

**Universidad de las Ciencias Informáticas  
"Facultad 3"**



**Título: Guía para el aseguramiento de la confiabilidad en el  
desarrollo de aplicaciones de escritorio de Gobierno  
Electrónico.**

Trabajo de Diploma para optar por el título de  
Ingeniera Informática

**Autora:** Dainelys Fernández De La Osa

**Tutor:** Ing. Raúl Velázquez Alvarez

**Co-tutora:** MSc Violena Hernández Aguilar

La Habana, Cuba

2012



*"La responsabilidad nuestra es luchar porque la calidad del producto que aquí se haga sea de los mejores y la mejor posible."*

*Ernesto Che Guevara*

## DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Dainelys Fernández De La Osa

---

Raúl Velázquez Alvarez

---

Violena Hernández Aguilar

---

## DATOS DE CONTACTO

Ing. Raúl Velázquez Alvarez

Centro de trabajo: Universidad de las Ciencias Informáticas.

Categoría docente: Instructor recién graduado.

Asignatura que imparte: Metodología de la Investigación Científica.

Asesor y Jefe del Grupo de la Calidad de Software del Centro de Gobierno Electrónico.

MSc Violena Hernández Aguilar

Centro de trabajo: Universidad de las Ciencias Informáticas.

Categoría docente: Asistente.

Asignatura que imparte: Ingeniería de Software y postgrado en las Maestría de Informática Aplicada y Gestión de Proyectos Informáticos.

Trabajó durante 4 años en CALISOFT y se desempeñó como jefa del grupo de Diseño de Pruebas, se ha desempeñado como líder de proyecto en proyectos con Venezuela.

## DEDICATORIA

---

### AGRADECIMIENTOS

*A mis padres porque sin ellos no hubiera podido cumplir este gran sueño de hacerme una profesional, porque siempre han sido mi apoyo y fuerza de inspiración, por ser lo más grande que tengo mi vida.*

*A mi hermanita por su amor, su cariño y por lo que representas para mí te quiero mucho mi niña linda.*

*A mi tío Rigoberto que aunque no este presente físicamente siempre estará en mi corazón.*

*A mi tío Mario por apoyarme en todo momento, por su cariño y por quererme como una hija más.*

*A mi familia en general, pues siempre han estado ahí en todo momento para darme apoyo y sabiduría.*

*A Raidel por darme todo su apoyo durante estos 5 años, por aguantarme mis malcriadeces y mis lágrimas.*

*A Daniel, Yanet y José Ángel mil gracias porque fueron las primeras personas en darme todo su apoyo para la realización de esta tesis cuando más lo necesitaba.*

*A mi tutora Violena por ser una persona especial, por comprenderme y enseñarme, por su paciencia y por estar siempre dispuesta a ayudarme.*

*A mi tutor Raúl por sus consejos, su forma de ser y por toda la ayuda brindada.*

*A Yaniselis por haber sido además de una profesora una madre por haberme ayudado a dar los primeros pasos en esta universidad.*

*A mis amistades: Dayana, Laritza, Meylin, Yarimis, Diamelis y Yosmany.*

## DEDICATORIA

---

*Un verdadero amigo es aquel que llega cuando todo el mundo se ha ido, es alguien que multiplica las alegrías, divide las penas y cuya honestidad es inviolable, que entiende nuestro silencio y que está presente en los momentos en que lo necesitas, te agradezco por ser tan especial, por formar parte de mi vida y por ocupar un lugar en mi corazón. A Yakşel muchas gracias.*

*A mis amistades en Venezuela que fueron una gran familia para mí de ellos pude aprender mucho.*

*A Michel por su amor, por su cariño, su comprensión y su apoyo en todo momento.*

*A la Universidad y su colectivo de profesores que me han formado durante estos cinco años.*

*A la Revolución y a Fidel por darme la oportunidad de realizar este sueño.*

*En fin, quiero agradecer a todas aquellas personas que de una forma u otra han contribuido al desarrollo de este trabajo de diploma y a mi formación tanto profesional como personal.*

## DEDICATORIA

---

### DEDICATORIA

*Le dedico este a esfuerzo a mi tío Rigoberto que estoy segura que si me hubiera visto se sentiría muy orgulloso de mí.*

*A mis padres José Rolando y Neysi por ser mi razón de ser en esta vida, quienes han sido el pilar de mis acciones y por los cuales puedo decir con total orgullo que son el mejor ejemplo que he recibido en mi paso por el mundo. Los amo.*

# RESUMEN

---

## RESUMEN

En este trabajo se presenta una guía para el aseguramiento de la confiabilidad en el desarrollo de aplicaciones de escritorio en el Centro de Gobierno Electrónico. Su realización estuvo motivada por los riesgos en los que pueden incidir los productos en el centro al no efectuarse un proceso de aseguramiento de la característica de calidad confiabilidad.

Para la confección del procedimiento se analizaron de manera exhaustiva algunos de los modelos y estándares de calidad, conceptos relacionados con la calidad, aseguramiento de la calidad y todo lo relacionado con la confiabilidad. Esto permitió la utilización de un conjunto de aspectos que se utilizaron para la elaboración de la guía.

La guía propuesta define un conjunto de actividades y elementos a tener en cuenta en cada una de las etapas del ciclo de vida del software. Esta estrategia se ha validado positivamente con especialistas en el tema que han dado un criterio acertado sobre la misma lo que da la posibilidad de servir de base para futuras investigaciones sobre el tema y se haga extensiva su aplicación y/o adaptación a todos los proyectos de la UCI.

## PALABRAS CLAVES

Aseguramiento, Calidad, Confiabilidad, Guía, Modelo.

# ÍNDICE

---

## TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	II
DEDICATORIA .....	IV
RESUMEN.....	V
INTRODUCCIÓN.....	1
<b>Introducción</b> .....	6
<b>1.1 Calidad de Software</b> .....	6
<b>1.2 Aseguramiento de la calidad de software orientado a la confiabilidad del producto</b> .....	7
<b>1.3 Confiabilidad</b> .....	7
1.3.1 Conceptos asociados a la confiabilidad .....	8
1.3.2 Ingeniería de la Confiabilidad.....	10
<b>1.4 Modelos y estándares internacionales a nivel de producto</b> .....	11
1.4.1 Modelo McCall .....	11
1.4.2 Modelo Boehm.....	12
1.4.3 Modelo FURPS.....	12
1.4.4 Modelo Dromey .....	14
1.4.5 Modelo de calidad ISO/IEC 25010SQuaRE .....	15
<b>1.5 Modelo integrado de capacidad y Madurez (CMMI)</b> .....	17
<b>1.6 Aplicaciones de escritorio</b> .....	18
<b>Conclusiones Parciales</b> .....	19
<b>CAPÍTULO 2 SOLUCIÓN PROPUESTA</b> .....	20
<b>Introducción</b> .....	20
<b>2.1 Ciclo de vida del proceso de desarrollo de software basado en el programa de mejora</b> .....	20
2.1.1 Estudio Preliminar .....	21
2.1.2 Modelación del Negocio.....	22
2.1.3 Requisitos.....	22
2.1.4 Análisis y Diseño .....	25
2.1.5 Implementación .....	27

# ÍNDICE

---

2.1.6 Pruebas.....	27
2.1.7 Despliegue.....	29
2.1.8 Soporte.....	30
<b>2.2 Guía de actividades a realizar por etapas. ....</b>	<b>30</b>
<b>Conclusiones Parciales.....</b>	<b>36</b>
<b>CAPÍTULO 3 VALIDACIÓN DE LA PROPUESTA.....</b>	<b>37</b>
<b>Introducción .....</b>	<b>37</b>
<b>3.1 Procedimiento utilizado para la aplicación del método.....</b>	<b>37</b>
<b>Conclusiones parciales .....</b>	<b>50</b>
CONCLUSIONES.....	51
RECOMENDACIONES .....	52
REFERENCIAS BIBLIOGRÁFICAS.....	53
BIBLIOGRAFÍA.....	54
ANEXOS.....	56

# INTRODUCCIÓN

---

## INTRODUCCIÓN

En la actualidad, la humanidad ha adquirido un considerable desarrollo tecnológico a gran escala debido al vertiginoso desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), siendo la segunda mitad del siglo XX el período de mayor evolución en innumerables sectores científicos. Es indudable que la informática se ha convertido en la ciencia clave para el progreso en casi todos los sectores económicos de un país y en el transcurso de los últimos años la producción de software ha ocupado un lugar significativo en el ámbito económico mundial.

La calidad es una de las metas propuestas por todas las empresas dedicadas a la producción de software y tiene una relación directamente proporcional a la satisfacción del cliente. Hoy en día, lograr la plena satisfacción del cliente es un requisito indispensable para ganarse un lugar en el mercado. Pero para lograrlo es muy importante que el producto que se obtenga cumpla con las características que realmente debe tener y es ahí donde la confiabilidad juega un papel fundamental en dicho proceso. Se puede definir a la confiabilidad como la capacidad que tiene un producto de realizar su función de la manera prevista sin incidentes por el período de tiempo especificado en el diseño del mismo. (1)

La confiabilidad no se mide con el mismo rigor en todos los productos de software porque no todos exigen un alto grado de confiabilidad, esto se puede evidenciar en un software hecho para ejecutarse una sola vez que no requiere el mismo nivel de confiabilidad que uno que va a ser explotado en un largo período de tiempo, pues estos últimos necesitan ser más confiables, seguros, de fácil mantenimiento para poder disminuir los costos a la hora de entrar en etapa de explotación.

Cuba, independientemente de no encontrarse entre los países desarrollados incorpora el uso de las TIC<sup>1</sup> a la vida cotidiana del pueblo. Se propone llevar a la sociedad cubana la revolución informática, haciendo uso de los principales medios de comunicación y de la computación para llevar adelante el conocimiento, creando los Joven Clubs de Computación y Electrónica, llevando a las escuelas las computadoras para realizar los procesos de aprendizaje mucho más interactivos. La Universidad de las Ciencias Informáticas (UCI) juega un papel importante en el proceso de informatización que se lleva a cabo en el país y constituye uno de los principales centros telemáticos existentes ya que cuenta con tecnología informática

---

<sup>1</sup>TIC: Tecnologías de la Informática y las Comunicaciones

# INTRODUCCIÓN

---

avanzada, con las cuales se gestiona la información de todos los estudiantes y profesores, tanto en la docencia como en la producción.

En la UCI, el Centro de Gobierno Electrónico (CEGEL) de la Facultad 3, tiene como misión satisfacer las necesidades de clientes gubernamentales mediante el desarrollo de productos, servicios y soluciones integrales de alta confiabilidad, calidad, competitividad, fidelidad y eficiencia. En una entrevista aplicada a arquitectos, líderes de proyecto y especialistas de calidad de los proyectos RAP, Fiscalía, CDA se pudo evidenciar que, no se aplican modelos y estándares de calidad, ni listas de chequeo, ni pruebas de software referentes a la confiabilidad, por lo que no hay actividades que respondan a su aseguramiento, trayendo como consecuencia el registro de no conformidades respecto a esta característica. En el Centro Nacional de Calidad de Software (Calisoft) de la Universidad específicamente en el Departamento de pruebas de software (DPSW) se evalúa la confiabilidad del software mediante una lista de chequeo pero es aplicada una vez terminado el software, por lo que en las etapas tempranas del ciclo de vida del producto no se responde por el cuidado de esta característica. Además un elemento importante es que la misma ha sido solamente diseñada para detectar no conformidades referentes a la recuperabilidad y la tolerancia a fallos del sistema dejando de analizar la madurez que es otro elemento de la confiabilidad. En dicho centro se aplicó la lista de chequeo a 5 proyectos terminados donde en 3 de ellos se detectaron un total de 9 no conformidades donde las acciones para identificarlas fueron: Quitar energía a la PC cliente, desconectar red a PC cliente, cerrar sistema, desconexión de dispositivos, tiempo de inactivación, ocurrencia de excepciones, cambiar hora a PC cliente, cambiar hora a PC servidor.

A partir de la **problemática** descrita se plantea como **problema de la investigación** ¿Cómo contribuir al aseguramiento de la confiabilidad en el desarrollo de las aplicaciones de escritorio de Gobierno Electrónico de manera que disminuyan las fallas en los mismos y se restauren ante una falla?

Siendo el **objeto de estudio** Aseguramiento de la calidad.

El **objetivo general** de la investigación es proponer una guía para el aseguramiento de la confiabilidad en el desarrollo de las aplicaciones de escritorio de Gobierno Electrónico de manera que disminuyan las fallas en los mismos y se restauren ante una falla.

Y como **campo de acción** Aseguramiento de la confiabilidad en el desarrollo de las aplicaciones de escritorio de Gobierno Electrónico.

# INTRODUCCIÓN

---

Se plantea para este trabajo la siguiente **idea a defender**: con la propuesta de una guía para el aseguramiento de la confiabilidad en el desarrollo de las aplicaciones de escritorio de Gobierno Electrónico se disminuirán las fallas en los mismos y podrán restaurarse ante una falla.

El objetivo se desglosa en una serie de **objetivos específicos**:

1. Elaborar el marco teórico de la investigación.
2. Desarrollar la propuesta de solución.
3. Validar la solución propuesta.

Se plantea una serie de **tareas de la investigación** para dar cumplimiento a los objetivos de la investigación:

- ✓ Realización de un estudio del estado del arte de la temática en Cuba y el mundo.
- ✓ Análisis de la Norma (ISO/IEC 25010) División Modelo de Calidad y otros estándares internacionales que traten el tema de las características de calidad de software.
- ✓ Análisis de los diferentes atributos existentes que conforman cada sub-característica.
- ✓ Definición de los atributos necesarios durante el ciclo de desarrollo, basado en el programa de mejora de la UCI que plantea CMMI<sup>2</sup>, para que las aplicaciones de escritorio de Gobierno Electrónico cumplan con la característica de confiabilidad.
- ✓ Definición de acciones o actividades a realizar a lo largo del ciclo de desarrollo de software, para asegurar la Confiabilidad en el producto final, basándose en los elementos anteriormente definidos.
- ✓ Evaluación de la propuesta mediante el criterio de especialistas.

A continuación se relacionan los **métodos científicos** utilizados en la investigación.

---

<sup>2</sup> CMMI: Modelo Integrado de Capacidad y Madurez

# INTRODUCCIÓN

---

**Métodos Teóricos:** permiten estudiar las características del objeto de investigación que no son observables directamente.

Analítico-Sintético: se utiliza para la realización de un estudio detallado de las teorías, tendencias y documentos existentes relacionados con la confiabilidad y el aseguramiento de la misma, donde se resumen los elementos más importantes y de mayor utilidad que permitirán el diseño correcto de una guía para asegurar la confiabilidad de software en las aplicaciones de escritorio.

Histórico-Lógico: se utiliza para constatar la evolución que ha tenido la confiabilidad durante los últimos años a nivel mundial a través del estudio de sus antecedentes y las tendencias actuales, haciendo un estudio además de los diferentes modelos que tratan esta característica de calidad en distintos períodos del desarrollo de software.

