{2}(kc + ka)$$
 Donde **k**: coeficiente de competencia.

kc es el coeficiente de conocimiento o información que tiene el especialista acerca del problema, calculado sobre la valoración del propio especialista en una escala del 0 al 10 y multiplicado por 0,1; de esta forma, la evaluación "0" indica que el especialista que no tiene absolutamente ningún conocimiento de la problemática correspondiente, mientras que la evaluación "10" significa que el especialista tiene pleno conocimiento de la problemática tratada.

ka es el coeficiente de argumentación o fundamentación de los criterios del especialista, obtenido como resultado de la suma de los puntos alcanzados a partir de una tabla patrón que se muestra a continuación:

Fuentes de argumentación	Grado de influencia de cada una de las fuentes en sus criterios.		
	Alto	Medio	Bajo
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

Para la interpretación de los resultados se debe tener en cuenta que:

Si $0,8 \leq k \leq 1$ el coeficiente de competencia es alto.

Si $0,5 \leq k < 0,8$ coeficiente de competencia medio.

Si $k < 0,5$ coeficiente de competencia bajo.

Después de la aplicación de la fórmula planteada con anterioridad se obtuvo los especialistas con disposición de integrar el panel que tenían las condiciones para conformar el mismo. Se muestra gráficamente los resultados logrados:

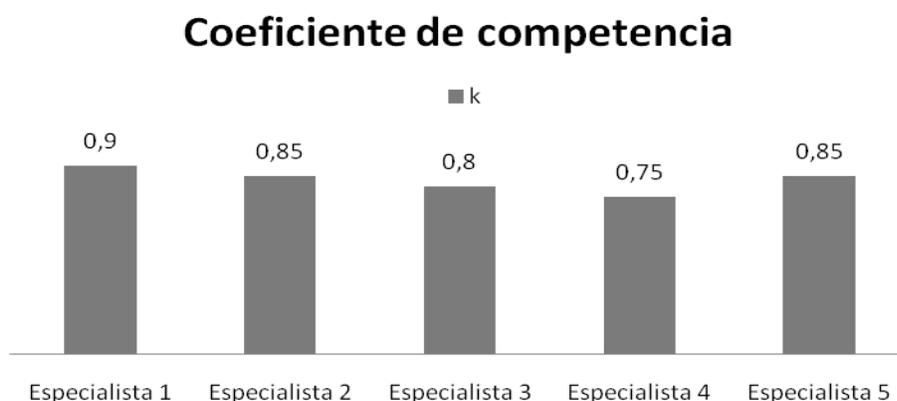


Figura 16. Coeficiente de competencia de los especialistas.

Se puede apreciar en la gráfica el nivel de competencia de cada especialista, a partir de donde se puede determinar que el promedio de coeficiente de competencia es de 0,83, lo que establece en el nivel alto de competencia los criterios de los especialistas.

Los especialistas seleccionados han trabajado directamente en proyectos de desarrollo, tanto de convenio con Venezuela como proyectos nacionales de gran envergadura. De los especialistas dos de ellos tienen 4 años de experiencia, uno con 3, uno con 5 y uno con 7 años respectivamente. El nivel de coeficiente de competencia de estos especialistas es alto para el caso de cuatro de ellos, y uno con nivel medio, por lo que sus criterios aportan validez a la solución. Se tomó el criterio del especialista con un nivel medio para que exista otro punto de vista de la solución, el mismo no afecta los resultados del método, pero sí puede brindar nuevos elementos a considerar en la propuesta. Los especialistas se desempeñan en roles como líder de proyecto, analista, arquitectos de software y administrador de calidad.

Aplicación del método.

Para aplicar el método se le envía a cada uno de los especialistas por separado para no condicionar las respuestas de los demás la encuesta, así como la propuesta de la guía. Es válido aclarar que el proceso de evaluación es iterativo, o sea, se realizan varias rondas en caso de ser necesario hasta que haya un

Capítulo 3. Validación de la propuesta de solución.

consenso entre los especialistas. En la encuesta se formulan seis preguntas dirigidas a valorar los elementos definidos para asegurar la funcionalidad, durante las etapas del ciclo de desarrollo de software basado en el programa de mejora que lleva a cabo la Universidad. A continuación se muestra gráficamente toda la información recopilada a partir de la aplicación del instrumento de manera que se puedan comprender mejor los resultados.

