

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
FACULTAD 5**



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

**Título:** *Procedimiento para evaluar la característica de funcionalidad de  
componentes software*

**Autora:**

Yuneisis Estrada Peña

**Tutoras:**

MSc. Liudmila Reyes Álvarez

Ing. Yadirá Morales Álamo

La Habana

Junio del 2012



*"El mundo está en las manos de aquellos que tienen el coraje de soñar y correr el riesgo de vivir sus sueños"*

*Ernesto Che Guevara*

## *Declaración de autoría*

Declaro ser autora 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 \_\_\_\_\_.

\_\_\_\_\_  
Firma de la Autora  
Yuneisis Estrada Peña

\_\_\_\_\_  
Firma de la Tutora  
MSc. Liudmila Reyes Álvarez

\_\_\_\_\_  
Firma de la Tutora  
Ing. Yadira Morales Álamo

## *Datos de contactos*

### **Tutoras:**

#### **MSc. Liudmila Reyes Álvarez**

Institución: Universidad de las Ciencias Informáticas (UCI)

Categoría Docente: Instructor

Años de experiencia: 5

Teléfono de contacto: 837 2741

Correo electrónico: lreyes@uci.cu

#### **Ing. Yadira Morales Álamo**

Institución: Universidad de las Ciencias Informáticas (UCI)

Categoría Docente: Instructor

Años de experiencia: 4

Teléfono de contacto: 837 3201

Correo Electrónico: yalamo@uci.cu

## *Agradecimientos*

*A dios: porque es mi guía y mi luz.*

*A mis padres y mi hermano: porque son mi más valioso regalo y bendición.*

*A Juana y Alberto por acompañar a mis padres y apoyarme en mi carrera.*

*A mi novio Alien: que amo tanto y me hace tan feliz.*

*A mi familia: por ser tan importantes para mí y yo para ellos.*

*A Yunerkis Prevot: por ser el mejor amigo del mundo y a quien le debo haber llegado hasta aquí.*

*A Adisleidys: por ser como una hermana para mí y por estar a mi lado en mis momentos buenos y malos.*

*A mis amigos: sin ellos no fuera nadie, les debo lo que soy y mis momentos de felicidad, sobre todo a Yenislely en su asesoramiento y apoyo.*

*A mis tutoras Liudmila y Yadira: porque fueron mi mejor apoyo en esta etapa.*

*A mis profesores: por haber cumplido en mi formación y superación como ingeniera, especialmente a la profesora Andrea y el profesor Millet por todo el cariño con que me han apoyado durante este tiempo.*

*A los especialistas del Grupo de Asistencia Técnica de la UCI en especial a Jeanlup por ayudarme sin titubeo en mi máquina.*

*A todos muchas GRACIAS!!!*

## *Dedicatoria*

*A mi mama Mariela y a mi papa Enrique. Esta victoria es de ellos también, a quien más he de dedicarle, si yo he sido el fruto de sus manos.*

*A mi hermano Yoel que quiero tanto y en todo momento ha estado de mi lado.*

*A toda mi familia, que siempre creyó y espero tanto de mí, en especial a mis abuelitas María y Celsa, mis tías (os) Mercedes, Celeste, Aida, Ester, Tico, Rolando, Luis, Chino, Julio y Felo, y mis primas (os) Dianne, Danay, Rolandito, Luisito y Luis Mario.*

*A mis amigos que fueron mi otra familia durante estos años, en especial a Berto que lo considero como un padre para mí.*

## *Resumen*

Los continuos avances en la Informática están haciendo cambiar la forma en la que se desarrollan las aplicaciones software, requiriendo reutilizar todo lo posible pero con la máxima calidad, lo que exige contar con estrategias que apoyen esta propiedad, una de ellas es la evaluación.

La presente investigación propone un procedimiento para evaluar la característica funcionalidad de componentes software que se desarrollan en el CEDIN. La cual representa la característica más importante de la calidad, se considera un atributo imprescindible para los productos software. Para la realización de la investigación se utilizan como guías las Normas ISO/IEC 9126 que se ocupa de la calidad de los productos y la ISO/IEC 14598 encargada de la evaluación de los productos de software.

El procedimiento está compuesto por 4 etapas, estas son: Establecimiento de los requisitos de la evaluación, Proceso de medición, Ejecución de la medición y Conclusión de la evaluación. Cada una de estas tiene definidas un grupo de actividades para darle cumplimiento al procedimiento.

La validación de la solución propuesta se hace en tres componentes software que utiliza el proyecto Sistema de Manejo Integral de Perforación de Pozos Petroleros (SIPP), estos componentes son Archivo, Gráfica y Seguridad, reutilizables y con interfaces definidas que les permiten vincularse a otros módulos.

**Palabras Clave:** calidad, componentes software, etapas, evaluación, funcionalidad

# *Índice*

Introducción .....	1
CAPÍTULO1 .....	5
1.1 Desarrollo de Software Basado en Componentes .....	5
1.2 Clasificación de los componentes software .....	7
1.3 ¿Qué es calidad? .....	9
1.3.1 Calidad del software .....	10
1.4 Modelos y normas de calidad de productos software .....	10
1.5 Evaluación de la calidad de los productos de software .....	14
1.6 Descripción de las normas a utilizar .....	15
1.7 Modelo para la calidad .....	17
1.8 Funcionalidad.....	17
1.9 Métricas de software .....	18
1.10 Técnicas para evaluar la funcionalidad de productos software .....	19
CAPÍTULO 2.....	22
2.1 Roles y responsabilidades de los participantes en el proceso de evaluación .....	22
2.2 Métricas Funcionales. Conceptos .....	24
2.3 Métricas de funcionalidad.....	24
2.3.1 Métricas del atributo idoneidad .....	25
2.3.2 Métrica del atributo exactitud .....	27
2.3.3 Métricas del atributo interoperabilidad .....	28
2.3.4 Métricas del atributo seguridad.....	29
2.4 Descripción del procedimiento .....	30
2.4.1 Nombre del procedimiento.....	30
2.4.2 Objetivos .....	30
2.4.3 Alcance.....	31
2.4.4 Referencias .....	31
2.4.5 Responsables.....	31
2.4.6 Términos y definiciones .....	31

2.4.7 Normas generales .....	32
2.4.8 Desarrollo del procedimiento .....	32
2.4.9 Etapa 1: Establecimiento de los requisitos de la evaluación .....	33
1. Planificación de la evaluación .....	34
2. Descripción del componente .....	34
2.4.10 Etapa 2: Proceso de medición .....	35
1. Establecer nivel de relevancia.....	36
2. Recopilación de datos.....	36
2.4.11 Etapa 3: Ejecución de la medición .....	37
1. Calcular métricas .....	38
2. Evaluación para cada métrica calculada .....	38
2.4.12 Etapa 4: Conclusión de la evaluación .....	39
1. Evaluación por subcaracterística .....	40
2. Evaluación de la funcionalidad.....	42
3. Actualización del Registro de Evaluaciones del Proyecto.....	45
CAPÍTULO 3.....	47
3.1 Descripción general del procedimiento .....	47
3.2. Proyecto seleccionado para realizar el procedimiento.....	49
3.3 Evaluación componente Archivo .....	49
1. Descripción del componente .....	49
2. Establecer nivel de relevancia .....	50
3. Calcular y evaluar métricas .....	50
4. Evaluación por subcaracterística.....	51
5. Evaluación de la funcionalidad .....	52
6. Actualización del Registro de Evaluaciones del Proyecto .....	52
3.4 Evaluación componente Gráfica.....	53
1. Descripción del componente .....	53
2. Establecer nivel de relevancia .....	54
3. Calcular y evaluar métricas .....	54
4. Evaluación por subcaracterística.....	56
5. Evaluación de la funcionalidad .....	56

---

6. Actualización del Registro de Evaluaciones del Proyecto .....	56
3.5 Evaluación componente Seguridad.....	57
1. Descripción del componente .....	57
2. Establecer nivel de relevancia .....	57
3. Calcular y evaluar métricas .....	58
4. Evaluación por subcaracterística.....	59
5. Evaluación de la funcionalidad .....	60
6. Actualización del Registro de Evaluaciones del Proyecto .....	60
3.6 Comparación de resultados .....	60
Conclusiones .....	62
Recomendaciones .....	63
Referencias Bibliográficas.....	64
Bibliografía.....	66
Anexos.....	68

## *Índice de Figuras*

Figura 1: Criterios asociados a los factores de calidad. McCall.....	11
Figura 2: Criterios asociados a los factores de calidad. Dromey .....	12
Figura 3: Criterios asociados a los factores de calidad. FURPS.....	12
Figura 4: Modelo de Boehm.....	13
Figura 5: Criterios asociados a los factores de calidad. ISO/IEC 9126.....	13
Figura 6: Proceso de evaluación según la norma 14598.....	15
Figura 7: Modelo de calidad según la ISO/IEC 9126.....	17
Figura 8: Flujo del Procedimiento.....	33
Figura 9: Actividades de la Etapa 1 .....	34
Figura 10: Actividades de la Etapa 2 .....	36
Figura 11: Actividades de la Etapa 3 .....	37
Figura 12: Actividades de la Etapa.....	40
Figura 13: Ejemplo resumen de funcionalidad .....	44
Figura 14: Resumen del componente Archivo.....	52
Figura 15: Resumen del componente Gráfica.....	56
Figura 16: Resumen del componente Gráfica.....	60
Figura 17: Por ciento de funcionalidad de los componentes Archivo y Gráfica.....	61

## *Índice de tablas*

Tabla 1: Familia del estándar ISO/IEC 9126.....	16
Tabla 2: Roles y responsabilidades de los evaluadores .....	23
Tabla 3: Descripción de los componentes .....	35
Tabla 4: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para el establecimiento del nivel. ....	36
Tabla 5: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para la Recopilación de Datos .....	37
Tabla 6: Ejemplo de un fragmento de la tabla Medición de la calidad Funcional para el cálculo de las métricas.....	38
Tabla 7: Tabla de escala.....	39
Tabla 8: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para establecer la evaluación alcanzada para cada métrica. ....	39
Tabla 9: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para establecer la evaluación alcanzada por cada subcaracterística. ....	41
Tabla 10: Criterios para evaluar la característica funcionalidad .....	43
Tabla 11: Ejemplo Tabla resumen de funcionalidad.....	44
Tabla 12: Posibles No Conformidades y Acciones Correctivas.....	46
Tabla 13: Etapas y actividades del procedimiento .....	47
Tabla 14: Métricas para realizar la evaluación.....	48
Tabla 15: No Conformidades detectadas en el componente Archivo .....	53
Tabla 16: No Conformidades detectadas en el componente Gráfica .....	57
Tabla 17: No Conformidades detectadas en el componente Seguridad .....	60

## *Introducción*

Los nuevos avances científicos y técnicos en las ramas de la comunicación, la electrónica y la informática han revolucionado el mundo en todos los aspectos.

El poderoso auge de las Tecnologías de la Información y la Comunicación (TIC) por sus siglas en español, han cambiado los paradigmas y estrategias reconocidas y establecidas por muchos años como válidas. Dentro de las TIC, la industria del software alcanza una posición relevante, por su característica de controlar o hacer accesible, en la mayoría de los casos, los adelantos electrónicos (1).

Muchas son las aplicaciones de la industria del software y cada vez son más aplicables a cualquier otra rama de la ciencia o la economía de cualquier país (1). Su creciente auge ha provocado que los sistemas de software sean cada vez más complejos, existiendo nuevos métodos, técnicas y herramientas que han sido creados para mitigar esta creciente complejidad de desarrollar software.

Una de las técnicas que ha surgido para mitigar dicha complejidad es la reutilización mediante el Desarrollo de Software Basado en Componentes (DSBC), el cual consiste en desarrollar elementos de software que puedan utilizarse más de una vez con la mínima cantidad de modificaciones, garantizando que al reutilizarlo este se encuentre libre de defectos, lo que implicará que el sistema que lo utilice no tendrá problema alguno en lo que respecta a dicho elemento.

Las bases del DSBC se fundamentan en el diseño y desarrollo de aplicaciones basadas en componentes software reutilizables, que son combinados apropiadamente para satisfacer las exigencias del sistema. Gran parte de su estructura técnica y tecnológica está bien fomentada, sin embargo es necesario abordar mejor sus características extra-funcionales y la evaluación de la calidad, aspectos que determinan la aceptación del producto final.

Cualquier organización que se dedique a la investigación, producción y comercialización de software debe tener en cuenta el aseguramiento de la calidad, mas hoy donde existe un mercado en el cual los clientes son cada vez más exigentes en cuanto a la eficiencia y confiabilidad que brindan los productos de software, este aspecto se ha convertido en estos tiempos en una realidad y ya no basta

con desarrollar sistemas, sino que cuenten con el debido nivel de calidad.

La norma ISO/IEC 9126 encargada de la calidad de los productos software describe un serie de características externas e internas de calidad del software, las cuales son la funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, refinándose en 27 subcaracterísticas, pero sería demasiado ambicioso y abarcador poderle medir o evaluar la calidad a todas estas características en un solo trabajo, por lo que se analizará la funcionalidad.

Funcionalidad es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando el software se usa bajo las condiciones especificadas (2).

En el Centro de Informática Industrial (CEDIN) de la Facultad 5 se han desarrollado varios componentes, los cuales han sido liberados por el grupo de pruebas del departamento de integración y despliegue del mismo centro. Como ejemplos de estos componentes se tienen: Aplicaciones\_v1.0, Middleware\_v1.0, Seguridad, Sonido 2D-3D, DIBI de la línea gestión industrial\_v1.0 y los componentes COMM3\_v1.0. A cada uno de estos componentes durante el proceso de pruebas se le han detectado una gran cantidad de no conformidades muchas de ellas graves ya que constituyen fallos en el componente. Teniendo en cuenta lo anterior se hace necesario fortalecer el trabajo del administrador de la calidad con la aplicación de revisiones de evaluación a cada componente con mayor periodicidad y que permita la evaluación de cada una de las características de calidad con que debe cumplir un componente y/o un producto software. Además de que las revisiones de evaluación de la calidad realizadas a estos componentes hasta la actualidad no son las más óptimas, dado que las métricas utilizadas solamente propician información de qué es lo que se va a evaluar pero no dicen cómo evaluar la calidad de software en cuanto a las características de calidad o requisitos no funcionales, trayendo consigo que se liberen productos con los siguientes problemas:

- Indebida capacidad para realizar funciones especificadas por el cliente.
- Los resultados obtenidos están desacordes a las necesidades para las cuales el componente fue concebido.
- Componentes interactuando ineficientemente con otros componentes.
- Existe una indebida protección de los datos.

Luego, como **problema científico** se formula la siguiente interrogante: ¿Cómo evaluar un componente software de acuerdo a la característica de funcionalidad en el CEDIN?

Por tanto, a partir del problema planteado, se define el siguiente **objeto de estudio**: evaluación de la calidad de componentes software.

Y para darle solución al problema científico se enuncia como **objetivo general** de la investigación: Elaborar un procedimiento que permita evaluar la característica de funcionalidad de los componentes software, el que determina el siguiente **campo de acción**: la evaluación de la calidad de los componentes software según la característica de funcionalidad.

A partir de todo lo antes expuesto en esta investigación se **defiende la idea** de que: si se elabora un procedimiento para evaluar la calidad de los componentes software en cuanto a la característica de funcionalidad, se debe contribuir a medir la idoneidad, exactitud, seguridad e interoperabilidad de dichos componentes, obteniendo un software con la calidad funcional requerida.