**Métodos Empíricos:** describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

Entrevista: se utiliza para obtener información acerca de la confiabilidad en los proyectos de CEGEL y en el departamento de pruebas de software de Calisoft.<sup>3</sup>, es aplicada a diferentes arquitectos de dichos proyectos y a especialistas de Calisoft.

Encuesta: se utiliza para realizar la validación de la solución propuesta mediante el criterio de especialistas.

La estructura de la investigación es la siguiente: introducción, 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas, anexos y glosario de términos.

Capítulo 1: Fundamentación Teórica, se realiza un estudio del estado del arte de las diferentes normas y modelos de calidad que harán permisible el desarrollo de la investigación, tales como: McCall, Dromey, Boehm, Hewlett-Packard y ISO/IEC 25010. Se explican también algunos conceptos como: calidad de software, aseguramiento de la calidad y todo lo relacionado con la confiabilidad.

---

<sup>3</sup>Calisoft: Centro Nacional de Calidad de Software

# INTRODUCCIÓN

---

Capítulo 2: Propuesta de solución, en este capítulo se describe el ciclo de vida por el cual se deben regir los proyectos de la universidad, se identifican los elementos críticos que afectan la confiabilidad en cada una de las etapas definidas y finalmente se presenta la guía propuesta para contribuir al aseguramiento de la misma.

Capítulo 3: Validación de la Propuesta, se realiza la validación de la propuesta a partir de la aplicación del criterio de especialistas, o sea, se presentan los resultados de un cuestionario después de haber sido aplicado a personas con experiencia profesional en el tema y que tienen el conocimiento necesario para emitir su criterio.

# CAPÍTULO 1: Fundamentación Teórica

---

## Capítulo 1 Fundamentación Teórica

### Introducción

Durante el desarrollo del presente capítulo se presentan los principales conceptos relacionados con la Calidad de software y el Aseguramiento de la calidad. Se realiza una caracterización de los modelos, estándares y normas de calidad existentes para así garantizar cuál de ellos aporta más elementos para dicha investigación. También se analiza la confiabilidad profundamente como campo en el cual se desarrolla esta investigación.

### 1.1 Calidad de Software

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. La calidad del software es medible y varía de un sistema a otro o de un programa a otro.

El Diccionario de la Real Academia de la Lengua Española define calidad como la *“propiedad o conjunto de propiedades inherentes a algo, que permiten apreciarla como igual, mejor o peor que las restantes de su especie.”* (2)

Para la IEEE es: *“Grado con el que un sistema, componente o proceso cumple los requisitos especificados y las necesidades o expectativas del cliente o usuario”.* (3)

Según Pressman<sup>4</sup>: *“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.”* (4)

Teniendo en cuenta las diferentes definiciones expuestas anteriormente, el presente trabajo se adscribe a la definición que da Pressman porque brinda una visión clara, profunda y la define de la manera más específica garantizando la calidad del producto y de la documentación. La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar costoso si se detectan problemas

---

<sup>4</sup>Pressman: Ingeniero de software

# CAPÍTULO 1: Fundamentación Teórica

---

derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta el aseguramiento de la calidad.

## 1.2 Aseguramiento de la calidad de software orientado a la confiabilidad del producto

El proceso de aseguramiento de la calidad del software es el destinado a cumplir con las especificaciones tanto del cliente como de calidad, el cumplimiento con los estándares preestablecidos proporcionando que el producto sea entregado en el tiempo establecido,

El SQA es una actividad de soporte y protección que se aplica durante todo el proceso de desarrollo de software pero es importante asegurarla desde el comienzo ya que en este momento los errores sin descubrir causan grandes fallas, es importante conocer y mitigar los errores desde etapas tempranas del ciclo de desarrollo del software, ya que su detección y mitigación tardía tiende a aumentar los costes, el tiempo. Está presente en métodos y herramientas de análisis, diseño, programación y prueba, inspecciones técnicas formales, ajuste de estándares, control de la documentación y de los cambios realizados durante todo el proceso de desarrollo del software.

La presente investigación se centra en el aseguramiento de calidad pero orientado a la característica de calidad confiabilidad y para ello se define como aseguramiento de la confiabilidad a *“Un diseño planificado de acciones o actividades a realizar a lo largo del ciclo de vida de software que se requieren para garantizar la confiabilidad en el producto final”*.

El proceso de aseguramiento es de suma importancia dentro del desarrollo del software, ya que brinda el margen para la obtención de un producto que cumpla con lo que realmente el cliente quiere y es muy conveniente que los ingenieros encuentren y corrijan inmediatamente los defectos introducidos, para asegurar la confiabilidad de un software.

## 1.3 Confiabilidad

La confiabilidad es uno de los principales aspectos que se tienen en cuenta cuando se construye un producto de software. A través de esta se pueden predecir las posibilidades de fallo de cualquier producto, de esta forma, se conoce con antelación los posibles errores y deficiencia que pueden presentar los

# CAPÍTULO 1: Fundamentación Teórica

---

productos. En algunas bibliografías sobre todo las que son en ingles fiabilidad y confiabilidad significan lo mismo, por lo tanto para esta investigación se asume que va ser lo mismo.

La confiabilidad “*Capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un período de tiempo establecido*”. (7) Es una medida del grado de operabilidad y capacidad de un sistema para prestar el servicio requerido en cualquier momento de su tiempo de misión, suponiendo su disponibilidad en el instante inicial.

Según la definición informática, el término Confiabilidad, representa una característica de los sistemas informáticos por la que se mide el tiempo de funcionamiento sin fallos. (8)

## 1.3.1 Conceptos asociados a la confiabilidad

**Deterioros:** son las circunstancias que provocan o resultan de la falta de confiabilidad y están compuestos por Falla, Error, Defecto.

**Falla:** Es un comportamiento incoherente del sistema que no cumple con el funcionamiento especificado, se manifiestan en el comportamiento externo del mismo pero son el resultado de errores internos. (9)

### Tipos de fallos

Fallos transitorios

- ✓ Desaparecen solos al cabo de un tiempo.

Ejemplo: interferencias en comunicaciones.

Fallos permanentes

- ✓ Permanecen hasta que se reparan.

Ejemplo: roturas de hardware, errores de diseño de software.

Fallos intermitentes

- ✓ Fallos transitorios que ocurren de vez en cuando.

# CAPÍTULO 1: Fundamentación Teórica

---

Ejemplo: calentamiento de un componente de hardware.

Error: estado interno de un sistema que difiere de uno válido, siendo susceptible a provocar una falla.

Defecto: condición mínima que provoca el error del sistema.

Medios: son los métodos, herramientas y soluciones requeridas para entregar un servicio confiable. Son usados en el proceso de construcción del software, dentro de ellos podemos encontrar a Evitación de fallos, Tolerancia a fallos, Eliminación de fallos, Predicción de fallos.

Evitación de fallos: su función es tratar de evitar que se introduzcan fallos en el sistema antes de que entre en funcionamiento.

Tolerancia a fallos: consiste en la capacidad del producto de software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interface especificada.

Eliminación de fallos: consiste en encontrar y eliminar los fallos que se producen en el sistema una vez construido.

Predicción de fallos: su objetivo es estimar la presencia de defectos, la ocurrencia y consecuencia de los fallos.

La confiabilidad agrupa un conjunto de atributos, es decir, modos y medidas por las cuales se puede estimar la calidad de un servicio confiable entre los que se encuentran disponibilidad, fiabilidad, seguridad, confidencialidad, integridad y mantenibilidad como se muestra en la Figura 1, que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales y libres de fallos en un período de tiempo establecido.

# CAPÍTULO 1: Fundamentación Teórica



**Figura 1. Atributos de la confiabilidad.**

## 1.3.2 Ingeniería de la Confiabilidad

La Ingeniería de la confiabilidad surge en 1970, es una práctica que permite planear y guiar el proceso de prueba del software de manera cuantitativa y está dedicada al estudio de los aspectos físicos y aleatorios del fenómeno falla. Se caracteriza por tener dos elementos fundamentales:

**El uso esperado relativo de las funcionalidades del sistema:** su objetivo es centrarse en caracterizar de manera cuantitativa el uso esperado del sistema mediante la definición del llamado perfil de operaciones que no es más que la caracterización cuantitativa del uso que se espera de las funcionalidades principales del sistema. Dicha caracterización permite optimizar el uso de los recursos en las funciones que tengan mayor impacto y mayor uso esperado dentro del sistema.

**Los requerimientos de calidad definidos por el cliente, que incluyen la confiabilidad, la fecha de liberación y costo del ciclo de vida del proyecto:** se refiere al enfoque al cliente mediante el establecimiento de objetivos cuantitativos asociados a la calidad del producto (representados con base a las fallas del producto). La satisfacción de dichos objetivos permite establecer un balance entre los costos del producto, así como la satisfacción de las necesidades del cliente.

Tomando en cuenta los conceptos planteados se puede llegar a la conclusión que, a través del estudio de la confiabilidad de un producto de software, es posible reconocer si el producto conserva buen desempeño ante las diversas circunstancias que se pueden presentar, verificando siempre que el producto reaccione positivamente ante las posibles fallas que pueda presentarse por defectos en el programa o causas

## CAPÍTULO 1: Fundamentación Teórica

---

externas, protegiendo siempre la integridad de la información. Por su parte los modelos y estándares de calidad que referencian a los productos permiten mayor organización y clasificación de las características de calidad.

### 1.4 Modelos y estándares internacionales a nivel de producto

Un modelo de calidad integra un grupo de características y relaciones entre ellas que proporcionan la base para especificar requisitos de calidad y evaluarla.

La ISO/IEC define un estándar como: *“Un estándar es un documento publicado. Se trata de un conjunto de reglas que se usan de manera habitual como los buenos principios, prácticas o directrices. Un estándar controla cómo las personas deben desarrollar y gestionar los materiales, productos, servicios, tecnologías, procesos y sistemas.”* (10)

Pressman indica que los factores que afectan a la calidad del software no cambian, por lo que resulta útil el estudio de los modelos de calidad que han sido propuestos en este sentido desde los años 70. Dado que los factores de calidad presentados para ese entonces siguen siendo válidos, se estudiarán los modelos más importantes propuestos hasta ahora: McCall (1977), Boehm (1978), SQuaRE, Dromey (1996), Hewlett-Packard y el estándar propuesto por la Norma ISO/IEC 9126 (1991).

#### 1.4.1 Modelo McCall

El modelo propuesto por **Jim McCall** en 1977, uno de los modelos más usados, se basa en tres perspectivas generales, que son utilizadas para definir e identificar la calidad de un producto de software: operación del producto, revisión del producto y transición del producto, desde los cuales el usuario puede contemplar la calidad de un producto. La fiabilidad en este modelo, se enmarca en tres atributos fundamentales.

**Precisión:** atributos del software que proporcionan el grado de precisión requerido en los cálculos y los resultados.

**Tolerancia a fallos:** atributos del software que posibilitan la continuidad del funcionamiento bajo condiciones no usuales.

## CAPÍTULO 1: Fundamentación Teórica

---

Simplicidad: atributos del software que posibilitan la implementación de funciones de la forma más comprensible posible. (11)

Sobre la confiabilidad McCall expresa que combina la tolerancia tanto a errores de hardware como a errores de software. Además considera que la confiabilidad puede ser medida en función del tiempo medio entre fallas, tiempo medio antes del mantenimiento, tiempo medio antes de la recuperación y probabilidad de falla.

### 1.4.2 Modelo Boehm

El Modelo Boehm es un modelo fijo sin posibilidad de ser modificado o adaptado por el usuario. Los criterios y factores son determinados y fijos, de forma que la medida de la calidad debe ajustarse a estas definiciones y a las relaciones entre criterios y factores de calidad que el modelo propone. El factor fiabilidad es uno de los criterios más importantes a la hora de medir el producto final, los atributos que lo evalúan son:

Exactitud: es la precisión de los cálculos y emisión de resultados.

Completitud: grado en que han sido implementadas las funciones para el sistema.

Consistencia: el grado en que las especificaciones externas se corresponden con las especificaciones internas. (6)

Boehm considera que para evaluar la confiabilidad se debe tomar en cuenta que los sistemas de software no son entidades estáticas.

### 1.4.3 Modelo FURPS

El modelo de McCall ha servido de base para modelos de calidad posteriores, y este es el caso del modelo **FURPS**, producto del desarrollo de Hewlett-Packard (Robert Grady, 1987). En este modelo se desarrollan un conjunto de factores de calidad de software, bajo el acrónimo de FURPS: funcionalidad (Functionality), usabilidad (Usability), confiabilidad (Reliability), desempeño (Performance) y capacidad de soporte (Supportability). La Tabla 1 presenta la clasificación de los atributos de calidad que se incluyen en el modelo, junto con las características asociadas a cada uno. (4)

## CAPÍTULO 1: Fundamentación Teórica

---

El modelo FURPS incluye, además de los factores de calidad y los atributos, restricciones de diseño y requerimientos de implementación, físicos y de interfaz. Su principal aporte es que divide los factores principales de evaluación en dos grupos: los basados en requerimientos funcionales y los basados en requerimientos no funcionales.

<b>Factor de calidad</b>	<b>Atributos</b>
<b>Funcionalidad</b>	<ul style="list-style-type: none"><li>• Características y capacidades del programa.</li><li>• Generalidad de las funciones.</li><li>• Seguridad del sistema.</li></ul>
<b>Facilidad de uso</b>	<ul style="list-style-type: none"><li>• Factores humanos.</li><li>• Factores estéticos.</li><li>• Consistencia de la interfaz.</li><li>• Documentación.</li></ul>
<b>Confiabilidad</b>	<ul style="list-style-type: none"><li>• Frecuencia y severidad de las fallas.</li><li>• Exactitud de las salidas.</li><li>• Tiempo medio de fallos.</li><li>• Capacidad de recuperación ante fallas.</li><li>• Capacidad de predicción.</li></ul>
<b>Rendimiento</b>	<ul style="list-style-type: none"><li>• Velocidad del procesamiento.</li><li>• Tiempo de respuesta.</li><li>• Consumo de recursos.</li><li>• Rendimiento efectivo total.</li><li>• Eficacia.</li></ul>
<b>Capacidad de Soporte</b>	<ul style="list-style-type: none"><li>• Extensibilidad.</li><li>• Adaptabilidad.</li><li>• Capacidad de pruebas.</li><li>• Capacidad de configuración.</li><li>• Compatibilidad.</li><li>• Requisitos de instalación.</li></ul>

## CAPÍTULO 1: Fundamentación Teórica

---

**Tabla 1: Atributos de calidad Modelo FURPS**

Acerca de la confiabilidad de un sistema plantea se complica a medida que este cerca. Este modelo habla de la confiabilidad de manera general, por lo que no brinda pocos elementos para esta investigación.