En las respuestas de los especialistas sobre la pregunta del cuestionario: ¿Considera que las necesidades del cliente y la planificación del proyecto son elementos fundamentales en la etapa de Estudio Preliminar que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico? evaluaron lo siguiente:

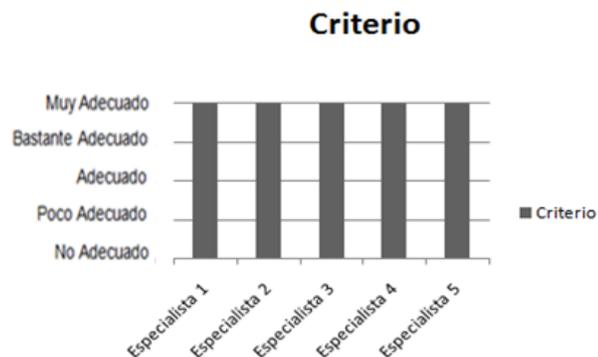


Figura 17. Criterio de evaluación de la pregunta 1 por especialistas.

Por ciento del criterio de evaluación de los especialistas



Figura 17. Por ciento del criterio de evaluación de los especialistas para la pregunta 1.

El 100 % de los especialistas valora como Muy Adecuado la definición de las necesidades del cliente y la planificación del proyecto como se aprecia en la Figura 18.

En las respuestas de los especialistas sobre la pregunta del cuestionario: ¿Considera que las Reglas del Negocio y Modelación del Negocio son elementos fundamentales en la etapa de Modelación del Negocio que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico? evaluaron lo siguiente:

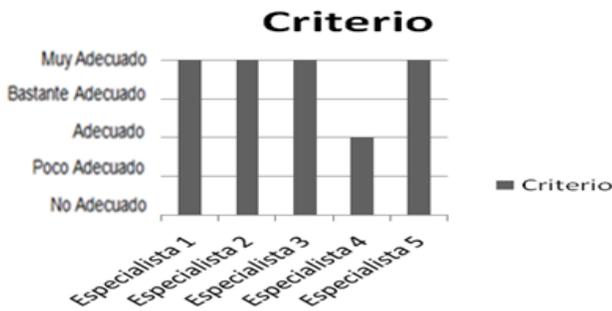


Figura 19. Criterio de evaluación de la pregunta 2 por especialistas.

Por ciento del criterio de evaluación de los especialistas

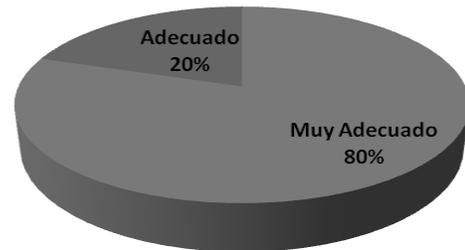


Figura 20. Por ciento del criterio de evaluación de los especialistas para la pregunta 2.

El 20 % y el 80 % de los especialistas valora como Adecuado y Muy Adecuado respectivamente la definición de las Reglas del Negocio y Modelación del Negocio como se aprecia en la Figura 20.

En las respuestas de los especialistas sobre la pregunta del cuestionario: ¿Considera que los requisitos funcionales, la planificación de la arquitectura, los requisitos no funcionales y la verificación de artefactos son elementos fundamentales en la etapa de Requisitos que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico? evaluaron lo siguiente:

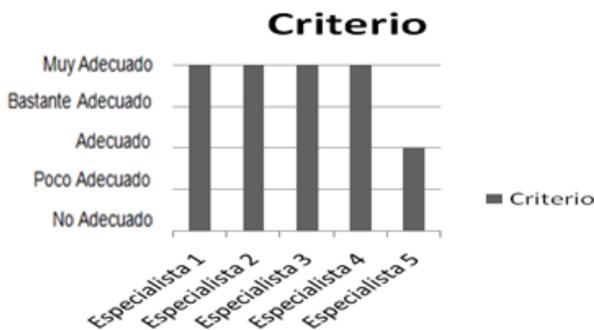


Figura 21. Criterio de evaluación de la pregunta 3 por especialistas.

Por ciento del criterio de evaluación de los especialistas



Figura 22. Por ciento del criterio de evaluación de los especialistas para la pregunta 3.

Capítulo 3. Validación de la propuesta de solución.

El 20 % y el 80 % de los especialistas valora como Adecuado y Muy Adecuado respectivamente la definición de los requisitos funcionales, la planificación de la arquitectura, los requisitos no funcionales y la verificación de artefactos como se aprecia en la Figura 22.

En las respuestas de los especialistas sobre la pregunta del cuestionario: ¿Considera que los patrones de diseño y el diseño de la base de datos son elementos fundamentales en la etapa de Análisis y Diseño que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico? evaluaron lo siguiente:



Figura 23. Criterio de evaluación de la pregunta 4 por especialistas.