Para darle solución al objetivo de la investigación se desarrollarán las siguientes **tareas**:

- Elaboración del marco teórico de la investigación a través del estudio del estado del arte existente sobre el tema.
- Descripción de los principales estándares o modelos de calidad con el objetivo de determinar el que más se adecue a la solución del procedimiento.
- Caracterización de las distintas técnicas y conceptos de medición y evaluación de la calidad de software en cuanto a la funcionalidad para un mejor entendimiento de la problemática.
- Estudio de la calidad de los componentes software para definir como se podrá evaluar la calidad de un componente en cuanto a su funcionalidad en el CEDIN.
- Confección del procedimiento para evaluar la calidad de los componentes en el CEDIN en cuanto a la funcionalidad.
- Validar el procedimiento por el método de caso de estudio en algún componente del CEDIN.

Los **métodos científicos** utilizados en el desarrollo de la investigación fueron:

### Teóricos

**Analítico-sintético**: este método permitirá analizar los modelos, normas y estándares

internacionales, posibilitando la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

**Análisis histórico-lógico:** este método posibilitará constatar teóricamente como se encuentra actualmente la calidad de los componentes, es decir conocer cuál es su estado de arte.

### **Empíricos**

**Observación:** el método facilitará observar cómo se comporta el procedimiento propuesto para evaluar la calidad de un componente en cuanto a su funcionalidad.

**Experimento:** mediante este método se comprobará la efectividad o validez de los resultados obtenidos.

**Método estadístico:** para la definición de las métricas.

Este trabajo de diploma está estructurado en tres capítulos los cuales son:

**Capítulo 1: Fundamentación teórica:** En este capítulo se hace un análisis de los diferentes conceptos y definiciones sobre calidad, calidad del software, desarrollo de software basado en componentes y medición de software, también se exponen los diferentes modelos y normas encargados de la calidad de los productos software, determinando las características cualitativas que evalúan a dichos productos.

**Capítulo 2: Elaboración del procedimiento:** Se define la propuesta del procedimiento para evaluar la calidad funcional de los componentes software.

**Capítulo 3: Validación del procedimiento:** En este capítulo se valida la solución propuesta de la investigación en tres componente del CEDIN.

# CAPÍTULO 1

## *Fundamentación Teórica*

### **Introducción**

Actualmente el término DSBC está siendo ampliamente difundido y utilizado. La industria de software cubana aun siendo joven reúne sus esfuerzos para desarrollar software con menor esfuerzo, pero con la máxima calidad, con el fin de insertarse en el mercado mundial. Para alcanzar este objetivo se incursiona en la aplicación de métricas, las cuales además de satisfacer necesidades de información, ofrecen una guía valiosa para el equipo de trabajo de cómo es el comportamiento de los procesos y productos de trabajo.

En el presente capítulo se abordará el marco teórico de la investigación, incluyendo además el marco conceptual sobre temas como DSBC, calidad de software, las características de calidad que definen los productos de este tipo, particularmente la funcionalidad. Se realizará una reseña de casi todos los modelos y normas que ofrecen guías de evaluación de la calidad y dentro de estos se encuentran los estándares ISO/IEC 9126 e ISO/IEC 14598, los cuales serán la base fundamental en que se apoyará este trabajo, en busca de lograr medir funcionalmente los componentes software que se desarrollan en el CEDIN.

### **1.1 Desarrollo de Software Basado en Componentes**

Durante muchos años el mundo ha sido testigo de los grandes y sorprendentes avances de la tecnología, siendo tan importantes e indispensables en la actualidad. La mayoría de dichos avances tecnológicos se han logrado gracias al software, ya que este ha permitido facilitar y agilizar los procesos.

Como resultado de presiones crecientes sobre la industria del software orientadas a reducir drásticamente el costo y tiempo de desarrollo de sistemas y aplicaciones, sin afectar los niveles de

calidad del producto, ha surgido un nuevo activo reutilizable denominado componente de software (3).

Existen varias definiciones de componentes realizadas por especialistas que han sido los facultados del desarrollo de esta metodología, ellos han tomado como base la metodología de la programación orientada objetos y el modelado a través de UML, estas son las siguientes:

- Un componente es una parte no trivial, casi independiente y reemplazable de un sistema que llena claramente una funcionalidad dentro de un contexto en una arquitectura bien definida. Un componente se conforma y provee la realización física por medio de un conjunto de interfaces (3).
- Un componente de software en tiempo de ejecución es un paquete dinámicamente vinculado con uno o varios programas manejados como una unidad y que son accedidos mediante interfaces bien documentadas que pueden ser descubiertos en tiempo de ejecución (4).
- Un componente de software es una unidad de composición con interfaces contractualmente especificadas y explícitas sólo con dependencias dentro de un contexto. Un componente de software puede ser desplegado independientemente y es sujeto a la composición de terceros (3).
- Un Componente de Negocio representa la implementación de software del concepto de un negocio “autónomo” o un proceso de negocio. Que consiste de artefactos de software necesarios para expresar, implementar y poner en marcha el concepto de elemento reusable de un sistema más grande de negocios (4).

En esencia estas definiciones expresan que los componentes software son módulos independientes con interfaces bien definidas que pueden ser vinculados a otros componentes.

Las características fundamentales para que un elemento pueda ser clasificado como componente son:

- **Identificable:** Debe tener una identificación que permita acceder fácilmente a sus servicios y que permita su clasificación (4).
- **Auto contenido:** Un componente no debe requerir de la utilización de otros para finalizar la función para la cual fue diseñado (4).

- Puede ser **reemplazado** por otro componente: Se puede reemplazar por nuevas versiones u otro componente que lo reemplace y mejore (4).
- Con **acceso** solamente a través de su interfaz: Debe asegurar que estas no cambiaran a lo largo de su implementación (4).
- **Sus servicios no varían**: Las funcionalidades ofrecidas en su interfaz no deben variar, pero su implementación sí (4).
- **Bien Documentado**: Un componente debe estar correctamente documentado para facilitar su búsqueda si se quiere actualizar, integrar con otros, adaptarlo, etc (4).
- **Es genérico**: Sus servicios debe servir para varias aplicaciones (4).
- **Reutilizado dinámicamente**: Puede ser cargado en tiempo de ejecución en una aplicación. Independiente de la plataforma: Hardware, Software, S.O (4).

El DSBC es una aproximación del desarrollo de software que describe, construye y utiliza técnicas de software para la elaboración de sistemas abiertos y distribuidos mediante el ensamblaje de partes software reutilizables. La aproximación DSBC es utilizada para reducir los costes, tiempos y esfuerzos de desarrollo del software, a la vez que ayuda a mejorar la fiabilidad, flexibilidad y la reutilización de la aplicación final (5).

Otro punto a tomar en cuenta es que el DSBC, pertenece al paradigma de programación de sistemas abiertos, los cuales son extensibles y tienen una interacción con componentes heterogéneos que ingresan o abandonan el sistema de forma dinámica, es decir que los componentes pueden ser reemplazados, por otros independientemente de su arquitectura y desarrollo (4).

La UCI no está al margen de este acontecimiento y ya es historia este término, específicamente en la Facultad 5, en la cual varios de sus proyectos, desarrollan sus productos mediante la metodología de DSBC.

### **1.2 Clasificación de los componentes software**

Los componentes software se clasifican en cuatro grandes grupos, los cuales a su vez contemplan

otras clasificaciones, estas son las siguientes (6):

### 1. Según su modificabilidad

- Caja negra
- Caja blanca

### 2. Según su granularidad

- Componentes de uso específico
- Componentes de negocio
- Marcos (Frameworks)
- Componentes de aplicación

### 3. Según su fabricante

- Componentes hechos en casa
- Commercial off-the-shelf (COTS)

### 4. Según la tecnología usada

- Componentes imperativos (Módulos, Funciones)
- Componentes Orientados a objetos (Clases)
- Componentes Distribuidos(Componentes CORBA, .NET, J2EE, Servicios Web)

En el CEDIN se desarrollan componentes según la tecnología usada de tipo Orientados a Objetos.

### Componentes Orientados a Objetos (Clases)

El Paradigma de Programación Orientado a Objetos (POO) es una técnica de programación que utiliza objetos, propiedades y métodos para diseñar aplicaciones.

La POO expresa un programa como un conjunto de objetos, que interactúan entre ellos para realizar tareas.

Un objeto es una unidad que contiene datos y las funciones que operan sobre esos datos.

Los datos que están contenidos dentro de un objeto se denominan atributos, definiéndose como las características o propiedades que posee un objeto.

Los métodos los constituyen las operaciones o procedimientos que operan sobre los atributos de los objetos.

Una clase es una colección de objetos similares y un objeto es una particularización de una definición de una clase.

Una clase es un tipo de dato definido por el usuario que determina las estructuras de datos y las operaciones asociadas con ese tipo. Las clases son las definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas. Cada vez que se construye un objeto de una clase, se crea una instancia de esa clase. En general, los términos objetos e instancias de una clase se pueden utilizar indistintamente.

### **1.3 ¿Qué es calidad?**

Como se explicó anteriormente el CEDIN desarrolla software basado en componentes software reutilizables, siendo tan favorable para su institución, pero se están requiriendo técnicas o métodos que ayuden a determinar en qué medida estos cuentan con la debida calidad, ya que estos podrán ser utilizados para cualquier otro producto que los necesite.

El Diccionario de la Real Academia Española, conceptúa la calidad como la "propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor" y es sinónimo de "buena calidad" la "superioridad o excelencia" (7).

La ISO (International Organization for Standardization) 8402, agrega otros aspectos importantes cuando define la calidad como el conjunto de propiedades y características de un producto o servicio, que le confieren aptitud para satisfacer unas necesidades expresadas o implícitas (7).

Por otra parte expertos en el tema definen la calidad de la siguiente forma:

**Philip Crosby:** "Calidad es cumplimiento de requisitos" (8)

**Joseph Jurán:** "Calidad es adecuación al uso del cliente" (8)

**Armand V. Feigenbaum:** "Satisfacción de las expectativas del cliente" (8)

**Walter A. Shewhart:** "La calidad como resultado de la interacción de dos dimensiones: dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece)" (8)

Estas definiciones revelan que calidad, resume las características, propiedades, cualidades y en general atributos propios de un producto, que determinan sobre este la ausencia de defectos y la conformidad de todo el personal que de una forma y otra se vinculan con él.

### **1.3.1 Calidad del software**

Existe gran cantidad de definiciones de calidad del software, ya que es un concepto complejo y por tanto difícil de definir, algunas de estas son:

#### ***Pressman 1992:***

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 (9).

#### ***Pressman 1998:***

Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario (10).

#### ***Norma IEEE 1601:***

La calidad de un producto de software se define como el grado que posee una combinación adecuada de determinadas características como son el rendimiento, la fiabilidad o la seguridad (11).

De acuerdo a lo planteado se puede deducir que calidad es desarrollar software cumpliendo con los requisitos y las normas establecidas para lograr satisfacer las necesidades del cliente, ya que los requisitos del software son la base de las medidas de calidad, por lo que si el producto no está en función de satisfacerlos entonces no contará con la debida calidad; y que las normas o estándares determinan un conjunto de procedimientos o procesos de desarrollo que rigen la forma de emplear la ingeniería del software para que este sea lo más coherente y fiable posible.

### **1.4 Modelos y normas de calidad de productos software**

Un modelo de calidad es “El conjunto de características y las relaciones entre ellas que proveen la base para la especificación de los requisitos de calidad y la evaluación de la calidad” (12). Los cuales

permiten tener una definición estructurada de criterios de evaluación, una especificación de requisitos con relación a ellos, descripción de componentes en un marco común y definición de métricas y prioridades.

Los modelos y normas de calidad de los productos software más reconocidos propuestos hasta el momento son:

1. *Modelo de McCall*: El modelo fue propuesto por McCall en 1977 y está orientado a los desarrolladores de sistemas, para ser utilizado durante el proceso de desarrollo, en el que describe la calidad como un concepto elaborado mediante relaciones jerárquicas entre factores de calidad, en base a criterios y métricas de calidad (13) (ver figura 1).

Factor	Criterio	Factor	Criterio
Correctitud	Rastreabilidad	Mantenibilidad	Simplicidad
	Complejidad		Concreción
	Consistencia	Capacidad de Prueba	Simplicidad
Confiabilidad	Consistencia		Instrumentación
	Exactitud		Auto-descriptividad
	Tolerancia a fallas		Modularidad
Eficiencia	Eficiencia de ejecución	Flexibilidad	Auto-descriptividad
	Eficiencia de almacenamiento		Capacidad de expansión
Integridad	Control de acceso		Generalidad
	Auditoría de acceso		Modularidad
Usabilidad	Operabilidad	Portabilidad	Auto-descriptividad
	Entrenamiento		Independencia del sistema
	Comunicación		Independencia de máquina
Interoperabilidad	Modularidad	Reusabilidad	Auto-descriptividad
	Similitud de comunicación		Generalidad
	Similitud de datos.		Modularidad
			Independencia de máquina

Figura 1: Criterios asociados a los factores de calidad. McCall

2. *Modelo de Dromey*: En 1996 Dromey sugiere una técnica genérica para construir un modelo de calidad determinada por los componentes del mismo (incluyendo documentos de requerimientos, guías de usuarios, diseños y código). Sugiere el uso de cuatro categorías que implican propiedades de calidad, que son: correctitud (corrección), internas, contextuales y descriptivas (13).

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

Figura 2: Criterios asociados a los factores de calidad. Dromey

1. *Modelo FURPS*: Hewlett-Packard lo presenta en 1987, desarrollando un conjunto de factores de calidad de software y sus respectivos atributos.

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

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

Figura 3: Criterios asociados a los factores de calidad. FURPS

2. *Modelo de Boehm*: Presentado por Boehm en 1978 (Pfleeger, 1998), incluye las necesidades de los usuarios, como lo hace McCall; sin embargo, incluye características de rendimiento de hardware que no se encuentran en el modelo de McCall.

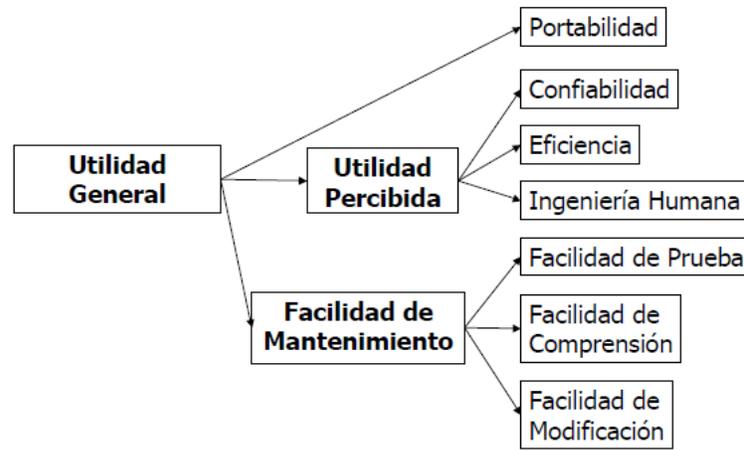


Figura 4: Modelo de Boehm

3. *Norma ISO/IEC 9126*: Esta norma define las características de calidad como un conjunto de atributos del producto de software a través de los cuales la calidad es descrita y evaluada. Estas características de calidad del software pueden ser precisadas a través de múltiples niveles de subcaracterística (ver figura 4).