### 1.4.4 Modelo Dromey

Dromey en el año 1996 propuso 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 solventar esta situación, Dromey sugiere el uso de cuatro categorías que implican propiedades de calidad, que son: correctitud, internas, contextuales y descriptivas. A continuación se muestra una tabla que presenta la relación que establece Dromey entre las propiedades de calidad del producto y los atributos de calidad de alto nivel.

Propiedades del producto	Atributos de la calidad
Correctitud	<ul style="list-style-type: none"><li>• Funcionalidad</li><li>• Confiabilidad</li></ul>
Internas	<ul style="list-style-type: none"><li>• Mantenibilidad</li><li>• Eficiencia</li><li>• Confiabilidad</li></ul>
Conceptuales	<ul style="list-style-type: none"><li>• Mantenibilidad</li><li>• Reusabilidad</li><li>• Portabilidad</li><li>• Confiabilidad</li></ul>
Descriptivas	<ul style="list-style-type: none"><li>• Mantenibilidad</li><li>• Reusabilidad</li><li>• Portabilidad</li><li>• Usabilidad</li></ul>

# CAPÍTULO 1: Fundamentación Teórica

---

## **Tabla 2 Relaciones entre las propiedades de calidad del producto y los atributos de calidad.**

El proceso de construcción de modelos de calidad propuesto por Dromey consta de 5 pasos, basados en las propiedades mencionadas. Los pasos del marco de referencia propuesto son:

1. Especificación de los atributos de calidad de alto nivel (por ejemplo, confiabilidad, mantenibilidad).
2. Determinación de los distintos componentes del producto a un apropiado nivel de detalle (por ejemplo, paquetes, subrutinas, declaraciones).
3. Para cada componente, determinación y categorización de sus implicaciones más importantes de calidad.
4. 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.
5. Iteración sobre los pasos anteriores, utilizando un proceso de evaluación y refinamiento.

Este modelo relaciona las propiedades de calidad del producto y los atributos de calidad pero no trata de manera específica las subcaracterísticas de confiabilidad.

### **1.4.5 Modelo de calidad ISO/IEC 25010SQuaRE**

El modelo SQuaRE (Software Product Quality Requirements and Evaluation) (Evaluación de los Requisitos de Calidad de los Productos de Software) es una revisión de ISO/IEC 9126-1:2001, este modelo conserva 6 características de calidad de software y le agrega 2 nuevas. La serie SQuaRE tiene 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.

## CAPÍTULO 1: Fundamentación Teórica

- ISO/IEC 2503n. División de requisitos de calidad.
- ISO/IEC 2504n. División de evaluación de calidad.

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. La división 2501n presenta un detallado modelo incluyendo características de calidad interna, externa y calidad en uso como se muestra a continuación en la Tabla 3. (1)

<i>Calidad del software (interna y externa)</i>							
<i>Funcionalidad</i>	<i>Seguridad</i>	<i>Interoperabilidad</i>	<i>Fiabilidad</i>	<i>Usabilidad</i>	<i>Eficiencia</i>	<i>Mantenibilidad</i>	<i>Portabilidad</i>
<i>Adecuación</i>	<i>Acceso resistente</i>	<i>Compatibilidad OSI</i>	<i>Madurez</i>	<i>Adecuación</i>	<i>Tiempo de respuesta</i>	<i>Capacidad de análisis</i>	<i>Adaptabilidad</i>
<i>Exactitud</i>	<i>Copia resistente</i>	<i>Compatibilidad Software</i>	<i>Tolerancia a fallos</i>	<i>Capacidad de aprendizaje</i>	<i>Utilización de recursos</i>	<i>Capacidad a cambios</i>	<i>Capacidad a instalación</i>
<i>Adherencia a normas</i>	<i>Cifrabilidad</i>	<i>Compatibilidad de datos</i>	<i>Recuperabilidad</i>	<i>Operabilidad</i>	<i>Adherencia a normas</i>	<i>Estabilidad</i>	<i>Capacidad a coexistencia</i>
	<i>Protección resistente</i>	<i>Trazabilidad</i>	<i>Adherencia a normas</i>	<i>Util</i>		<i>Capacidad a testing</i>	<i>Capacidad a reemplazo</i>
	<i>Robustez</i>	<i>Adherencia a normas</i>		<i>Agradable</i>		<i>Adherencia a normas</i>	<i>Adherencia a normas</i>
	<i>Adherencia a normas</i>			<i>Adherencia a normas</i>			

**Tabla 3 Modelo de calidad para la calidad interna y externa para la versión ISO/IEC25010.**

SQuaRE no habla del término confiabilidad como tal pero sí de la **fiabilidad** que se refiere a la capacidad del producto de software para mantener un nivel de desempeño especificado bajo condiciones específicas y dentro de esta incluye a **Madurez, Tolerancia a defectos, Recuperabilidad, Fiabilidad conforme.**

**Madurez:** capacidad del producto de software para evitar fallos como resultado de defectos en el software.

**Tolerancia a fallos:** se refiere a la capacidad del producto de software para mantener un nivel de desempeño especificado en casos de fallas del software o de infracción de su interfaz especificada.

# CAPÍTULO 1: Fundamentación Teórica

---

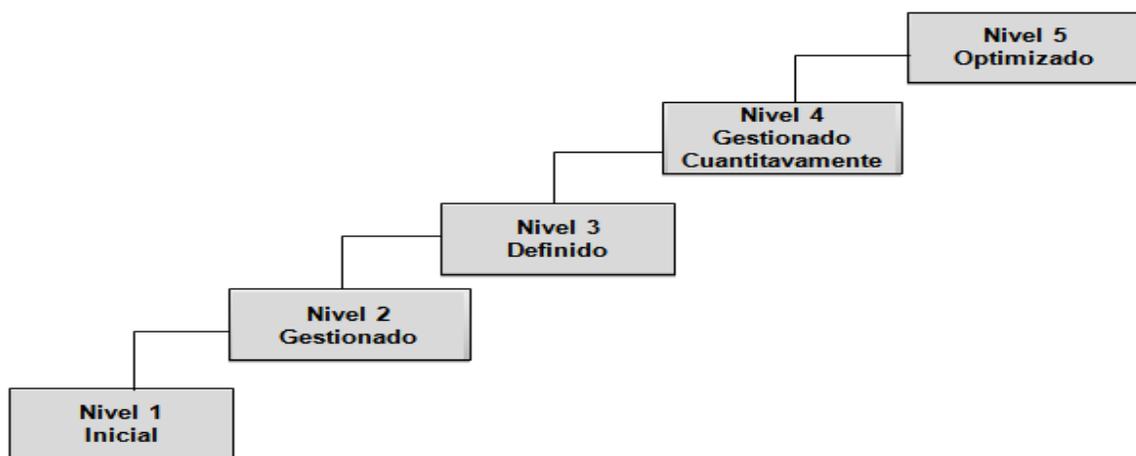
**Recuperabilidad:** es la capacidad del producto de software para restablecer un nivel especificado de desempeño no dejando perder los datos directamente afectados en caso de falla.

**Fiabilidad conforme:** se define como la capacidad del producto de software para adherirse a los estándares, convenios o regulaciones relacionadas con la fiabilidad.

El estudio de los modelos y estándares de calidad ha permitido profundizar en los beneficios que pueden obtenerse con la aplicación de los mismos, se determinó utilizar como referencia el Modelo SQuaRE en su versión 25010, porque brinda una visión profunda de las características y sub-características de confiabilidad, demuestra lo novedoso y estructurado que está en comparación con el resto de los modelos que tratan a la característica pero de manera superficial.

## 1.5 Modelo integrado de capacidad y Madurez (CMMI)

CMMI es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Representa un camino de mejoramiento y permite determinar la madurez, la identificación de las fortalezas, las debilidades y hacer convertir las debilidades en fortalezas. Establece 5 niveles de madurez y 22 áreas de procesos de forma escalonada.

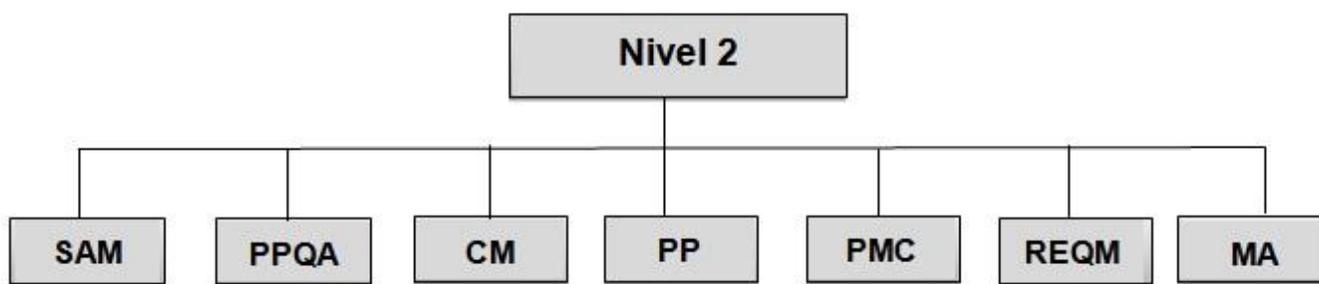


## CAPÍTULO 1: Fundamentación Teórica

---

**Figura 2 Representación escalonada de CMMI.**

Actualmente los proyectos certificados en la universidad se encuentran en el nivel 2 que 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).



**Figura 3 Áreas de procesos del nivel 2 de CMMI.**

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. (12)

Dada la importancia de la aplicación de este modelo la universidad se dio a la tarea de definir un ciclo de vida para el desarrollo de software, ciclo por el cual se rige la investigación para lograr el aseguramiento de la confiabilidad en las aplicaciones de escritorio.

### 1.6 Aplicaciones de escritorio

Se entiende por aplicación de escritorio a aquella aplicación que ya está instalada en el ordenador de un usuario que se ejecuta directamente en un sistema operativo, cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM, disco duro, memoria de video, es un programa de computadora que se utiliza como herramienta para una operación o tarea específica. A diferencia de otros

## CAPÍTULO 1: Fundamentación Teórica

---

programas como los sistemas operativos, los lenguajes de programación y otros, la aplicación tiene el único y principal fin de realizar una tarea específica, a menudo básica y de rápido y fácil uso para el usuario común no avanzado.

### 1.6.1 Ventajas de las aplicaciones de escritorio

- ✓ En su ejecución no requieren comunicación con el exterior, sino que se realiza de forma local. Esto repercute en mayor velocidad de procesamiento, y por tanto en mayores capacidades a la hora de programar herramientas más complicadas o funcionales.
- ✓ Suelen ser más robustas y estables que las aplicaciones Web.
- ✓ Rendimiento: el tiempo de respuesta es muy rápido.
- ✓ Seguridad: pueden ser muy seguras (dependiendo del desarrollador).

### 1.6.2 Desventajas de las aplicaciones de escritorio

- ✓ Su acceso se limita al ordenador donde están instaladas.
- ✓ Son dependientes del sistema operativo que utilice el ordenador y sus capacidades (video, memoria, etc).
- ✓ Requieren instalación y actualización personalizada.
- ✓ Suelen tener requerimientos especiales de software y librerías.

### Conclusiones Parciales

En el presente capítulo, a partir de la investigación realizada se analizaron los principales conceptos referentes a calidad de software y aseguramiento de la calidad definiéndose los más adecuados para esta investigación. Se realizó un estudio de todo lo relacionado con la confiabilidad y de todos los modelos y estándares que incluyen a esta característica de la calidad escogiendo para esta investigación la ISO/ICE 25010 SQuaRE como guía para el desarrollo de la propuesta de solución.

## CAPÍTULO 2: Solución Propuesta

---

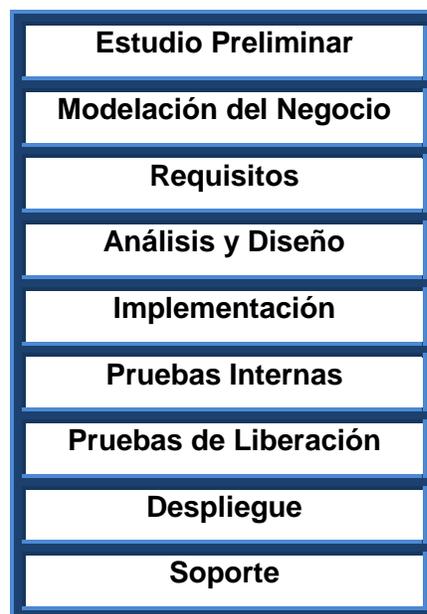
### CAPÍTULO 2 SOLUCIÓN PROPUESTA

#### Introducción

Cuando se tiene la idea de desarrollar un software se pretende que los analistas, diseñadores, desarrolladores y todo el personal que interviene en el proceso, utilicen correctamente características de calidad para que el producto final sea capaz de satisfacer las necesidades del cliente. En el presente capítulo se presenta una guía para asegurar la confiabilidad en el desarrollo de las aplicaciones de escritorio de gobierno electrónico.

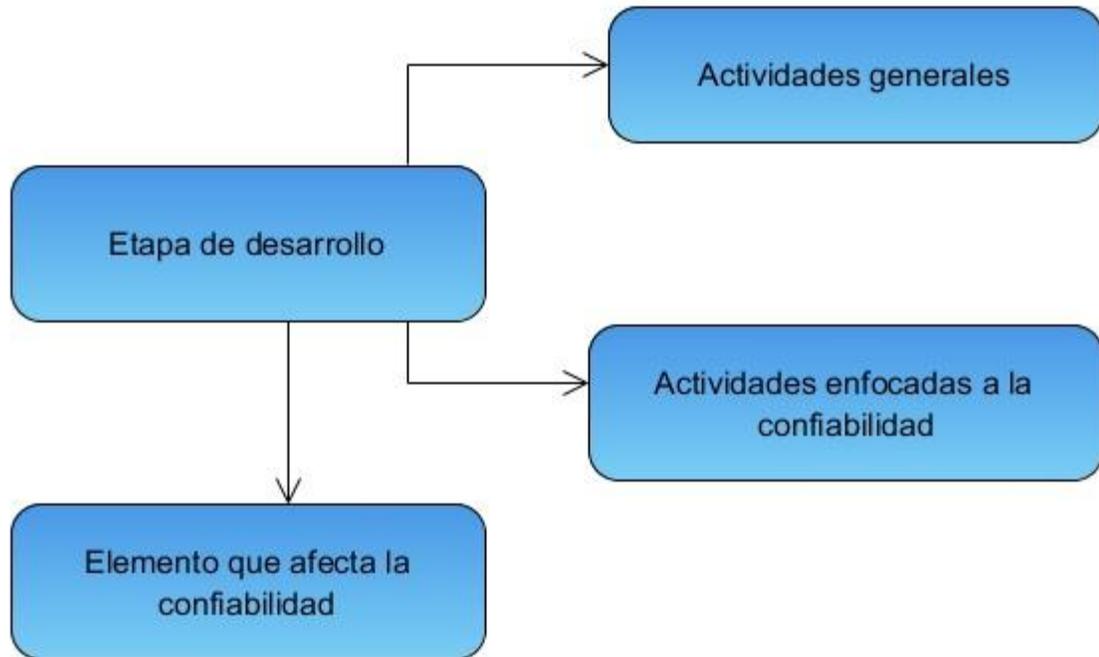
#### 2.1 Ciclo de vida del proceso de desarrollo de software basado en el programa de mejora

Como parte de la institucionalización que lleva a cabo la universidad es necesario realizar un estudio del ciclo de vida del proceso de desarrollo de software.