Por ciento del criterio de evaluación de los especialistas



Figura 24. Por ciento del criterio de evaluación de los especialistas para la pregunta 4.

El 40 %, el 20 % y el 40 % de los especialistas valora como Adecuado, Bastante Adecuado y Muy Adecuado respectivamente la definición de los patrones de diseño y el diseño de la base de datos como se aprecia en la Figura 24.

En las respuestas de los especialistas sobre la pregunta del cuestionario: ¿Considera que la implementación de funcionalidades, la comprobación del estado de la funcionalidad y el acceso a datos son elementos fundamentales en la etapa de Implementación que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico? evaluaron lo siguiente:

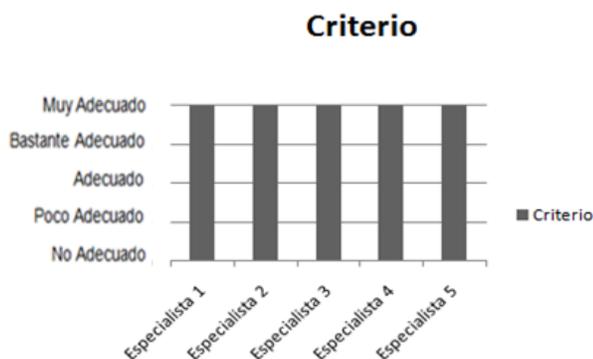


Figura 25. Criterio de evaluación de la pregunta 5 por especialistas.

Por ciento del criterio de evaluación de los especialistas



Figura 26. Por ciento del criterio de evaluación de los especialistas para la pregunta 5.

El 100 % de los especialistas valora como Muy Adecuado la definición de la comprobación del estado de la funcionalidad y el acceso a datos como se aprecia en la Figura 26.

En las respuestas de los especialistas sobre la pregunta del cuestionario: ¿Considera que los métodos de evaluación es un elemento fundamental en la etapa de pruebas internas que influye en la funcionalidad de los sistemas informáticos de Gobierno Electrónico? evaluaron lo siguiente:

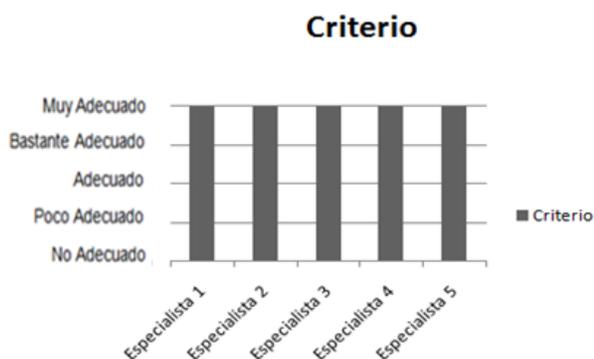


Figura 27. Criterio de evaluación de la pregunta 6 por especialistas.

Por ciento del criterio de evaluación de los especialistas



Figura 28. Por ciento del criterio de evaluación de los especialistas para la pregunta 6.

El 100 % de los especialistas valora como Muy Adecuado métodos de evaluación como se aprecia en la Figura 28.

Capítulo 3. Validación de la propuesta de solución.

La primera ronda arrojó de manera general los siguientes resultados:

	Muy Adecuado	Bastante Adecuado	Adecuado	Poco adecuado	No Adecuado	Total
Pregunta 1	5					5
Pregunta 2	4		1			5
Pregunta 3	4		1			5
Pregunta 4	2	1	2			5
Pregunta 5	5					5
Pregunta 6	5					5

Figura 29. Resumen de respuestas.

Visualmente se puede apreciar que los resultados arrojados por la encuesta demuestran la aceptación de los elementos críticos por parte de los especialistas, pero para mayor confirmación se analizan los resultados estadísticamente, por lo que se procede a determinar el grado de concordancia entre los especialistas, con respecto a las evaluaciones que hicieron. Para ello se determinó utilizar el coeficiente de Kendall (W), de la herramienta Paquete Estadístico para las Ciencias Sociales (SPSS) para la explotación de los datos, alcanzándose como resultado:

Kendall's W Test

Ranks	
	Mean Rank
P1	2,80
P2	4,40
P3	3,70
P4	4,40
P5	4,40
P6	1,30

Test Statistics	
N	5
Kendall's W ^a	,722
Chi-Square	18,056
df	5
Asymp. Sig.	,003

a. Kendall's Coefficient of Concordance

Figura 3018. Tabla estadística.

