

**Universidad de las Ciencias Informáticas**

**Facultad 3**



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

**Título:** Guía para asegurar la portabilidad en los sistemas informáticos de Gobierno Electrónico

**Autor:** Edgar Leal Faunde

**Tutor:** Ing. Raúl Velázquez Alvarez

Junio 2012



*Ningún mortal debe hacer ilusión de ser más temido que los dioses.*

*Fidel Castro Ruz*

### **Declaración de autoría**

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

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

### **Autores:**

\_\_\_\_\_

Edgar Leal Faunde

### **Tutor:**

\_\_\_\_\_

Ing. Raúl Velázquez Alvarez

**Datos de contacto**

Raúl Velázquez Álvarez

Ingeniero en Ciencias Informáticas

**Centro de trabajo:** Universidad de las Ciencias Informáticas

**Cargo que desempeña:** Asesor Técnico Docente de Calidad del Centro de Gobierno Electrónico de la facultad 3 y líder del Proyecto Calidad del mismo centro.

**Correo electrónico:** rvelazquez@uci.cu

## **Agradecimientos**

A mi familia y en especial a mis padres por su inmenso amor y cariño, a ellos les debo lo que soy y gracias a ellos he llegado hasta donde estoy.

A mi hermana por ser una amiga en todo momento.

A mis tíos Isnay y Carlos, siempre dispuestos a brindarme su ayuda.

A mi abuela por siempre estar cuidando de mí.

A mi novia Yanetsi que me brindo su mano para cruzar obstáculos que solo no supe rebasar y por quererme tal y como soy.

A mi tutor Raúl por ofrecerme todo su apoyo incondicional durante la realización de este trabajo.

A mis amigos y a mis compañeros de grupo por su apoyo, por cada consejo que me supieron dar en el momento oportuno, en especial a los que están conmigo desde el inicio compartiendo malos y buenos momentos.

### **Dedicatoria**

Le dedico este trabajo a mis padres, pues nunca podré agradecerles todo lo que han hecho por mí, no hay palabras que describan lo que ustedes significan para mí, los quiero mucho.

A mi hermana Yissel, quien siempre estará presente para mí.

A tíos Isnay y Carlos en los que siempre podre confiar.

A mi abuela por quererme tanto.

## **Resumen**

Hoy en día la calidad se hace un factor fundamental en el desarrollo de software ante la necesidad de crear software de calidad para poder satisfacer a los clientes que hacen uso de estos. El aseguramiento de la calidad es un aspecto fundamental en este sentido, pues garantiza que el software cuente con diferentes características de calidad, entre ellas la portabilidad. En el presente trabajo se propone una guía para el aseguramiento de la portabilidad en los sistemas informáticos de Gobierno Electrónico que se desarrollan en la Universidad de las Ciencias Informáticas.

Para la elaboración de la guía se tuvo en cuenta los resultados arrojados en la entrevista aplicada a arquitectos de los proyectos del Centro de Gobierno Electrónico. Como sustento a la guía se realizó un estudio de diversos conceptos de calidad de software y aseguramiento de la calidad, se analizaron modelos y estándares de calidad y se investigó sobre la portabilidad.

La guía propuesta define un conjunto de actividades para diversos elementos que afectan la portabilidad, definidos en las etapas del ciclo de vida de desarrollo del software. La misma fue validada satisfactoriamente por especialistas en el tema.

**Palabras Claves:** aseguramiento, calidad, guía, portabilidad, sistemas informáticos.

**Índice**

Introducción .....	1
Capítulo 1: Fundamentación teórica .....	4
1.1 Introducción.....	4
1.2 Calidad de software.....	4
1.3 Aseguramiento de la calidad de software orientado a la portabilidad del producto .....	5
1.4 Portabilidad del Software .....	6
1.4.1 ¿Qué es la portabilidad? .....	6
1.4.2 Fases de la portabilidad.....	7
1.4.3 Principales tipos de portabilidad .....	7
1.4.4 ¿Por qué se debe tener en cuenta la portabilidad? .....	7
1.4.5 ¿Por qué en ocasiones no se debe hacer el software portable? .....	8
1.5 Modelos y/o estándares internacionales para las características de calidad de software.....	8
1.5.1 Modelo de MCCALL (1977).....	8
1.5.2 Modelo de Boehm (1978).....	11
1.5.3 Modelo de DROMEY (1996) .....	12
1.5.4 ISO 9126. Modelo de Calidad .....	13
1.5.5 Modelo de calidad ISO/IEC 25010 SQuaRE (Software Product Quality Requirements and Evaluation).....	15
1.6 Modelo de Capacidad y Madurez Integrado (CMMI). Modelo de calidad a nivel de proceso ....	18
1.7 Conclusiones parciales .....	20
2.1 Introducción.....	21
2.2 Características de los proyectos del Centro CEGEL.....	21
2.3 Ciclo de vida del proceso de desarrollo de software basado en el programa de mejora. Elementos	

definidos por etapas .....	21
2.3.1    Estudio Preliminar.....	22
2.3.2    Modelación del negocio .....	22
2.3.3    Requisitos.....	23
2.3.4    Análisis y Diseño .....	24
2.3.5    Implementación .....	26
2.3.6    Pruebas Internas .....	26
2.3.7    Pruebas de Liberación .....	26
2.3.8    Despliegue.....	27
2.3.9    Soporte.....	27
2.4    Conclusiones parciales .....	32
3.1    Introducción .....	33
3.2    Conclusiones Parciales.....	48
Conclusiones generales.....	49
Recomendaciones .....	50
Glosario de términos.....	51
Referencias bibliográficas y bibliografías consultadas. ....	52
Anexos.....	54

## Introducción

Hoy en día el mundo está inmerso en una nueva revolución tecnológica basada en la informática, una herramienta importante para el desarrollo de los países. El aprovechamiento de las tecnologías, la informática y las comunicaciones, propicia la mejoría de los niveles de bienestar y permite aumentar la productividad y competitividad de las naciones, además ha contribuido a que culturas y sociedades se transformen aceleradamente, tanto política como social y económicamente, con el objetivo fundamental de alcanzar con plenitud sus potencialidades.

Para lograr una sociedad informatizada se necesita que los productos informáticos empleados para este fin sean seguros, confiables y que tengan calidad. Con el propósito de evitar detectar que una vez terminado un software este no cumpla con los requisitos de calidad necesarios, se debe tener en cuenta, durante todo el ciclo de desarrollo de estos, diferentes características que determinan la calidad del producto final, entre ellas la portabilidad, característica enfocada en la capacidad del software para ser trasladado y adaptado en diferentes ambientes. Se plantea que el software es portable si en la medida en que el costo de cambio de operador es menor que el costo de la reconstrucción. La portabilidad es cada vez más un atributo deseable para los sistemas informáticos, debido a que la mayoría del software diseñado para una amplia utilidad se enfrentará a la necesidad de ser transferido a nuevos ambientes a lo largo de su vida.

Cuba, con el objetivo de informatizar su sociedad y lograr un lugar respetado en el mundo del desarrollo de software, ha realizado grandes inversiones, un ejemplo de esto lo constituye el surgimiento de la Universidad de las Ciencias Informáticas (UCI), en la actualidad uno de los centros vanguardias en la producción de software. En esta institución se han desarrollado varios proyectos para contribuir a la informatización de la sociedad cubana entre los que se encuentra el Centro de Gobierno Electrónico (CEGEL) creado para desarrollar soluciones para entidades gubernamentales y jurídicas, favoreciendo de esta manera la construcción del gobierno electrónico en el país.

A partir de una entrevista realizada a arquitectos de cinco proyectos del centro se detectó que durante el proceso de aseguramiento de la calidad, no se ven reflejados en los planes actividades definidas a asegurar la portabilidad, específicamente en el acápite de uso de estándares que posibiliten el uso de buenas prácticas para contribuir a que los productos desarrollados cuenten con esta característica. Además el 100 % de los proyectos entrevistados planteaba que la

portabilidad se ve afectada por la falta de personal especializado en el tema.

Teniendo en cuenta lo expuesto anteriormente se plantea el siguiente **problema a resolver**: el proceso de aseguramiento de la calidad que se lleva a cabo en los proyectos del Centro de Gobierno Electrónico no contribuye a asegurar la portabilidad en el producto durante todo el ciclo de desarrollo.

Donde se tiene como **objeto de estudio** el aseguramiento de la portabilidad en los sistemas informáticos de Gobierno Electrónico.

Para dar solución al problema planteado se define como **objetivo general**: proponer una guía el aseguramiento de la portabilidad en los sistemas informáticos de Gobierno Electrónico.

Donde se tiene como **campo de acción** la portabilidad en los sistemas informáticos de Gobierno Electrónico.

Esta investigación plantea como **idea a defender**: con la propuesta de una guía que permita el aseguramiento de la portabilidad durante todo el ciclo de desarrollo en los sistemas informáticos de Gobierno Electrónico se contribuirá a que los mismos cuenten con esta característica.

Los **objetivos específicos** para dar cumplimiento al objetivo general son:

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

### **Tareas 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 las sub-características existentes en dependencia del estudio de los estándares y modelos definidos internacionalmente que traten el tema de las características de calidad de software.
- Definición de los atributos necesarios durante el ciclo de desarrollo, basado en el programa de mejora de la UCI que plantea CMMI, para que las aplicaciones de Gobierno Electrónico cumplan con la característica de Portabilidad.

- Definición de acciones o actividades a realizar a lo largo del ciclo de desarrollo de software, para asegurar la portabilidad en el producto final, basándose en los elementos anteriormente definidos.
- Aplicación del método Delphi utilizando el criterio de especialistas para comprobar la validez de la propuesta de solución.

### **Métodos de Investigación:**

#### **Métodos Teóricos**

##### Analítico-Sintético:

Posibilita realizar el análisis de trabajos y documentos facilitando la obtención de conocimientos acerca de la calidad del software y la portabilidad en los sistemas informáticos. Se sintetizan los elementos más importantes y de mayor utilidad para el desarrollo de la investigación.

##### Histórico-Lógico:

Permite constatar teóricamente la evolución de la portabilidad en los sistemas informáticos, desde su surgimiento hasta la actualidad, así como la necesidad que existe de aplicar la portabilidad en los sistemas informáticos.

#### **Métodos Empíricos**

##### Entrevista:

Se utiliza con el objetivo de obtener información acerca de cómo se trabaja la portabilidad en el desarrollo de software.

##### Encuesta:

Se utiliza para determinar la eficacia de la propuesta de guía.

### **Posibles resultados:**

- Una guía con los elementos que son necesarios tener en cuenta durante todo el ciclo de desarrollo de software para que el producto cumpla con la característica portabilidad, así como un conjunto de actividades por cada etapa encaminadas a asegurar que el producto cuente con esta característica.

## Capítulo 1: Fundamentación teórica

### 1.1 Introducción

En este capítulo quedan plasmados diversos conceptos de Calidad de Software y Aseguramiento de la Calidad. Se tratan diversos aspectos referentes a la portabilidad para tener un mayor entendimiento de esta. Se realiza un estudio de una variedad de modelos y estándares de calidad existentes que aportan gran cantidad de datos en cuanto a calidad y sus características se refiere, por lo que son fundamentales para definir aspectos importantes y necesarios que permitan un mayor entendimiento para la contribución en el aumento de la portabilidad en las soluciones informáticas.

### 1.2 Calidad de software

En la actualidad la calidad está de moda en todos los aspectos, pero especialmente en el desarrollo de software. La creciente complejidad en el desarrollo de software ha planteado la necesidad, cada vez mayor, de obtener técnicas y métodos para mejorar la calidad del software que se construye. El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más estrictos a la hora de seleccionar un software, rechazando los productos con baja calidad o que realmente no dan respuesta a sus necesidades. Para comprender mejor en qué consiste la calidad de software se tienen en cuenta diferentes definiciones planteadas por disímiles autores.

La IEEE define la calidad de software como: *“Calidad de software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”*. (IEEE, 1990)

Según la norma ISO/IEC 9126 (International Standart Organization u Organización Internacional de Estándares en español) define a la calidad de software *“como la totalidad de rasgos y atributos de un producto de software que le apoyan en su capacidad de satisfacer sus necesidades explícitas o implícitas”*. (ISO/IEC 9126, 1998)

La Calidad de Software para Pressman es *“la concordancia con los requisitos funcionales y de rendimiento establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado de forma profesional”*.

(Pressman, 2002)

La Calidad del Software es *“el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia, la cual plantea un adecuado balanceo de eficiencia, confiabilidad, facilidad de mantenimiento, portabilidad, facilidad de uso, seguridad e integridad”*. (SCALONE, 2006)

Después de realizar un estudio de los diversos conceptos de calidad de software planteados, esta se puede definir como el cumplimiento de las expectativas del cliente, con productos que cumplan con las normas establecidas y cuenten con un balanceo adecuado de las características (mantenibilidad, portabilidad, eficiencia, usabilidad, fiabilidad, interoperabilidad, seguridad, funcionalidad) con que todo software debe contar.