**Figura 4 Ciclo de vida del software**

A partir de dicho ciclo de vida se definen en cada una de las etapas elementos en los que puede influir la confiabilidad del sistema. A continuación se muestra un gráfico que recoge los elementos que se tienen en cuenta en la guía propuesta.



**Figura 5 Elementos que contiene la guía.**

A continuación se detalla el significado de cada elemento:

**Etapa de desarrollo:** son las etapas de desarrollo definidas en el ciclo de vida del programa de mejora.

**Elemento que afecta la confiabilidad:** es el elemento que influye sobre la confiabilidad del sistema en la etapa de desarrollo.

**Actividades generales:** son las actividades que propone la metodología a utilizar en esa etapa.

**Actividades enfocadas a la confiabilidad:** actividades que se realizan para lograr el aseguramiento de la confiabilidad en cada etapa.

### **2.1.1 Estudio Preliminar**

En esta etapa se llevan a cabo las actividades relacionadas con la planeación del proyecto. Además se realiza un estudio inicial del cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo.

## CAPÍTULO 2: Solución Propuesta

---

Elemento crítico de la etapa:

- ✓ Alcance del sistema: es necesario lograr un entendimiento entre el cliente y la gerencia del proyecto y definir la magnitud del sistema a implementar. Uno de los problemas más frecuentes que influyen en la confiabilidad de un sistema son los cambios de alcance, es común subvalorar la magnitud del sistema a implementar, luego comienzan a aparecer nuevos requisitos y el sistema crece sobrepasando las capacidades de la arquitectura y por ende el resultado final no se corresponde con el esperado.

### 2.1.2 Modelación del Negocio

Esta etapa está destinada a comprender los procesos de negocio de la organización. Se comprende cómo funciona el negocio que se desea automatizar y se garantiza que el software que va a ser desarrollado cumple con su propósito. Para la descripción y modelado de negocio pueden ser utilizadas diferentes técnicas como el Modelado de Casos de Uso del Negocio.

Elemento crítico de la etapa:

- ✓ Las reglas del negocio: ayudan al analista a definir los flujos de validación y comportamiento del sistema. Si las reglas del negocio no quedan bien descritas entonces el sistema puede presentar comportamientos indeseados por el cliente afectando la confiabilidad del mismo.

### 2.1.3 Requisitos

Esta etapa tiene como objetivo fundamental desarrollar el modelo del sistema del software que se va a construir. Se describen un conjunto de casos de uso, que estos responden a los requisitos funcionales del sistema. Además la especificación de requisitos incluye requisitos no funcionales.

Elementos críticos de esta etapa:

- ✓ Requisitos funcionales: durante la captura de requisitos el analista debe entender las necesidades del cliente y transformarla en requisitos entendibles por el equipo de desarrollo, evitar la ambigüedad y mantener la perspectiva de que el mismo responde a una funcionalidad del sistema. Un proceso incorrecto de captura de requisitos funcionales se convierte en funcionalidades del sistema con comportamientos incorrectos o erróneos, aspecto que afecta la confiabilidad. En los

## CAPÍTULO 2: Solución Propuesta

---

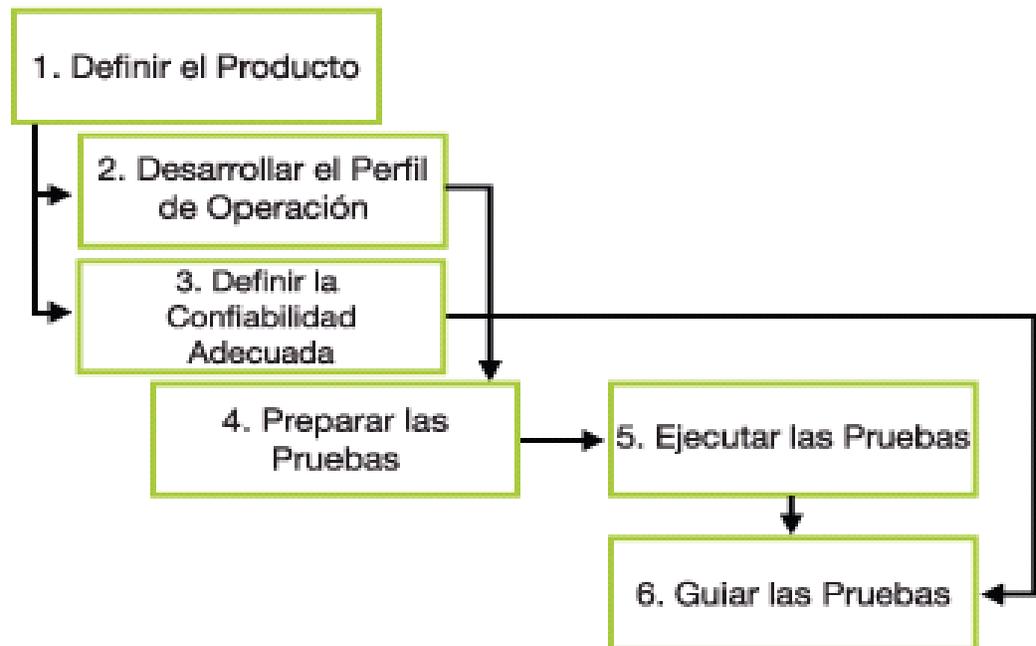
requisitos funcionales se van a definir los casos de uso arquitectónicamente significativos en dependencia de la metodología utilizada, mediante la realización de pruebas de caja negra, no pueden fallar ya que son las funcionalidades de mayor peso en el sistema.

- ✓ Requisitos no funcionales: los requisitos no funcionales de fiabilidad definen una serie de características que debe cumplir el software que deben ser especificadas con claridad para que el equipo de desarrollo las entienda y las pueda poner en práctica. En el caso de los requisitos relacionados con los tiempos de respuesta y el comportamiento del sistema ante los errores y su posterior recuperación influyen directamente en la recuperabilidad del sistema, es importante señalar que estos requisitos en su mayoría deben ser soportados por la arquitectura del sistema y es responsabilidad del arquitecto identificarlos y luego velar por su cumplimiento y correcta implementación.
- ✓ Nivel de confiabilidad: los niveles de confiabilidad permiten obtener el nivel de confiabilidad que tiene el software y se clasifican en:
  - Seguridad crítica: si existe algún fallo, provoca una lesión, pérdida de vida o algún daño la ambiente. Programas que controlan una nave espacial.
  - Misión crítica: si en algún momento falla el sistema, provoca anomalías en las actividades que se encargan de controlar la misión de un objeto que se encuentra dirigido hacia un sector en específico. Sistemas de pronósticos meteorológicos.
  - Negocio crítico: si falla el sistema en algún momento se desata una caída en los negocios que utilizan dicho sistemas. Programas de gestión de un Banco, programa de Toma de decisiones.
  - Media: si falla provoca retraso en actividades asociadas al negocio.
  - Baja: Si falla el sistema las actividades que gestiona se pueden resolver manualmente sin afectar el negocio en cuestión.
- ✓ Perfil operacional: La construcción de un perfil operacional mejora el análisis de requerimientos dado que el proceso de cuantificar las probabilidades de uso de las operaciones del producto exige

## CAPÍTULO 2: Solución Propuesta

---

un proceso adicional de reflexión y trabajo con el cliente. A continuación se muestra un gráfico con un ejemplo de cómo realizar un perfil operacional.



**Figura 6 Ejemplo de perfil operacional.**

Desarrollar el Perfil de Operación: se define el conjunto completo de operaciones (tareas o funcionalidades lógicas principales del sistema) con su correspondiente probabilidad de ocurrencia o uso esperado. En esta etapa, la administración de los recursos toma un nivel cuantitativo basado en la importancia de cada operación del sistema.

Para ello es necesario buscar los casos de uso y ordenarlos de forma ascendente (de menor a mayor probabilidad) en cuanto a la probabilidad de uso de la funcionalidad, la cual se calcula de la siguiente forma:

## CAPÍTULO 2: Solución Propuesta

---

$$P = \frac{\text{\# de la funcionalidad}}{\text{Total de funcionalidades}}$$

P: probabilidad de uso o frecuencia.

# de la funcionalidad: va a ser el número que tenga el caso delante el caso después de haber sido ordenado de forma ascendente, dándole menor probabilidad a los primeros y dejando los arquitectónicamente significativos para el final.

Total de funcionalidades: cantidad de funcionalidades que va a tener el sistema.

### 2.1.4 Análisis y Diseño

Durante esta etapa, de considerarse necesario, los requisitos descritos pueden ser refinados y estructurados para conseguir una comprensión más precisa de los mismos y una descripción que sea fácil de mantener y ayude a estructurar el sistema (incluyendo su arquitectura).

También en esta etapa se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Esto contribuye a una arquitectura sólida que se convierte en un plano para la implementación. Los modelos desarrollados en esta etapa son más formales y específicos de una implementación. También se desarrolla el documento de arquitectura, diagramas de clases, diagramas de entidad relación, diagrama de despliegue, se diseña la base de datos entre otros.

Elementos críticos de esta etapa:

- ✓ Descripción de los casos de uso: en algunos casos es común que el analista omita pasos de los flujos alternos, sobre todo los relacionados con la validación de las entradas, en otros casos se omite la descripción relacionada con el comportamiento del sistema ante un error determinado. Esto influye de manera negativa en el producto final, si los casos de uso no se corresponden con las funcionalidades descritas en los requisitos funcionales y los flujos no quedan descritos de manera explícita entonces temas medulares en un sistema como el control de flujo y el tratamiento de excepciones quedan a disposición de la creatividad de los desarrolladores, incluso en proyectos

## CAPÍTULO 2: Solución Propuesta

---

grandes no se logra una uniformidad en el manejo de los errores y las respuestas del sistema a los mismos. Es responsabilidad del analista describir los casos de uso de forma clara, precisa y abarcar todos los flujos, disminuyendo entonces la responsabilidad del desarrollador.

- ✓ Estilos arquitectónicos: los estilos arquitectónicos están basados en los patrones de arquitectura que se usen, agrupan clases englobando una serie de estilos arquitectónicos que comparten características que sirven para sintetizar la estructura de las aplicaciones y permiten evaluar arquitecturas alternativas con ventajas y desventajas conocidas ante diferentes conjuntos de requisitos no funcionales.
- ✓ Patrones de diseño: los patrones del diseño son los que definen la estructura del proyecto sobre la que van a implementar los desarrolladores, además constan de subsistemas, sus responsabilidades e interrelaciones. Presentan una serie de directivas para organizar los componentes del sistema.
- ✓ Modelos de evaluación de arquitectura: sirven como medio para la obtención de indicadores de calidad del producto contribuyendo a identificar los errores del sistema. A continuación se explica cada uno de estos modelos.
  - SAMM (Método de Análisis de arquitectura de software): Es un modelo basado en escenarios, donde los interesados comprenden las arquitecturas evaluadas y la documentación es mejorada. Se sugiere utilizar cuándo el atributo de calidad modificabilidad es el de mayor interés.
  - ATAM (Método de Análisis de Acuerdos de Arquitectura): Este método de evaluación obtiene su nombre no solo porque nos dice cuán bien una arquitectura particular satisface las metas de calidad, sino que también provee ideas de cómo esas metas de calidad interactúan entre ellas, como realizan concesiones mutuas entre ellas. Es el más profundo para evaluar aspectos más relacionados con la arquitectura, como la performance y la confiabilidad.
  - ARID (Revisión Activa para el Diseño Intermedio): Utilizado para la evaluación de diseños detallados de unidades del software como los componentes o módulos. Las preguntas giran

## CAPÍTULO 2: Solución Propuesta

---

en torno a la calidad y completitud de la documentación y la suficiencia, el ajuste y la conveniencia de los servicios que provee el diseño propuesto. Este método evalúa mejor la factibilidad de la arquitectura.

### 2.1.5 Implementación

En la implementación a partir de los resultados del análisis y diseño se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ejecutables y similares.

Elementos críticos de esta etapa:

- ✓ La omisión de las pautas definidas por la arquitectura para el manejo de los errores y el control del flujo: descripción incorrecta de los casos de uso donde se omiten los flujos de recuperación ante los errores. Puede ser responsabilidad del arquitecto velar por el cumplimiento de las pautas para el tratamiento de los errores definidas previamente.
- ✓ Buenas Prácticas de Programación: sirven para mejorar y comprender la claridad del código.
- ✓ Tratamiento de errores y respuesta del sistema a los mismos: el desarrollador debe realizar las verificaciones necesarias para evitar la ocurrencia de errores, igual debe utilizar un mecanismo de captura de excepciones siguiendo las pautas definidas por la arquitectura y darle solución a los errores mediante la descripción de los flujos relacionados con la recuperación ante los errores previamente definidos en las descripciones de los casos de uso.

### 2.1.6 Pruebas

En el ciclo de vida del programa de mejora las pruebas están comprendidas en 2 etapas, una de los proyectos y otra de un ente externo al proyecto pero al final las 2 etapas utilizan los mismos mecanismos para llevar a cabo el proceso, por tanto se definió que los elementos críticos a tener en cuenta son los mismos en ambas etapas. (12)

#### Pruebas Internas

Durante esta etapa el proyecto verifica el resultado de la implementación probando según sea necesario cada componente o unidad de prueba, incluyendo tanto los componentes internos como intermedios, así

## CAPÍTULO 2: Solución Propuesta

---

como las versiones finales a ser liberadas. Además se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posibles componentes de prueba ejecutables para automatizar las pruebas.

Elemento crítico de esta etapa:

- ✓ Mecanismos para realizar pruebas: durante la etapa de pruebas se deben definir los mecanismos que permitan probar el comportamiento del sistema ante situaciones anormales, los resultados de las pruebas deben ser documentados y discutidos con el equipo de desarrollo, en dependencia del tipo de error detectado entonces el arquitecto definirá si los cambios se realizarán en la estructura del sistema que gestiona las excepciones o si deben ser corregidos de manera local en el caso de uso que corresponda a la funcionalidad incorrecta.
- ✓ Preparar las Pruebas: se definen los casos de prueba y los métodos de prueba a partir de la información de los perfiles operacionales. Esta actividad puede integrarse con el proceso de pruebas del modelo de desarrollo que se tenga. Lo importante en esta etapa es la decisión de qué cosas se van a probar y qué datos se usaran en los casos de prueba.
- ✓ Ejecutar las Pruebas: se asignan los tiempos para las pruebas entre los sistemas, los tipos de prueba (características, carga y regresión) así como su ejecución.
- ✓ Guiar las Pruebas: se procesa la información obtenida en la ejecución de las pruebas con varias intenciones. El primero es monitorear el crecimiento de la confiabilidad del sistema (o la reducción de las intensidades de falla) mientras se van reparando los defectos encontrados que generaron las fallas. Otro propósito es el de poder determinar si es necesario seguir probando.
- ✓ Las pruebas de confiabilidad: tiene el objetivo de detectar y eliminar errores luego de haberle realizado pruebas al software, debido a la complejidad de algunos sistemas puede que no se encuentren todos los posibles errores. Existen varios tipos de pruebas de confiabilidad:
  - Pruebas de Estrés: consiste en la simulación de grandes cargas de trabajo para observar de qué forma se comporta la aplicación ante situaciones de uso intenso. Dentro de ellas se encuentran las pruebas de integración que no son más que las interacciones con otras estructuras de datos, procesos y servicios de los componentes internos y externos del

## CAPÍTULO 2: Solución Propuesta

---

sistema y las pruebas de componentes que se dedican a forzar cada componente de forma aislada más de lo que la aplicación podría experimentar en condiciones normales.