Esta tabla muestra el número de casos válidos (N), el valor del estadístico W (W de Kendall), su transformación en Chi-cuadrado (Chi-Square), sus grados de libertad (df) y el nivel crítico (Asymp. Sig.).

Para saber si W de Kendall es significativamente distinta de 0 se realizó una prueba de hipótesis donde:

Capítulo 3. Validación de la propuesta de solución.

H_0 = No hay concordancia entre los especialistas.

H_1 = Hay concordancia entre los especialistas.

El coeficiente W de Kendall es una medida de la concordancia de los especialistas y por definición del Método Delphi, el resultado debe moverse en un rango de 0 a 1 y debe ser siempre $W > 0,5$ porque cuanto más se acerque el coeficiente a 1, mayor será el grado de concordancia entre los especialistas. En el estudio realizado resultó ser un aproximado de $W = 0,72$, por tanto la propuesta resultó ser aceptada y con un nivel de concordancia alto con respecto a los criterios que fueron evaluados.

La tabla muestra que el valor del nivel crítico (0,003) es menor que 0,05, por tanto se puede rechazar la hipótesis de concordancia nula y concluir que entre las puntuaciones de las variables estudiadas existe asociación significativa.

Tras la primera ronda se constató que no era necesario realizar más rondas, pues existía un nivel de concordancia entre los especialistas evaluando los elementos seleccionados entre Muy Adecuados y Adecuados. Se resaltó como actividad imprescindible el diseño de la base de datos por parte de los especialistas para contribuir a la funcionalidad del software.

3.3 Conclusiones parciales.

En el presente capítulo se validó la propuesta de la guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico mediante el Método Delphi. A través de dicho método se realizó el análisis de los criterios de cada uno de los especialistas consultados y se determinó el índice de aceptación que tiene la propuesta, obteniéndose concordancia en el criterio de cada uno de ellos acerca de la investigación realizada y una alta probabilidad de éxito de ser implementada la guía en el centro CEGEL, cumpliéndose así la hipótesis H_1 : Hay concordancia entre los especialistas. La aceptación de la guía por parte de los especialistas es una muestra sólida de los beneficios que puede traer la misma al ser aplica en el centro, contribuyendo al aseguramiento de la funcionalidad como característica a cumplir por el software desarrollado.

Conclusiones.

Conclusiones

Con la realización del presente trabajo de diploma se pudo determinar que el centro CEGEL no consta de una guía para contribuir al aseguramiento de la funcionalidad como característica de la calidad del software. Para dar respuesta a esta situación se seleccionó como estándar para desarrollar la propuesta al ISO/IEC 25010 SQuaRE. Además se tuvo en cuenta el modelo CMMI, ya que por el mismo se encuentra basado el ciclo de vida del proceso de mejora.

Para cada una de las etapas del ciclo de vida del proceso de mejora que se encuentran en el desarrollo del software se determinaron elementos a tener en cuenta para la planificación de actividades que respondan a los mismos. Dichos elementos influyen directamente en la funcionalidad del producto, por lo que son imprescindibles en el desarrollo del mismo.

Es importante resaltar que el proceso de mejora plantea actividades concretas a cumplir en respuesta a las diferentes etapas y artefactos generados en cada una de ellas, por lo que la guía constituye un artefacto de apoyo destinado específicamente a la funcionalidad del producto.

Con la aplicación del método Delphi se constató la aceptación de los elementos críticos, los cuales permitieron planificar actividades en respuesta a los mismos para cada una de las etapas. Estas actividades contribuyen al aseguramiento de la funcionalidad como característica de la calidad del software. El método Delphi mostró un nivel de aceptación por parte de los especialistas entre Muy Adecuado y Adecuado, cumpliéndose así la hipótesis H_1 : Hay concordancia entre los especialistas, mostrando con un nivel de aceptación de 99,7 %, de que los resultados alcanzados después de la aplicación de la guía sean satisfactorios, por tanto la aplicación de la misma en el centro CEGEL tendrá efectos positivos en el desarrollo del software de los proyectos de este centro.

Recomendaciones

Recomendaciones

- ✓ Aplicar la guía propuesta en el centro CEGEL y analizar los resultados arrojados tras la aplicación de la misma para una actualización en caso de ser necesario.
- ✓ Realizar periódicamente el estudio y la investigación de la funcionalidad en los sistemas informáticos de Gobierno Electrónico.
- ✓ Definir un plan de capacitación para todas las personas inmersas en el proceso de desarrollo del software de manera que puedan utilizar la guía.
- ✓ Planificar actividades para las etapas de Despliegue y Soporte con un estudio profundo del papel que juega la funcionalidad en estas etapas.