Características	Atributos
Funcionalidad	Adaptabilidad Exactitud Interoperabilidad Seguridad
Usabilidad	Comprensibilidad Aprendizaje Operabilidad Atractivo
Mantenimiento	Análisis Cambio Estabilidad Prueba
Confiabilidad	Madurez Tolerancia a cambio Recuperabilidad
Eficiencia	Comportamiento del tiempo Uso de los recursos
Portabilidad	Adaptabilidad Instalación Coexistencia Reemplazo

Figura 5: Criterios asociados a los factores de calidad. ISO/IEC 9126

4. *Norma ISO-25000*: fue creada con el objetivo de organizar y unificar las series que cubren dos procesos principales: especificación de requerimientos de calidad del software y evaluación de la calidad del software, soportada por el proceso de medición de calidad del software. Las

características de calidad y sus mediciones asociadas pueden ser útiles no solamente para evaluar el producto software sino también para definir los requerimientos de calidad. Esta norma reemplaza a dos estándares relacionados: ISO/IEC 9126 e ISO/IEC 14598 (14).

### **1.5 Evaluación de la calidad de los productos de software**

La norma ISO/IEC 14598 se encarga de la evaluación de los productos de software, estableciendo un proceso de evaluación para desarrolladores, arquitectos y evaluadores, a partir de cuatro subprocesos básicos (ver figura 9), estos son:

1. Establecer los requisitos de la evaluación.
2. Especificar la evaluación.
3. Diseñar la evaluación.
4. Ejecutar la evaluación.

La norma brinda amplios conceptos de evaluación y a pesar de ser estos tan generales que no quedan bien definidos algunos elementos, por ejemplo los relacionados con Establecer criterios de valoración y Valorar resultados, será una valiosa guía en la que se apoyará esta investigación para definir su proceso de evaluación.

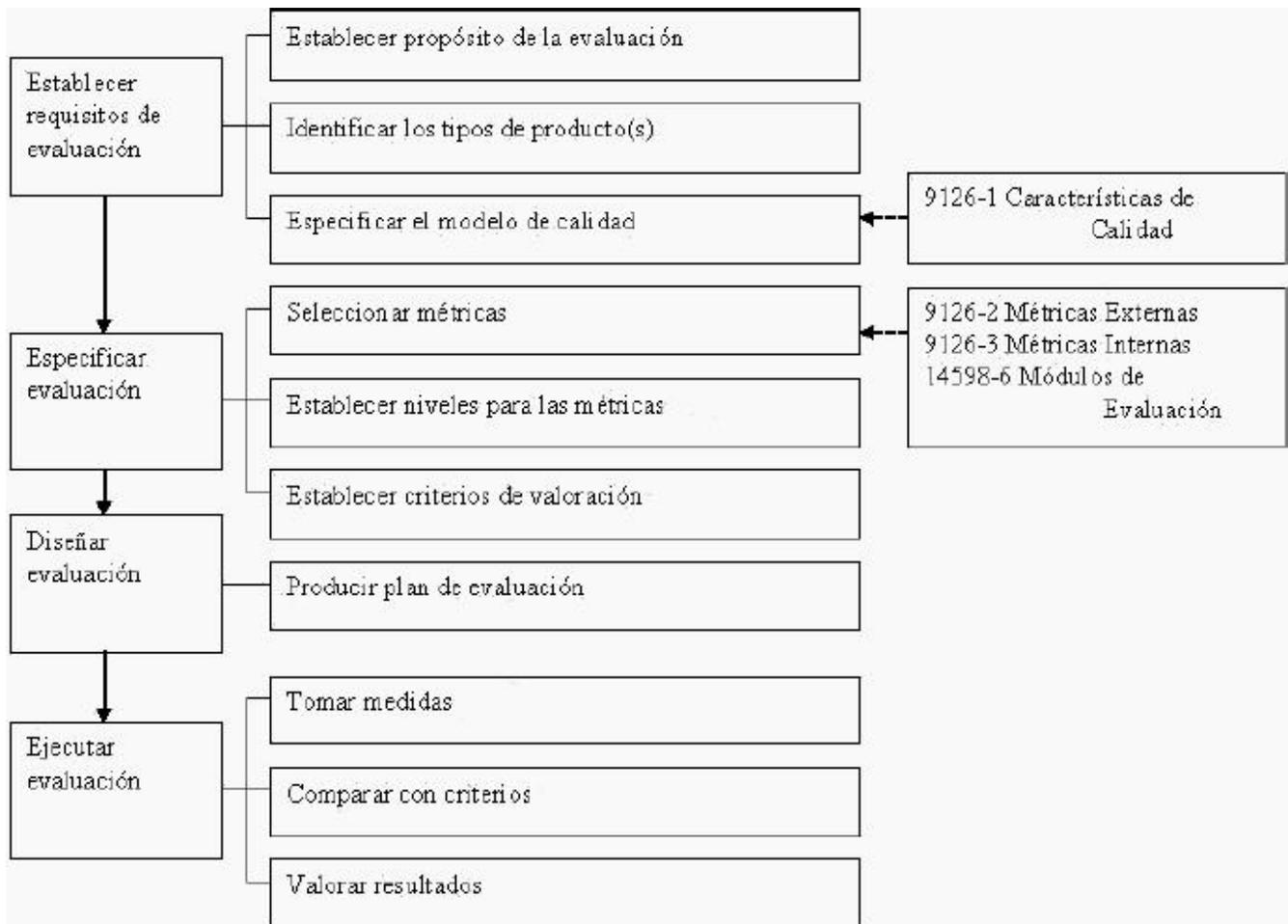


Figura 6: Proceso de evaluación según la norma 14598

## 1.6 Descripción de las normas a utilizar

En los apartados anteriores se mencionaron propuestas que se han realizado con el objetivo de tener una guía para obtener la calidad requerida del software, pero no existe consenso a la hora de definir y clasificar las características de calidad que debe presentar un producto software, por tanto, este trabajo diploma se basará en los estándares internacionales, específicamente en el ISO 9126 y el ISO 15498, ya que identifican los atributos claves de calidad para un producto de software, además aportan una valiosa base para medidas indirectas y una excelente lista para determinar la calidad de un sistema, así como el método de evaluación de dichos productos, los cuales se adaptarán a los

condiciones de los componentes software que se desarrollan en el CEDIN, para finalmente lograr evaluar la calidad funcional de estos.

Es conveniente aclarar una vez más que no se medirá un software, sino componentes software, por lo que se realizará una abstracción de las normas seleccionadas y se extraerán los elementos fundamentales y convenientes que permitirán confeccionar el procedimiento que será aplicado en el centro.

El estándar ISO/IEC 9126 está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso, como se muestra en la tabla 1.

Estándar	Objetivo
ISO/IEC 9126-1	Modelo de calidad
ISO/IEC 9126-2	Métricas externas
ISO/IEC 9126-3	Métricas internas
ISO/IEC 9126-4	Métricas de calidad en uso

Tabla 1: Familia del estándar ISO/IEC 9126

Por su parte la ISO/IEC 14598 guía el proceso de valoración de la calidad del software según los criterios de la ISO/IEC 9126, dedicándose a la evaluación de dichos productos. Cuenta con seis partes, las cuales son:

Parte 1: Visión General

Parte 2: Planificación y Gestión

Parte 3: El Proceso para Desarrolladores

Parte 4: El Proceso para Compradores

Parte 5: El Proceso para Evaluadores

Parte 6: Documentación de los Módulos de Evaluación

*NOTA:* El estándar ISO/IEC 9126 también define un proceso para evaluar la calidad del software.

### 1.7 Modelo para la calidad

Los parámetros que permiten establecer los niveles mínimos que el producto debe alcanzar, los ofrecerá la norma ISO 9126 en su primera parte, clasificando la calidad de productos software en características, las que se definen como un conjunto de propiedades mediante las cuales se evalúa y describe su calidad (15). Una característica se puede refinar en múltiples niveles de subcaracterísticas, estas son funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. A continuación se muestran las características, con sus subcaracterísticas:

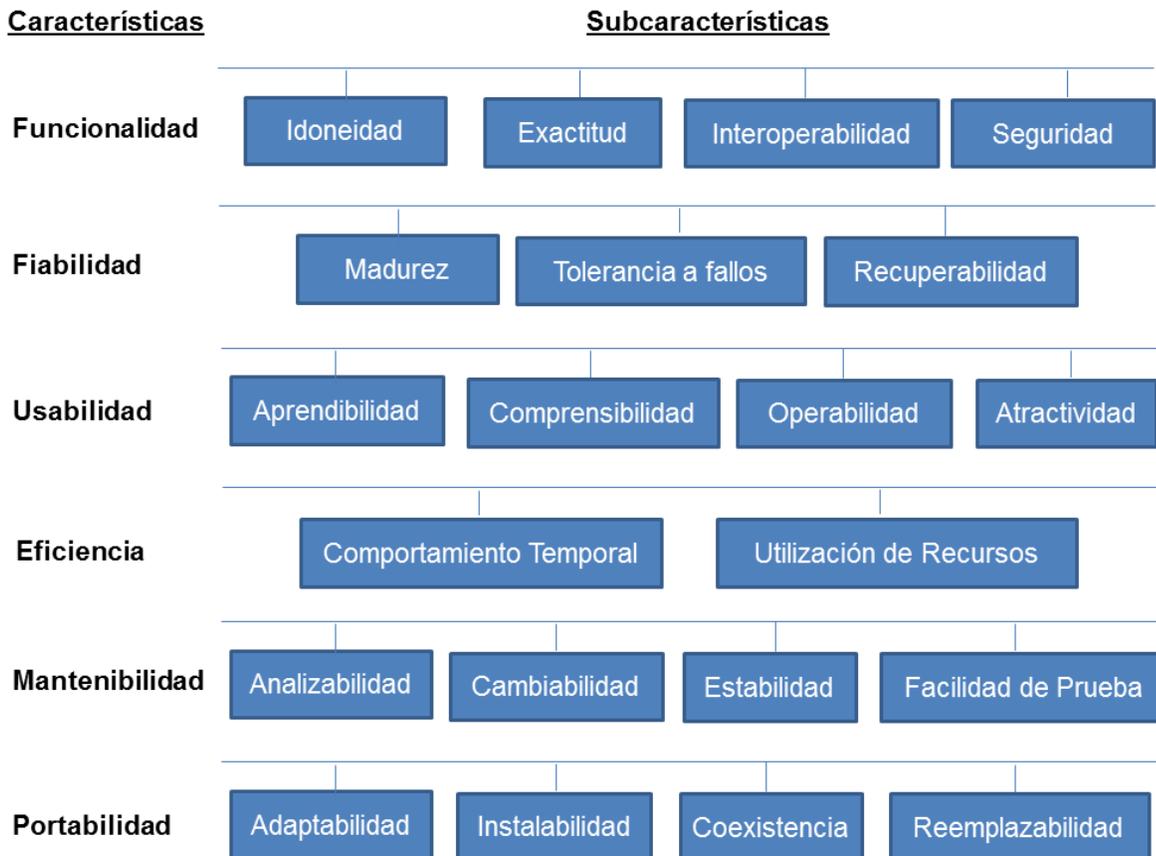


Figura 7: Modelo de calidad según la ISO/IEC 9126

### 1.8 Funcionalidad

Basándose en el objeto de estudio de esta investigación, este trabajo se centrará en una sola característica de las que se explicaron en el epígrafe anterior, la cual es funcionalidad.

Esta característica está relacionada con lo que hace el software para satisfacer las necesidades, tanto las descritas como las asumidas por el equipo de trabajo. Se divide en las subcaracterísticas: idoneidad, precisión, interoperabilidad y seguridad.

A continuación se explicará cada una de las subcaracterísticas de la funcionalidad para una mejor comprensión de las mismas.

- La **idoneidad** indica si están presentes las funciones que soportan las tareas especificadas (16).
- La **precisión** muestra el grado de exactitud de los efectos del sistema (16).
- La **interoperabilidad** demuestra si el sistema puede interactuar con otros sistemas (16).
- La **seguridad** indica el grado en que un acceso no autorizado (accidental o deliberado) se prevenga y se permita un acceso autorizado (16).

Esta característica mantiene el mismo sentido para los componentes que para un producto software, por lo que sobre estos conceptos se basará este trabajo para realizar la evaluación de la funcionalidad de los componentes.

La funcionalidad es uno de los factores más importante de los componentes y/o productos software, ya que es donde evaluamos verdaderamente si estos cumplen con las funciones para los cuales fueron diseñados.

### **1.9 Métricas de software**

Desde hace algunos años las métricas han cobrado gran importancia en las empresas desarrolladoras de software, reconociendo las ventajas que tienen las mediciones para cuantificar y por consiguiente gestionar de forma más efectiva la calidad de los productos de software.

Sin embargo existen grandes dificultades para realizar métricas óptimas, debido a diferentes situaciones, así como todavía existen empresas que no tienen una cultura adecuada sobre la medición, desconociendo el alcance de calidad que pudiera conseguir el producto final si este se midiera.

Una métrica se define como una medida del grado en que un sistema, componente o proceso posee un atributo dado, mientras que medida proporciona una indicación cuantitativa de la cantidad, dimensiones o tamaño de algunos atributos de un producto; y medición es el acto de determinar una medida (17).

Entendiendo por atributo a una propiedad de calidad a la que puede asignarse una métrica (18).

Las métricas externas miden el software en sí mismo, mientras que las métricas internas evalúan el comportamiento del sistema.

Las métricas ayudan a entender tanto el proceso que se realiza durante el desarrollo del software como el producto al final de su implementación y ejecución, por tanto esto facilitará en caso de algún problema tomar alguna acción correctiva que mejore las condiciones siempre que sea posible, con el objetivo de mejorar la calidad del producto.

La medición constituye una disciplina fundamental en cualquier ingeniería, ya que a pesar de ser esta aún reciente ha dado pasos gigantescos para lograr incrementar la calidad y la productividad en el desarrollo y mantenimiento del software, ya que permite controlar qué es lo que sucede en los proyectos y prever su esfuerzo y duración.

### **1.10 Técnicas para evaluar la funcionalidad de productos software**

Mundialmente se han desarrollado numerosas técnicas que tienen como objetivo evaluar la calidad de los productos software, pero tienen la desventaja que son privativas, las más relevantes se mencionarán a continuación:

- La Herramienta computacional para la evaluación de calidad de productos software enmarcados en actividades de investigación: es una herramienta desarrollada en Visual Basic que utiliza las directrices dadas por los estándares ISO 14598, ISO 9126 e ISO 25000. Actualmente se está utilizado en la evaluación de la calidad de desarrollos terminados en diferentes grupos de investigación de Colombia, como el Grupo de Investigación en Sistemas y Tecnologías de la Información y el de Ingeniería Biomédica, permitiéndoles conocer el grado de calidad de sus desarrollos y así plantear oportunidades de mejora.

- El Modelo Sistémico de Calidad (MOSCA): este es un modelo con su herramienta desarrollado en el Laboratorio de Investigación en Sistemas de Información (LISI) de la Universidad Simón Bolívar de Venezuela, para evaluar la calidad de los sistemas software, que integra el modelo de calidad del producto y el modelo de calidad del proceso de desarrollo. El modelo de calidad enfocado hacia el producto enlaza las ideas del estándar ISO/IEC 9126 y del Modelo de Dromey
- El modelo MECHDAV: con su herramienta para la evaluación técnica de la calidad de las herramientas RAD (Rapid Applications Development) para ambientes visuales, basado en el cumplimiento de las características de calidad del software propuestas en el estándar ISO/IEC 9126.
- La metodología Mede-Pros: surge del Centro de Tecnología de la Información Renato Archer de Sao Paulo - Brasil, para contribuir con la mejoría de la calidad de productos de software, basado en las ISO/IEC 9126, ISO/IEC 12119 e ISO/IEC 14598, el cual obtuvo Premio ASSESPRO (Premio al Mejor Software del Año) y el Llamado Nacional SOFTEX de Sociedad para la Promoción de la excelencia del software brasileño.
- Metodología producto OK!: fue desarrollada por el Instituto de Estudios Económicos en Software en Brasil con el objetivo principal de conocer el nivel de satisfacción de los clientes con el producto evaluado y con la empresa desarrolladora.