- **Pruebas Reales:** se realizan con el objetivo de asegurarse de que la nueva aplicación puede ejecutarse en el espacio del servidor, con la configuración final, y con plena experiencia en el perfil de eventos del cliente previsto.
- **Pruebas de destrucción aleatorias:** consiste en realizar la entrada de datos de forma aleatoria e intenta por todos los medios bloquear la aplicación o que esta produzca fallas.
- **Pruebas de Integración:** su objetivo es identificar errores introducidos por la combinación de programas probados unitariamente. Además verificar que las especificaciones de diseño sean alcanzadas.
- **Pruebas Estructurales:** las pruebas estructurales son conocidas también como las pruebas de caja blanca o las pruebas basadas en código, donde se enfocan en probar cada una de las estructuras de código, para que su comportamiento sea el esperado.
- **Pruebas de Recuperación y Tolerancia a fallos:** consiste en introducir determinados errores en el sistema para comprobar la recuperación, respuesta y el grado de tolerancia del mismo ante fallos. (13)

### **Pruebas de Liberación**

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

#### **2.1.7 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.

## CAPÍTULO 2: Solución Propuesta

---

Elemento crítico de esta etapa:

- ✓ Los escenarios de despliegue: son los lugares donde se va a utilizar el software que se está desarrollando y deben cumplir con los requisitos de software y hardware definidos.

### 2.1.8 Soporte

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

Elemento crítico de esta etapa:

- ✓ Elementos definidos en la implementación: en esta etapa se trata de las modificaciones que se le van a realizar al software por lo que se deben tener en cuenta los elementos definidos en la implementación para así garantizar la confiabilidad del mismo.

### 2.2 Guía de actividades a realizar por etapas.

Cualquier proyecto de desarrollo de software sigue una metodología determinada ya sea ágil o robusta, específicamente en los proyectos que se va a implantar esta guía son los que desarrollan soluciones siguiendo el ciclo de vida de software definido por el programa de mejora. En este punto hay que aclarar que la guía va a ser desarrollada por etapas de desarrollo y para cada una de estas se identifica el elemento que afecta la confiabilidad y se definen las actividades a desarrollar para garantizarla, estas actividades no aparecen definidas en ninguna metodología y se deben realizar de conjunto con las actividades propuestas por la metodología que use el proyecto.

<b>Etapas de desarrollo</b>	<b>Elemento que afecta la confiabilidad</b>	<b>Actividades para garantizar la confiabilidad</b>
-----------------------------	---	---

## CAPÍTULO 2: Solución Propuesta

---

<b>Estudio Preliminar</b>	Alcance del sistema	Definir el alcance del sistema a implementar.
		Definir los posibles usuarios y clientes que va a tener el sistema.
		Identificar el entorno de la organización donde se va a realizar el proyecto.
<b>Modelación del Negocio</b>	Reglas del Negocio	Identificar y definir las reglas del negocio con claridad y sin ambigüedad, definiéndolos flujos de validación y comportamiento del sistema.
		Definir reglas del negocio relacionadas con: <ul style="list-style-type: none"><li>✓ Entidades.</li><li>✓ Actividades.</li><li>✓ Posibles usuarios que contribuyan a mejorar la fiabilidad.</li></ul>
		Definir los diferentes tipos de usuarios que utilizarán el sistema.

## CAPÍTULO 2: Solución Propuesta

---

<b>Requisitos</b>	Requisitos funcionales	Entender las necesidades del cliente y transformarlas en requisitos entendibles por el equipo de desarrollo.
		Definir las entradas y salidas del sistema (datos, tipo de datos y restricciones de los mismos).
		Definir el nivel de fiabilidad que va a tener el sistema.
		Realizar el perfil de operaciones. Ver Anexo 2
	Requisitos no funcionales	Definir los requisitos no funcionales que garantizan la disponibilidad y la confiabilidad.
		Elegir el hardware en correspondencia con la tecnología de software definida. <ul style="list-style-type: none"> <li>✓ RAM</li> <li>✓ Microprocesador</li> <li>✓ Disco Duro</li> <li>✓ Tarjeta Gráfica</li> <li>✓ Conectividad</li> <li>✓ Periféricos</li> </ul>
		Definir de forma medible el

## CAPÍTULO 2: Solución Propuesta

---

		rendimiento de la aplicación que estará dado en el tiempo de respuesta de las funcionalidades arquitectónicamente significativas.
		Definir cuántas transacciones va a realizar el sistema en un tiempo determinado.
		Definir la cantidad de transacciones en paralelo que soporta el sistema.
<b>Análisis y Diseño</b>	Casos de uso	Describir los casos de uso de forma clara, precisa y abarcar todos los flujos, tanto el flujo básico como todos los alternativos.
	Estilos arquitectónicos	Diseñar los mecanismos para garantizar el cumplimiento de los requisitos identificados en la etapa de requisitos a través de componentes del diseño.
	Patrones de diseño	<ul style="list-style-type: none"> <li>✓ Seleccionar correctamente los estilos y patrones arquitectónicos                             <ul style="list-style-type: none"> <li>• Definir el patrón de organización general.</li> <li>• Identificar las interacciones que se</li> </ul> </li> </ul>

## CAPÍTULO 2: Solución Propuesta

---

	Modelos de evaluación de la arquitectura	<p style="text-align: center;">establecen entre los componentes.</p> <p>Utilizar modelos de evaluación de arquitectura.</p>
<b>Implementación</b>	Buenas Prácticas de Programación	<p>Utilizar entornos de desarrollo adecuados para gestionar componentes.</p> <p>Utilizar buenas prácticas de programación como son los estándares de codificación.</p>
	Pautas definidas por la arquitectura	<p>Realizar una correcta implementación en la aplicación de las funciones descritas en las especificaciones.</p> <p>Tener en cuenta las pautas definidas por la arquitectura para el manejo de los errores y el control del flujo.</p>
	Tratamiento de errores y respuesta del sistema	<p>Tratamiento de errores y respuesta del sistema a los mismos.</p> <ul style="list-style-type: none"> <li>✓ Realizar las validaciones necesarias para evitar la ocurrencia de errores.</li> <li>✓ Usar procedimientos de control de cambios.</li> </ul>

## CAPÍTULO 2: Solución Propuesta

---

		<ul style="list-style-type: none"> <li>✓ Utilizar un mecanismo de captura de excepciones siguiendo las pautas definidas por la arquitectura.</li> </ul>
<p style="text-align: center;"><b>Prueba de Internas</b></p>	<p style="text-align: center;">Mecanismos para realizar pruebas</p>	<p>Definir los mecanismos que permitan probar el comportamiento del sistema ante situaciones anormales.</p>
		<p>Discutir y documentar con el equipo de desarrollo los resultados de las pruebas realizadas.</p>
	<p style="text-align: center;">Pruebas de Confiabilidad</p>	<p>Realizar las pruebas de Confiabilidad.</p> <ul style="list-style-type: none"> <li>✓ Pruebas de Estrés.</li> <li>✓ Pruebas Reales.</li> <li>✓ Pruebas de destrucción aleatorias.</li> <li>✓ Pruebas de Integración.</li> <li>✓ Pruebas Estructurales.</li> <li>✓ Pruebas de Recuperación y Tolerancia a fallos.</li> </ul>
		<p>Diseñar casos de prueba para probar el perfil de operaciones.</p> <ul style="list-style-type: none"> <li>✓ Utilizar herramientas.</li> </ul>

## CAPÍTULO 2: Solución Propuesta

---

		<ul style="list-style-type: none"> <li>✓ Realizar pruebas de estrés.</li> </ul>
		<p>Pruebas de aceptación por parte del cliente.</p> <ul style="list-style-type: none"> <li>✓ Comprobar que el sistema cumple con las reglas del negocio y que las funcionalidades responden a los requisitos sin producir fallas.</li> </ul>
<b>Despliegue</b>	Escenarios de despliegue	Comprobar que el escenario de despliegue cumple con los requisitos no funcionales.
		Realizar pruebas pilotos.
<b>Soporte</b>	Elementos definidos en la implementación	Tener en cuenta los elementos definidos en la etapa de implementación.

**Tabla 4 Guía de actividades a realizar por etapa.**

### Conclusiones Parciales

En la solución planteada del capítulo quedó evidenciada la guía de actividades garantizando que cada etapa del ciclo de desarrollo de software, contenga las actividades necesarias para lograr la confiabilidad en las aplicaciones de escritorio y así tener menos errores una vez terminado el producto.

## CAPÍTULO 3: Validación de la propuesta

---

### CAPÍTULO 3 VALIDACIÓN DE LA PROPUESTA

#### Introducción

El presente capítulo está dirigido a la aceptación de la guía propuesta para evaluar la confiabilidad en el desarrollo de las aplicaciones de escritorio de gobierno electrónico, se seleccionarán un grupo de especialistas a los cuales se les realizará una encuesta para comprobar la efectividad y eficiencia de la misma.

El criterio de especialistas es considerado como uno de los métodos subjetivos de pronósticos más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de expertos en el tema tratado. El mismo permite rebasar el marco de las condicionantes actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos, las cuales podrán estar fundamentadas, tanto en un análisis estrictamente lógico como en su experiencia intuitiva.

#### **3.1 Procedimiento utilizado para la aplicación del método.**

Después de haber decidido aplicar el método de especialistas, se realiza la selección de los expertos o especialistas en el tema a evaluar. Estos serán grupos de personas con un gran conocimiento sobre el tema tratado, que podrán emitir criterios incuestionables de cualquier problema y valoraciones importantes con un alto nivel de conocimiento.

- ✓ En este caso se escoge como área de conocimiento la arquitectura de software y la calidad de software ya que es donde se trabajan las características de calidad de un software específicamente para esta investigación, la confiabilidad.
- ✓ Posteriormente se identifican a través de un muestro intencional los 5 especialistas que evaluarán la propuesta en correspondencia con el nivel de competencia en dicho tema.
  - Nivel de competencia.

## CAPÍTULO 3: Validación de la propuesta

---

Para la selección de los especialistas que realizaron la valoración de la guía para el aseguramiento de la confiabilidad propuesta, se tuvo en cuenta una serie de características que posibilitaron obtener información de la trayectoria y actualizada sobre los posibles especialistas. Basándose principalmente en la participación de estos en eventos científicos, la labor que desempeñan actualmente, los años de experiencia en el tema arquitectura, calidad de software y los conocimientos teóricos que tienen del tema tratado.

Basándose en las informaciones antes expuestas, la lista de especialistas a consultar quedó conformada por 5 especialistas, que de una forma u otra están vinculados al área de arquitectura y calidad de software y cuentan con una ardua experiencia en el tema. A continuación se muestra la Tabla 7 con el listado de especialistas.

Listado de especialistas		Cargos
1	Eduardo Aranda Pierre	Arquitecto-DESPROD-CEIGE
2	Eliecer Cabrera Casas	Arquitecto
3	Jorge Alberto Mora Julián	Especialista en calidad
4	Yunier Pérez Barroso	Arquitecto Principal CEGEL
5	José Angel Martínez García	Arquitecto Principal Proyecto CDA

**Tabla 5 Listado de especialistas.**

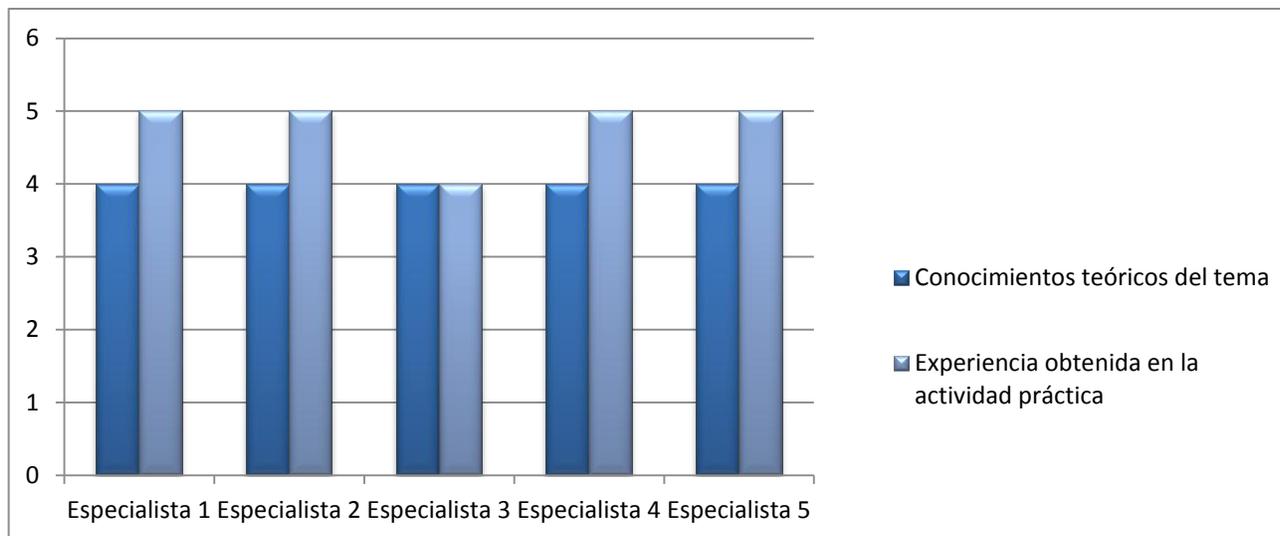
De acuerdo al cuestionario aplicado se obtuvo el siguiente resultado con respecto a la autovaloración del nivel de competencia de cada uno de los especialistas:

## CAPÍTULO 3: Validación de la propuesta



**Figura 7 Nivel de autovaloración de los especialistas en una escala del 1 al 5**

Para conocer dos elementos fundamentales como son los conocimientos teóricos del tema y la experiencia obtenida en la actividad práctica el instrumento aplicado aportó la siguiente información.



**Figura 8 Nivel de competencia de los especialistas**

En la misma se evidencia que en una escala del 1 al 5, el promedio de conocimiento teórico que poseen los especialistas es de 4. En cuanto a la experiencia en la actividad práctica el promedio es de 4,8.