Para lograr que estas características se pongan de manifiesto en el producto final es necesario tener en cuenta desde etapas tempranas del desarrollo un conjunto de acciones que contribuyan al aseguramiento de las mismas, en este caso, la portabilidad.

### **1.3 Aseguramiento de la calidad de software orientado a la portabilidad del producto**

El aseguramiento de la calidad del software es uno de los temas más actuales e importantes dentro de la disciplina de la Ingeniería de Software. Esta se identifica con tener todos los procesos bien definidos para asegurar al cliente que puede confiar en que los productos y servicios cumplirán fielmente con sus requisitos o especificaciones.

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

Según Roger Pressman el aseguramiento de la calidad del software *“es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza que el software satisfará los requisitos dados de calidad”* (Pressman, 2002).

Como bien lo mencionan Pressman y la norma ISO, el aseguramiento de la calidad de software tiene la finalidad de satisfacer los requisitos de calidad en un producto, que vienen manifestados parcialmente a través de las características de calidad. Como se había mencionado anteriormente esta investigación se centra en el aseguramiento de la portabilidad como característica de calidad.

Por tanto, para el autor del presente trabajo, el aseguramiento de la portabilidad es *“un diseño*

*planificado de acciones o actividades a realizar a lo largo del ciclo de vida de software que se requieren para garantizar la portabilidad en el producto final”.*

### 1.4 Portabilidad del Software

La mayoría de los desarrolladores de software coinciden en que la portabilidad es cada vez más un atributo deseable para los proyectos de software. La vida útil de una aplicación, por ejemplo, es probable que se extienda, y su base de usuarios aumente, si se pueden migrar a varias plataformas durante su vida.

La continua proliferación del software y arquitecturas de hardware asegura que la mayoría de los paquetes de software con el tiempo se enfrentará a la necesidad de ser portado a fin de mantener y ampliar su viabilidad. Esta necesidad puede ser lo más eficazmente posible si la portabilidad se ha tenido en cuenta durante el proceso de desarrollo.

En la Universidad se han realizado cuatro investigaciones referentes a esta característica pero de manera general no se hace un análisis durante todo el ciclo de desarrollo del programa de mejora de los elementos importantes que afectan la misma. En dos de los casos se han centrado a evaluar la portabilidad, una referente a los datos en el desarrollo del software y la otra específicamente en el código fuente de las aplicaciones web. De las restantes, una se centra en una estrategia para despliegue en aplicaciones web de tecnología de Microsoft en plataforma Linux, donde se analizan sólo elementos técnicos de ambas tecnologías y la otra está referida específicamente a la obtención de dos subcaracterísticas en las soluciones informáticas de apoyo a la toma de decisiones que utilizan almacenes de datos basadas en PostgreSQL. Esta última plantea elementos importantes que se tienen en cuenta en la presente investigación y contribuye en cierta medida a la definición de la solución.

#### 1.4.1 ¿Qué es la portabilidad?

*“La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos de hardware o de software”.* (ISO/IEC 9126, 1998)

*“Una unidad de software es portable a través de una clase de entornos en la medida en que el costo de transportar y adaptar a un nuevo entorno es menor que el costo de la reconstrucción”.* (Mooney, 1990)

*“Es la habilidad del sistema para ser ejecutado en diferentes ambientes de computación. Estos*

---

# Capítulo 1: Fundamentación teórica

*ambientes pueden ser hardware, software o una combinación de los dos". (Rick Kazman, 2001)*

## **Conceptos claves en la portabilidad:**

Unidad de software: término para indicar una aplicación de un programa o un componente de un programa.

Entorno: este término se refiere a la colección completa de elementos externos con los que una unidad de software interactúa. Estos pueden incluir otro software, sistemas operativos, hardware, sistemas remotos, documentos y personas.

Clase de entornos: se utiliza este término para enfatizar que se busca portabilidad no sólo en un conjunto de entornos específicos, sino en todos los entornos, incluso en aquellos que aún no se han desarrollado.

### **1.4.2 Fases de la portabilidad**

Se distinguen dos fases principales en la portabilidad: el transporte y la adaptación. El transporte se refiere al movimiento físico del software y los artefactos asociados. La adaptación incluye la mayoría de las modificaciones que hay que hacer con el software original.

### **1.4.3 Principales tipos de portabilidad**

Portabilidad Binaria (Portar en formato ejecutable): ofrece varias ventajas pero solo es posible entre entornos muy similares.

Portabilidad Fuente (Portar código fuente): da la posibilidad de adaptar una unidad de software a una amplia variedad de entornos.

### **1.4.4 ¿Por qué se debe tener en cuenta la portabilidad?**

- Existe una gran diversidad de hardware y de plataformas de software.
- Cuando los usuarios cambian a nuevos entornos, desean tener un software familiar.
- Facilita la migración a nuevas versiones de los sistemas y a nuevos entornos.
- Más usuarios pueden usar el mismo producto.

Pero la razón más importante para desarrollar el software de forma que se pueda implementar fácilmente en otros equipos es que, cada 4 años o menos, las organizaciones clientes compran un

nuevo hardware, y todo su software a continuación, debe ser adaptado para funcionar con el nuevo hardware.

### 1.4.5 ¿Por qué en ocasiones no se debe hacer el software portable?

- Un pequeño costo extra o demora en conseguir el producto no se considera aceptable.
- Una pequeña reducción en el rendimiento o la eficiencia del producto puede ser no aceptada.
- A veces una unidad de software está estrechamente ligada a un organismo que es poco probable trasladarlo a un nuevo entorno.

Debido a la importancia que tiene esta característica en los sistemas informáticos, la misma se ha venido tomando en cuenta en el transcurso de los años en modelos y estándares de calidad que se han desarrollado a nivel mundial para evaluar los productos de software.

## 1.5 Modelos y/o estándares internacionales para las características de calidad de software

Hoy en día todas las empresas productoras de software se ven en la necesidad de tener en cuenta los estándares y modelos de calidad durante el proceso de desarrollo de sus productos con el fin de que estos cuenten con la calidad necesaria, debido a que los estándares y modelos de calidad definen un conjunto de criterios a tener en cuenta durante el desarrollo de los productos. Si no se tiene en cuenta ningún estándar o modelo siempre habrá falta de calidad en el software.

Stufflebeam considera los estándares como “*principios ampliamente compartidos para determinar el valor o calidad de un servicio profesional*” (Stufflebeam, 1997).

Estándar de calidad es el que reúne los requisitos mínimos en busca de la excelencia dentro de una organización institucional (Pulido, 2004).

Un modelo de calidad tiene en cuenta criterios para satisfacer las necesidades de los desarrolladores, mantenedores, adquirentes y usuarios finales (ISO/IEC 9126, 2001).

### 1.5.1 Modelo de MCCALL (1977)

En este modelo la calidad se describe como un concepto elaborado mediante relaciones

---

## Capítulo 1: Fundamentación teórica

jerárquicas entre factores de calidad, en base a criterios. Estos factores se organizan en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto.

1-Operación del producto.

2-Revisión del producto.

3-Transición del producto.

Son 11 factores de calidad, que se organizan entorno a los tres ejes de la siguiente forma:

### **Operación del producto:**

- Facilidad de uso (¿Puedo ejecutarlo?)
- Integridad (¿Es seguro?)
- Corrección (¿Hace el software lo que yo quiero?)
- Fiabilidad (¿Lo hace de forma exacta todo el tiempo?)
- Eficiencia (¿Se ejecutará sobre mi hardware lo mejor posible?)

### **Revisión del producto**

- Facilidad de mantenimiento (¿Puedo arreglarlo?)
- Facilidad de prueba (¿Puedo probarlo?)
- Flexibilidad (¿Puedo modificarlo?)

### **Transición del producto**

- Facilidad de reutilización (¿Podré reutilizar parte del software?)
- Interoperabilidad (¿Podré comunicarlo con otros sistemas?)
- Portabilidad (¿Podré utilizarlo en otra máquina?)

En este modelo la portabilidad es tratada dentro de la capacidad de transición del producto. La misma es determinada por una serie de criterios (Jim A. McCall, 1977):

- ✓ Auto-descriptividad: atributos del software que proporcionan una explicación de la aplicación de una función.
- ✓ Modularidad: propiedades del software que proporcionan una estructura de módulos

altamente independientes.

- ✓ Independencia del software: atributos del software que determinan su dependencia en el entorno de software (sistemas operativos).
- ✓ Independencia del hardware: esos atributos del software que determinan su dependencia en el hardware del sistema.

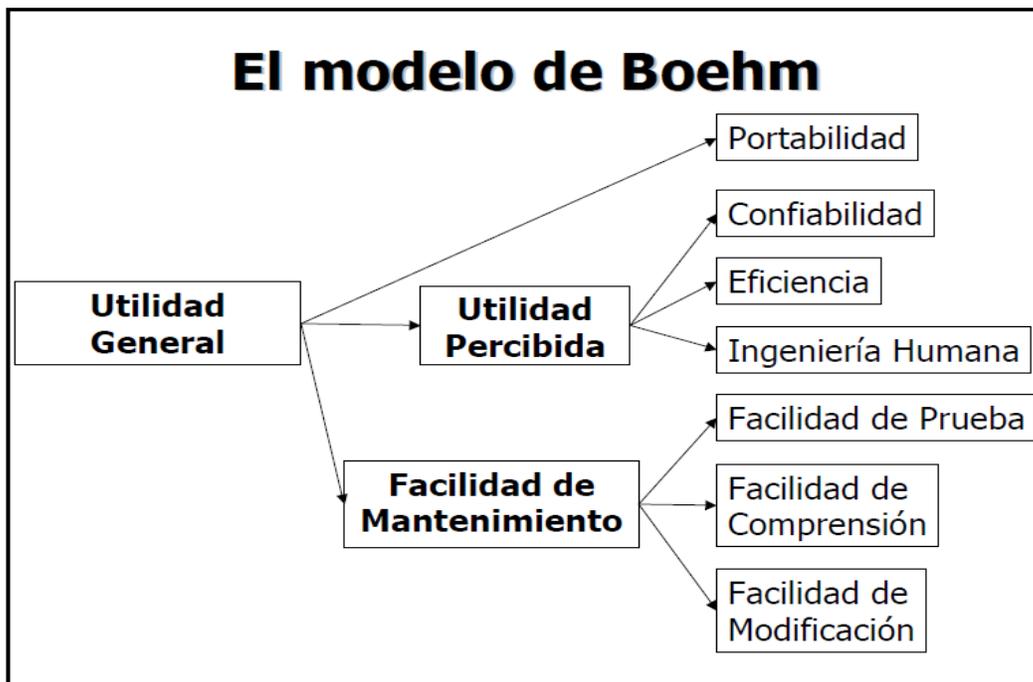


**Figura 1. Modelo de McCall**

## 1.5.2 Modelo de Boehm (1978)

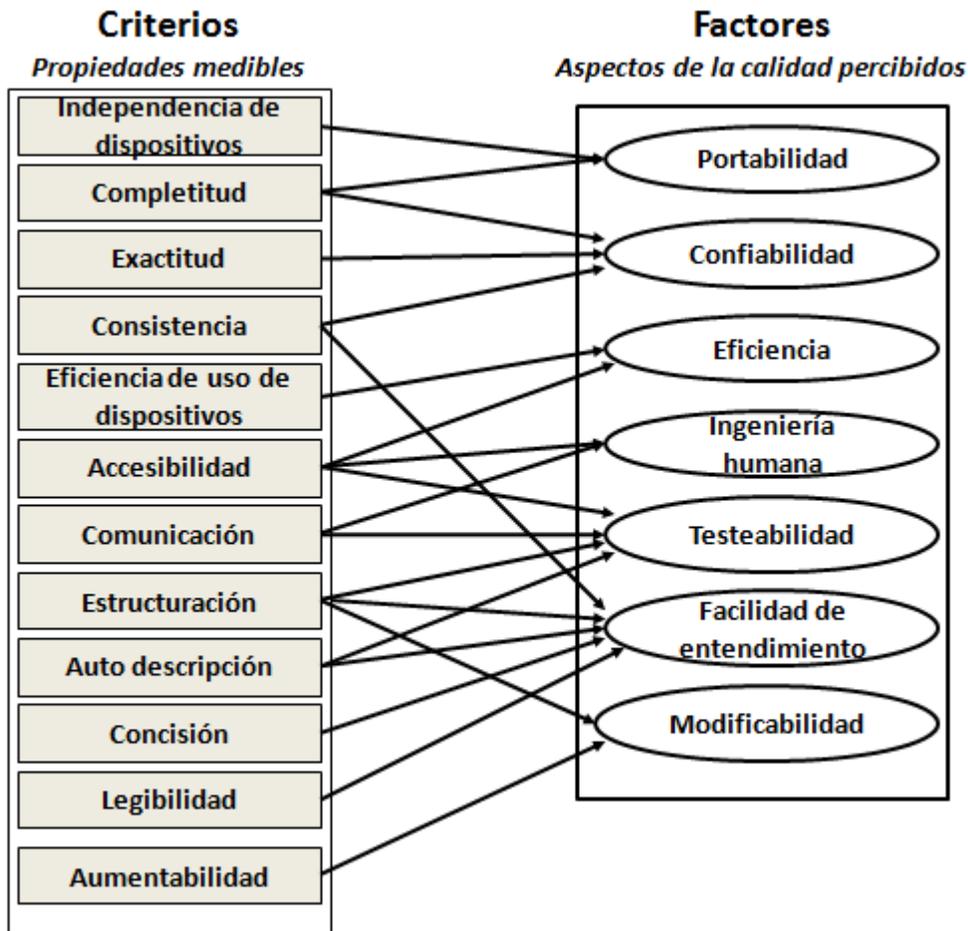
El modelo de Boehm (1978) agrega algunas características a las existentes en el modelo de McCall y representa una estructura jerárquica de características, cada una contribuye a la calidad total. Consiste en un modelo de descomposición de características de calidad de software en 3 niveles (usos principales, componentes intermedios y componentes primitivos) previos a la aplicación de métricas. Este modelo plantea factores de calidad formados por criterios de calidad y métricas respectivas.