### **Conclusiones del capítulo**

Contar con la exactitud, idoneidad, interoperabilidad y seguridad de los componentes software beneficiará el trabajo que se desarrolla en el CEDIN, tanto a nivel de procesos como de productos, ya que a pesar de contar con modelos de calidad estos no son suficientes para evaluar objetivamente el avance y terminación de los productos, hasta incluso muchas veces conlleva a confiarse en el proceso de desarrollo de los componentes, obviando aspectos extra-funcionales que determinan las exigencias que deben ser alcanzada.

En este capítulo se definieron conceptos relacionados a la calidad y medición de software, haciendo mención de normas y estándares de calidad, destacando las características cualitativas del software, en específico de la funcionalidad. Se fundamentó sobre los temas de desarrollo de software basado en la utilización de componentes reutilizables. Se mencionaron diferentes técnicas internacionales encargadas de evaluar la calidad del software. Todos estos aspectos fundamentales que aportan una valiosa base de información y sugerencias que se materializarán en la propuesta del procedimiento para evaluar la calidad funcional de los componentes software del CEDIN.

# CAPÍTULO 2

## *Procedimiento para evaluar la funcionalidad de Componentes Software*

### **Introducción**

Cada vez más cobra auge el desarrollo de aplicaciones basadas en componentes, por tanto aumenta la necesidad de disponer de métodos y procedimientos que permitan valorar su calidad, específicamente la calidad funcional.

El objetivo del presente capítulo es definir un procedimiento, funcional y fácil de aplicar para evaluar componentes software, que se pueda usar para describir y valorar las subcaracterísticas específicas de la funcionalidad y que puedan ayudar en el proceso de selección, mejora y comparación de componentes. Se explica detalladamente el procedimiento propuesto y se describe la estructura del mismo, así como las actividades y artefactos que se utilizan, también se define un modelo para la evaluación de calidad funcional que recolecta los datos y resultados fundamentales en la evaluación.

### **2.1 Roles y responsabilidades de los participantes en el proceso de evaluación**

Los participantes en el proceso de evaluación de la característica funcionalidad son los roles que pertenecen al área de calidad, estos son los administradores de calidad de los proyectos del CEDIN y los revisores líderes.

A continuación se muestra una tabla con el rol y la responsabilidad de dichos ejecutores como se tiene definido en el documento 0516\_Roles y Responsabilidades del programa de mejora, que se encuentra publicado en <http://calisoft.uci.cu>, y adaptadas durante esta investigación a las responsabilidades que deben tener estos dos roles para una correcta evaluación de la característica funcionalidad.

Rol	Responsabilidad
<b>Administrador de la calidad</b>	<ul style="list-style-type: none"><li>• Elabora el Plan de Aseguramiento de la calidad donde incluye, cómo evaluar la característica funcionalidad de un componente software.</li><li>• Elabora el Plan de Mediciones incluyendo métricas para evaluar la idoneidad, exactitud, interoperabilidad y seguridad.</li><li>• Participa en la elaboración del Plan de Monitoreo y en el monitoreo y análisis de las áreas de procesos según la fase en la que se encuentre el componente verificando cómo se comporta la funcionalidad del mismo.</li><li>• Participa en el análisis y recolección de los datos para aplicar las métricas de la característica de funcionalidad.</li><li>• Colabora en las auditorías que se les realicen al proyecto comprobando si el componente cumple con las características de funcionalidad.</li><li>• Crea una cultura de calidad especificando la importancia que tiene la funcionalidad para los componentes que se desarrollen en el CEDIN.</li></ul>
<b>Revisor Líder</b>	<ul style="list-style-type: none"><li>• Planificar y organizar el trabajo de equipo de revisores.</li><li>• Llevan a cabo la revisión dentro del horario acordado.</li><li>• Verifican la exactitud de la información recopilada.</li><li>• Confirman que la evidencia es apropiada para apoyar las No Conformidades.</li><li>• Evalúan aquellos factores que puedan afectar la fiabilidad de las No Conformidades y conclusiones.</li></ul>

Tabla 2: Roles y responsabilidades de los evaluadores

Los administradores de calidad son estudiantes seleccionados para desempeñar ese rol en cada uno de los proyectos de la facultad, mientras que los revisores líderes son profesores con experiencia en el trabajo de calidad, encargados de dirigir y orientar el trabajo de los administradores de calidad.

## **2.2 Métricas Funcionales. Conceptos**

El procedimiento propuesto consta de métricas externas e internas que permitirán evaluar la característica funcionalidad de los componentes. A continuación se definen como deben ser las métricas para cada subcaracterística funcional.

Las métricas externas para la medición de la característica funcionalidad deben ser capaces de medir un atributo como es el comportamiento funcional de los componentes, mientras que las métricas internas se utilizan para predecir si el componente satisface los requisitos funcionales establecidos y las necesidades implícitas de usuario.

Las métricas de idoneidad por su parte deben medir un atributo como es la ocurrencia de un funcionamiento u operación insatisfactoria, que no se realice de forma como se definió en la Especificación de Requisitos, o no sea razonablemente aceptable.

Las métricas de precisión medirán un atributo como es la frecuencia con que el usuario se encuentre con la ocurrencia de una falta de exactitud o de precisión, como puede ser: resultados incorrectos o imprecisos causados por datos inadecuados; por ejemplo, un dato con pocos dígitos significativos para un cálculo de precisión.

Por su parte las métricas de interoperabilidad evaluarán un atributo como es el número de funciones o la ocurrencia de la menor incomunicación que involucre a datos y comandos o instrucciones que sean transferidos entre los componentes que estén conectados.

Y las métricas de seguridad deben ser capaces de medir un atributo como el número de funciones o problemas de seguridad como: falla en la seguridad de salida de información o datos, falla en la prevención de pérdida de datos y fallas en denegar accesos ilegales u operaciones no permitidas.

## **2.3 Métricas de funcionalidad**

En la norma ISO/IEC 9126 se definen métricas internas y externas para cada atributo de la característica funcionalidad, después de haber realizado un análisis de cada una de ellas se escogieron las métricas necesarias, suficientes y más importantes para una correcta evaluación subjetiva de las

subcaracterísticas, se tuvieron en cuenta las necesidades específicas de la evaluación y que estas fueran lo suficientemente precisas para permitir realizar comparaciones fiables entre los atributos.

En los siguientes sub-epígrafes se realiza una descripción de cada una de estas métricas organizados por subcaracterísticas; a continuación se muestran la estructura común que se planteó para todas las métricas:

- a) Nombre de la métrica: Denominación de la métrica.
- b) La métrica se propone medir: Se expresa como la interrogante a ser respondida por la aplicación de la métrica.
- c) Método de aplicación: Provee una secuencia de pasos para la aplicación de la métrica.
- d) Medición (fórmula): Provee la fórmula de medición y el significado de los datos empleados.
- e) Interpretación del valor obtenido: Provee valores preferidos y su delimitación.
- f) Escala: Tipo de escala usada por la métrica. Los tipos son: escala valorativa, y escala absoluta (ejemplo número de líneas de código).

### 2.3.1 Métricas del atributo idoneidad

Estas métricas les indicarán al evaluador cuán adecuada, completa y correcta ha sido la implementación de las funcionalidades de los componentes a evaluar, permitiéndole establecer un criterio cuantitativo y cualitativo sobre la idoneidad de dichos componentes.

#### 1. Adecuación funcional

- a) Nombre de la métrica: Adecuación funcional
- b) La métrica se propone medir: ¿Cuán adecuada es la función evaluada?
- c) Método de aplicación: Contar el número de funciones que son adecuadas para realizar las tareas específicas y compararlas con el número de funciones evaluadas.
- d) Medición (fórmula):  $X = 1 - A/B$ 

A - Número de funciones en las cuales se detectaron problemas en la evaluación: estas son el número de exámenes que dieron lugar a No Conformidades cuando se evaluó el componente.

B - Número de funciones evaluadas: estas constituyen la cantidad total de los exámenes que se realizaron al componente según el documento Modelo de casos de uso del sistema.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mejor.

f) Escala: Absoluta

## **2. Completitud de la implementación funcional**

a) Nombre de la métrica: Completitud de la implementación funcional

b) La métrica se propone medir: ¿Cuán completa ha sido la implementación y su conformidad con la especificación de requisitos?

c) Método de aplicación: Contar el número de funciones que faltan detectadas en la evaluación y compararlas con el número de la función descrita en las especificaciones de requisitos

d) Medición (fórmula):  $X = 1 - A/B$

A - Número de funciones perdidas detectadas en la evaluación: constituyen la cantidad de requisitos funcionales que no se implementaron en el componente.

B - Número de funciones descritas en la especificación de requisitos: es la cantidad total de requisitos funcionales descritos en el documento Especificación de requisitos del componente.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mejor.

f) Escala: Absoluta

## **3. Cobertura de la implementación funcional**

a) Nombre de la métrica: Cobertura de la implementación funcional

b) La métrica se propone medir: ¿Cuán correcta ha sido la implementación funcional?

c) Método de aplicación: Contar el número de funciones realizadas incorrectamente o no detectadas en la evaluación y compararlas con el número total de funciones que se describen en las especificaciones de requisitos.

Contar el número de funciones que son completas en comparación con las que no lo son.

d) Medición (fórmula):  $X = 1 - A/B$

A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas: es la cantidad de requisitos funcionales que se implementaron incorrectamente y los que no se llegaron a implementar de los descritos en el documento Especificación de requisitos.

B - Número de funciones descritas en la especificación de requisitos: es la cantidad total de requisitos funcionales descritos en el documento Especificación de requisitos del componente.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mejor.

f) Escala: Absoluta

### **2.3.2 Métrica del atributo exactitud**

Estas métricas les mostrarán al evaluador con qué frecuencia se encuentran cálculos y resultados inexactos en los componentes evaluados, permitiéndole establecer un criterio cuantitativo y cualitativo sobre la exactitud de dichos componentes.

#### **1. Exactitud de los cálculos**

a) Nombre de la métrica: Exactitud de los cálculos

b) La métrica se propone medir: ¿Con qué frecuencia los evaluadores encuentran resultados incorrectos?

c) Método de aplicación: Registre el número de cálculos incorrectos basado en las especificaciones.

d) Medición (fórmula):  $X = A / T$

A - Número de cálculos incorrectos encontrados por los evaluadores: constituyen la cantidad de cálculos diferentes de cómo se describe en los documentos Modelo de casos de uso del sistema y Especificación de requisitos.

T - Tiempo de operación: es el tiempo que se demora el sistema en realizar el cálculo.

e) Interpretación del valor obtenido:  $0 \leq X$

A mayor cercanía al 0 resultará mejor.

f) Escala: Valorativa.

#### **2. Precisión**

a) Nombre de la métrica: Precisión

b) La métrica se propone medir: ¿Con qué frecuencia los evaluadores encuentran resultados con la precisión inadecuada?

c) Método de aplicación: Registre el número de resultados con una precisión inadecuada.

d) Medición (fórmula):  $X = A / T$

A - Número de resultados incorrectos encontrados por los evaluadores con un nivel de precisión diferente del requerido: constituyen la cantidad de resultados diferentes de cómo se describe en los documentos Modelo de casos de uso del sistema y Especificación de requisitos.

T - Tiempo de operación: es el tiempo que se demora el sistema en mostrar los resultados.

e) Interpretación del valor obtenido:  $0 \leq X$

A mayor cercanía al 0 resultará mejor.

f) Escala: Valorativa.

### **2.3.3 Métricas del atributo interoperabilidad**

Con la aplicación de estas métricas el evaluador conocerá si se establece un correcto intercambio de datos entre los componentes y si son correctas las funciones de interfaces para una transferencia de datos específica, facilitándole al evaluador establecer un criterio cuantitativo y cualitativo sobre la interoperabilidad de los componentes.

#### **1. Intercambiabilidad de datos, en base su formato**

a) Nombre de la métrica: Intercambiabilidad de datos, en base su formato.

b) La métrica se propone medir: ¿Cuán correctamente ha sido implementado el intercambio de funciones de interfaces para una transferencia de datos específica?

c) Método de aplicación: Examinar cada registro de salida de las funciones de interfaces de acuerdo con la especificación de los campos de datos. Contar el número de formatos de datos que deben ser intercambiados con otros componentes en comparación con el número total.

d) Medición (fórmula):  $X = A / B$

A - Número de formatos de datos intercambiados exitosamente con otros componentes durante las pruebas del intercambio de datos: es la cantidad total de formatos de datos intercambiados exitosamente con otros componentes según el documento plantilla de Especificaciones de campos del componente.

B - Número total de formatos de datos a intercambiar: representa la cantidad total de formatos de datos a intercambiar según la plantilla de Especificaciones de campos.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mayor intercambiabilidad.

f) Escala: Absoluta.

## **2. Intercambiabilidad de datos, en base al éxito del intento**

a) Nombre de la métrica: Intercambiabilidad de datos, en base al éxito del intento.

b) La métrica se propone medir: ¿Cuán frecuentemente falló el intento de intercambio de datos entre el componente objeto de la evaluación y otros componentes?

c) Método de aplicación: Contar el número de casos en que las funciones de interfaces fueron usadas y fallaron.

d) Medición (fórmula):  $X = 1 - A / B$

A - Número de casos en que se falló al proceder un intercambio de datos con otros componentes: cantidad de veces en que falló el intercambio de datos entre el componente evaluado y otros que se relacionen con él.

B - Número de casos en que se intentó proceder a un intercambio de datos: cantidad total de intercambio de datos entre el componente evaluado y otros que se relacionen con él.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mejor.

f) Escala: Absoluta

### **2.3.4 Métricas del atributo seguridad**

Con la aplicación de estas métricas se conocerá si el sistema posee la capacidad de control de acceso a los datos y de los usuarios al sistema, permitiéndole al evaluador establecer un criterio cuantitativo y cualitativo sobre la exactitud de dichos componentes.

#### **1. Capacidad de revisión de cuentas de acceso**

a) Nombre de la métrica: Capacidad de revisión de cuentas de acceso.

b) La métrica se propone medir: ¿Cuán completa es el rastro de auditoría del acceso del usuario al componente y a los datos?

c) Método de aplicación: Evaluar la cantidad de accesos que el componente guardó en la base de datos histórica de acceso.

d) Medición (fórmula):  $X=A/B$

A - Número de acceso del usuario al componente y a los datos registrados en la base de datos del historial: es la cantidad total de accesos del evaluador guardados en la base de datos de accesos de los usuarios durante la evaluación.

B - Número de accesos del usuario al componente y a los datos durante la evaluación: es la cantidad total de accesos aplicados al componente durante la evaluación.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mejor.

f) Escala: Absoluta

## **2. Control del acceso.**

a) Nombre de la métrica: Control de acceso.

b) La métrica se propone medir: ¿Cuán controlable es el acceso al componente?

c) Método de aplicación: Contar el número de operaciones ilegales detectadas en comparación con el número de operaciones ilegales, como en las especificaciones

d) Medición (fórmula):  $X=A/B$

A - Número de las diferentes operaciones ilegales detectadas: es la cantidad de operaciones ilegales que el componente rechaza o no permitió durante la evaluación.