## CAPÍTULO 3: Validación de la propuesta

---

Estos datos demuestran un nivel de competencia adecuado para la validación de esta propuesta, lo que da alto valor a sus criterios con respecto a cada una de las preguntas realizadas en el cuestionario.

- ✓ Se analizan los resultados de la aplicación del cuestionario donde se evalúa por parte de los especialistas cada elemento definido en la solución.

En el cuestionario se formulan 8 preguntas destinadas a valorar los elementos definidos para asegurar la confiabilidad, durante las etapas del ciclo de desarrollo de software basado en el programa de mejora que lleva a cabo la universidad. Dichas respuestas se presentan a continuación para que sean comprendidas de la mejor manera.

	<b>Muy Adecuado</b>	<b>Bastante Adecuado</b>	<b>Adecuado</b>	<b>Poco Adecuado</b>	<b>No Adecuado</b>
Pregunta # 1	<b>1</b>	<b>1</b>	<b>3</b>		
Pregunta # 2	<b>3</b>	<b>2</b>			
Pregunta # 3	<b>4</b>	<b>1</b>			
Pregunta # 4		<b>4</b>	<b>1</b>		
Pregunta # 5	<b>3</b>	<b>1</b>	<b>1</b>		
Pregunta # 6	<b>4</b>	<b>1</b>			
Pregunta # 7		<b>3</b>	<b>2</b>		
Pregunta # 8	<b>1</b>	<b>2</b>	<b>2</b>		

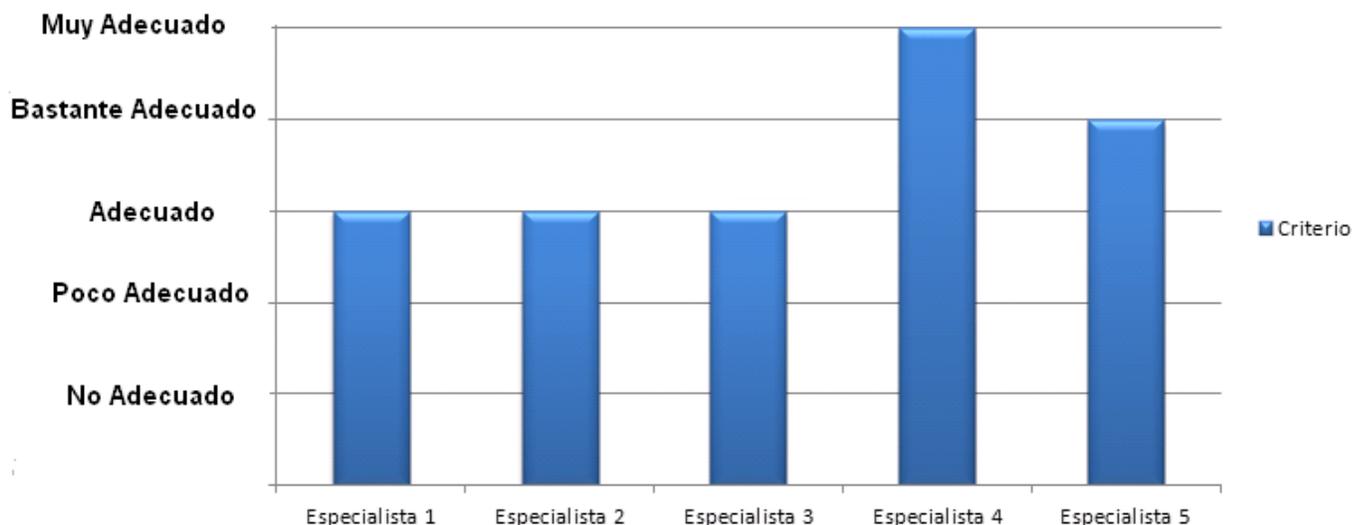
## CAPÍTULO 3: Validación de la propuesta

**Tabla 6 Respuestas a las preguntas aplicadas.**

El proceso de evaluación es iterativo, se realizan varias rondas hasta que haya un consenso entre los especialistas, en este caso solo fue necesario realizar una ronda. Hay que decir además, que el cuestionario se le envía a cada uno de los especialistas por separado para no condicionar las respuestas de los demás. Los datos que se muestran a continuación están referidos a la ronda que se realizó. A continuación se muestra toda la información recopilada a partir de la aplicación del instrumento.

### Resultados de la aplicación del cuestionario

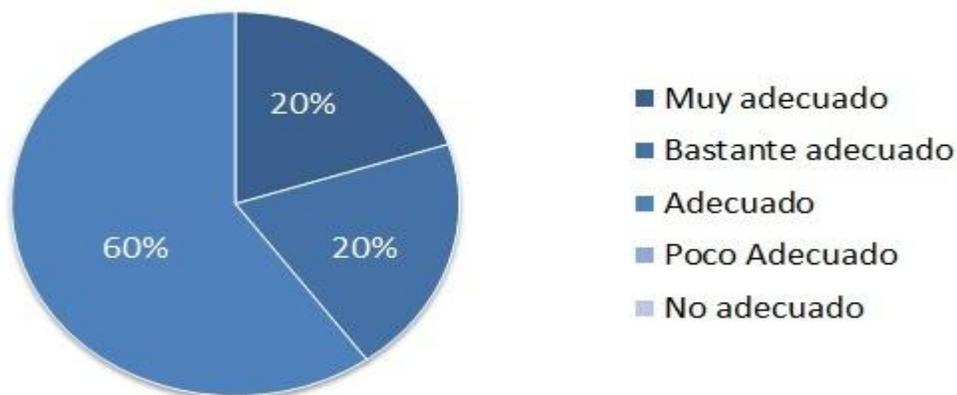
En la respuesta a la primera pregunta acerca de la definición del alcance del sistema como elemento significativo en la etapa de estudio preliminar que influye en la confiabilidad del sistema, los especialistas evaluaron lo siguiente:



**Figura 9 Criterio de evaluación de la pregunta 1 por especialista.**

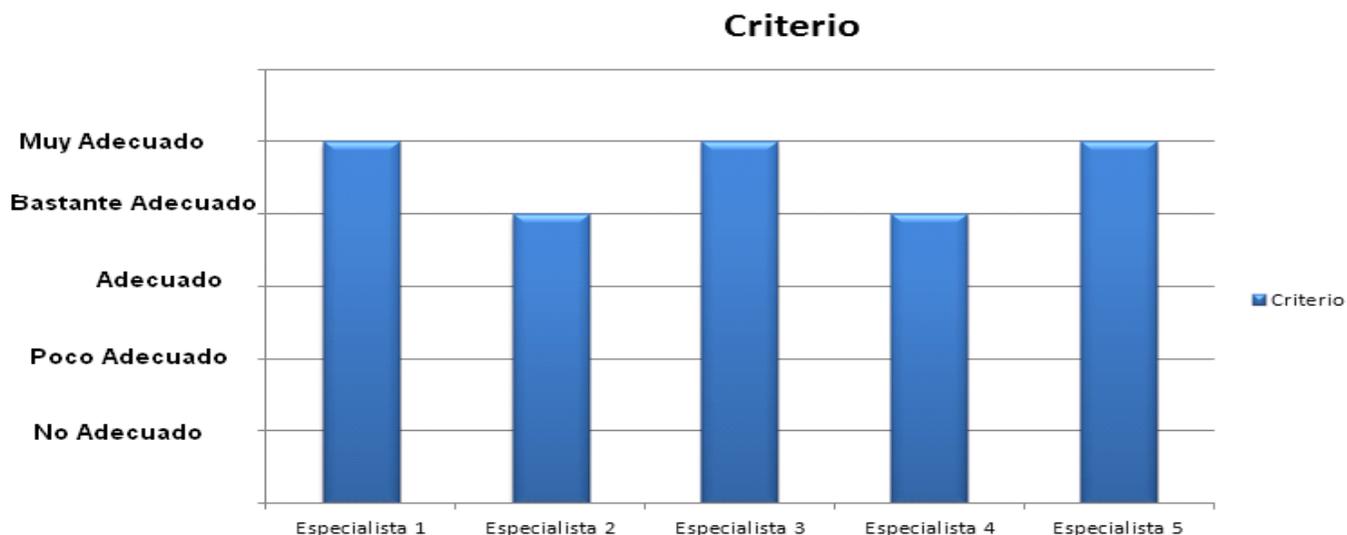
El 100% de los especialistas valora entre Muy adecuado y Adecuado la definición del alcance del sistema como se aprecia en la Figura 10.

## CAPÍTULO 3: Validación de la propuesta



**Figura 10 Por ciento del criterio de evaluación de los especialistas para la pregunta 1.**

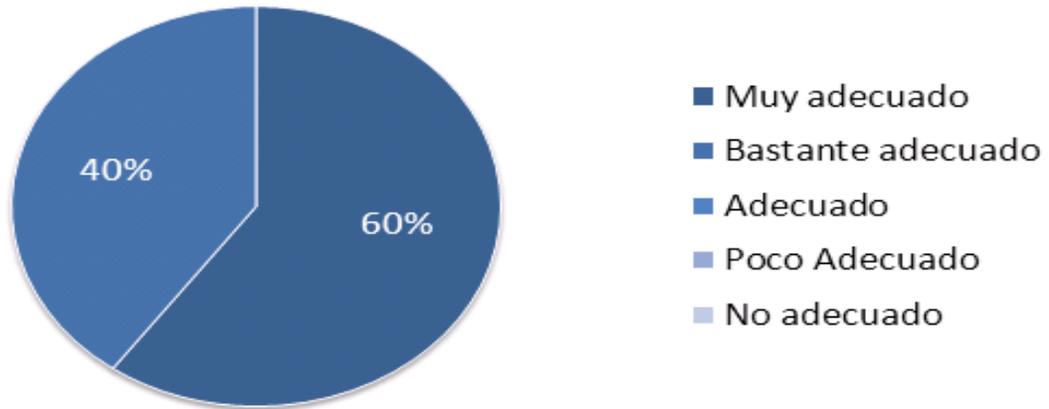
En la respuesta a la segunda pregunta del cuestionario acerca de la definición de las Reglas del negocio como elemento importante en la etapa de Modelación de Negocio que influye en la confiabilidad del sistema, los especialistas evaluaron lo siguiente:



**Figura 11 Criterio de evaluación de la pregunta 2 por especialista.**

## CAPÍTULO 3: Validación de la propuesta

El 100% de los especialistas valora entre Muy adecuado y Bastante Adecuado la definición de las reglas del negocio como se aprecia en la Figura 12.



**Figura 12 Por ciento del criterio de evaluación de los especialistas para la pregunta 2.**

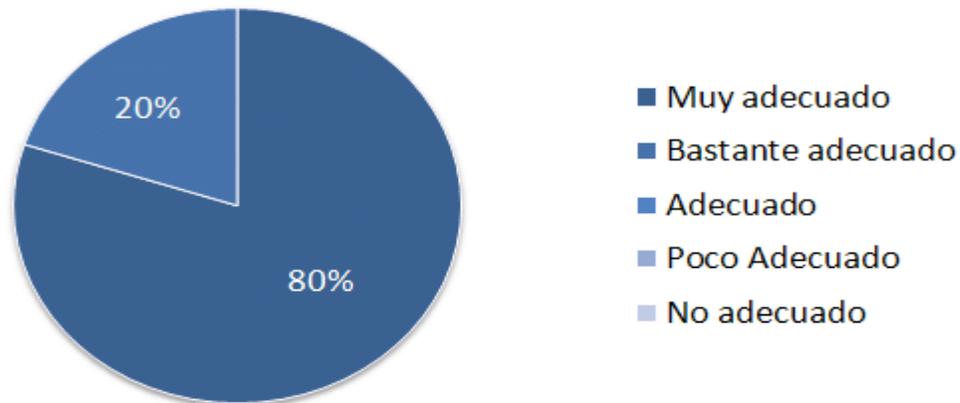
En la respuesta a la tercera pregunta del cuestionario acerca la definición de los requisitos funcionales y no funcionales en la etapa de requisitos afecta la confiabilidad del sistema, los especialistas evaluaron lo siguiente:



**Figura 13 Criterio de evaluación de la pregunta 3 por especialista.**

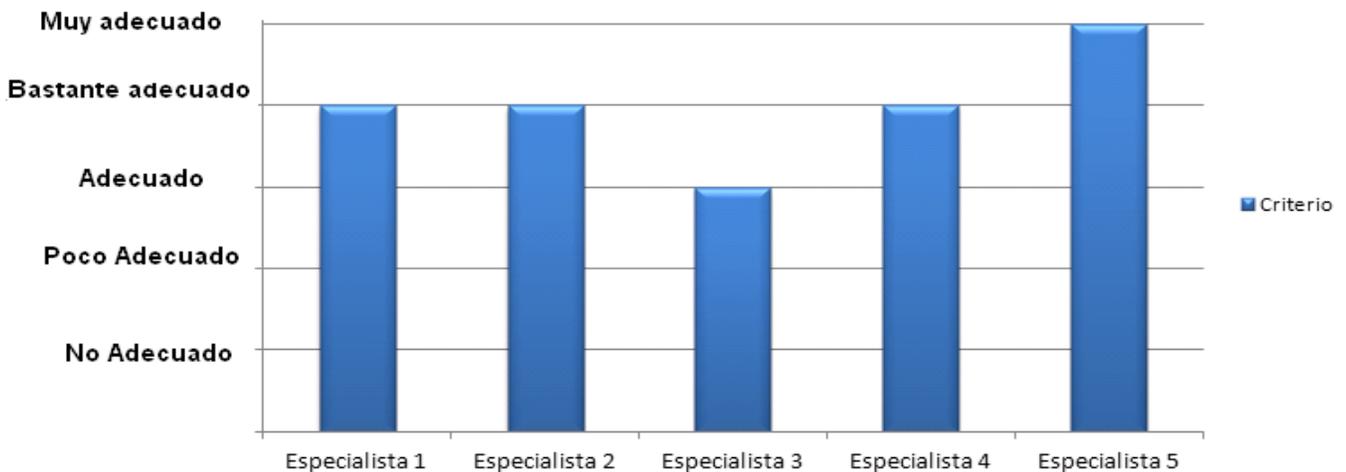
## CAPÍTULO 3: Validación de la propuesta

El 100% de los especialistas valora entre Muy adecuado y Bastante Adecuado la definición de los requisitos funcionales y no funcionales como se aprecia en la Figura 14.