Bibliografía

Bibliografía

- ✓ 9126, ISO/IEC. 2001.. ISO/IEC 9126-1 Software engineering—Product quality—Part 1: Quality Model. 2001.
- ✓ 9126-1, ISO/IEC. 2005. Norma Cubana INGENIERÍA DE SOFTWARE—CALIDAD DEL PRODUCTO—PARTE 1: MODELO DE LA CALIDAD, ISO/IEC 9126-1. 2005.
- ✓ Axentia. 2006. Una introducción a CMMI. [En línea] 2012. [Consultado el: 1 de noviembre de 2011.] <http://es.scribd.com/doc/69546265/WP03-Una-Introduccion-a-CMMI>.
- ✓ Bichachi, Diana S. 2004. El uso de las Listas de Chequeo (Check-List) como herramienta para controlar la calidad de la ley. 2004. [En línea] 2004. [Consultado el: 23 de diciembre de 2011.] http://www.salvador.edu.ar/vrid/iiefqs/tr_check_list.pdf
- ✓ CALISOFT. 2012. Centro Nacional de Calidad del Software. [En línea] 2012. [Consultado el: 15 de marzo de 2012.] <http://calisoft.uci.cu/index.php/proceso-de-mejora>.
- ✓ Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. 2004. ARQUITECTURAS DE SOFTWARE. 2004.
- ✓ Centro Nacional de Calidad del Software. 2009. Libro de Proceso para la Administración de Acuerdos con Proveedores. La Habana : s.n., 2009.
 - . 2009. Libro de Proceso para la Administración de Requisitos. La Habana : s.n., 2009.
 - . 2009. Libro de Proceso para la Planeación del Proyecto. La Habana : s.n., 2009.
 - . 2009. Libro de Proceso para Medición y Análisis. La Habana : s.n., 2009.
 - . 2009. Libro de Proceso para Monitoreo y Control de Proyecto. La Habana : s.n., 2009.
 - . 2009. Libro de Proceso para PPQA. La Habana : s.n., 2009.
 - . 2008. Lineamientos de Calidad de Software. La Habana : s.n., 2008.
 - . 2008. Roles y responsabilidades. La Habana : s.n., 2008.

Bibliografía

- ✓ *Departement of Software and IT Engineering. 2007CD 25010, Software engineering-Software product Quality Requirements and Evaluation (SQuaRE) Quality model. Canada. 2007*
- ✓ *García., Ó. G. (2009). SQuaRE: Una unificación de normas para la especificación de requisitos y la evaluación de la calidad.*
- ✓ *Herrera, L., Ortiz, M., & Blank, I. (2005). Pruebas Funcionales.*
- ✓ *ISO 9126 vs. SQuaRE. 2008*
- ✓ *Machado Peña, Yadira, y otros. 2012. Estrategia de pruebas funcionales aplicada en CALISOFT para garantizar la calidad del software. La Habana : s.n., 2012.*
- ✓ *Método Delphi para la identificación de prioridades de ciencia e innovación tecnológica. Rodríguez Perón, Dr.C. José Miguel; Aldana Vilas, Dra. Laura; Villalobos Hevia, MSc. Nelson. 2010. La Habana. ISSN 1561-3046*
- ✓ *MG Soluciones Informaticas. 2011. MG Soluciones Informáticas. [En línea] [Consultado el: 9 de febrero de 2012.] <http://www.mginformatica.com.ar/modelo-de-calidad.htm>*
- ✓ *Minguet Melián, Jesús M y Hernadez Ballesteros, Juan Francisco. 2003. La calidad del software y su medida. Madrid : CENTRO DE ESTUDIOS RAMON ARECES, S.A, 2003.*
- ✓ *Modelos de calidad web. Clasificación de métricas.2006*
- ✓ *Modelo de Madurez de la Capacidad del Software. Pérez, Carmelo López. 2004. 2004, Revista de Ingeniería Informática del CIIRM. ISSN: 1698-8841*
- ✓ *Muñoz, Coral Calero. 2005. MODELOS DE CALIDAD. España : s.n., 2005.*
- ✓ *Oficina presidencial de tecnología de la Información y Comunicación. [En línea] [Consultado el: 31 de marzo de 2012.] <http://www.optic.gob.do/eGobierno/SobreeGob/tabid/76/Default.aspx>*
- ✓ *Pressman, Roger S. 2005. Ingeniería de Software. Un enfoque práctico. La Habana : Felix Varela, 2005.*