B - Número de operaciones ilegales como en la especificación: es la cantidad total de operaciones ilegales intentadas para acceder al componente o a los datos del mismo.

e) Interpretación del valor obtenido:  $0 \leq X \leq 1$

Nota: A mayor cercanía al 1 resultará mejor.

f) Escala: Absoluta

## **2.4 Descripción del procedimiento**

### **2.4.1 Nombre del procedimiento**

Evaluación de la calidad funcional de los componentes software.

### **2.4.2 Objetivos**

Los objetivos del procedimiento Evaluación de la calidad funcional de componentes software son:

- Proveer una guía para evaluar la característica funcionalidad de los componentes software del CEDIN.
- Definir el modelo necesario para evaluar la característica funcionalidad de los componentes.
- Formalizar toda la información que se genera durante el proceso de evaluación de la característica de funcionalidad.
- Documentar toda la información que se genera durante el proceso de evaluación de la característica de funcionalidad.

### **2.4.3 Alcance**

Dicho procedimiento es aplicable a todos los componentes software que se desarrollan en el CEDIN.

### **2.4.4 Referencias**

Manual de procedimientos IPP-1000:2008.

ISO/IEC 14598 Evaluación de los productos de software.

ISO/IEC 9126-2 Métricas externas.

ISO/IEC 9126-3 Métricas internas.

### **2.4.5 Responsables**

**Ejecuta:** Administrador de la Calidad y Revisor Líder.

**Responsable de su ejecución:** Jefe del Grupo de Gestión de la Calidad del CEDIN.

**Revisa y actualiza este procedimiento:** Jefe del Grupo de Gestión de la Calidad del CEDIN.

**Fiscaliza su cumplimiento:** Administrador de la Calidad.

### **2.4.6 Términos y definiciones**

**Característica de calidad de un producto software:** conjunto de propiedades mediante las cuales se evalúa y describe su calidad. Una característica se puede refinar en múltiples niveles de subcaracterísticas.

**Atributo:** una propiedad de calidad a la que puede asignarse una métrica

**Métrica:** es un procedimiento que examina un componente y produce un dato simple, un símbolo (p.e. Excelente, S, No) o un número.

**Procedimiento:** Serie de pasos bien definidos que permitirán y facilitarán la realización de un trabajo de la manera más correcta y exitosa posible.

**Rol:** Responsabilidades de cada uno de los miembros del proyecto.

**Etapa:** Fase en el desarrollo de una acción u obra.

**Fase:** Representa un ciclo de desarrollo.

**Acciones Correctivas:** Métodos que se utilizan para eliminar las causas que provocan que existan No Conformidades en el software desarrollado.

**No Conformidad (NC):** Incumplimiento de un requisito.

**Fallos:** Incumplimiento de un requisito asociado a un uso previsto o especificado.

**Proceso de Mejora:** El proceso de mejora está encaminado a que la UCI alcance una certificación internacional del nivel 2 del modelo CMMI. Hecho que la convertiría en la primera empresa cubana certificada con este modelo.

**CMMI (Capability Maturity Model Integration):** CMMI es un modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de Administración como de Ingeniería de Sistemas y Software.

### 2.4.7 Normas generales

- El grupo de aseguramiento de la calidad debe de ser diferente del equipo de desarrollo de los componentes.
- El ejecutor del procedimiento debe contar con las competencias necesarias para poder aplicar el procedimiento de manera exitosa.
- Se debe planificar la revisión de evaluación de la característica funcionalidad de los componentes al finalizar su fase implementación y antes de llegar a la fase pruebas internas definidas en el programa de mejora.

### 2.4.8 Desarrollo del procedimiento

El procedimiento propuesto está basado en las normas ISO/IEC 9126 y en la ISO/IEC 14598, estos dos

estándares proponen una guía de lo que se debe hacer para evaluar la calidad, pero no dice cómo realizarlo, el procedimiento que se describirá en los próximos sub-epígrafes es una propuesta de cómo evaluar la característica funcionalidad.

Para elaborar dicho procedimiento se realizó una adaptación de lo que proponían estas normas a lo que se necesitaba en el centro, definiéndose las diferentes etapas que componen el proceso de evaluación propuesto para la aplicación de las métricas, el cual consta de cuatro etapas y una serie de actividades, a continuación se muestra un diagrama con el flujo del procedimiento:

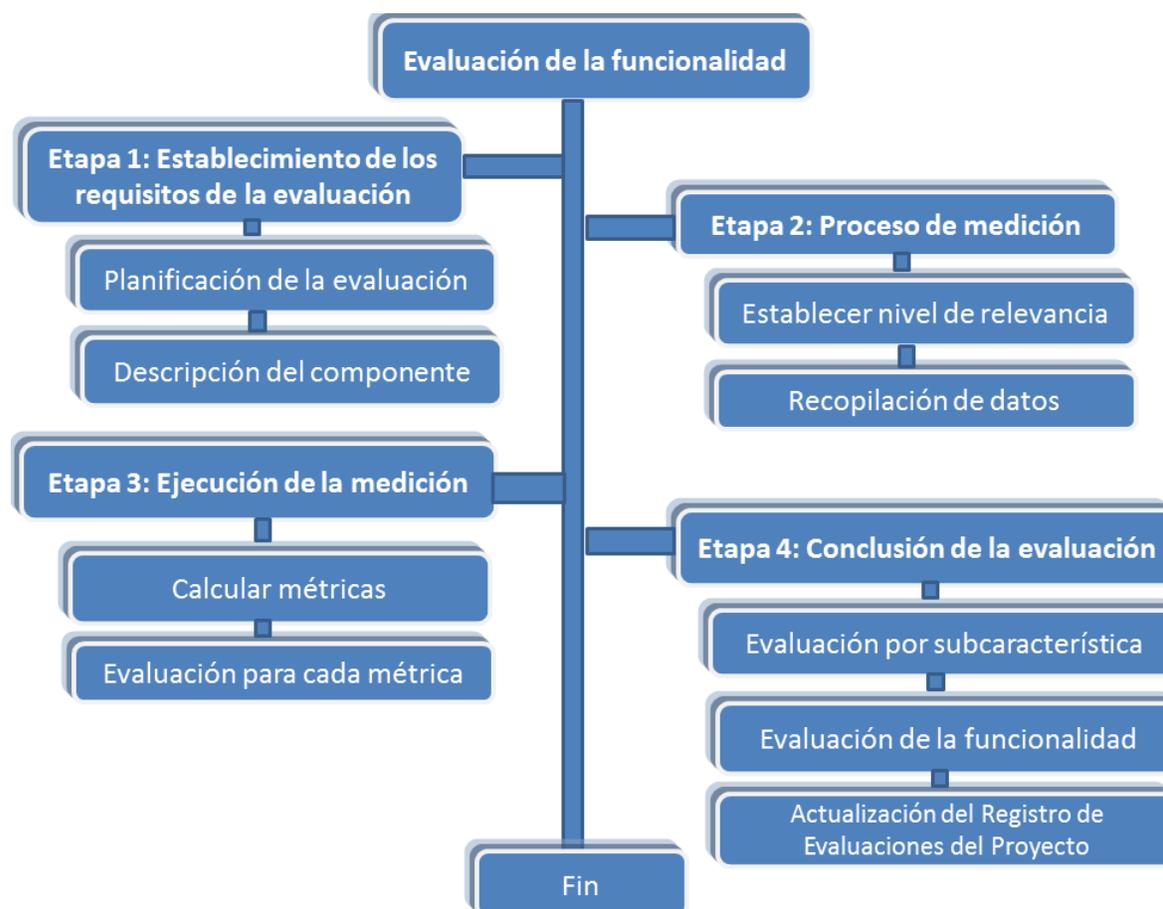


Figura 8: Flujo del Procedimiento

#### 2.4.9 Etapa 1: Establecimiento de los requisitos de la evaluación

Esta etapa Establecimiento de los requisitos de la evaluación tiene gran significación para el proceso de

medición, es donde se realizará una correcta planificación de la evaluación de la característica funcionalidad en el ciclo de vida de los componentes y donde se recopilarán datos esenciales de los mismos, permitiendo un adecuado conocimiento de su estado.

Definiéndose dos pasos o actividades primordiales: Planificación de la evaluación y Descripción del componente.

En el próximo diagrama se muestra la secuencia de actividades de esta etapa:

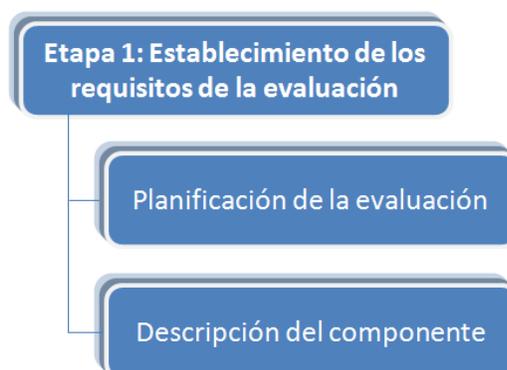


Figura 9: Actividades de la Etapa 1

### 1. Planificación de la evaluación

El administrador de la calidad en conjunto con el jefe del proyecto deben programar la evaluación de la característica funcionalidad, para ello se debe de incluir dicha actividad en el Plan de Desarrollo de Software, en el apéndice del Plan de Aseguramiento de la Calidad (PPQA), según el cronograma del proyecto. Para probar correctamente todas las funciones de los componentes esta evaluación se debe realizar después de concluida la etapa de Implementación de los componentes y antes de llegar a la fase Pruebas Internas definida en el proceso de mejora.

### 2. Descripción del componente

El propósito fundamental de la actividad Descripción del componente software a evaluar es conocer funcionalmente el componente que se analizará, es decir recolectar la información inicial necesaria para realizar la evaluación de esta característica de calidad.

En esta actividad se debe especificar con claridad la fecha de la evaluación, el nombre de dicho componente, la Fase del ciclo de vida del componente en el momento de la evaluación (ver anexo 1), el propósito de la evaluación, por último rol y nombre del evaluador, los roles definidos para esta evaluación son los que se establecieron en el apartado 2.1 Roles y responsabilidades de los participantes en el proceso de evaluación.

Estos datos son recogidos en el Modelo Evaluación de la Funcionalidad en la tabla Descripción de los componentes (ver tabla 3):

No	Fecha de evaluación	Nombre del componente	Fase	Propósito de la evaluación	Rol del evaluador	Nombre del evaluador

Tabla 3: Descripción de los componentes

El propósito de la evaluación pueden ser alguno de los que se nombra a continuación, independientemente que la mayoría de las veces se quiera determinar si el componente presenta un adecuado cumplimiento funcional.

- Asegurar el cumplimiento funcional del componente
- Definir cuándo declarar terminado el componente
- Decidir cuándo mejorar o reemplazar el componente
- Compararlo con otros componentes
- Decidir si se acepta o no el componente
- Otro

#### **2.4.10 Etapa 2: Proceso de medición**

Una vez planeada la evaluación e identificados los aspectos iniciales de los componentes los evaluadores procederán a determinar los pesos de cada una de las subcaracterísticas funcionales y los datos de las variables implicadas en la medición. Por lo que esta etapa tiene definida dos actividades que se muestran en el siguiente diagrama:

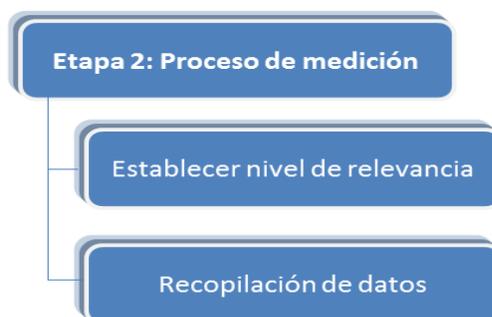


Figura 10: Actividades de la Etapa 2

### 1. Establecer nivel de relevancia

En esta actividad el evaluador le otorgará un nivel a cada subcaracterística funcional del componente sometido a evaluación, para reflejar los resultados de este análisis se hará uso del Modelo Evaluación de la Funcionalidad, la tabla Medición de la calidad funcional (ver tabla 4).

Este nivel lo establece el evaluador en dependencia de la importancia que tenga cada sub-característica sobre el componente, estos pueden ser Alto, Medio o Nulo, en este último caso la subcaracterística no es medible, por lo que no se evaluará en la medición.

Subcaracterísticas	Nivel de relevancia	Nombre de la métrica
Idoneidad	Alta	a) Adecuación funcional
		b) Completitud de la implementación funcional
		c) Cobertura de la implementación funcional

Tabla 4: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para el establecimiento del nivel.

Las subcaracterísticas idoneidad, interoperabilidad y seguridad deben ser de Alta relevancia, debido a la importancia que representan para el correcto funcionamiento de los componentes, mientras que exactitud puede ser Media.

### 2. Recopilación de datos

Luego de asignar los niveles de los atributos se procede a colocar los datos de las variables implicadas en

las fórmulas de medición de las métricas (ver tabla 5). Para realizar este paso se utilizará del Modelo Evaluación de la Funcionalidad la tabla Medición de la calidad funcional, donde en la columna Datos el evaluador pondrá los números de las variables de cada fórmula.

Nombre de la métrica	Métrica	Variables	Datos
a) Adecuación funcional	$X = 1 - A/B$	A - Número de funciones en las cuales se detectaron problemas en la evaluación	
		B - Número de funciones evaluadas	
b) Completitud de la implementación funcional	$X = 1 - A/B$	A - Número de funciones perdidas detectadas en la evaluación.	
		B - Número de funciones descritas en la especificación de requisitos	
c) Cobertura de la implementación funcional	$X = 1 - A/B$	A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas	
		B - Número de funciones descritas en la especificación de requisitos	

Tabla 5: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para la Recopilación de Datos

### 2.4.11 Etapa 3: Ejecución de la medición

En esta tercera etapa se realizan dos actividades vitales en esta medición, una es el cálculo de las métricas y la otra es otorgarle la evaluación correspondiente a las métricas según los resultados obtenidos. Aquí se comienzan a establecer evaluaciones que son indispensables ya que al final se pueden detectar cuáles fueron los atributos que no arrojaron buenos resultados. A continuación se muestra el diagrama que representa esta etapa:

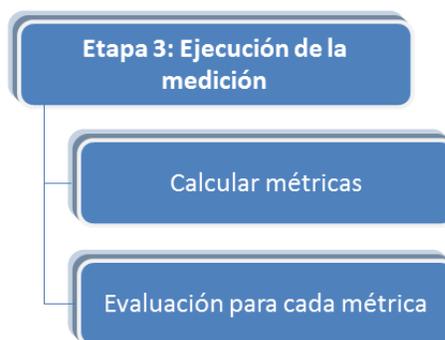


Figura 11: Actividades de la Etapa 3

## 1. Calcular métricas

La actividad calcular métricas se realizará automáticamente en el Modelo Evaluación de la Funcionalidad, en la tabla Medición de la calidad funcional en la columna Resultados, este cálculo está basado en las fórmulas de la métricas y en los datos introducidos por el evaluador, un ejemplo de este cálculo se puede observar en la siguiente tabla:

Nombre de la métrica	Métrica	Variables	Datos	Resultados
a) Adecuación funcional	$X = 1 - A/B$	A - Número de funciones en las cuales se detectaron problemas en la evaluación	1	0,9
		B - Número de funciones evaluadas	10	
b) Completitud de la implementación funcional	$X = 1 - A/B$	A - Número de funciones perdidas detectadas en la evaluación.	0	1
		B - Número de funciones descritas en la especificación de requisitos	3	
c) Cobertura de la implementación funcional	$X = 1 - A/B$	A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas	1	0,7
		B - Número de funciones descritas en la especificación de requisitos	3	

Tabla 6: Ejemplo de un fragmento de la tabla Medición de la calidad Funcional para el cálculo de las métricas.