Los componentes o constructores del modelo se centran en el producto final. Se identifican características de calidad desde el punto de vista del usuario. El software debe hacer lo que el usuario quiere que haga, utilizar recursos de la computadora correcta y eficientemente, fácil de aprender y usar para los usuarios y debe estar bien diseñado, bien codificado y ser probado y mantenido fácilmente. (SCALONE, 2006)



**Figura 2. Modelo de Boehm**

Factores tales como portabilidad, confiabilidad, facilidad de mantenimiento y facilidad de modificación, son descompuestos en varios criterios.



**Figura 3. Criterios y Factores de Calidad según Boehm**

El modelo trata la portabilidad como un factor a tener en cuenta dentro de la utilidad general del software. Esta es dividida en 2 criterios:

- ✓ Independencia de dispositivos: independencia del hardware.
- ✓ Completitud: independencia de los elementos del software.

### 1.5.3 Modelo de DROMEY (1996)

El modelo de Dromey tiene el propósito de trabajar con una estructura que permite construir y utilizar un modelo de calidad práctico para evaluar las etapas de determinación de los requerimientos, diseño e implementación. Esta información puede ser usada para elaborar, comparar y evaluar la calidad de los productos de software. Este modelo plantea la calidad del

---

## Capítulo 1: Fundamentación teórica

producto por medio de la definición de subcaracterísticas que pueden ser medidas y evaluadas como características. También, permite aumentar el entendimiento respecto de la relación entre las características y las subcaracterísticas de calidad.

Las características de calidad planteadas en este modelo son: Eficiencia, Confiabilidad, Facilidad de mantenimiento, Portabilidad, Facilidad de uso y Funcionalidad.

Factor	Criterio
Correctitud	Funcionalidad Confiabilidad
Internas	Mantenibilidad Eficiencia Confiabilidad
Contextuales	Mantenibilidad Reusabilidad Portabilidad Confiabilidad
Descriptivas	Mantenibilidad Reusabilidad Portabilidad Usabilidad

**Figura 4. Modelo de Dromey**

En el modelo de calidad de Dromey la portabilidad no se tiene en cuenta como una característica. Esta es enfocada como un criterio a tener en cuenta dentro de dos de los factores propuestos en el modelo (Contextuales y Descriptivas).

### 1.5.4 ISO 9126. Modelo de Calidad

ISO/IEC 9126 es un estándar internacional que permite evaluar la calidad del producto software. Este estándar actualmente se encuentra siendo guiado y supervisado por el ISO/IEC 25000:2005 y una serie de estándares agrupados dentro del proyecto SQuaRE (Software product Quality Requirement and Evaluation).

El mismo permite especificar y evaluar la calidad del software desde distintas perspectivas, las cuales están asociadas a la adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoría del software. Puede ser usada por

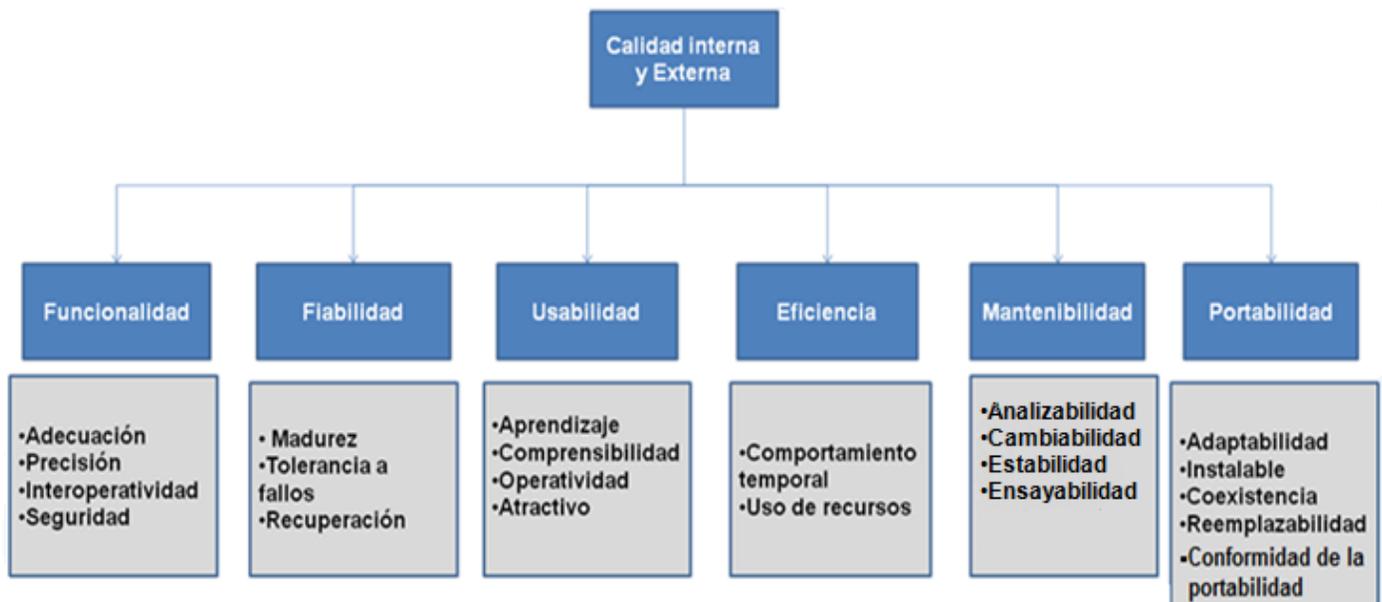
## Capítulo 1: Fundamentación teórica

desarrolladores, evaluadores independientes y grupos de aseguramiento de la calidad, responsables de especificar y evaluar la calidad del software.

El estándar está conformado de 4 partes: un modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad ISO 9126-1 es un modelo jerárquico, la primera parte del modelo especifica 6 características de calidad interna y externa, las cuales están divididas en subcaracterísticas, son manifestadas externamente cuando el software es utilizado como parte de un sistema, y son un resultado de atributos internos del software.

La calidad externa evalúa que el software satisfaga las necesidades del usuario, esta calidad es medible en el comportamiento del producto. La calidad interna evalúa el total de atributos que un software debe satisfacer, esta calidad es medible a partir de las características intrínsecas. Las características definidas son aplicables a todo tipo de software. Las características y subcaracterísticas proveen una terminología consistente respecto de la calidad del producto del software.



**Figura 5. Modelo de Calidad Interna y Externa de ISO/IEC 9126-1**

En este modelo a la portabilidad se le atribuye una gran importancia y es tratada como una

característica de calidad que influye de forma directa en la calidad del producto. Se definen para esta 5 subcaracterísticas:

**Adaptabilidad:** capacidad del producto software para ser adaptado en ambientes determinados.

**Facilidad de instalación:** la capacidad del producto software para ser instalado en un ambiente determinado.

**Coexistencia:** la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.

**Conformidad de la portabilidad:** la capacidad del producto de software para adaptarse a estándares relacionados con la portabilidad.

**Reemplazabilidad:** capacidad del producto de software de ser usado en lugar de otro producto de software especificado para los mismos fines y en el mismo ambiente.

### 1.5.5 Modelo de calidad ISO/IEC 25010 SQuaRE (Software Product Quality Requirements and Evaluation)

Este modelo es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598, que surge con el fin de eliminar inconsistencias que existían entre ambos modelos producto a la utilización de ciclos de vida separados para la especificación de requisitos y la evaluación del producto. SQuaRE (norma ISO/IEC 25000) contiene un modelo de calidad de software genérico que puede ser aplicado a cualquier producto de software. Su objetivo es ayudar a aquellos que se encuentran en el proceso de desarrollo y selección de productos de software, estableciendo los criterios para la especificación de software, requisitos de calidad de los productos, su medición, y evaluación. Este modelo conserva 6 características de calidad de software existentes en ISO/IEC 9126-1:2001 y le agrega 2 nuevas. SQuaRE consiste en documentos agrupados en 5 divisiones (familias de estándares), estos documentos conforman la arquitectura de SQuaRE.

<b>Requisitos de Calidad 2503n</b>	<b>Modelo de Calidad 2501n</b>	<b>Evaluación de Calidad 2504n</b>
	<b>Gestión de Calidad 2500n</b>	
	<b>Medición de Calidad 2502n</b>	

**Figura 6. Arquitectura de SQuaRE**

# Capítulo 1: Fundamentación teórica

La división 2501n presenta un detallado modelo incluyendo características de calidad interna, externa y calidad en uso.

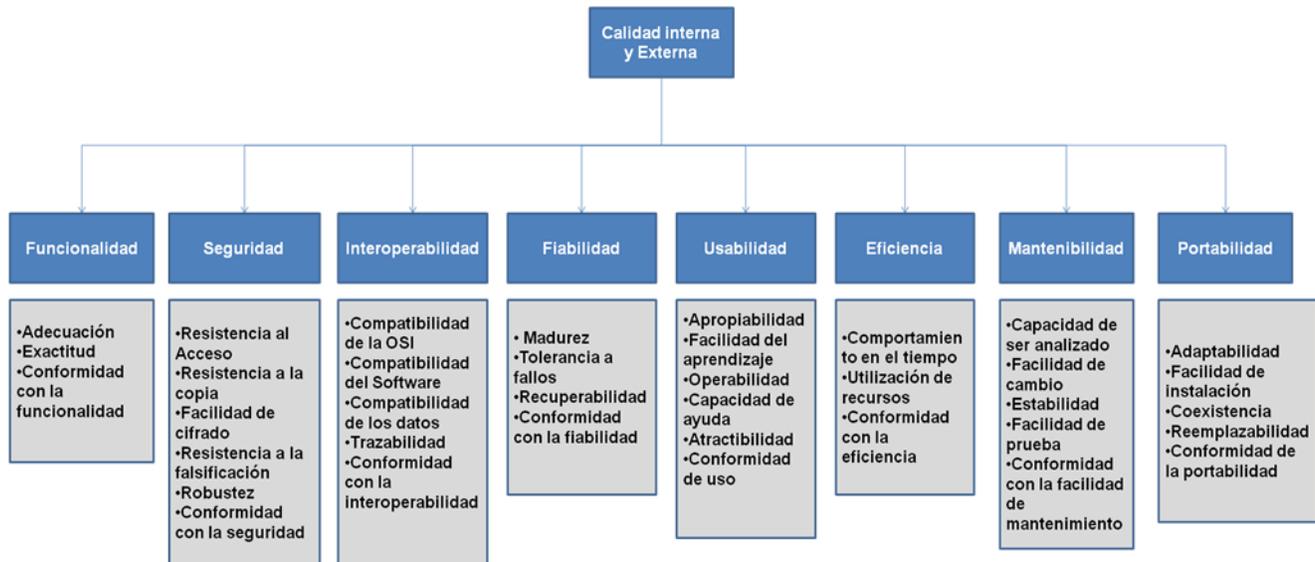


Figura 7. Modelo de calidad para la calidad externa e interna

**Descripción de las características de calidad presentes en el Modelo de calidad ISO/IEC 25010 SQuaRE.**

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

---

## Capítulo 1: Fundamentación teórica

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

### **Subcaracterísticas de la portabilidad presentes en este modelo.**

**Adaptabilidad:** la capacidad del producto software para ser adaptado en ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.