Bibliografía

- ✓ RAE. 2012. *Real Academia de la Lengua Española*. [En línea] 2012. [Consultado el: 24 de enero de 2012.] <https://www.rae.es>.
- ✓ Rodríguez, I. A. (2011). *Catálogo automatizado de métricas de calidad para evaluar los productos en las pruebas*. La Habana.
- ✓ Scalone, Fernanda. 2006. *Estudio comparativo de los modelos y estándares del software*. Buenos Aires : s.n., 2006.
- ✓ *Sistemas de gestión de la calidad y desempeño, importancia de las prácticas de gestión del conocimiento y de I+D*. Camisón Zornoza, César, Boronat Navarro, Montse y Villar López, Ana. 2009. 1, 2009, Vol. 18. ISSN 1019-6838.
- ✓ Software Engineering Institute. SIE. [En línea] [Consultado el: 22 de enero de 2012.] <https://www.sei.emu.edu>.
- ✓ Vega Lebrún, Carlos, Rivera Prieto, Laura Susana y García Santillán, Arturo. 2008. *Mejores prácticas para el establecimiento y aseguramiento de la calidad de software*. 2008.

Anexo 1. Entrevista aplicada a los proyectos del Centro para verificar cómo aseguran la funcionalidad en sus productos.

Conociendo que funcionalidad del software no es más que “la capacidad del producto de software para proveer las funciones que satisfacen las necesidades explícitas e implícitas cuando el software se utiliza bajo condiciones específicas”.

Responda las siguientes preguntas.

1. ¿Utilizan modelos u estándares para asegurar la calidad del software?
¿Cuál? _____
2. ¿Utilizan algún modelo u estándar para asegurar la funcionalidad del software en sus productos?
__ Modelo FURPS
__ Modelo DROMEY
__ Modelo WQM
__ MODELO CMMI
__ Estándar ISO 9126
__ Estándar ISO/IEC 25010 SQUARE
__ Otros. ¿Cuál? _____
3. ¿Aplican métricas para medir la funcionalidad del software?
__ Adecuación ¿En qué etapa del desarrollo de software?
__ Exactitud ¿En qué etapa del desarrollo de software?
__ Adherencia a normas ¿En qué etapa del desarrollo de software?
4. ¿Durante el proceso de desarrollo comprueban la funcionalidad del software con el usuario?
__ Pruebas de comunicaciones. ¿En qué etapa del desarrollo?
__ Pruebas de rendimiento. ¿En qué etapa del desarrollo?
__ Pruebas de Volumen ¿En qué etapa del desarrollo?
__ Pruebas de sobrecarga. ¿En qué etapa del desarrollo?
__ Pruebas de facilidad de uso. ¿En qué etapa del desarrollo?
__ Pruebas de Configuración. ¿En qué etapa del desarrollo?
__ Pruebas de Almacenamiento. ¿En qué etapa del desarrollo?
__ Lista de chequeo de presentación y funcionalidad. ¿En qué etapa del desarrollo?

__Otros. ¿Cuál?_____ ¿En qué etapa del desarrollo?

Anexo 2. Métricas para evaluar funcionalidad propuestas en la tesis de maestría: “Catálogo automatizado de métricas de calidad para evaluar los productos en las pruebas”

Métricas para la medición de la característica funcionalidad.

✓ Métrica Idoneidad.

Nombre de la métrica	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Adecuación funcional	$X = 1 - A/B$ A - Número de problemas detectados en las funcionalidades del sistema. B - Número de especificaciones de requisitos evaluadas.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará más adecuada	X = Contable/ Contable A = Contable B = Contable
Compleitud de la implementación funcional	$X = 1 - A/B$ A - Número de funciones perdidas detectadas en la evaluación. B - Número de funciones descritas en especificación de requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	X = contable/ Contable A = Contable B = Contable
Cobertura de la implementación funcional	$X = 1 - A/B$ A - Número de funciones	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará	X = Contable/ Contable A = Contable

Anexos

	<p>incorrectamente implementadas o funciones perdidas detectadas.</p> <p>B - Número de funciones descritas en la especificación de requisitos.</p>	mejor	B = Contable
<p>Estabilidad en las especificaciones funcionales</p>	<p>$X=1-A/B$</p> <p>A- Número de funcionalidades actualizadas después de entrar en funcionamiento.</p> <p>B- Número de funcionalidades descritas en la especificación de requisitos.</p>	<p>$0 \leq X \leq 1$</p> <p>A mayor cercanía al 1 resultará mejor</p>	<p>X=Contable / Contable</p> <p>A=Contable</p> <p>B=Contable</p>

✓ Métricas de Exactitud.