**Figura 14 Por ciento del criterio de evaluación de los especialistas para la pregunta 3.**

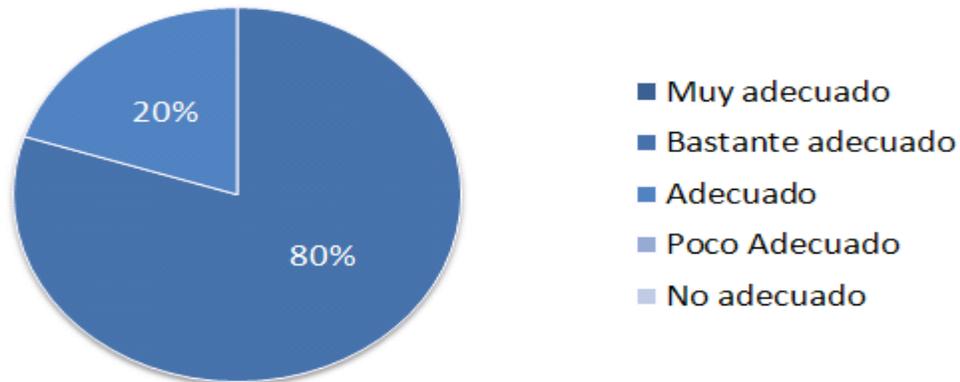
En la respuesta a la cuarta pregunta del cuestionario acerca de la descripción de los casos de uso, así como seleccionar los estilos, patrones arquitectónicos y modelos de evaluación de arquitectura son elementos primordiales en la etapa de análisis que garantizan el aseguramiento de la confiabilidad.



**Figura 15 Criterio de evaluación de la pregunta 4 por especialista.**

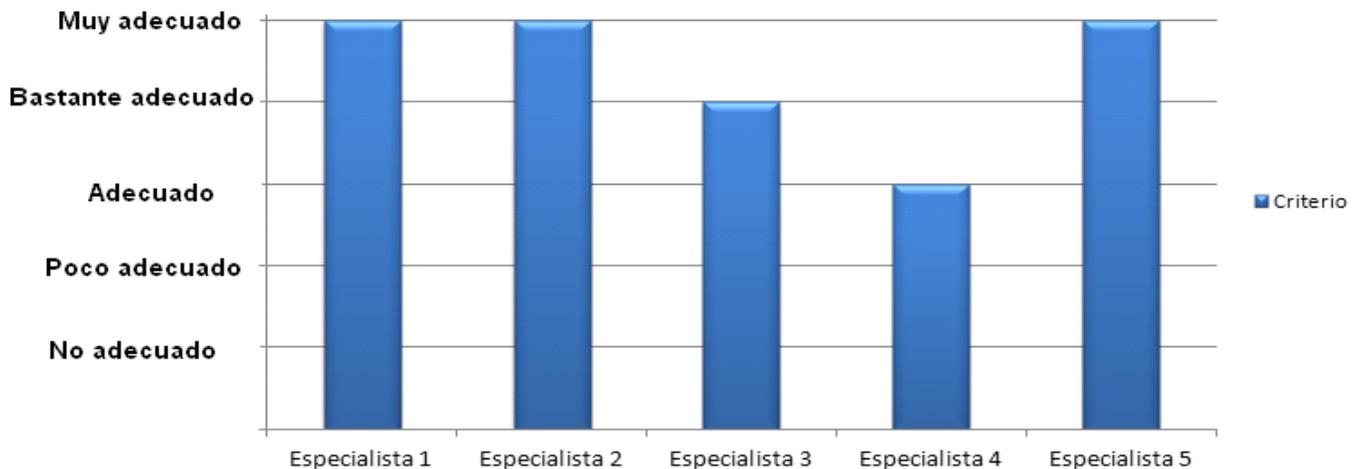
## CAPÍTULO 3: Validación de la propuesta

El 100% de los especialistas valora entre Bastante Adecuado y Adecuado la descripción de los casos de uso, seleccionar los estilos, patrones arquitectónicos y modelos de evaluación de arquitectura como se aprecia en la Figura 16.



**Figura 16 Por ciento del criterio de evaluación de los especialistas para la pregunta 4.**

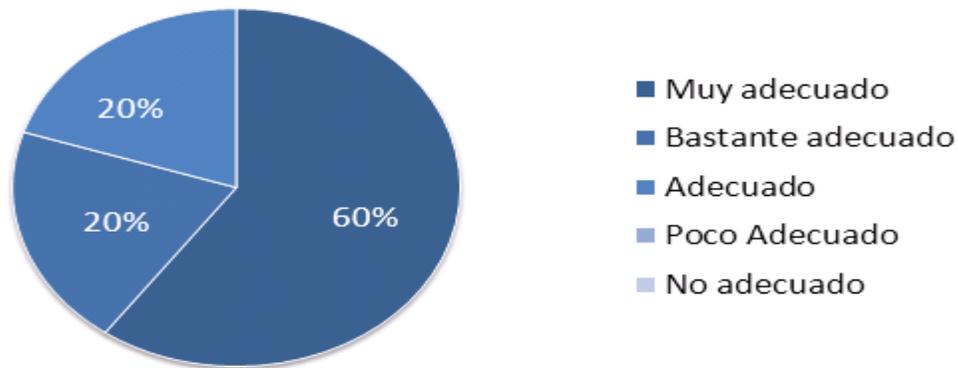
En la respuesta a la quinta pregunta del cuestionario acerca las pautas definidas y el manejo de los errores, las buenas prácticas de programación, el tratamiento de errores y respuesta del sistema a los mismos son elementos que influyen en la confiabilidad del sistema en la etapa de Implementación.



**Figura 17 Criterio de evaluación de la pregunta 5 por especialista.**

## CAPÍTULO 3: Validación de la propuesta

El 100% de los especialistas valora entre Muy adecuado y adecuado las pautas definidas y el manejo de los errores, las buenas prácticas de programación, el tratamiento de errores y respuesta del sistema a los mismos como se aprecia en la Figura 18.



**Figura 18 Por ciento del criterio de evaluación de los especialistas para la pregunta 5.**

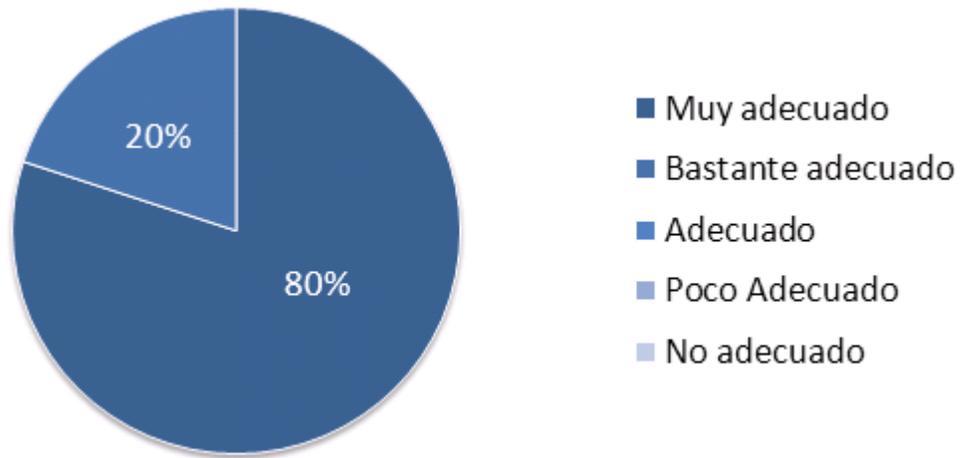
En la respuesta a la sexta pregunta del cuestionario sobre el comportamiento del sistema y las pruebas de confiabilidad (Pruebas de Estrés, Pruebas Reales, Pruebas de destrucción aleatorias, Pruebas de Integración, Pruebas Estructurales, Pruebas de Recuperación y Tolerancia a fallos) son el elemento que influye en la confiabilidad del sistema en la etapa de pruebas.



**Figura 19 Criterio de evaluación de la pregunta 6 por especialista.**

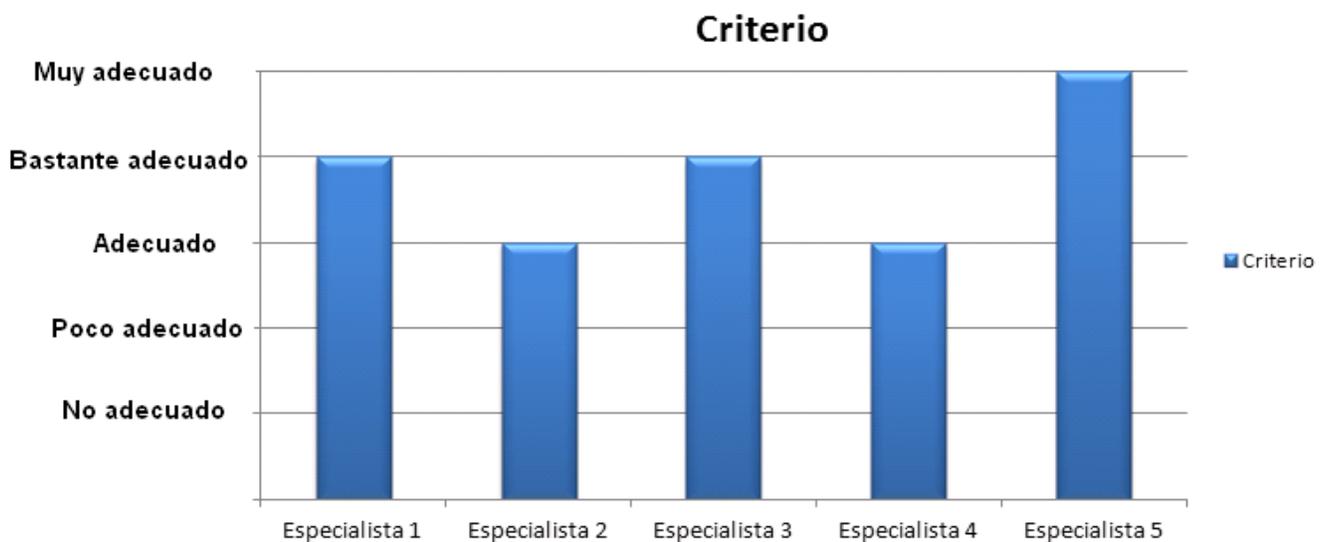
## CAPÍTULO 3: Validación de la propuesta

El 100% de los especialistas valora entre Muy adecuado y Bastante adecuado el comportamiento del sistema y las pruebas de confiabilidad como se aprecia en la Figura 20.



**Figura 20 Por ciento del criterio de evaluación de los especialistas para la pregunta 6.**

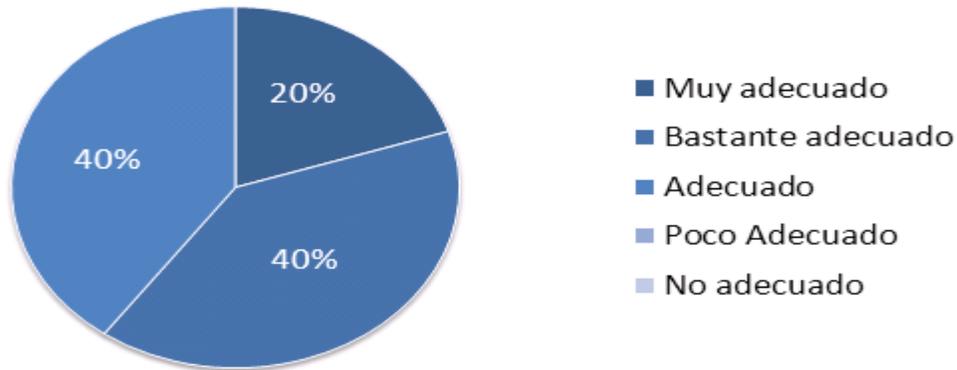
En la respuesta a la séptima pregunta del cuestionario acerca de los escenarios de despliegue como elemento que influye en la confiabilidad del sistema en la etapa de Despliegue.



**Figura 21 Criterio de evaluación de la pregunta 7 por especialista.**

## CAPÍTULO 3: Validación de la propuesta

El 100% de los especialistas valora entre Muy adecuado y Adecuado tener en cuenta los escenarios de despliegue como se aprecia en la Figura 22.



**Figura 22 Por ciento del criterio de evaluación de los especialistas para la pregunta 7.**

En la respuesta a la octava pregunta del cuestionario acerca de tener en cuenta los elementos definidos en la etapa de implementación son el elemento que garantiza la confiabilidad del sistema en la etapa de soporte.

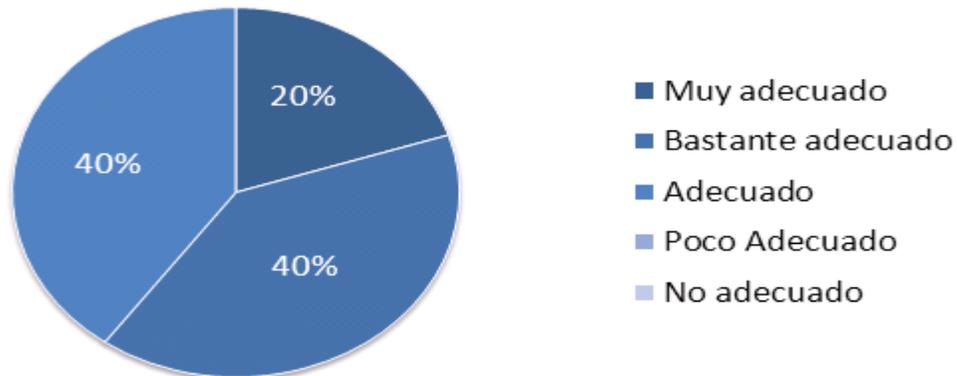


**Figura 23 Criterio de evaluación de la pregunta 8 por especialista.**

## CAPÍTULO 3: Validación de la propuesta

---

El 100% de los especialistas valora entre Muy adecuado y Adecuado tener en cuenta los elementos definidos en la etapa de implementación como se muestra en la Figura 24.



**Figura 24 Por ciento del criterio de evaluación de los especialistas para la pregunta 8.**

Como se puede evidenciar por lo antes mostrado 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, para determinar el grado de concordancia entre los especialistas mediante el coeficiente de Kendall (W) y la prueba de hipótesis. Para ello se utilizó herramienta Paquete Estadístico para las Ciencias Sociales (SPSS) para la explotación de los datos, obteniéndose como resultado:

**Kendall's W Test**

Test Statistics

N	5
Kendall's W <sup>a</sup>	,822
Chi-Square	,056
df	5
Asymp. Sig.	,003

a. Kendall's Coefficient of Concordance

**Figura 25 Tabla estadística.**

## CAPÍTULO 3: Validación de la propuesta

---

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:

$H_0$  = 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  $W = 0,82$ , 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.

### **Conclusiones parciales**

Luego de haber analizado los resultados de los especialistas que estuvieron valorados entre Muy adecuado y Adecuado y que existe un nivel de concordancia alto entre los especialistas, se puede decir que la guía para el aseguramiento de la confiabilidad en el desarrollo de aplicaciones de escritorio cumple con los objetivos propuestos.