**Facilidad de instalación:** la capacidad del producto software para ser instalado en un ambiente determinado.

**Coexistencia:** la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.

**Reemplazabilidad:** la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.

**Conformidad:** la capacidad del producto de software para adaptarse a estándares relacionados con la portabilidad.

Después de realizar un estudio de los modelos de calidad propuestos en la investigación, que tratan la portabilidad, se puede plantear que el modelo de McCall y el de Boehm son modelos antiguos que enfocan la portabilidad a las necesidades del momento en que fueron propuestos. El modelo de Dromey no trata la portabilidad como una característica directamente sino como un criterio a tener en cuenta dentro de dos de los factores propuestos por este. La portabilidad en el modelo de calidad de la ISO\IEC 9126 es tratada como una característica de calidad y es determinada por un conjunto de cinco subcaracterísticas. El modelo de calidad SQuaRE surge a partir de la ISO\IEC 9126, trata la portabilidad como una característica que influencia de manera

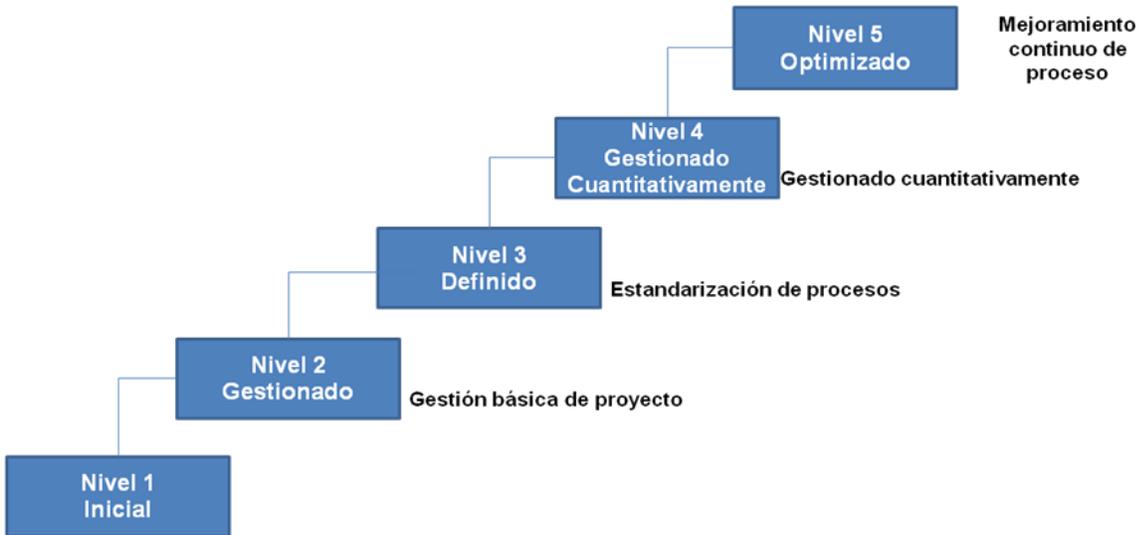
directa en la calidad del producto, es el modelo más novedoso y mejor estructurado por lo que se selecciona para esta investigación.

Para asegurar la calidad del producto y que en este se manifiesten las características anteriormente mencionadas es necesario asegurar y mejorar la calidad del proceso.

### **1.6 Modelo de Capacidad y Madurez Integrado (CMMI). Modelo de calidad a nivel de proceso**

CMMI (Capability Maturity Model Integration) es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios. Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento. El propósito de CMMI para desarrollo es ayudar a las organizaciones a mejorar sus procesos de desarrollo y de mantenimiento, tanto para los productos como para los servicios. (Mary Beth Chrissis, 2009)

Este modelo establece cinco niveles de madurez para clasificar a las organizaciones, en función de qué áreas de procesos consiguen sus objetivos y se gestionan con principios de ingeniería. Para que una organización se encuentre en un determinado nivel es necesario cumplir con todas las actividades definidas para ese nivel y para los niveles anteriores. Los mismos describen un camino evolutivo recomendado para una organización que quiera mejorar los procesos que utiliza para desarrollar y mantener sus productos y servicios, caracterizan a la mejora desde un estado mal definido hasta un estado que utiliza información cuantitativa para determinar y gestionar las mejoras que se necesitan para satisfacer los objetivos de negocio de una organización.



**Figura 8. Representación escalonada**

Los proyectos de la UCI se encuentran certificados en el nivel 2 de CMMI, a partir de la puesta en práctica del programa de mejora basado en este modelo, diseñado en la Universidad. Lo que se pretende con este nivel es conseguir que en los proyectos de la organización haya una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados.

Áreas de proceso del Nivel 2 de CMMI

- 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).
- Medición y Análisis (MA).

Con la aplicación del nivel 2 de CMMI la universidad ha obtenido los siguientes beneficios:

- Calendarios y presupuestos predecibles en los proyectos.
- Mejora del ciclo de vida dentro del desarrollo de software.

- Mayor productividad.
- Mayor calidad de los productos y servicios que ofrece la Universidad a sus clientes y por ende la satisfacción de los mismos.
- Mejorar la moral del personal que labora en el centro.

Esta investigación se rige por el ciclo de vida que se definió en el programa de mejora de la Universidad para la propuesta de solución.

### **1.7 Conclusiones parciales**

En el presente capítulo se realizó un análisis de diferentes conceptos de calidad del software y aseguramiento de la calidad que permitieron crear conceptos propios referentes a calidad de software y aseguramiento de la portabilidad. Se realizó un estudio de los modelos y/o estándares internacionales de calidad que tienen en cuenta la portabilidad y se escogió el modelo SQuaRE para la realización de este trabajo. Se tiene en cuenta además el modelo CMMI ya que la Universidad lo ha tomado para la mejora de los procesos en el desarrollo de software definiendo un programa de mejora en el cual se plantea el ciclo de vida por el cual se rige la investigación.

### **Capítulo 2: Propuesta de solución**

#### **2.1 Introducción**

En este capítulo se presenta la guía para asegurar la portabilidad en los sistemas informáticos de gobierno electrónico, basado en el ciclo de vida de los proyectos definidos en el nivel 2 de CMMI como parte de la institucionalización que lleva a cabo la universidad. La carencia de una guía para asegurar la portabilidad en el desarrollo de software implicó la definición de esta propuesta de solución.

#### **2.2 Características de los proyectos del Centro CEGEL**

Actualmente el Centro cuenta con dos departamentos, el de Informática Jurídica y el de Gestión Gubernamental. Los proyectos del centro desarrollan sus soluciones siguiendo el ciclo de vida definido por el programa de mejora que se lleva a cabo en la Universidad y se definió de manera general por cada etapa elementos a seguir para asegurar la portabilidad, se debe tener en cuenta que la definición de portabilidad de cada proyecto es propia y que los proyectos definen internamente qué nivel de importancia le dan a la característica en dependencia de las características que posea la aplicación. Los sistemas informáticos de gobierno electrónico tienen una larga vida útil por lo que se pueden ver sometidos a cambios de entornos durante esta, es por eso que deben ser desarrollados teniendo en cuenta la portabilidad.

#### **2.3 Ciclo de vida del proceso de desarrollo de software basado en el programa de mejora. Elementos definidos por etapas.**

Como parte de la institucionalización que se lleva a cabo en la Universidad es necesario realizar un estudio del ciclo de vida del proceso de desarrollo de software y utilizarlo como punto de partida para definir por etapa del mismo, las actividades que se deben tener en cuenta para el aseguramiento de la portabilidad. (Calisoft, 2009)

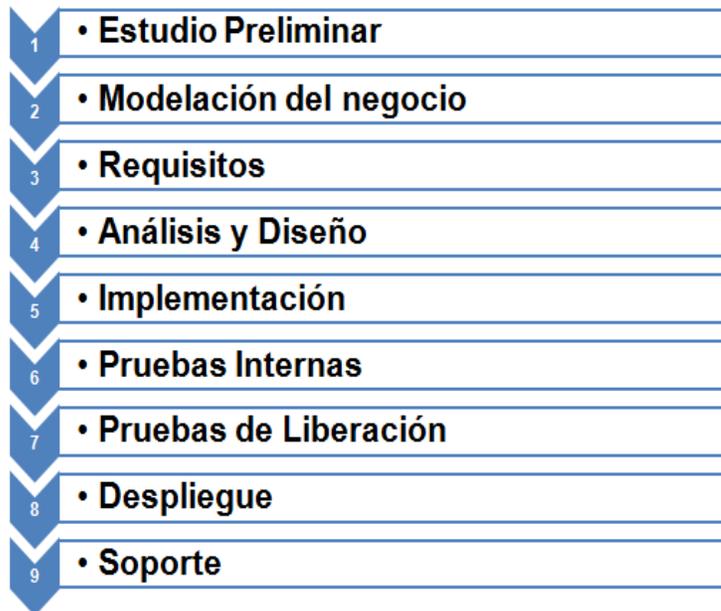


Figura 9. Ciclo de Vida

### 2.3.1 Estudio Preliminar

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

En esta etapa no se define ningún elemento a tener en cuenta para asegurar la portabilidad.

### 2.3.2 Modelación del negocio

Etapa 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 y como lenguaje de modelado se utiliza Business Process Modeling Notation (BPMN).

Elemento crítico de la etapa:

Reglas del negocio: para identificar de forma correcta los requisitos del negocio primero se debe comprender cómo funciona, es decir cuáles son las reglas del negocio.

### 2.3.3 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 la etapa:

- Requisitos de Portabilidad: definen qué características deberá tener el software para que sea fácil utilizarlo en otra máquina o bajo otro sistema operativo.
- Plan de gestión de requisitos: especifica la información que se debe recopilar y los mecanismos de control que deben utilizarse para medir, informar y controlar los cambios de los requisitos del producto.
- Requisitos de Actualización de Software: están relacionados a la capacidad del software de poder ser actualizado. Una actualización es una revisión o reemplazo completo del software. Con las actualizaciones se pueden realizar las siguientes funciones:
  - Mejorar la seguridad.
  - Aumentar la compatibilidad del software.
  - Optimizar la forma en que el sistema maneja los recursos de la máquina.
  - Añadir funcionalidades.
  - Quitar funciones y herramientas anticuadas.

Por lo tanto las actualizaciones mejoran entre otras cosas la reemplazabilidad y coexistencia del software.

### 2.3.4 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 la etapa:

Patrones: los patrones son un amplio repertorio de principios generales basados en la experiencia que guían la creación de un software. En el desarrollo de un software en ocasiones se emplean patrones sin la menor idea de lo que estos propician, cada patrón o estilo favorece algún elemento en particular. Se deben usar patrones para lograr bajo acoplamiento, alta cohesión y modularidad. Tanto el bajo acoplamiento, como la alta cohesión y la modularidad desempeñan un papel fundamental en la portabilidad, pues son de gran importancia para el mantenimiento, control, corrección de errores y modificación para futuras versiones del software. Cuando existe un bajo acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.

Patrones recomendados:

- Patrones Arquitectónicos:

Layers: Consiste en estructurar aplicaciones que pueden ser descompuestas en grupos de subtareas, las cuales se clasifican de acuerdo a un nivel particular de abstracción.

Broker: Puede ser usado para estructurar sistemas de software distribuido con componentes desacoplados que interactúan por invocaciones a servicios remotos. Un componente broker es responsable de coordinar la comunicación, como el reenvío de solicitudes, así como también la transmisión de resultados y excepciones.

---

## Capítulo 2: Propuesta de solución

Microkernel: Se aplica para sistemas de software que deben estar en capacidad de adaptar los requerimientos de cambio del sistema. Separa un núcleo funcional mínimo del resto de la funcionalidad y de partes específicas pertenecientes al cliente.

- Patrones de Diseño:

Patrones GRAPS (General Responsibility Assignment Software Patterns):

- ✓ Patrón Bajo Acoplamiento: Propone que el nivel de acoplamiento entre los objetos sea bajo, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas.
- ✓ Patrón Alta Cohesión: Asigna una responsabilidad de manera que la cohesión permanezca alta. Una clase con alta cohesión tiene un número relativamente pequeños de métodos, con funcionalidad altamente relacionada. Este tipo de clases, por su pequeño número de métodos, resultan fáciles de mantener, de entender y de reutilizar.

Diseño de Interfaces: es uno de los elementos clave en la realización del programa. Si la forma en que las interfaces se presentan varía en los diferentes ambientes de destino, surge un problema importante, la portabilidad.