## 2. Evaluación para cada métrica calculada

Una vez que el evaluador tiene todos los resultados de las métricas pasará a darle una evaluación cualitativa a cada una de ellas. Para realizar esta actividad este hará uso de la Tabla de escala.

Esta tabla tiene el objetivo de mostrar en qué estado se encuentran los resultados obtenidos y está definida en el criterio de que cuando más cercano esté de uno mejor resultará, ya que los valores de las métricas se encuentran entre 0 y 1.

La Tabla de escala consta de 3 rangos que oscilan del 0 al 1, que es donde el evaluador ubicará los resultados de las métricas, el por ciento que representa dicho rango y la evaluación que le corresponde al rango (ver tabla 7).

Rango	Por ciento	Evaluación
$0,8 \leq X \leq 1$	80% - 100%	Bien
$0,5 \leq X \leq 0,8$	50% - 80%	Regular
$X < 0,5$	Menor o igual que 49%	Mal

Tabla 7: Tabla de escala

La evaluación que le corresponda a cada uno de los resultados encontrados se colocará en el Modelo Evaluación de la Funcionalidad en la tabla Medición de la calidad funcional en la columna Evaluación (ver tabla 8).

Nombre de la métrica	Métrica	Variables	Datos	Resultados	Evaluación
a) Adecuación funcional	$X = 1 - A/B$	A - Número de funciones en las cuales se detectaron problemas en la evaluación			
		B - Número de funciones evaluadas			
b) Completitud de la implementación funcional	$X = 1 - A/B$	A - Número de funciones perdidas detectadas en la evaluación.			
		B - Número de funciones descritas en la especificación de requisitos			
		B - Número de funciones descritas en la especificación de requisitos			

Tabla 8: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para establecer la evaluación alcanzada para cada métrica.

#### 2.4.12 Etapa 4: Conclusión de la evaluación

En esta etapa finalmente se obtendrá la evaluación exacta de la característica funcionalidad; para lograr esto el evaluador calculará la alcanzada por cada una de las subcaracterísticas. Y para finalizar el procedimiento actualizará el Registro de Evaluaciones del Proyecto, en el que expondrá las NC detectadas en caso de que existan, luego le entregará la actualización del registro al Líder del proyecto, el que definirá las acciones correctivas que se aplicarán para eliminar dichos problemas, siendo el administrador de la calidad el responsable de darle seguimiento al cumplimiento de dichas acciones.



Figura 12: Actividades de la Etapa

### 1. Evaluación por subcaracterística

En esta actividad el evaluador le asignará una evaluación a cada subcaracterística, tanto cuantitativa como cualitativa. Para esto el evaluador calculará el por ciento que representan los resultados alcanzados en sus métricas, una vez encontrado lo ubicará en la Tabla de escala (ver tabla 7) y según en el rango que se encuentre el número en la columna Por ciento de dicha tabla es la evaluación que le corresponde, esta puede ser Bien, Regular o Mal.

Este cálculo se realizará en la tabla Medición de la calidad funcional del Modelo Evaluación de la Funcionalidad en la columna Por ciento alcanzado (ver tabla 9). La métrica que se utilizará para encontrar el Por ciento es la siguiente:

$$P_{sub} = \frac{\Sigma \text{ResultM}}{\text{CantM}} * 100$$

**Psub:** Por ciento alcanzado por cada subcaracterística.

**ResultM:** Resultados de los cálculos de las métricas.

**CantM:** Cantidad de métricas calculadas

Métrica	Variables	Datos	Resultados	Evaluación	Porcentaje alcanzado	Evaluación por Subcaracterística
$X = 1 - A/B$	A - Número de funciones en las cuales se detectaron problemas en la evaluación					
	B - Número de funciones evaluadas					
$X = 1 - A/B$	A - Número de funciones perdidas detectadas en la evaluación.					
	B - Número de funciones descritas en la especificación de requisitos					
$X = 1 - A/B$	A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas					
	B - Número de funciones descritas en la especificación de requisitos					

Tabla 9: Ejemplo de un fragmento de la tabla Medición de la calidad funcional para establecer la evaluación alcanzada por cada subcaracterística.

Como existen dos niveles requeridos en las métricas para calcular las diferentes subcaracterísticas, se dificulta el cálculo del por ciento por lo que fue necesario lograr una similitud en los resultados. Para esto se utilizó la siguiente propiedad de la probabilidad en las métricas cuando el nivel requerido es 0:

$$P(A^c) = 1 - P(A)$$

Donde  $A^c$  significa complemento de A y es que dado un evento A, el evento que no ocurra A, se denomina evento complemento de A.

De esta manera se le resta 1 al resultado de la métrica, obteniendo su complemento, por ejemplo:

Métrica de Precisión:

$$X = A / T$$
$$X = 1 / 5$$
$$X = 0.2$$

Por tanto su complemento sería:

$$P(A^c) = 1 - 0.2$$

$$P(A^c) = 0.8$$

## 2. Evaluación de la funcionalidad

En esta actividad se presenta la evaluación final de la característica funcionalidad, para emitir dicha evaluación el evaluador calculará el por ciento alcanzado entre todas las subcaracterísticas evaluadas y luego le otorgará una evaluación final, para realizar este último paso deberá analizar las evaluaciones alcanzadas por cada subcaracterística y tendrá en cuenta los criterios que se definen en la tabla 10, sino es el caso de alguno de estos el promedio final de la funcionalidad se lleva a la tabla de escala y es el que le corresponda.

Criterios	Evaluación
Si se evaluaron las subcaracterísticas idoneidad y exactitud.	Se tiene en cuenta la evaluación alcanzada por la idoneidad, si es Mal o Regular la funcionalidad es Mal o Regular, sino se lleva el resultado a la Tabla de escala.
Si se evaluaron las subcaracterísticas idoneidad, exactitud e interoperabilidad.	Se tienen en cuenta las evaluaciones obtenidas por idoneidad e interoperabilidad, si al menos alguna de ellas es Mal, la evaluación final es Mal también, si ambas son regulares, la evaluación es Regular, sino se lleva el resultado a la Tabla de escala.
Si se evaluaron las subcaracterísticas idoneidad, exactitud, interoperabilidad y	Se tienen en cuenta las evaluaciones obtenidas por idoneidad, interoperabilidad y seguridad, si al menos alguna de ellas es Mal, la evaluación final es Mal

seguridad.	también, y si al menos dos son regulares, la evaluación es Regular, sino se lleva el resultado a la Tabla de escala.
Si se evaluaron las subcaracterísticas idoneidad, interoperabilidad y seguridad.	Si al menos una de ellas es Mal, la evaluación final es Mal, o si dos de ellas son regulares, la evaluación es Regular, sino se lleva el resultado a la Tabla de escala.
Si se evaluaron las subcaracterísticas idoneidad e interoperabilidad.	Si alguna de ellas es Mal o Regular, la evaluación final de la funcionalidad es Mal o Regular.
Si se evaluaron las subcaracterísticas idoneidad y seguridad.	Si alguna de ellas es Mal o Regular, la evaluación final de la funcionalidad es Mal o Regular.

*Tabla 10: Criterios para evaluar la característica funcionalidad*

Para calcular el por ciento final de la funcionalidad se utilizará la métrica siguiente:

$$P_{fun} = \frac{\sum P_{sub}}{CantSub}$$

P<sub>fun</sub>: Por ciento de funcionalidad.

P<sub>sub</sub>: Por ciento obtenido en las subcaracterísticas

CantSub: Cantidad de subcaracterísticas

El resultado obtenido se ubica al final de la tabla Medición de la calidad funcional debajo de los por cientos alcanzados por las subcaracterísticas, y la evaluación que le corresponda en la parte derecha de esa columna.

De esta manera se tiene una evaluación cuantitativa y cualitativa de la funcionalidad, además de los por cientos alcanzados por cada una de las subcaracterísticas con su evaluación, pudiéndose identificar claramente aquellas subcaracterísticas que no cumplieron los niveles requeridos.

Para una mayor visualización gráfica de dicho cumplimiento el evaluador colocará en la tabla Resumen de funcionalidad del Modelo Evaluación de la Funcionalidad el peso que le asignó a cada subcaracterística, el por ciento que alcanzó cada una y la evaluación que le corresponde (ver tabla 11).

Subcaracterísticas	Peso	Por ciento alcanzado	Evaluación
Idoneidad	2 (Alta Relevancia)	80%	Bien
Exactitud	1 (Media Relevancia)	90%	Bien
Interoperabilidad	2 (Alta Relevancia)	63%	Regular
Seguridad	1 (Media Relevancia)	79%	Regular

Tabla 11: Ejemplo Tabla resumen de funcionalidad

Una vez completada esta tabla se obtendrá actualizado el gráfico de dicha tabla resumen, del cual se muestra un ejemplo a continuación:

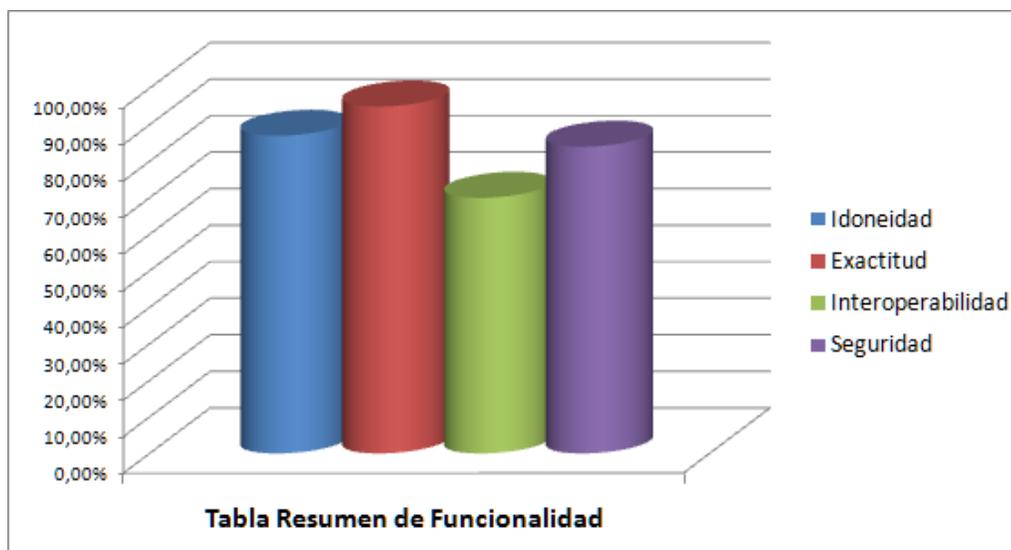


Figura 13: Ejemplo resumen de funcionalidad

### 3. Actualización del Registro de Evaluaciones del Proyecto

Para finalizar este procedimiento el evaluador actualizará el Registro de Evaluaciones del Proyecto que se encuentra en la siguiente dirección del expediente de proyecto:

#### **Expediente de Proyecto del PM 3.3**

- 3. Soporte

- 3.1 Aseguramiento de la calidad

- 0312\_Plantilla de Registro de Evaluaciones del proyecto.xls

En este registro el evaluador pondrá las No Conformidades (NC) que encontró, buscando en las subcaracterísticas todos los errores encontrados en la evaluación de cada indicador.

Una vez concluida esta actividad el evaluador le entregará este registro al Líder del proyecto, el cual presentará las Acciones Correctivas (AC) que solucionaran los errores encontrados, con su fecha de cumplimiento.

A continuación se proponen posibles NC y AC para corregirlas:

<b>Posibles No conformidades</b>	<b>Posibles Acciones Correctivas</b>
Funcionamiento inadecuado de las funciones implementadas.	Verificar la veracidad de la descripción de los requisitos, en caso de estar correctos arreglar la función inadecuada
Implementación funcional incompleta.	Completar todas las funciones descritas en el Modelo Especificación de Requisitos
Dificultades en los resultados esperados.	Verificar que el componente cumpla con los requisitos funcionales que se definieron en el artefacto Especificación de Requisitos. Insertarle alguna funcionalidad que permita solucionar el problema.
Indebida implementación del intercambio de funciones de interface.	Revisar la implementación de las funciones de interfaz para la transferencia; insertar alguna función que

	solvente el error.
Incapacidad de intercambio de datos con otros componentes.	Revisar la implementación de las funciones de interfaces para la transferencia de datos e insertar alguna función que solvente el error.
Indebida capacidad de control de acceso al sistema.	Verificar que el componente cumpla con los requisitos no funcionales de seguridad que se definieron en el artefacto Especificación de Requisitos. Insertarle alguna funcionalidad que permita solucionar el problema.
Dificultades en la prevención de la corrupción de datos.	Verificar que el componente cumpla con los requisitos no funcionales de seguridad que se definieron en el artefacto Especificación de Requisitos. Insertarle alguna funcionalidad que permita solucionar el problema.

Tabla 12: Posibles No Conformidades y Acciones Correctivas

### **Conclusiones del capítulo**

En este capítulo se presentó el diseño del procedimiento para la evaluación de la funcionalidad en el CEDIN, también un conjunto de métricas planteadas para la evaluación de cada subcaracterística funcional. Contribuyendo que en las revisiones de calidad de los componentes se puedan detectar para luego mitigar posibles errores que estos puedan presentar, aportando una guía paso a paso de aplicación para los administradores de la calidad y revisores líderes, así como un modelo para la recolección y cálculo de los datos.

# CAPÍTULO 3

## *Validación del procedimiento*

### **Introducción**

En este capítulo se realiza la validación del procedimiento propuesto en el capítulo anterior, la cual se efectúa en tres componentes software utilizados en el proyecto Sistema de Manejo Integral de Perforación de Pozos Petroleros del CEDIN ellos son: el componente Archivo, Gráfica y Seguridad. Se le aplicará a cada componente por separado y luego se hará una comparación de los resultados.

### **3.1 Descripción general del procedimiento**

El procedimiento Evaluación de la calidad funcional de los componentes software consta de cuatro etapas, conformadas por actividades, las cuales paulatinamente van mostrando el estado de calidad funcional que presenta el componente, estas se muestran en la siguiente tabla:

<b>Etapas</b>	<b>Actividades</b>
1. Establecimiento de los requisitos de la evaluación	✓ Planificar la evaluación ✓ Descripción del componente
2. Proceso de medición	✓ Establecer nivel de relevancia
3. Ejecución de la medición	✓ Calcular métricas ✓ Evaluación para cada métrica calculada
4. Conclusión de la evaluación	✓ Evaluación por subcaracterística ✓ Evaluación de la funcionalidad ✓ Actualización del Registro de Evaluaciones del Proyecto

Tabla 13: Etapas y actividades del procedimiento

Los roles encargados de ejecutar el procedimiento para los componentes son los definidos en la sección 2.1 del capítulo 2 del presente trabajo, las subcaracterísticas a evaluar son:

- ✓ *Idoneidad*
- ✓ *Exactitud*
- ✓ *Interoperabilidad*
- ✓ *Seguridad*

La subcaracterística exactitud no será medible en la evaluación del componente Archivo y Seguridad debido a que no hay posibilidad de la ocurrencia de una falta de exactitud o de precisión en algún cálculo o resultado.

La subcaracterística seguridad no será medible en la evaluación de los componentes Archivo y Gráfica debido a que los componentes no son los encargados de realizar ninguna función de seguridad en el sistema, ni poseen seguridad por si solos.