## CONCLUSIONES

---

### CONCLUSIONES

Al finalizar la presente investigación se dio cumplimiento al objetivo general concebido, arrojando a las siguientes conclusiones:

- ✓ Se desarrolló un estudio exhaustivo de la bibliografía relacionada con el tema como parte de la fundamentación teórica definiéndose la posición del investigador.
- ✓ Se comprobó a través de una entrevista que los proyectos del centro CEGEL no utilizan ningún modelo para asegurar la confiabilidad.
- ✓ Se escogió un modelo de calidad que abarca las subcaracterísticas de confiabilidad como son: la madurez, la tolerancia a fallos, la recuperabilidad y la fiabilidad conforme, que sirvió como guía para el desarrollo de la propuesta de solución.
- ✓ Se determinó en cada una de las etapas del ciclo de vida del software elementos que afectan la confiabilidad del sistema.
- ✓ Se desarrolló una guía para el aseguramiento de la confiabilidad en los proyectos del centro CEGEL, con el propósito de determinar la aplicabilidad de la guía y verificar el cumplimiento de los objetivos para la cual fue creada, la misma fue validada satisfactoriamente por especialistas en el tema con un criterio valorado de un 100% entre Muy adecuado y Adecuado.

## RECOMENDACIONES

---

### RECOMENDACIONES

A modo general los objetivos trazados al inicio del trabajo han sido logrados, pero al mismo tiempo, a lo largo del proceso de desarrollo, se hacen las siguientes recomendaciones:

- ✓ Aplicar la guía en los proyectos del CEGEL.
- ✓ Extender la guía propuesta para garantizar el aseguramiento de la confiabilidad en todos los proyectos de la UCI, con el fin de lograr una mayor calidad en los productos de software realizados en la Universidad.
- ✓ Continuar la investigación para incluirle a la guía nuevas ideas que colaboren en su refinamiento y agregar la forma de evaluar estos elementos.

## REFERENCIAS BIBLIOGRÁFICAS

---

### REFERENCIAS BIBLIOGRÁFICAS

1. **Morilla, José Joaquín Ruiz.** ISO 9126 vs. SQuaRE. [En línea] 2008. [Citado el: 25 de marzo de 2012.] <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Joaquin%20Ruiz.pdf>.
2. Real Academia Española. [En línea] 2001. [Citado el: 10 de abril de 2012.] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=calidad](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=calidad).
3. **LuisFuertes, Jose.** [En línea] [Citado el: 10 de abril de 2012.] [http://www.iimv.org/actividades2/08\\_Tecnologia/Jose%20LuisFuertesCalidad%20del%20Software.pdf](http://www.iimv.org/actividades2/08_Tecnologia/Jose%20LuisFuertesCalidad%20del%20Software.pdf)...
4. **Pressman, Roger.** *Ingeniería del software: un enfoque práctico.* [ed.] McGraw-Hill. 2002. Vol. Quinta edición. 2002.
5. **Lovelle, Juan Manuel Cueva.** Calidad del Software. [En línea] 21 de Octubre de 1999. [Citado el: 20 de Febreo de 2012.] [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
6. **Quiñones, Ernesto.** Modelos de calidad de software y software libre . [En línea] 20 de 10 de 2006. [Citado el: 20 de 2 de 2012.]
7. ACIMED. [En línea] [Citado el: 12 de abril de 2012.] [http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci_arttext).
8. **Núñez Camallea, Noel Luis y Ronald, Coutin.** *Diccionario de Informática.* s.l. : Científico, 2005.
9. **Puente, Juan Antonio de la.** [En línea] 2 de febrero de 2002. [Citado el: 2012 de abril de 20.]
10. ISO Standards Definition. [En línea] [Citado el: 10 de marzo de 2012.] [http://www.ehow.com/facts\\_5762097\\_iso-standards-definition.html](http://www.ehow.com/facts_5762097_iso-standards-definition.html).
11. **Fillotrani, Pablo R.** *Calidad en el desarrollo del software "Modelos de calidad del software"*.
12. Calisoft 2012. [En línea] [Citado el: 1 de abril de 2012.] <http://calisoft.uci.cu/index.php/proceso-de-mejora>.
13. **Carolina Zibert van Gricken y Ivan Boucchechter.** [En línea] 5 de mayo de 2005. [Citado el: 22 de 4 de 2012.]

## BIBLIOGRAFÍA

---

### BIBLIOGRAFÍA

1. **Morilla, José Joaquín Ruiz.** ISO 9126 vs. SQuaRE. [En línea] 2008. [Citado el: 25 de marzo de 2012.] <http://alarcos.inf-cr.uclm.es/doc/cmsi/trabajos/Joaquin%20Ruiz.pdf>.
2. Real Academia Española. [En línea] 2001. [Citado el: 10 de abril de 2012.] [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=calidad](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=calidad).
3. **LuisFuertes, Jose.** [En línea] [Citado el: 10 de abril de 2012.] [http://www.iimv.org/actividades2/08\\_Tecnologia/Jose%20LuisFuertesCalidad%20del%20Software.pdf](http://www.iimv.org/actividades2/08_Tecnologia/Jose%20LuisFuertesCalidad%20del%20Software.pdf)...
4. **Pressman, Roger.** *Ingeniería del software: un enfoque práctico.* [ed.] McGraw-Hill. 2002. Vol. Quinta edición. 2002.
5. ACIMED. [En línea] [Citado el: 12 de abril de 2012.] [http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94352009001200003&script=sci_arttext).
6. **Núñez Camallea, Noel Luis y Ronald, Coutin.** *Diccionario de Informática.* s.l. : Científico, 2005.
7. **Puente, Juan Antonio de la.** [En línea] 2 de febrero de 2002. [Citado el: 2012 de abril de 20.]
8. ISO Standards Definition. [En línea] [Citado el: 10 de marzo de 2012.] [http://www.ehow.com/facts\\_5762097\\_iso-standards-definition.html](http://www.ehow.com/facts_5762097_iso-standards-definition.html).
9. **Fillottrani, Pablo R.** *Calidad en el desarrollo del software "Modelos de calidad del software".*
10. **Quiñones, Ernesto.** Modelos de calidad de software y software libre . [En línea] 20 de 10 de 2006. [Citado el: 20 de 2 de 2012.]
11. Calisoft 2012. [En línea] [Citado el: 1 de abril de 2012.] <http://calisoft.uci.cu/index.php/proceso-de-mejora>.
12. **Carolina Zibert van Gricken y Ivan Boucchechter.** [En línea] 5 de mayo de 2005. [Citado el: 22 de 4 de 2012.]
13. **Lovelle, Juan Manuel Cueva.** Calidad del Software. [En línea] 21 de Octubre de 1999. [Citado el: 20 de Febreo de 2012.] [http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad\\_software.PDF](http://gidis.ing.unlpam.edu.ar/downloads/pdfs/Calidad_software.PDF).
14. **Axentia.** Una Introducción a CMMI. [En línea] 2006. [Citado el: 18 de abril de 2012.] <http://es.scribd.com/doc/69546265/WP03-Una-Introduccion-a-CMMI>.
15. **Sommerville, Ian.** *Ingeniería del Software.* 2006.

## BIBLIOGRAFÍA

---

16. **Juan Antonio de la Puente.** [En línea] 2006. [Citado el: 2012 de abril de 18.]  
<http://laurel.datsi.fi.upm.es/~ssoo/STR/Fiabilidad.pdf>.
17. **Irene Eusgeld, Falk Fraikin, Matthias Rohr, Felix Salfne, Ute Wappler.** [En línea] 2008. [Citado el: 20 de abril de 2012.] <http://se.inf.tu-dresden.de/pubs/papers/Eus2008.pdf>.
18. **Maceo, Raydel Daniel Abreu.** Guía para el aseguramiento de la eficiencia en los sistemas de gobierno electrónico. La Habana : s.n., 2011.
19. **LLanes, Kathrin Rodríguez.** PROPUESTA PARA LOGRAR FIABILIDAD EN LAS APLICACIONES BIOINFORMATICAS VISTA DESDE LA OBTENCION DE. 2007.
20. **Katy Espronceda Jor, Yelina León Sosa.** Diseño de un procedimiento para evaluar la confiabilidad de los productos con tecnología Postgres, de DATEC. 2010.

# ANEXOS

---

## ANEXOS

### Anexo 1

**Entrevista aplicada a los proyectos del Centro para verificar cómo aseguran la confiabilidad en sus productos.**

1. ¿Cómo evalúan la confiabilidad del software? \_\_\_\_\_
2. ¿Utilizan algún modelo u estándar para asegurar la confiabilidad en sus productos? \_\_\_\_\_
3. ¿Cuáles son las principales fallas que ha presentado el software? \_\_\_\_\_

### Anexo 2 Pasos para realizar un perfil de operación.

#### 1-Desarrollar perfil de operaciones

No. de la funcionalidad	Nombre de la Funcionalidad	Probabilidad
1	[Nombre del Caso de Uso o requisito funcional, historia de usuario u otro artefacto que defina la Metodología utilizada ordenado ascendentemente en dependencia de la importancia de la funcionalidad para la arquitectura] Un caso de uso es arquitectónicamente significativo si: <ul style="list-style-type: none"><li>• El caso de uso resuelve un objetivo del negocio.</li><li>• Las clases o componentes involucrados en la implementación del caso de uso son claves en la arquitectura.</li><li>• El caso de uso es importante para el cliente</li></ul>	$\frac{\text{No. de la Funcionalidad}}{\text{Total de las funcionalidades}}$
2		
n		

## ANEXOS

---

2- **Definir el Nivel de Falla:** se define lo que se considera como “falla” para el producto en desarrollo así como los medios para identificarla. Esta definición es crítica para el proceso y debe ser constante durante todo el ciclo de vida.

<b>Severidad de la Falla</b>	<b>Falla</b>	<b>Impacto al Sistema</b>	<b>Cuantificación de la intensidad de la falla</b>
Alta Media Baja	Descripción de la Falla	Consecuencias para el sistema si ocurre la falla	Cantidad de transacciones, páginas impresas, llamadas a funciones, accesos.

### Anexo 3

Encuesta aplicada a especialista

**Encuesta a especialistas para someter a sus criterios la propuesta de la Guía para el aseguramiento de la confiabilidad en el desarrollo de aplicaciones de escritorio 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 confiabilidad en el desarrollo de aplicaciones de escritorio de Gobierno Electrónico.

Nombre y Apellidos: \_\_\_\_\_

Fecha de graduación: \_\_\_\_\_ Puesto de trabajo actual: \_\_\_\_\_

Calificación profesional: Ingeniero\_\_\_ Licenciado en Educación\_\_\_ Master\_\_\_ Doctor\_\_\_

Categoría Docente: Prof. Instructor\_\_\_ Prof. Asistente\_\_\_ Prof. Auxiliar\_\_\_ Prof. Titular\_\_\_ Prof.

Adjunto\_\_\_\_\_

1. Seleccione en una escala del 1 – 5 el valor que corresponda con el grado de conocimientos que

## ANEXOS

usted posee acerca del tema de investigación que desarrollamos (aseguramiento de la confiabilidad como característica de calidad en el desarrollo de las aplicaciones de escritorio gobierno electrónico), considerando 1 como no tener ningún conocimiento y 5 el de pleno conocimiento de la problemática tratada.

1	2	3	4	5

2. Valore el grado de influencia que cada una de las fuentes que le presentamos a continuación ha tenido en su conocimiento y criterios sobre el tema que se investiga.

Fuentes de Argumentación	Grado de influencia de cada fuente		
	Alto	Medio	Bajo
Conocimientos teóricos que posea acerca del tema			
Su experiencia obtenida en la actividad práctica			

La propuesta de la guía para el aseguramiento de la confiabilidad se encuentra adjunta a esta encuesta. Para su análisis y mejor comprensión se le informa que consta con 2 partes fundamentales, la primera parte (columna izquierda) es la relación de las etapas del ciclo de vida de los proyectos y la segunda parte (columna derecha) es las actividades propuestas para cada etapa del ciclo de vida.

3. Le pedimos su criterio acerca del elemento definido en la etapa de Estudio Preliminar

**Criterio de Especialistas**

## ANEXOS

Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera que el alcance del sistema es el elemento fundamental en la etapa estudio preliminar que influye en la confiabilidad del sistema?					

4. Le pedimos su criterio acerca del elemento definido en la etapa de Modelación del Negocio.

Criterio de Especialistas					
Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera usted que las reglas del negocio son el elemento fundamental que influye en la confiabilidad del sistema en la etapa de Modelación del Negocio?					

5. Le pedimos su criterio acerca del elemento definido en la etapa de Requisitos.

## ANEXOS

<b>Criterio de Especialistas</b>					
Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera usted que la definición de los requisitos funcionales y no funcionales en la etapa de requisitos influye en la confiabilidad del sistema?					

6. Le pedimos su criterio acerca del elemento definido en la etapa de Análisis y Diseño.

<b>Criterio de Especialistas</b>					
Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera que la descripción correcta de los casos, así como seleccionar los estilos y patrones arquitectónicos son elementos primordiales que					

## ANEXOS

---

garantizan el aseguramiento de la confiabilidad durante la etapa de Análisis y Diseño?					
--	--	--	--	--	--

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

Criterio de Especialistas					
Pregunta	C1	C2	C3	C4	C5
	Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
¿Considera usted que tener en cuenta las pautas definidas y el manejo de los errores, las buenas prácticas de programación y el tratamiento de errores y respuesta del sistema a los mismos son elementos para garantizar la confiabilidad en la etapa de Implementación?					

8. Le pedimos su criterio acerca del elemento definido en la etapa de Pruebas.

## ANEXOS

<b>Criterio de Especialistas</b>					
Pregunta	C1	C2	C3	C4	C5
	Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
¿Considera usted que en la etapa de pruebas el comportamiento del sistema y las pruebas de confiabilidad (Pruebas de Estrés, Pruebas Reales, Pruebas de destrucción aleatorias, Pruebas de Integración, Pruebas Estructurales, Pruebas de Recuperación y Tolerancia a fallos) son el elemento que influye en la confiabilidad del sistema?					

9. Le pedimos su criterio acerca del elemento definido en la etapa de Despliegue.

<b>Criterio de Especialistas</b>					
Pregunta	C1	C2	C3	C4	C5
	Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
¿Considera usted					

## ANEXOS

---

que los escenarios de despliegue durante la etapa de Despliegue es un elemento para garantizar la confiabilidad?					
--	--	--	--	--	--

10. Le pedimos su criterio acerca del elemento definido en la etapa de Soporte.

Criterio de Especialistas					
Pregunta	C1	C2	C3	C4	C5
	Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
¿Considera usted que las tener en cuenta los elementos definidos en la etapa de implementación son el elemento que garantiza la confiabilidad del sistema en la etapa de soporte?					

## GLOSARIO DE TÉRMINOS

---

- 1 **CMMI:** Capability Maturity Model Integration.
- 2 **BPMN:** Business Process Modeling Notation.
- 3 **IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos.
- 4 **SQA:** Software Quality Assurance.
- 5 **SQuaRE:** Software Product Quality Requirements and Evaluation.
- 6 **HCI:** Human-Computer Interaction.
- 7 **IPO:** Interacción Persona – Ordenador.
- 8 **ISO:** International Organization for Standardization. Es la Organización Internacional para la
- 9 Normalización; responsable para la normalización a escala mundial.