Estilos arquitectónicos: sirven para sintetizar la estructura de las aplicaciones y definen los posibles patrones para estas aplicaciones. Además permiten evaluar arquitecturas alternativas con ventajas y desventajas conocidas ante diferentes conjuntos de requisitos no funcionales, como la portabilidad.

Lenguaje de Programación: este factor se refiere primeramente a una correcta selección del lenguaje de programación para después hacer un buen uso de este siguiendo las normas de portabilidad asociadas al mismo, logrando con esto llevar la portabilidad al producto final. Si el código cuenta con gran portabilidad entre plataformas y arquitecturas, el programa creado con este tendrá una mayor portabilidad. Es decir si un programa es escrito con un lenguaje de programación para el cual existen pocos compiladores, es muy probable que el código de este tenga que volverse a escribir en otro lenguaje a la hora de ser portado.

Los elementos planteados en esta etapa afectan la reemplazabilidad y adaptabilidad del producto.

### 2.3.5 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 la etapa:

Estándares de programación: conjunto de reglas a tener en cuenta en la implementación de un producto software. El uso de estos permite que cualquier programador entienda y pueda mantener la aplicación, además mejoran la legibilidad y reutilización del código.

Complejidad de los algoritmos: se deben usar buenas prácticas de programación y seguir los aspectos definidos por el arquitecto, evitar métodos de mucha complejidad, con accesos a datos y demasiados ciclos anidados, de manera que facilite una mayor reutilización del código.

Los elementos definidos en esta etapa afectan la reemplazabilidad y adaptabilidad del sistema.

### 2.3.6 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 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:

Pruebas de portabilidad: incluyen la posibilidad de instalar el sistema en otro entorno de hardware y software, estas pruebas tienen por objeto comprobar que la aplicación podrá ejecutarse sin problemas en soportes diferentes al empleado para su desarrollo.

### 2.3.7 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.

En esta etapa de pruebas no se define ningún elemento crítico ya que son realizadas por un ente externo al proyecto y no se tiene ningún control sobre las mismas.

### 2.3.8 Despliegue

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

#### Elemento crítico de la etapa:

Escenarios de despliegue: son aquellos donde se va a utilizar el software en cuestión y deben estar en correspondencia con los que se definió en los requisitos no funcionales de hardware.

Documentación de portabilidad: se debe documentar todo lo relacionado al proceso de portar el software desarrollado.

### 2.3.9 Soporte

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

Consiste en dar mantenimiento al sistema una vez que se haya implantado. Esta etapa incluye mantenimiento y mejoras que se pueden realizar en versiones anteriores.

#### Elemento crítico de la etapa:

Modificaciones al software: se refieren al tamaño y la complejidad de las mismas, deben estar en el rango de los acuerdos con los clientes, nunca más allá ya que pueden afectar el código que no está implicado en estas modificaciones.

## Capítulo 2: Propuesta de solución

A continuación se muestra una figura que recoge los elementos que se deben tener en cuenta durante todo el ciclo de desarrollo de software, en este caso solo las etapas en las que se puede influir en la portabilidad del producto.



Figura 10. Elementos por etapa

La definición de los elementos tiene dentro un conjunto de criterios que pueden determinar la portabilidad del sistema y para que esta no se vea afectada se proponen un grupo de actividades que contribuyen al aseguramiento de esta característica.

Etapas	Actividades
--------	-------------

---

## Capítulo 2: Propuesta de solución

<b>Modelación del negocio</b>	<p>Reglas del negocio</p> <ul style="list-style-type: none"><li>➤ Definir con claridad y precisión las reglas del negocio, evitando ambigüedades que pudieran afectar el desarrollo del sistema.</li></ul>
<b>Requisitos</b>	<p>Requisitos de Portabilidad</p> <ul style="list-style-type: none"><li>➤ Documentar de manera clara y precisa lo referente a la portabilidad, para asegurar correctamente su desarrollo en etapas posteriores.</li><li>➤ Establecer la portabilidad como un objetivo y determinar qué importancia se le debe atribuir a esta.</li></ul> <p>Plan de Gestión de Requisitos</p> <ul style="list-style-type: none"><li>➤ Analizar rigurosamente la información y los mecanismos de control que se utilizarán para medir, informar y controlar los cambios efectuados en los requisitos.</li></ul> <p>Requisitos de Actualización de Software</p> <ul style="list-style-type: none"><li>➤ Tener en cuenta requisitos que involucren la actualización futura del software.</li></ul>

<p><b>Análisis y Diseño</b></p>	<p>Patrones</p> <ul style="list-style-type: none"><li>➤ Realizar una correcta identificación, con el objetivo de propiciar un bajo acoplamiento, alta cohesión y modularidad en el sistema.</li></ul> <p>Diseño de Interfaces</p> <ul style="list-style-type: none"><li>➤ Diseñar las interfaces teniendo en cuenta desde el inicio como objetivo un bajo acoplamiento y una alta cohesión.</li><li>➤ Desarrollar las interfaces de forma tal que dependan lo menos posible del sistema operativo.</li><li>➤ Evitar al máximo la ocurrencia de errores en las interfaces y crear un diseño sencillo.</li></ul> <p>Estilos arquitectónicos</p> <ul style="list-style-type: none"><li>➤ Seleccionar un estilo arquitectónico que contribuya a la modularidad del producto.</li><li>➤ Definir tipos de componentes presentes en el estilo.</li><li>➤ Identificar las interacciones que se establecen entre los componentes.</li></ul> <p>Lenguaje de Programación.</p> <ul style="list-style-type: none"><li>➤ Seleccionar un lenguaje de programación estándar que tribute a la portabilidad del producto de software.</li><li>➤ Examinar las debilidades del lenguaje para evitar problemas durante el desarrollo.</li></ul>
---------------------------------	---

<b>Implementación</b>	<p>Estándares de Programación</p> <ul style="list-style-type: none"><li>➤ Tener en cuenta el uso de los estándares internacionales de programación.</li></ul> <p>Complejidad de los algoritmos</p> <ul style="list-style-type: none"><li>➤ Realizar actividades que permitan la revisión del código y el seguimiento de errores.</li><li>➤ Utilizar algoritmos lo menos complejos posible.</li><li>➤ Asegurar la legibilidad en el código.</li></ul>
<b>Prueba</b>	<p>Pruebas de portabilidad</p> <ul style="list-style-type: none"><li>➤ Instalar el software en los diferentes entornos en los que se va a implantar.</li><li>➤ Comprobar que el software funciona de la manera esperada en los diversos entornos para los que fue desarrollado.</li><li>➤ Documentar los errores encontrados para evitar que estos se repitan.</li></ul>
	<p>Escenarios de despliegue</p> <ul style="list-style-type: none"><li>➤ Verificar la correspondencia del escenario de</li></ul>

<b>Despliegue</b>	despliegue con los requisitos no funcionales. Documentación de portabilidad <ul style="list-style-type: none"><li>➤ Documentar las posibles dependencias del sistema y los tipos de adaptación que pueden realizarse a la hora de portar el software.</li><li>➤ Identificar y separar las partes del sistema dependientes e independientes en distintas secciones o documentos separados, para una nueva implementación, sólo la documentación limitada a las dependencias del sistema debe rediseñarse.</li></ul>
<b>Soporte</b>	Modificaciones al software <ul style="list-style-type: none"><li>➤ Reducir la cantidad de modificaciones al software.</li></ul>

**Tabla 1. Guía de actividades para los elementos definidos por etapas.**

### 2.4 Conclusiones parciales

En este capítulo se definieron elementos a tener en cuenta en las etapas del ciclo de vida de desarrollo de software que influyen sobre la portabilidad. Se propuso una serie de actividades para asegurar la portabilidad en cada elemento seleccionado, por lo tanto se confeccionó una guía para el aseguramiento de la portabilidad en los sistemas informáticos de gobierno electrónico.

## Capítulo 3: Validación de la propuesta solución

### 3.1 Introducción

Este capítulo está enfocado a la validación de la guía anteriormente propuesta mediante las valoraciones y criterios de una población de especialistas seleccionada. El análisis de las respuestas de estos se visualizan mediante representaciones gráficas. Para obtener la información necesaria para esta validación fue a través del método Delphi utilizando la variante del criterio de especialistas, este fue seleccionado por la capacidad que posee para conformar, valorar y enriquecer criterios, guías, concepciones, modelos y estrategias.

Características presentes en el método Delphi.

**Anonimato:** durante la aplicación del método ningún especialista conoce la identidad de los otros que componen el grupo de debate.

**Iteración y realimentación controlada:** la iteración se consigue al presentar varias veces el mismo cuestionario, con lo que se logra que los especialistas vayan conociendo los diferentes puntos y puedan ir modificando su opinión.

#### Táctica utilizada para la aplicación del método

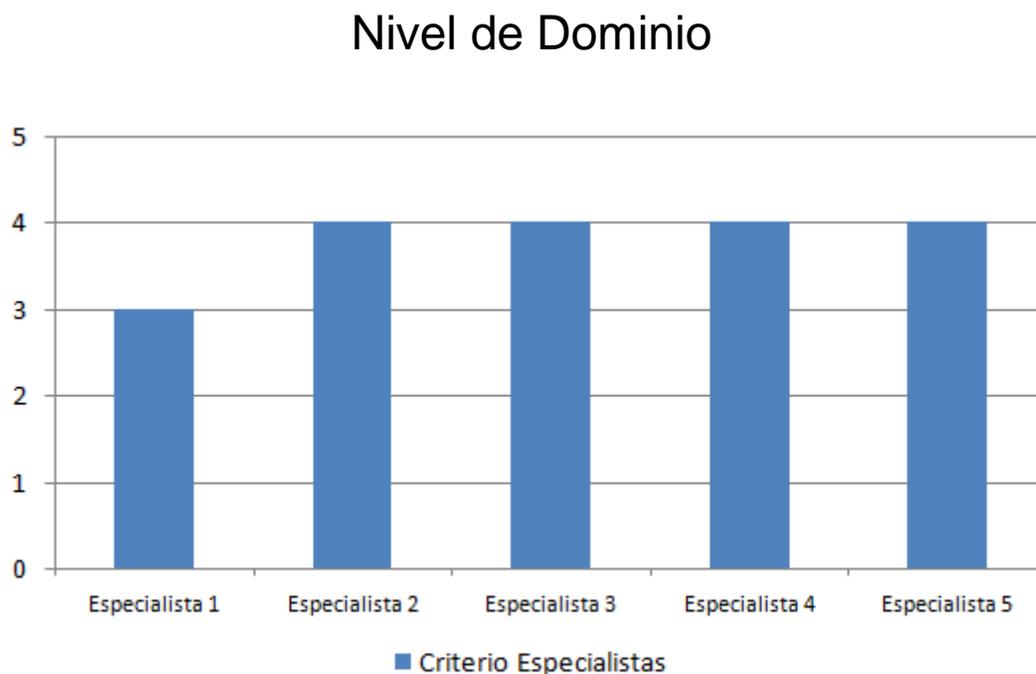
1. Se determina el área del conocimiento que los especialistas deben dominar y así determinar la población de especialistas apta para evaluar la propuesta.

- En este caso el área del conocimiento seleccionada es Arquitectura de Software ya que es donde se trabajan las características de calidad de un software, haciendo énfasis en la portabilidad.

- Muestra: 4 arquitectos de software y 1 jefe de proyecto.

2. Se contactó con cada especialista seleccionado para ver su disposición para realizar la encuesta, se determina el nivel de dominio que estos poseen sobre el tema de la portabilidad y la influencia de las fuentes de argumentación sobre el conocimiento de los mismos. Determinando así el nivel de competencia de cada uno.

El nivel de dominio del tema de los especialistas fue determinado a través del cuestionario aplicado y la autovaloración de cada uno de ellos.