Para realizar la evaluación de dichas subcaracterísticas el procedimiento tiene incluida una selección de métricas adecuadas para dar una valoración objetiva del comportamiento funcional del sistema, a continuación se muestran en una tabla organizada por las subcaracterísticas:

Subcaracterística	Métrica
Idoneidad	Adecuación funcional
	Compleitud de la implementación funcional
	Cobertura de la implementación funcional
	Obtener por ciento de la subcaracterística
Precisión	Precisión
	Obtener por ciento de la subcaracterística
Interoperabilidad	Intercambiabilidad de datos, en base su formato
	Intercambiabilidad de datos, en base éxito del intento
	Obtener por ciento de la subcaracterística
Seguridad	Control del acceso
	Obtener por ciento de la subcaracterística
-----	Por ciento de funcionalidad.

Tabla 14: Métricas para realizar la evaluación

La métrica Capacidad de control de acceso del componente Seguridad no se analizará en este procedimiento debido a que dicho componente no cuenta con una base de datos que almacene el historial de acceso de los usuarios.

En el componente Gráfica no se analizará la métrica Exactitud en los cálculos de la subcaracterística exactitud debido a que el mismo no efectúa ningún tipo de cálculo en sus funcionalidades.

### **3.2. Proyecto seleccionado para realizar el procedimiento**

Dentro del CEDIN se encuentran varios proyectos que desarrollan componentes software reutilizables, uno de ellos es el Sistema de Manejo Integral de Perforación de Pozos Petroleros (SIPP), el cual tiene como objetivo desarrollar una aplicación web para CUPET (Empresa Cuba de Petróleo) que se encarga de automatizar la gestión de información, así como la elaboración de partes y reportes diarios de información, tanto en los pozos de petróleo como en la oficina de despacho de la DIPP (Dirección de Intervención y Perforación de Pozos).

Actualmente este proyecto se encuentra en la fase Pruebas de Liberación definida en el proceso de mejora y a pesar de haber pasado por pruebas funcionales, sería un aporte detectar posibles No Conformidades y fallos de las subcaracterísticas (idoneidad, exactitud, interoperabilidad y seguridad) de funcionalidad que los componentes podrían tener y proponer acciones correctivas para el mejoramiento de las NC. Además de conocer el por ciento de funcionalidad que posee cada uno de ellos.

### **3.3 Evaluación componente Archivo**

#### **1. Descripción del componente**

Una vez planificada la evaluación de la característica funcionalidad del componente Archivo en el documento Plan de Desarrollo de Software, el apéndice del Plan de Aseguramiento de la Calidad (PPQA), se procede a realizar la medición.

Para esto el evaluador posee el Modelo de Evaluación de la Funcionalidad y el acceso a la aplicación, donde el primer paso es describir el componente y quién está efectuando dicha comprobación. En caso de tener dudas en alguno aspecto de la descripción este se apoyará en el líder del proyecto

donde se utiliza el componente.

La información obtenida del componente es la siguiente:

Fecha de evaluación: 30/04/2012

Nombre del componente: Archivo

Fase: Pruebas de Liberación

Propósito de la evaluación: Asegurar el cumplimiento funcional del componente

Rol del evaluador: Administrador de la calidad

Nombre del evaluador: Yuneisis Estrada Peña

Archivo: es un componente que permite gestionar en los Módulos principales que son Pozos y SSIP la importación y exportación de información (ver Anexos 2 y 3).

## **2. Establecer nivel de relevancia**

A partir de este paso se hará uso de la tabla Medición de la calidad funcional del Modelo Evaluación de la Funcionalidad, lo primero es determinar el nivel de cada subcaracterística, el cual el administrador seleccionará en el campo Nivel de relevancia (ver tabla 4), definiéndose de la siguiente forma:

Idoneidad: Alta

Exactitud: Nula

Interoperabilidad: Alta

Seguridad: Nula

## **3. Calcular y evaluar métricas**

### **Idoneidad:**

- a) Adecuación funcional

$$X = 1 - 6/48$$

$$X = 0.9 \text{ Evaluación: Bien}$$

Se probaron 48 funcionalidades de las cuales 6 presentaron problemas en la evaluación.

- b) Completitud de la implementación funcional

$$X = 1 - 0/2$$

$$X = 1 \text{ Evaluación: Bien}$$

Este componente posee dos funciones descritas en el documento Especificación de requisitos

(RF 2.43), las cuales están implementadas.

- c) Cobertura de la implementación funcional

$$X = 1 - 1/2$$

X= 0.5 Evaluación: Regular

De las dos funciones descritas en el documento Especificación de requisitos (RF 2.43) una de ellas Exportar Archivo presenta problemas funcionales.

Métrica general de la subcaracterística idoneidad

$$P_{Sub} = 0.9 + 1 + 0.5 / 3 * 100$$

$$P_{Sub} = 80 \%$$

#### **Interoperabilidad**

- a) Intercambiabilidad de datos, en base su formato

$$X = 8/9$$

X=0.9 Evaluación: Bien

De un total de 9 formatos de datos a intercambiar con los otros módulos (Pozos y SSIP), 8 fueron intercambiados exitosamente, debido a que en Partes el campo Seleccionar Partes tuvo problemas de exportación.

- b) Intercambiabilidad de datos, en base al éxito del intento

$$X = 1 - 1/16$$

X=0.9 Evaluación: Bien

En el módulo DIPP falló la exportación en Partes del Operativo de Perforación de las 6.00

Métrica general de la subcaracterística Interoperabilidad

$$P_{Sub} = 0.9 + 0.9 / 2 * 100$$

$$P_{Sub} = 90\%$$

#### **4. Evaluación por subcaracterística**

Idoneidad: alcanzó un 80% por lo que obtiene la evaluación de Bien

Interoperabilidad: alcanzó un 90 % por lo que obtiene la evaluación de Bien

## 5. Evaluación de la funcionalidad

$$PFun = 80 + 90 / 2$$

$$PFun = 85\%$$

Analizando los resultados de todas las subcaracterísticas, las cuales todas cumplieron con su nivel requerido, se llega a la conclusión que el componente Archivo está funcionando al 85 % Obteniendo la evaluación de Bien.

A continuación se muestra una gráfica con los resultados de las subcaracterísticas:



Figura 14: Resumen del componente Archivo

## 6. Actualización del Registro de Evaluaciones del Proyecto

No Conformidades	Acción correctiva
Al seleccionar inclinometrías en la DIPP el campo fecha se debe ocultar	Revisar la implementación de la función fecha.
No se muestra un mensaje de error al no seleccionar un Reporte al exportar un archivo en la DIPP de tipo Reporte, formato Excel.	Revisar la implementación de la función Reporte.
No se muestra un mensaje de error al no seleccionar un	Revisar la implementación de la

Reporte al exportar un archivo en la DIPP de tipo Partes, formato Excel.	función Reporte.
Falta por añadir en el subsistema Pozo el reporte Operativo de Perforación de las 6.00 am	Revisar la implementación de la función Reporte.
Al seleccionar el formato TXT en el módulo Pozo, solo se habilitan los reportes de inclinometría (real y plan), no se muestran las otras opciones de reportes	Revisar la implementación de la función Reporte.
En la DIPP no se exporta el parte Operativo de las 6.00 am del tipo Partes formato Excel	Revisar la implementación de la función Parte.

Tabla 15: No Conformidades detectadas en el componente Archivo

### **3.4 Evaluación componente Gráfica**

#### **1. Descripción del componente**

Una vez planificada la evaluación de la característica funcionalidad del componente Gráfica en el documento Plan de Desarrollo de Software, el apéndice del Plan de Aseguramiento de la Calidad (PPQA), se procede a realizar la medición.

Para esto el evaluador posee el Modelo de Evaluación de la Funcionalidad y el acceso a la aplicación, donde el primer paso es describir el componente y quién está efectuando dicha comprobación. En caso de tener dudas en algún aspecto de la descripción este se apoyará en el líder del proyecto donde se utiliza el componente.

La información obtenida del componente es la siguiente:

Fecha de evaluación: 03/05/2012

Nombre del componente: Gráfica

Fase: Pruebas de Liberación

Propósito de la evaluación: Asegurar el cumplimiento funcional del componente

Rol del evaluador: Administrador de la calidad

Nombre del evaluador: Yuneisis Estrada Peña

Gráfica: es un componente que permite visualizar la información de los diferentes indicadores

ubicados en los reportes de perforación petrolera (ver Anexos 2 y 3).

## **2. Establecer nivel de relevancia**

A partir de este paso se hará uso de la tabla Medición de la calidad funcional del Modelo Evaluación de la Funcionalidad, lo primero es determinar el nivel de cada subcaracterística, el cual el administrador seleccionará en el campo Nivel de relevancia (ver tabla 4), definiéndose de la siguiente forma:

Idoneidad: Alta

Exactitud: Media

Interoperabilidad: Alta

Seguridad: Nula

## **3. Calcular y evaluar métricas**

### **Idoneidad:**

- a) Adecuación funcional

$$X = 1 - 1/25$$

$$X = 1 \text{ Evaluación: Bien}$$

Se probaron 25 funcionalidades de las cuales en el módulo pozo en Dirección Horizontal no graficó mostrando un error.

- b) Completitud de la implementación funcional

$$X = 1 - 0/6$$

$$X = 1 \text{ Evaluación: Bien}$$

Este componente tiene seis funciones descritas en el documento Especificación de requisitos (RF 2.43), las cuales están implementadas.

- c) Cobertura de la implementación funcional

$$X = 1 - 1/6$$

$$X = 0.8 \text{ Evaluación: Bien}$$

No graficó en el módulo pozo la Dirección Horizontal.

Métrica general de la subcaracterística idoneidad

$$P_{\text{Sub}} = 1 + 1 + 0.8 / 3 * 100$$

$$P_{\text{Sub}} = 93.33 \%$$

### **Exactitud**

- a) Precisión

$$X = 1 - (0/2)$$

$$X=1 \quad \text{Evaluación: Bien}$$

En un tiempo de operación dos segundos el sistema es capaz de graficar los datos de los módulos DIPP y Pozo con la precisión requerida.

Métrica general de la subcaracterística Exactitud

$$P_{\text{Sub}} = 1 * 100$$

$$P_{\text{Sub}} = 100 \%$$

### **Interoperabilidad**

- a) Intercambiabilidad de datos, en base su formato

$$X = 2/2$$

$$X=1 \quad \text{Evaluación: Bien}$$

Los 2 formatos de datos a intercambiar con los otros módulos (Pozos y SSIP) fueron intercambiados exitosamente.

- b) Intercambiabilidad de datos, en base al éxito del intento

$$X = 1 - 0/14$$

$$X=1 \quad \text{Evaluación: Bien}$$

No existió ningún fallo al proceder el intercambio de datos con otros componentes.

Métrica general de la subcaracterística interoperabilidad

$$P_{\text{Sub}} = 1 + 1 / 2 * 100$$

$$P_{\text{Sub}} = 100 \%$$

#### 4. Evaluación por subcaracterística

$$PFun = 93.33 + 100 + 100 / 3$$

$$PFun = 97.77$$

Idoneidad: alcanzó un 93.33% por lo que obtiene la evaluación de Bien

Exactitud: alcanzó un 100% por lo que obtiene la evaluación de Bien

Interoperabilidad: alcanzó un 100% por lo que obtiene la evaluación de Bien

#### 5. Evaluación de la funcionalidad

Analizando los resultados de todas las subcaracterísticas, las cuales todas cumplieron con su nivel requerido, se llega a la conclusión que El componente Archivo está funcionando al 97.77 % Obteniendo la evaluación de Bien.

A continuación se muestra una gráfica con los resultados de las subcaracterísticas:

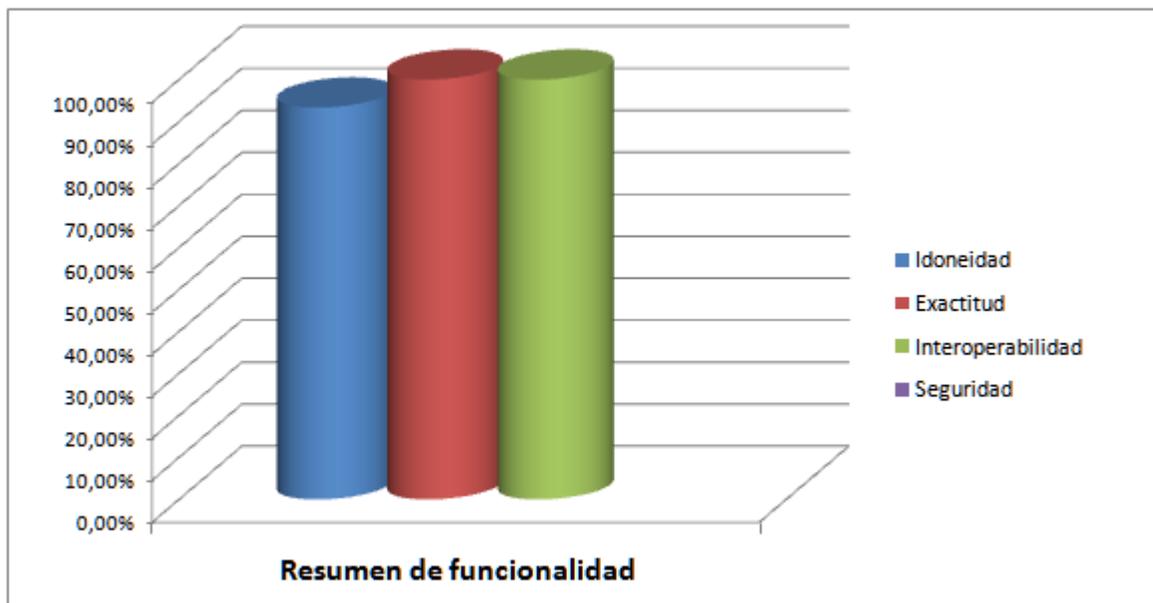


Figura 15: Resumen del componente Gráfica

#### 6. Actualización del Registro de Evaluaciones del Proyecto

No Conformidades	Acción correctiva
No graficó en el módulo Pozo la Dirección Horizontal	Revisar la función Graficar Dirección

	Horizontal
--	------------

Tabla 16: No Conformidades detectadas en el componente Gráfica

### **3.5 Evaluación componente Seguridad**

#### **1. Descripción del componente**

Una vez planificada la evaluación de la característica funcionalidad del componente Seguridad en el documento Plan de Desarrollo de Software, el apéndice del Plan de Aseguramiento de la Calidad (PPQA), se procede a realizar la medición.

Para esto el evaluador posee el Modelo de Evaluación de la Funcionalidad y el acceso a la aplicación, donde el primer paso es describir el componente y quién está efectuando dicha comprobación. En caso de tener dudas en algún aspecto de la descripción este se apoyará en el líder del proyecto donde se utiliza el componente.

La información obtenida del componente es la siguiente:

Fecha de evaluación: 06/07/2012

Nombre del componente: Seguridad

Fase: Despliegue

Propósito de la evaluación: Asegurar el cumplimiento funcional del componente

Evaluador: Administrador de la calidad

Nombre del evaluador: Yuneisis Estrada Peña

Seguridad: es un componente que además que permite gestionar los usuarios y los roles del proyecto es el encargado de la seguridad de los demás subsistemas, previniendo de accesos no autorizados.

#### **2. Establecer nivel de relevancia**

A partir de este paso se hará uso de la tabla Medición de la calidad funcional del Modelo Evaluación de la Funcionalidad, lo primero es determinar el nivel de cada subcaracterística, el cual el administrador seleccionará en el campo Nivel de relevancia (ver tabla 4), definiéndose de la siguiente forma:

Idoneidad: Alta

Exactitud: Nula

Interoperabilidad: Alta

Seguridad: Alta

### **3. Calcular y evaluar métricas**

#### **Idoneidad:**

- a) Adecuación funcional

$$X = 1 - 1/64$$

X= 1 Evaluación: Bien

Se probaron 64 funcionalidades y la función cancelar la operación de Adicionar usuario al pozo no se implementó.