Nombre de la métrica	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Exactitud esperada	<p>$X = A/T$</p> <p>A - Número de casos encontrados con diferencias entre los resultados razonablemente</p>	<p>$0 \leq X$</p> <p>A mayor cercanía al 0 resultará mejor</p>	<p>X = Contable/ Tiempo</p> <p>A = Contable</p> <p>T =Tiempo</p>

Anexos

	esperados y aquellos resultantes más allá de lo permisible. T - Tiempo de operación.		
Precisión	X=A/T A- Número de resultados encontrados por los usuarios con un nivel de precisión diferente de la necesaria. T- Tiempo de Operación.	$0 \leq X$ A mayor cercanía al 0 resultará mejor	X=Contable/ Tiempo A= Contable T=Tiempo

✓ Métrica de Seguridad.

Nombre de la métrica	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Prevención de la corrupción de los datos	a) X=1-A/N A- Número de veces que un evento de corrupción de los datos importantes se produce. N- Número de casos de prueba que provocó los	a) $0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor b) $0 \leq Y \leq 1$ A mayor cercanía al 1 resultará mejor c) $0 \leq Z$ A mayor cercanía al 1 resultará	X = Contable/ Tiempo A = Contable T =Tiempo X=Contable/ Contable A=Contable B=Contable N=Contable

Anexos

	<p>eventos de mejor</p> <p>corrupción de</p> <p>datos</p> <p>b) $Y=1-B/N$</p> <p>B- Número de veces que un evento de corrupción de datos de menor importancia se produjeron.</p> <p>c) $Z= A/T$ ó B/T</p> <p>T- Período de tiempo de funcionamiento.</p>		<p>Y=Contable/ Contable</p> <p>T=Tiempo</p> <p>Z=Contable/ Contable o</p>
Capacidad de control de acceso	<p>$X=A/B$</p> <p>A- Número de detectar diferentes tipos de operaciones ilegales.</p> <p>B- Número de tipos de operaciones ilegales que están en las condiciones.</p>	<p style="text-align: center;">$0 \leq X \leq 1$</p> <p>A mayor cercanía al 1 resultará mejor</p>	<p>X=Contable/ Contable</p> <p>A=Contable</p> <p>B=Contable</p>

✓ Métricas de Funcionalidad.

Nombre de la métrica	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
----------------------	--------------------	-----------------------------------	----------------

Cumplimiento funcional	$X=1-A/B$ A- Número de elementos cumplidos de las funcionalidades especificadas que no se han aplicado durante las pruebas. B- Número total de elementos cumplidos con funcionalidad especificada.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	$X = \text{Contable} / \text{Contable}$ $A = \text{Contable}$ $B = \text{Contable}$
Cumplimiento de estándar de interfaz	$X=A/B$ A- Número de interfaces con problemas detectados. B- Número total de Casos de pruebas utilizados.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	$X = \text{Contable} / \text{Contable}$ $A = \text{Contable}$ $B = \text{Contable}$

Anexo 3. Métrica para aplicaciones web.

Considerando el peso en kilobytes, se define la siguiente métrica, llamada Índice de Funcionalidad (IF):

$$IF_{\text{Tipo}} = \frac{\sum \text{peso}(\text{archivo funcional}_{\text{tipo}})}{\sum \text{peso}(\text{archivo funcional}) + \sum \text{peso}(\text{archivo funcional}_{\text{tipo}})}$$

Donde *tipo* puede corresponder a uno de los tres tipos de archivos funcionales en un sistema: de servidor, de cliente o incrustada. Dado esto, se define un Índice de Funcionalidad Neto (IFN) como una combinación lineal de los valores anteriores:

Anexos

$$IFN = \alpha IF_{servidor} + \beta IF_{cliente} + \delta IF_{embebido}$$

$$\text{con } \alpha + \beta + \delta = 1$$

Esta métrica pretende medir (con un valor entre 0 y 1) la funcionalidad presente en un sistema Web y provee mayor precisión para poder hablar de sitios o aplicaciones Web. Si aplicamos la métrica a un sistema Web, según el valor del índice podemos decir lo siguiente:

- ✓ Si $IFN < 0,2$, hablamos de un sitio Web no funcional.
- ✓ Si $0,2 \leq IFN < 0,5$, hablamos de un sitio Web funcional.
- ✓ Si $0,5 \leq IFN$, hablamos de software en Web o aplicación Web.

Anexo 4. Encuesta para la selección de especialistas.

Nombre: _____

1. Es graduado de:
2. Seleccione en una escala del 1 – 10 el valor que corresponda con el grado de conocimientos que usted posee sobre del aseguramiento de la funcionalidad como característica de la calidad del software.