**Figura 11. Nivel de dominio del tema de cada especialista del 1 al 5**

Para determinar el grado de influencia de las fuentes de argumentación en el criterio de los especialistas se tuvieron en cuenta 3 factores fundamentales, los conocimientos teóricos del tema, la experiencia obtenida en la práctica y trabajos de autores internacionales consultados sobre el tema. El instrumento aplicado proporcionó la siguiente información:

Para las siguientes gráficas de barra se deben entender los valores numéricos de la siguiente manera:

- 1-Bajo
- 2 -Medio
- 3 -Alto

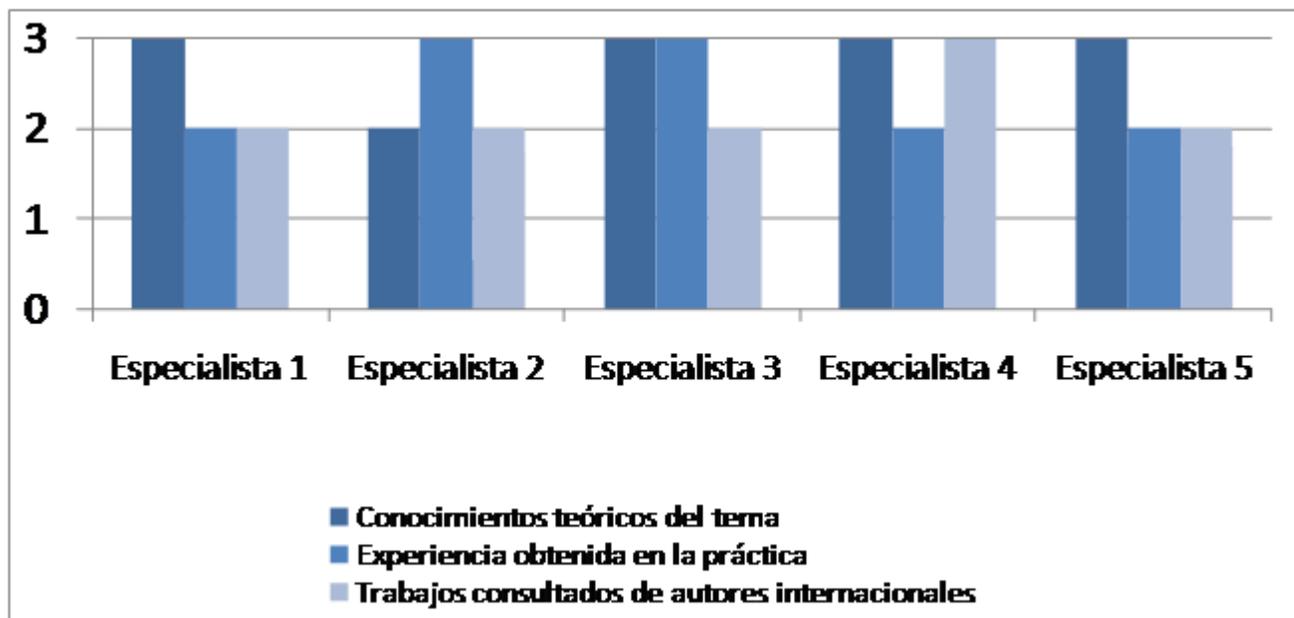


Figura 12. Influencia de las fuentes de argumentación en el criterio de los especialistas en una escala del 1 al 3

Analizando los resultados de la figura 11 y 12 se procede a determinar el nivel de competencia de cada uno de los especialistas seleccionados para realizarle la encuesta para comprobar que cada uno cuenta con el nivel requerido.

Esto fue posible mediante la aplicación de la siguiente fórmula.

$$k = \frac{1}{2}(kc + ka)$$

Donde **k**: coeficiente de competencia.

**Kc**: es el coeficiente de conocimiento o información que tiene el especialista acerca del problema, calculado sobre la valoración del propio especialista en una escala del 0 al 5 y multiplicado por 0,2; de esta forma, la evaluación "0" indica que el especialista que no tiene absolutamente ningún conocimiento de la problemática correspondiente, mientras que la evaluación "5" significa que el especialista tiene pleno conocimiento de la problemática tratada.

---

## Validación de la propuesta solución

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

<i>Fuentes de argumentación</i>	<i>Grado de influencia de cada una de las fuentes en sus criterios.</i>		
	Alto	Medio	Bajo
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores extranjeros	0.2	0.15	0.1

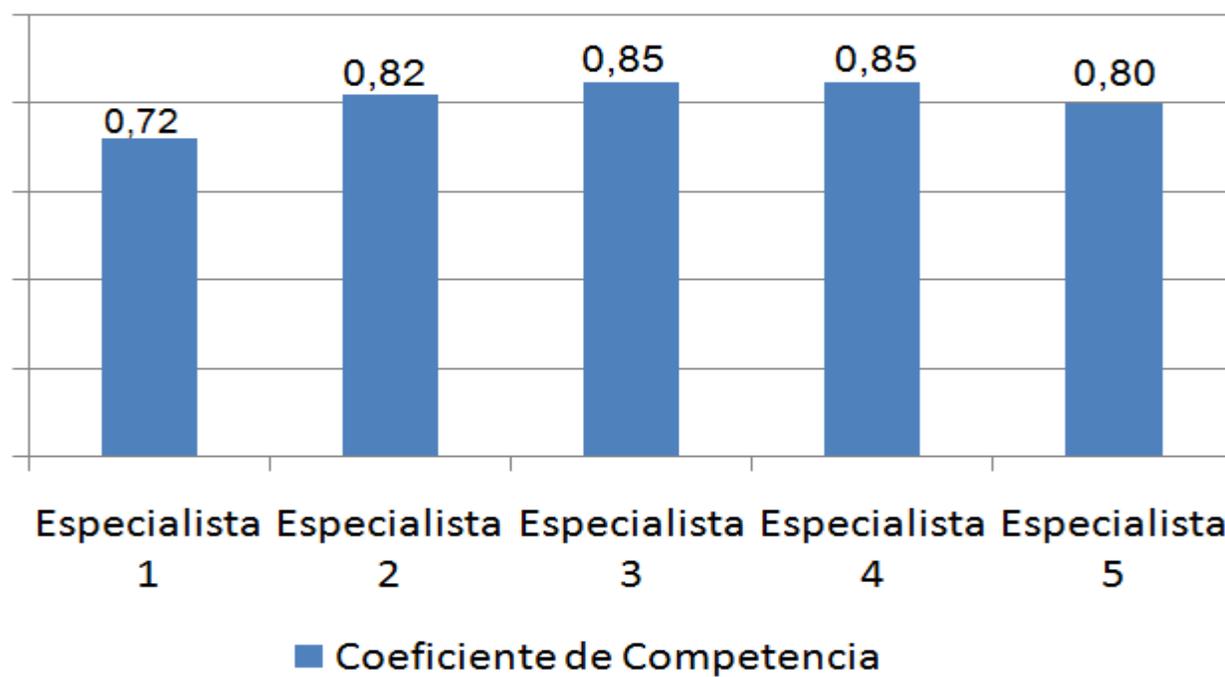
Para la interpretación de los resultados se debe tener en cuenta que:

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

Si  $0,5 \leq k < 0,8$  el nivel de competencia es medio.

Si  $k < 0,5$  el nivel de competencia es bajo.

### Coeficiente de Competencia



**Figura 13. Resultados obtenidos mediante la aplicación de la fórmula planteada**

Analizando los resultados obtenidos se determina que el coeficiente de competencia de los especialistas seleccionados es alto promediando entre todos 0,808. Estos datos demuestran un alto nivel de competencia de los mismos, lo que da alto valor a sus criterios con respecto a cada una de las preguntas realizadas en el cuestionario.

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

El cuestionario está compuesto por siete preguntas dirigidas a valorar los elementos definidos para asegurar la portabilidad, durante las etapas del ciclo de desarrollo de software basado en el programa de mejora que lleva a cabo la universidad.

Las respuestas a las preguntas han sido graficadas de manera que se puedan comprender mejor los resultados en la siguiente tabla.

---

## Validación de la propuesta solución

	Muy Adecuado	Bastante Adecuado	Adecuado	Poco Adecuado	No Adecuado
Pregunta #1		3	2		
Pregunta #2	2	2	1		
Pregunta #3	3	1	1		
Pregunta #4	3	2			
Pregunta #5	4	1			
Pregunta #6	3	2			
Pregunta #7		2	3		

**Tabla 2. Evaluación por preguntas de los especialistas**

### Resultados de la aplicación del cuestionario

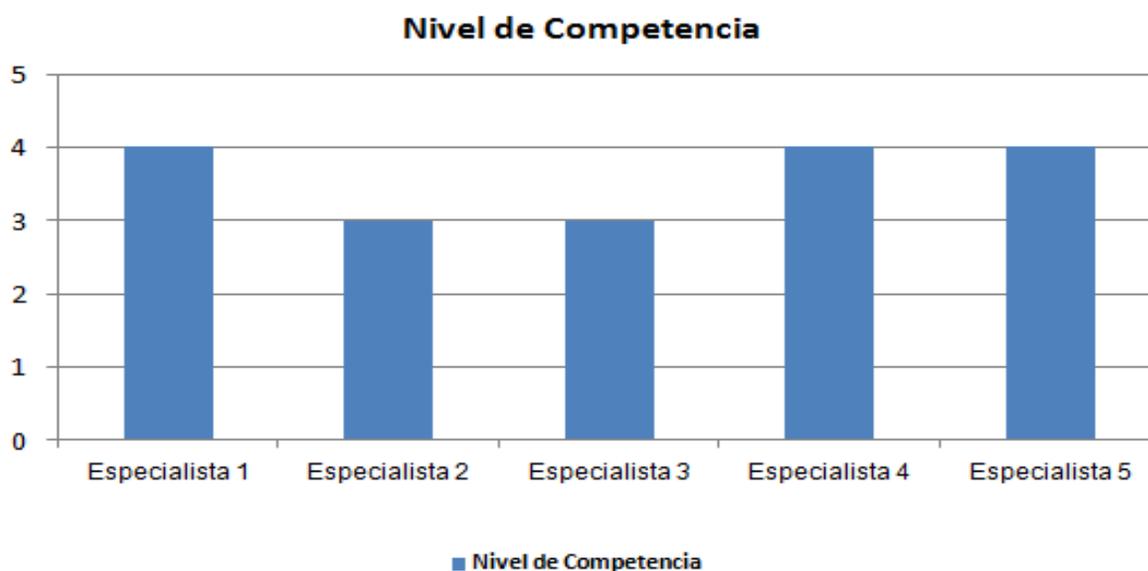
Los datos obtenidos con la realización de la encuesta serán mostrados a través de gráficas para un mayor entendimiento de estos.

Para las siguientes gráficas de barra se deben entender los valores numéricos de la siguiente manera:

- 1- No adecuado
- 2- Poco adecuado

- 3- Adecuado
- 4- Bastante adecuado
- 5- Muy adecuado

La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la primera pregunta de la encuesta. En la primera pregunta se plantea la definición de las Reglas del Negocio como elemento importante en la etapa de Modelación del Negocio que influye en la portabilidad del producto final.



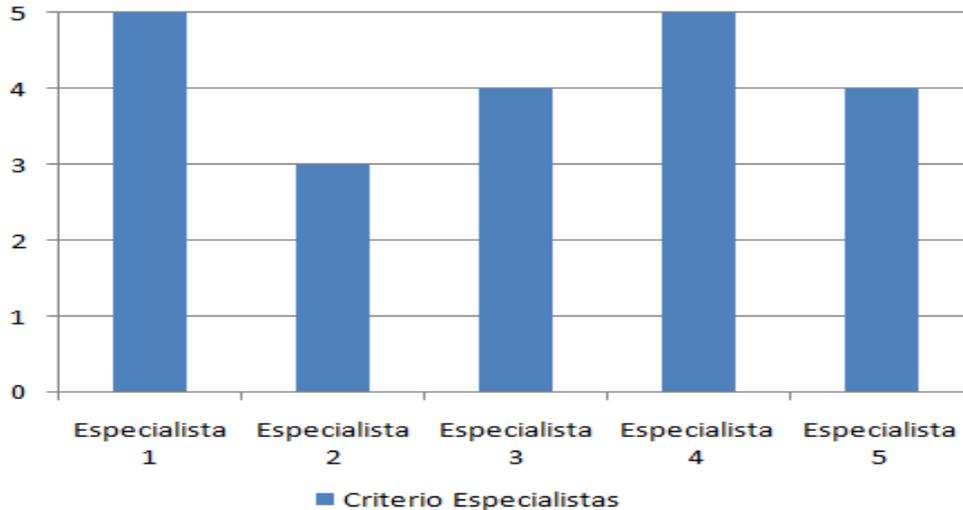
**Figura 14. Criterio de evaluación de la pregunta 1 por especialista**

El 100 % de los especialistas valora entre adecuado y bastante adecuado la definición de las Reglas del Negocio en la etapa de Modelación del Negocio como elemento que influye en la portabilidad del producto final.



**Figura 15. Por ciento del criterio de evaluación de los especialistas para la pregunta 1**

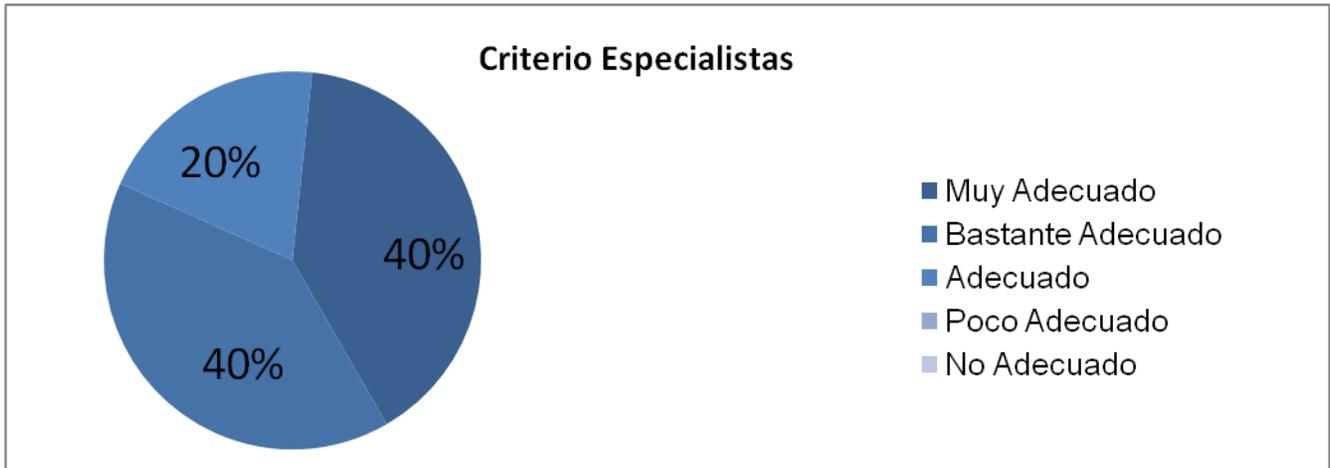
La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la segunda pregunta de la encuesta. En la segunda pregunta se plantean los requisitos de portabilidad, el plan de gestión de requisitos y los requisitos de actualización de software como elementos de la etapa de Requisitos que determinan de manera general la portabilidad del producto final.



**Figura 16. Criterio de evaluación de la pregunta 2 por especialista**

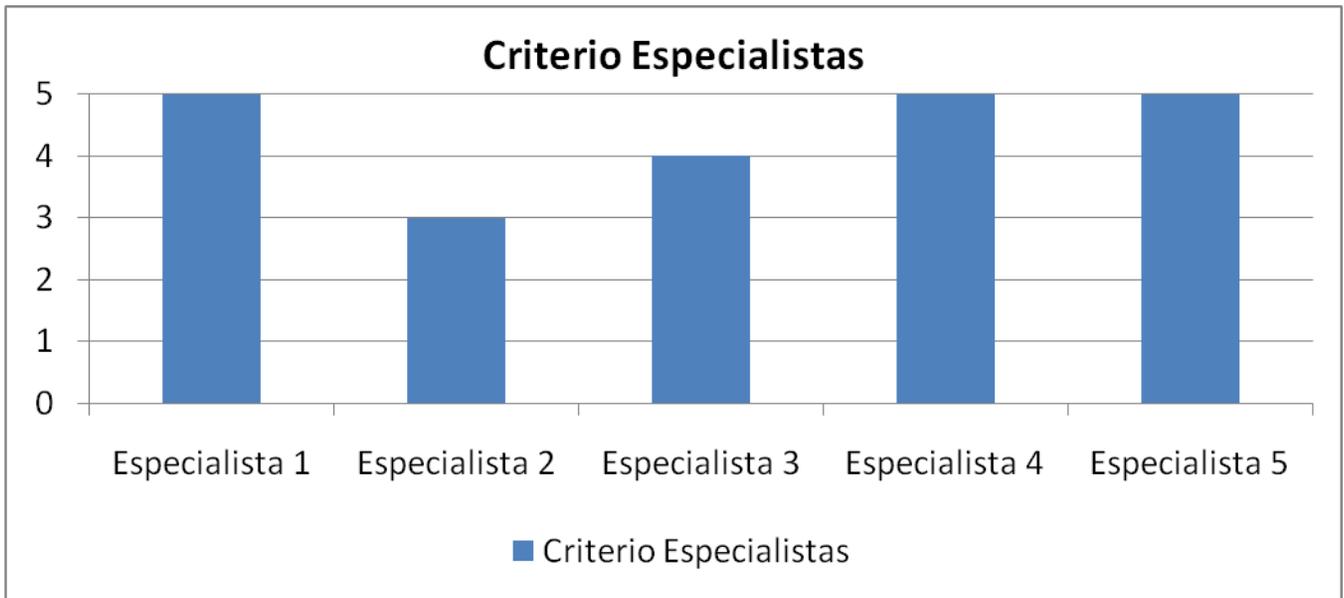
El 100 % de los especialistas valora entre adecuado y muy adecuado la definición correcta de los requisitos de portabilidad, el plan de gestión de requisitos y los requisitos de actualización de

software en la etapa de Requisitos como elementos que determinan de manera general la portabilidad del producto final como se muestra en la siguiente figura.



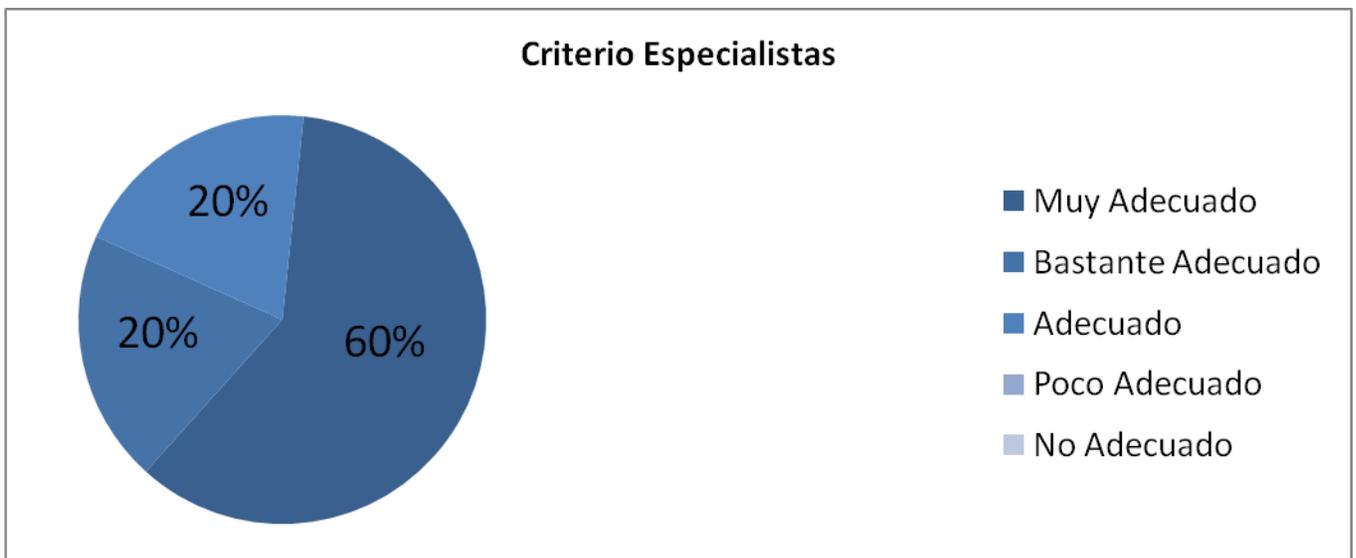
**Figura 17. Por ciento del criterio de evaluación de los especialistas para la pregunta 2**

La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la tercera pregunta de la encuesta. En la tercera pregunta se plantean los estilos arquitectónicos, los patrones, el diseño de interfaz y los lenguajes de programación como los elementos primordiales en el aseguramiento de la portabilidad durante la etapa de Análisis y Diseño.



**Figura 18. Criterio de evaluación de la pregunta 3 por especialista**

El criterio de los especialistas es 100% entre adecuado y muy adecuado respecto a la pregunta 3, donde se plantean a los estilos arquitectónicos, los patrones y el diseño de interfaz como los elementos primordiales en el aseguramiento de la portabilidad durante la etapa de Análisis y Diseño. Ver figura siguiente.



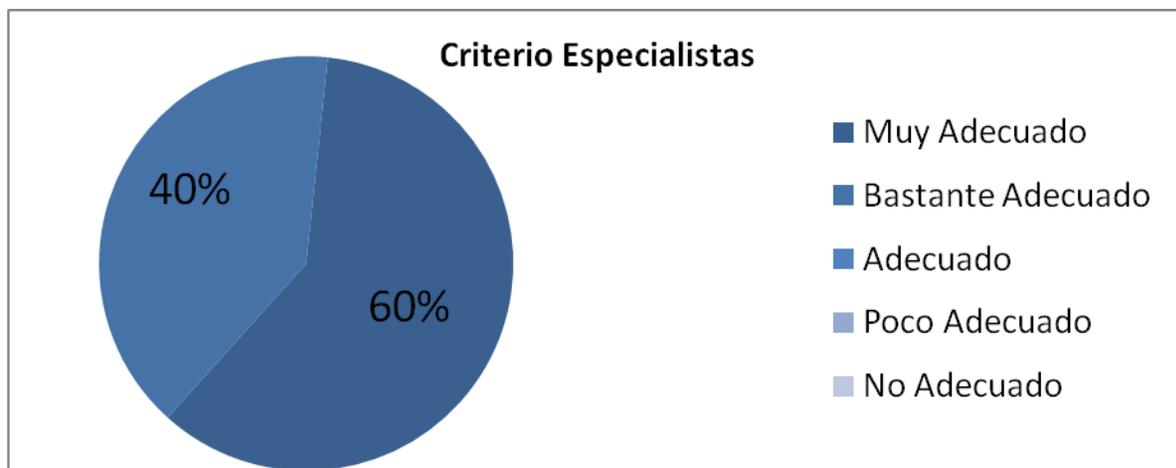
**Figura 19. Por ciento del criterio de evaluación de los especialistas para la pregunta 3**

La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la cuarta pregunta de la encuesta. En la cuarta pregunta se plantean la complejidad de los algoritmos y los estándares de programación como los elementos críticos de la etapa de Implementación que pueden influir en la portabilidad de los productos de software.



**Figura 20. Criterio de evaluación de la pregunta 4 por especialista**

El 100% de los especialistas valoró entre adecuado y muy bastante adecuado los elementos definidos en la etapa de Implementación. Estos resultados se exponen en la siguiente gráfica.



**Figura 21. Por ciento del criterio de evaluación de los especialistas para la pregunta 4**

La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la quinta pregunta de la encuesta. En la quinta pregunta se definen las pruebas de portabilidad como elemento de la etapa de Pruebas que influencia en la portabilidad de los sistemas.



**Figura 22. Criterio de evaluación de la pregunta 5 por especialista**

Se valoró por el 100% de los especialistas entre adecuado y muy bastante adecuado a las pruebas de portabilidad como elemento crítico que influye en la portabilidad final de los sistemas.



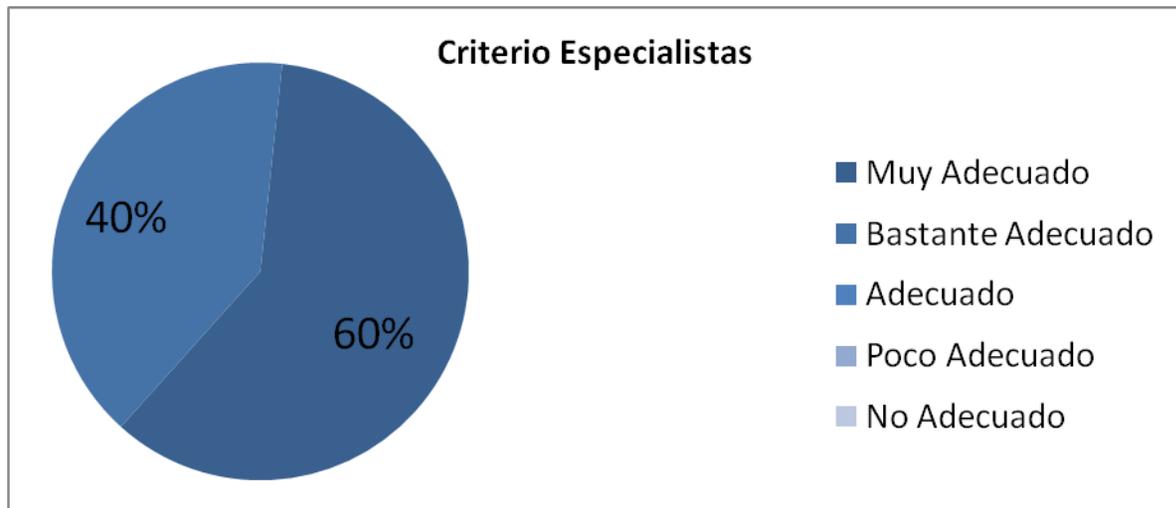
**Figura 23. Por ciento del criterio de evaluación de los especialistas para la pregunta 5**

La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la sexta pregunta de la encuesta. En la sexta pregunta se plantea la definición de los escenarios de despliegue durante la etapa de Despliegue como elemento crítico para asegurar la portabilidad.