1	2	3	4	5	6	7	8	9	10

3. Seleccione con una X en qué grado clasifica sus criterios.

Fuentes de argumentación	Grado de influencia de cada una de las fuentes en sus criterios.		
	Alto	Medio	Bajo
Análisis teóricos realizados por usted			
Su experiencia obtenida			
Trabajos de autores nacionales			
Trabajos de autores extranjeros			
Su propio conocimiento del estado del problema en el extranjero			
Su intuición			

Anexo 5. Encuesta para evaluar la guía para el aseguramiento de la funcionalidad en los sistemas informáticos de Gobierno Electrónico.

Estimado profesor(a):

La presente encuesta forma parte de la aplicación del Método de Valoración de Especialistas. Con este fin solicitamos su valiosa colaboración, y de antemano le aseguramos, que sus opiniones se tendrán en cuenta para la aplicación de la Guía para el aseguramiento de la funcionalidad como característica de calidad en los sistemas informáticos de Gobierno Electrónico.

Muchas Gracias.

Nombre y Apellidos: _____

Fecha de graduación: _____ Puesto de trabajo actual: _____

Calificación profesional: Ingeniero ___ Licenciado en Educación ___ Máster ___ Doctor ___

Rol que desempeña: _____

Años de experiencia en el rol: _____

Categoría Docente: Prof. Instructor ___ Prof. Asistente ___ Prof. Auxiliar ___ Prof. Titular ___

Prof. Adjunto _____

La propuesta de la guía se encuentra representada a través de una tabla, para su análisis y mejor comprensión se le informa que consta con 4 partes fundamentales, la primera parte (columna izquierda) es la relación de las etapas del ciclo de vida del proceso de mejora, la segunda parte (segunda columna) son las actividades propuestas para cada etapa del ciclo de vida, la tercera parte (tercera columna) la descripción de la actividad y la última parte (columna final) la subcaracterística del estándar ISO/IEC 25010 SQuaRE a la que responde la actividad se. Las tres últimas columnas son encabezadas por el elemento crítico que se definió para la etapa, y en respuesta al mismo son las actividades planificadas.

1. Le pedimos su criterio acerca de los elementos definidos en la etapa Estudio Preliminar.

Pregunta	Criterio del especialista				
	C1 <i>Muy adecuado</i>	C2 <i>Bastante adecuado</i>	C3 <i>Adecuado</i>	C4 <i>Poco adecuado</i>	C5 <i>No adecuado</i>
¿Considera que las necesidades del					

cliente y la planificación del proyecto son elementos fundamentales en la etapa de Estudio Preliminar que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico?					
--	--	--	--	--	--

2. Le pedimos su criterio acerca de los elementos definidos en la etapa Modelación del Negocio.

Pregunta	Criterio del especialista				
	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera que las Reglas del Negocio y Modelación del Negocio son elementos fundamentales en la etapa de Modelación del Negocio que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico?					

3. Le pedimos su criterio acerca de los elementos definidos en la etapa Requisitos.

Pregunta	Criterio del especialista				
	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera que los requisitos funcionales, la planificación de la arquitectura, los requisitos no funcionales y la verificación de artefactos son elementos					

Anexos

fundamentales en la etapa de Requisitos que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico?					
--	--	--	--	--	--

4. Le pedimos su criterio acerca de los elementos definidos en la etapa Análisis y Diseño.

Pregunta	Criterio del especialista				
	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera que los patrones de diseño y el diseño de la base de datos son elementos fundamentales en la etapa de Análisis y Diseño que influyen en la funcionalidad de los sistemas informáticos de Gobierno Electrónico?					

5. Le pedimos su criterio acerca de los elementos definidos en la etapa Implementación.

Pregunta	Criterio del especialista				
	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera que la implementación de funcionalidades, la comprobación del estado de la funcionalidad y el acceso a datos son elementos fundamentales en la etapa de Implementación que influyen en la					

Anexos

funcionalidad de los sistemas informáticos de Gobierno Electrónico?				
---	--	--	--	--

6. Le pedimos su criterio acerca de los elementos definidos en la etapa de Pruebas Internas.

<i>Pregunta</i>	<i>Criterio del especialista</i>				
	<i>C1 Muy adecuado</i>	<i>C2 Bastante adecuado</i>	<i>C3 Adecuado</i>	<i>C4 Poco adecuado</i>	<i>C5 No adecuado</i>
¿Considera que los métodos de evaluación es un elemento fundamental en la etapa de pruebas internas que influye en la funcionalidad de los sistemas informáticos de Gobierno Electrónico?					