**Figura 24. Criterio de evaluación de la pregunta 6 por especialista**

Se valoró por el 60% de los especialistas como muy adecuado y por el resto como bastante adecuado los elementos de la etapa de Despliegue



**Figura 25. Por ciento del criterio de evaluación de los especialistas para la pregunta 6**

La siguiente gráfica está vinculada a la información obtenida por la respuesta dada por cada especialista respecto a la séptima pregunta de la encuesta. En la séptima pregunta se plantea que las modificaciones al software en la etapa de Soporte pueden influir en la portabilidad del sistema, los especialistas evaluaron lo siguiente:



**Figura 26. Criterio de evaluación de la pregunta 7 por especialista**

La gráfica muestra el resultado en por ciento de la evaluación por los especialistas del elemento definido en la etapa de Soporte. El 40% consideró el elemento como bastante adecuado y el resto como adecuado.



**Figura 27. Por ciento del criterio de evaluación de los especialistas para la pregunta 7**

Como se puede observar en los resultados de la aplicación del cuestionario a los especialistas, el 100% de estos evaluó la guía entre adecuada y muy adecuada. Para dar mayor veracidad a los resultados estos se analizan estadísticamente, por lo que se determina el grado de concordancia entre las respuestas de los analistas. Se decidió utilizar para esto el coeficiente de Kendall de la herramienta Paquete Estadístico para las Ciencias Sociales (SPSS). Los resultados obtenidos son los siguientes:

### Kendall's W Test

	Mean Rank
P1	1,90
P2	3,90
P3	4,50
P4	5,20
P5	5,70
P6	5,20
P7	1,60

### Test Statistics

N	5
Kendall's W <sup>a</sup>	,802
Chi-Square	24,059
df	6
Asymp. Sig.	,001

a. Kendall's Coefficient of Concordance

**Figura 28. Resultados arrojados por la herramienta**

Los resultados de la aplicación de la herramienta SPSS muestran el número de casos válidos (N),

el valor estadístico  $W$  ( $W$  de Kendall), su transformación en Chi-cuadrado (Chi-Square), sus grados de libertad ( $df$ ) y el nivel crítico (Asymp. Sig.) como se puede observar en la figura 28.

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.

Para esta investigación  $W = 0,802$ , por lo que se puede afirmar que el nivel de concordancia entre los criterios de los especialistas es alto. El nivel crítico obtenido fue de 0,001 menor que 0,05, por lo tanto se rechaza la hipótesis nula y se acepta la hipótesis alternativa demostrándose que el nivel de concordancia de los especialistas es significativo.

### **3.2 Conclusiones Parciales**

En el presente capítulo se empleó el método Delphi con la variante del criterio de especialistas para la validación de la propuesta de solución. Los resultados obtenidos con la aplicación de este método demostraron que el nivel de concordancia entre los especialistas es significativo luego de haber sido corroborada la prueba de hipótesis.

### **Conclusiones generales**

- ✓ Se realizó un estudio de diferentes conceptos de calidad de software y aseguramiento de la calidad definiendo conceptos propios para el desarrollo de la investigación.
- ✓ Se analizaron diferentes modelos y estándares de calidad a nivel de producto y se seleccionó el modelo ISO/IEC 25010 SQuaRE para tener en cuenta la portabilidad y sus subcaracterísticas para el desarrollo de la propuesta de solución.
- ✓ Se definieron 14 elementos que tributan a la portabilidad en las diferentes etapas del ciclo de vida de desarrollo de software que plantea el programa de mejora y 25 actividades desglosadas para estos elementos.
- ✓ Se validó la propuesta de solución a través de la utilización del método Delphi utilizando el criterio de especialistas, donde luego de haber aplicado el coeficiente de Kendall se obtuvo un 0,802 de concordancia entre los especialistas, corroborándose mediante la prueba de hipótesis que la propuesta definida tiene un 99,9% de efectividad.

### **Recomendaciones**

- Aplicar la guía propuesta en el centro CEGEL y analizar los resultados obtenidos con la aplicación de esta.
- Capacitar al personal inmerso en el proceso de desarrollo del software que de alguna manera interfieran en los posibles resultados de la aplicación de la guía.
- Mantener la guía actualizada con nuevos elementos que puedan surgir de acuerdo a lo que se defina en cada Centro o departamento y a las nuevas tendencias que surjan en el mundo.

## **Glosario de términos**

**Cliente:** Una persona u organización, interna o externa a la organización productora que toma responsabilidad financiera por el sistema. El cliente es el último destinatario del producto desarrollado y sus artefactos.

**CEGEL:** Centro de Gobierno Electrónico.

**CMMI:** Modelo de Capacidad y Madurez Integrado (Capability Maturity Model Integration).

**ISO:** (International Organization for Standardization) Es la Organización Internacional para la Normalización; responsable para la normalización a escala mundial.

**Especialista:** El término se relaciona con el ejercicio de la profesión en relación con el objeto de estudio de la investigación. Es especialista el que practica una rama determinada de la ciencia.

## Referencias bibliográficas y bibliografías consultadas.

### Referencias bibliográficas

**Calisoft. 2009.** Calisoft- Centro de Calidad Para Soluciones Informáticas. [Online] 2009. [Cited: 3 20, 2012.] [http://calisoft.uci.cu/index.php/proceso-de-mejora/046\\_046\\_Procesos y Guías.](http://calisoft.uci.cu/index.php/proceso-de-mejora/046_046_Procesos_y_Guías)

**IEEE. 1990.** *Standard Glossary of Software Engineering.* 1990.

**ISO/IEC, 9126. 1998.** *Software Engineering - Product Quality - Part 1: Quality.* 1998.

**ISO/IEC. 9126. 2001.** *Software Product Evaluation-Quality Characteristics and Guidelines for their Use.* 2001.

**ISO9000. 2000.** *Sistema de Gestión de la Calidad - Principios y Vocabulario.* 2000.

**Jim A. McCall, Paul K. Richards, Gene F. Walters. 1977.** *FACTORS IN SOFTWARE QUALITY. Concept and Definitions of Software Quality.* 1977.

**Mary Beth Chrissis, Mike Konrad, Sandy Shrum. 2009.** *Guía para la integración de procesos.* 2009.

**Mooney, James D. 1990.** *Bringing Portability to the Software Process.* 1990.

**Pressman, Roger. 2002.** *Ingeniería de software. Un Enfoque Práctico.* 2002.

**Rick Kazman, Hoh In, David Olson. 2001.** *From Requirements Negotiation to Software Architectural Decisions.* 2001.

**SCALONE, LIC. FERNANDA. 2006.** *ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE.* 2006.

**Stufflebeam, D.L. 1997.** *A Standards-Based Perspective on Evaluation. Advances in Program Evaluation. Vol.3 Jay Press.* 1997.

### Bibliografía consultada

**Smith, Williams Connie. 2001.** *Introduction to software performance.* 2001.

**Aleksander González, Marizé Mijares, Luis E. Mendoza, Anna Grimán, María Pérez. 2005.** *Método de Evaluación de Arquitecturas de Software Basadas en Componentes.* 2005.

- Antonio Calero, Paco Castro, David García Hugo Mora, Miguel Ángel Vicedo. 2009.** *Visión innovadora de la calidad del producto software.* 2009.
- Barchini, Gaciela E., Alvarez, Margarita M. and Palliotto, Diana. 2009.** *Evaluación de la Calidad de los Sistemas de Información basados en ontologías.* 2009.
- Coral Calero, Ismael Caballero, M<sup>a</sup> Ángeles Moraga, Manuel Serrano. 2009.** *Introducción a la calidad. Modelos de calidad. ISO 9126. Ejemplos de modelos de calidad: PQM, WQM.* 2009.
- FABIO CARDESO, GABRIEL NUÑEZ. 2004.** *ARQUITECTURAS DE SOFTWARE.* 2004.
- Grupo Investigación en Sistemas de Información. 2012.** *ASEGURAMIENTO DE LA CALIDAD PARA PROYECTOS DE EXPLOTACIÓN DE INFORMACIÓN.* 2012.
- Laboratorio Nacional de Calidad del Software. 2009.** *GUÍA DE CERTIFICACIÓN.* 2009.
- Lovelle, Juan Manuel Cueva. 1999.** *Calidad del Software.* 1999.
- M.G. Piattini, J.A. Calvo-Manzano, J. Cerveza, y L. Fernández. 2006.** *Análisis y diseño detallado de aplicaciones informáticas de gestión.* 2006.
- McConnell, S. 1993.** *Code Complete: Apractical Handbook of Software Construction, Microsoft Press.* 1993.
- Ospina Gutiérrez, Luz María, Botero Arbeláez, Marcela and Mendoza Vargas, Jairo Alberto. 2008.** *Importancia de la metrología al interior de las empresas para el aseguramiento de la calidad.* 2008.
- Pulido, Javier Hernán. 2004.** *ESTÁNDARES DE CALIDAD.* 2004.
- Quiñones, Ernesto. 2009.** *Modelos de Calidad de Software y Software Libre.* 2009.
- SALANOVA, PILAR ESPARZA. 2006.** *MODELOS DE CALIDAD WEB.* 2006.

## **Anexos**

### **Anexo 1**

Entrevista realizada a los proyectos del Centro con el objetivo de determinar cómo se trata la portabilidad durante el desarrollo de sus aplicaciones.

- 1- ¿Cuentan con personal especializado en el tema de la portabilidad?
- 2- ¿Hacen uso de algún modelo o estándar para asegurar la portabilidad en los productos que desarrollan?
- 3- ¿Realizan alguna actividad durante el desarrollo de los productos que tribute a la portabilidad final de los mismos?

### **Anexo 2**

## **Encuesta aplicada a especialistas**

**Encuesta a especialistas para someter a sus criterios la propuesta de la Guía para el aseguramiento de la portabilidad como característica de calidad en los sistemas informáticos de Gobierno Electrónico.**

Estimado profesor(a):

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

Muchas Gracias.

1. Seleccione en una escala del 1 – 5 el valor que corresponda con el grado de conocimientos que usted posee acerca del tema de investigación que desarrollamos (aseguramiento de la portabilidad como característica de calidad en sistemas informáticos de 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 posee acerca del tema			
Su experiencia obtenida en la actividad práctica			
Trabajos consultados de autores internacionales			

La propuesta de la guía para el aseguramiento de la portabilidad 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 Modelación del Negocio.

Criterio de especialista					
Pregunta	C1	C2	C3	C4	C5
	Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
¿Considera que la correcta definición de las Reglas del Negocio es un elemento fundamental en la etapa de Modelación del Negocio que influya en la portabilidad de los sistemas?					

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

Criterio de especialista					
Pregunta	C1	C2	C3	C4	C5
	Muy adecuado	Bastante adecuado	Adecuado	Poco adecuado	No adecuado
¿Considera que la correcta definición de los requisitos de portabilidad, el plan de gestión de requisitos y tener en cuenta requisitos que involucren la actualización del software son elementos en la etapa de Requisitos que determinaban de manera general la portabilidad de los sistemas?					

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

Criterio de especialista					
Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera usted que la correcta selección de los estilos arquitectónicos, los patrones, el correcto diseño de las interfaces y la selección de los lenguajes de programación son los elementos primordiales en el aseguramiento de la portabilidad durante la etapa de Análisis y Diseño?					

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

<b>Criterio de especialista</b>					
<b>Pregunta</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>
	<b>Muy adecuado</b>	<b>Bastante adecuado</b>	<b>Adecuado</b>	<b>Poco adecuado</b>	<b>No adecuado</b>
¿Considera usted que la complejidad de los algoritmos y los estándares de programación son elementos críticos de la etapa de Implementación que pueden influir en la portabilidad del sistema?					

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

Criterio de especialista					
Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera usted que las pruebas de portabilidad es un elemento fundamental en la etapa de Pruebas que influya en la portabilidad final de los sistemas.					

8. Le pedimos su criterio acerca de los elementos definidos en la etapa de Despliegue.

Criterio de especialista					
Pregunta	C1 Muy adecuado	C2 Bastante adecuado	C3 Adecuado	C4 Poco adecuado	C5 No adecuado
¿Considera usted que los escenarios de despliegue y la documentación de la portabilidad durante					

la etapa de Despliegue son los elementos críticos para asegurar la portabilidad?					
--	--	--	--	--	--

9. Le pedimos su criterio acerca de los elementos definidos en la etapa de Soporte.

<b>Criterio de especialista</b>					
<b>Pregunta</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>	<b>C5</b>
	<b>Muy adecuado</b>	<b>Bastante adecuado</b>	<b>Adecuado</b>	<b>Poco adecuado</b>	<b>No adecuado</b>
¿Considera usted que las modificaciones al software en la etapa de Soporte pueden influir en la portabilidad del sistema?					