- b) Completitud de la implementación funcional

$$X = 1 - 2/9$$

X= 0.8 Evaluación: Bien

Este componente tiene nueve requisitos descritos en el documento Especificación de requisitos, de los cuales dos faltan por implementar.

- c) Cobertura de la implementación funcional

$$X = 1 - 3/9$$

X= 0.7 Evaluación: Regular

En esta métrica se detectaron dos requisitos funcionales sin realizar y además gestionar usuario por terminar su implementación.

Métrica general de la subcaracterística idoneidad

$$P_{Sub} = 1 + 0.8 + 0.7 / 3 * 100$$

$$P_{Sub} = 83.33 \%$$

#### **Interoperabilidad**

- a) Intercambiabilidad de datos, en base a su formato

$$X = 46/46$$

X=1 Evaluación: Bien

Los 46 formatos de datos a intercambiar con los otros módulos (Pozos y SSIP) fueron intercambiados exitosamente.

- b) Intercambiabilidad de datos, en base al éxito del intento

$$X=1 - 0/28$$

X=1 Evaluación: Bien

No existió ningún fallo al proceder el intercambio de datos con otros componentes.

Métrica general de la subcaracterística interoperabilidad

$$P_{Sub} = 1 + 1 / 2 * 100$$

$$P_{Sub} = 100 \%$$

### **Seguridad**

- b) Capacidad de control de acceso

$$X= 4/4$$

X=1 Evaluación: Bien

Se realizaron 4 intentos para tratar de introducirse sin autorización al sistema, los cuales fallaron sin aportar información vulnerable, estos fueron:

- Cualquier usuario y contraseña
- Usuario legítimo y cualquier contraseña
- Solo clic en el botón Entrar
- Tratando de introducir una inyección SQL que sea una condición lógica y siempre sea verdadera, por ejemplo: en el campo usuario se introduce “'1' OR 1 = 1 --”

Métrica general de la subcaracterística seguridad

$$P_{Sub} = 1 * 100$$

$$P_{Sub} = 100 \%$$

### **4. Evaluación por subcaracterística**

$$P_{Fun} = 83.33 + 100 + 100 / 3$$

$$P_{Fun} = 94.44\%$$

Idoneidad: alcanzó un 83.33% por lo que obtiene la evaluación de Bien

Interoperabilidad: alcanzó un 100% por lo que obtiene la evaluación de Bien

Seguridad: alcanzó un 100% por lo que obtiene la evaluación de Bien

### 5. Evaluación de la funcionalidad

Analizando los resultados de todas las subcaracterísticas, las cuales todas cumplieron con su nivel requerido, se llega a la conclusión que El componente Archivo está funcionando al 94.44 % Obteniendo la evaluación de Bien.

A continuación se muestra una gráfica con el resumen de dichos resultados:

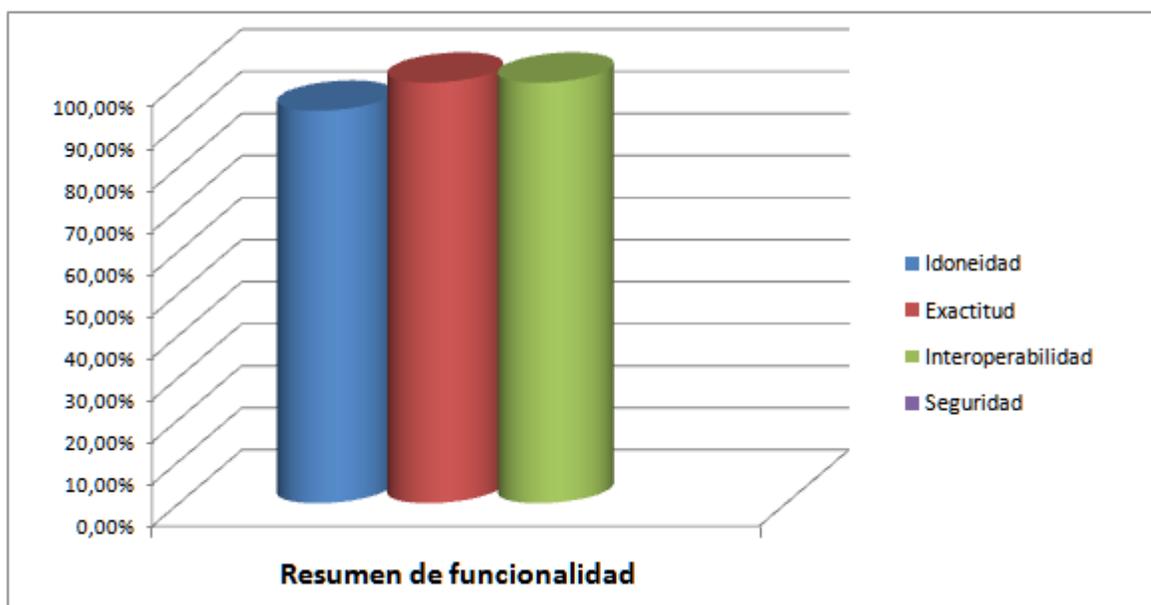


Figura 16: Resumen del componente Gráfica

### 6. Actualización del Registro de Evaluaciones del Proyecto

No Conformidades	Acción correctiva
No se permite cancelar la operación de Adicionar usuario al pozo	Revisar la implementación de la función Gestionar usuario.
Implementar el requisito Guardar Registros de Seguridad	Realizar la implementación
Implementar el requisito Consultar Registro de Seguridad	Realizar la implementación

Tabla 17: No Conformidades detectadas en el componente Seguridad

### 3.6 Comparación de resultados

Los componentes Archivo, Gráfica y Seguridad se encuentran en estado terminado y están integrados

y listos para usarse con todos los software necesarios para que funcionen con éxito. Presentan altos índices de idoneidad, exactitud, interoperabilidad y seguridad, por lo que se llega a la conclusión que poseen un alto por ciento de funcionalidad.

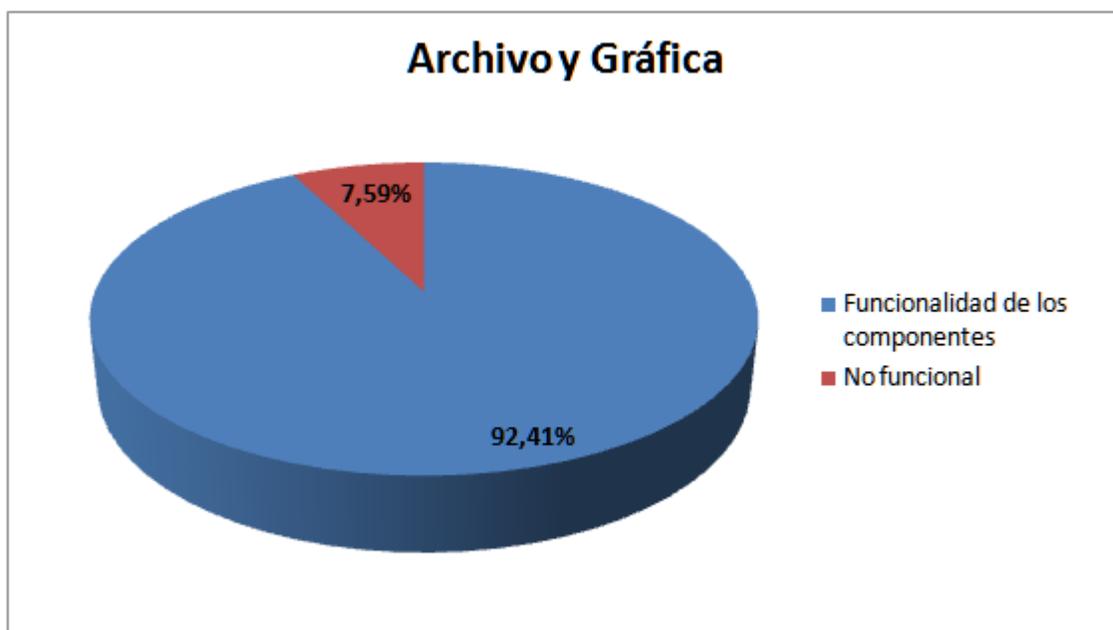


Figura 17: Por ciento de funcionalidad de los componentes Archivo y Gráfica

### **Conclusiones del capítulo**

En este capítulo se realizó la validación del procedimiento propuesto en los componentes Archivo, Gráfica y Seguridad del proyecto SIPP obteniéndose resultados favorables en los tres casos, con niveles altos de funcionalidad, pero aun así se detectaron algunas no conformidades que disminuyen la calidad del trabajo realizado y son convenientes solucionar. Demostrando la validez de la necesidad de que el grupo de calidad del centro cuente con un procedimiento que permita guiar la revisión de evaluación de la característica funcionalidad de los componentes durante su proceso de desarrollo.

## *Conclusiones*

Con el desarrollo del procedimiento y luego de los resultados obtenidos en su aplicación se le dio cumplimiento al objetivo por el cual fue definida la investigación:

- Se investigó la bibliografía relacionada con la calidad de los productos software, en especial la funcional. Obteniéndose las normas encargadas del modelo y evaluación de la calidad.
- Se analizaron las métricas funcionales propuestas en el estándar ISO 9126, seleccionándose las que más se adecuan a las características de los componentes del CEDIN.
- Se definieron roles y responsabilidades, así como etapas con sus actividades que guiarán el proceso de evaluación de los componentes.
- Se especificó el modelo necesario para la recogida de información y la salida de los resultados.
- Se validó la propuesta a través de su aplicación a tres de los componentes desarrollados en el CEDIN, con el objetivo de probar la eficacia de la propuesta.

## *Recomendaciones*

Para extender la investigación presentada en el presente trabajo de diploma se recomienda:

- Aplicar el procedimiento propuesto a todos los componentes que se desarrollen en el CEDIN, para de esta forma contribuir a mejorar la calidad funcional de dichos componentes.
- Incorporarle otras métricas a cada subcaracterística que se ajusten a las condiciones y particularidades de los componentes que se desarrollan en la facultad.
- Elaborar una herramienta que permita gestionar el proceso de evaluación de la propuesta realizada.

## *Referencias Bibliográficas*

1. Grajales, Gilberto. Líneas de Productos de Software. Líneas de Productos de Software. [En línea] 2009. <http://www.sg.com.mx/content/view/830>.
2. NC ISO/IEC 9126-1: 2005 Ingeniería de software calidad del producto. Vedado, Ciudad de La Habana: Cuban National Bureau of Standards, 2005. ICS: 35.080.
3. C, Jonás A. Montilva. Desarrollo de Software Basado en Componentes. Mérida – Venezuela : s.n., 2012.
4. Rojas, Maribel Ariza. Introducción y principios básicos del desarrollo de software basado en componentes. 2004.
5. Desarrollo de Software Basado en Componentes. Desarrollo de Software Basado en Componentes. [En línea] 2009. <http://www.slideshare.net/toryneutral/desarrollo-sw-basado-en-componentes>.
6. Figueroa, Viridiana. Slideshare. Slideshare. [En línea] <http://www.slideshare.net/viri88/ingenieria-de-dominio>.
7. Informandodecalidad. Informandodecalidad. [En línea] 2008.
8. Scribd. Scribd. [En línea] 2011. <http://es.scribd.com/doc/18014848/Definiciones-de-Calidad>.
9. Ecured. Ecured. [En línea] 2011. [http://www.ecured.cu/index.php/Calidad\\_de\\_Software](http://www.ecured.cu/index.php/Calidad_de_Software).
10. Buenastareas . Buenastareas. [En línea] 2011. <http://www.buenastareas.com/ensayos/Modelo-Calidad-Software/2899163.html>.
11. Maritzatrujilloarango. Maritzatrujilloarango. [En línea] 2010. <https://sites.google.com/site/maritzatrujilloarango/testimonials-1/definiciondecalidaddelsoftware>.
12. Vega, Juan Pablo Carvallo. Introducción a los modelos de calidad del software. 2011.
13. Camacho, Erika. Arquitecturas de software. 2004.
14. Scribd. Scribd . [En línea] 2011. <http://es.scribd.com/doc/50097036/Estandares-de-calidad-de-software-IEEE>.

15. Ecured. Ecured. [En línea] 2011.  
[http://www.ecured.cu/index.php/Calidad\\_en\\_el\\_Desarrollo\\_Software\\_Basado\\_en\\_Componentes](http://www.ecured.cu/index.php/Calidad_en_el_Desarrollo_Software_Basado_en_Componentes).
16. ITI, Grupo SQUaC del. Software Quality Usability and Certification(SQUAac). Software Quality Usability and Certification(SQUAac). [En línea] 2012. <http://squac.iti.upv.es/glosario-calidad/>.
17. Agudelo, Laura Posada. Métricas de calidad de software. 2012.
18. Bertoa, Manuel F. y Vallecillo, Antonio. Atributos de Calidad para Componentes COTS. 2012.

## *Bibliografía*

- Ariza, Maribel. kioskea. kioskea. 2004. <http://es.kioskea.net/contents/poo/classe-instance.php3>.
- Baccarelli, Rodolfo Evaluación de Software Producto OK!. Buenos Aires, 2005.
- Escuela de Ingeniería de Sistemas, Departamento de Computación. Práctica de Laboratorio 4 Programación Orientada a Objetos, 2011.
- Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales, Herramientas de Software, 2005.
- Fuentes, Lidia. Desarrollo de Software Basado en Componentes. Málaga-España : s.n., 2010.
- Ferré, Xavier. Desarrollo Orientado a Objetos con UML. Facultad de Informática – UPM.
- ISO/IEC JTC1 /SC7. Software engineering –Product quality – Part 2: External metrics. Canadá, 2002.
- ISO/IEC. Software engineering- Product quality- part 3: Internal metrics, 2003.
- ISO / IEC. Information technology-Software product evaluation- Part 5> Process for evaluators, 1998.
- Mendoza, Luis E. Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software. Venezuela, 2004.
- Mendoza, Luis E. Prototipo de Modelo Sistémico de Calidad (Mosca) del Software. Venezuela, 2004.
- Romayna, Aliaga. Auditoria de la calidad. 2012.
- NC. Ingeniería de Software—Calidad del producto— Parte 1: Modelo de la Calidad, Ciudad de la Habana-Cuba, 2005.
- Programación orientada a objetos, Programación orientada a objetos [http://java.ciberaula.com/articulo/tecnologia\\_orientada\\_objetos](http://java.ciberaula.com/articulo/tecnologia_orientada_objetos).

- S., Viridiana Figueroa. Slideshare. 2011. <http://www.slideshare.net/viri88/ingenieria-de-dominio>.
- Scalone, Fernanda. Estudio Comparativo de los Modelos y Estándares de Calidad de Software. Buenos Aires, 2006.
- Universidad Tecnológica, Herramienta Computacional para la Evaluación de Calidad de Productos Software Enmarcados en Actividades de Investigación, Pereira 2011.
- UCI. 0516\_Roles y responsabilidades.
- UCI. Elaboración y aprobación de los procedimientos y lineamientos para la actividad productiva.
- Vargas, Laura Silvia. MECHDAV: Un modelo y su herramienta para la evaluación técnica de la calidad de las herramientas RAD para ambientes visuales. México, 2006.
- Vega, Juan Pablo Carvallo. Introducción a los modelos de calidad del software. 2011.
- Villalva de Benito, María Teresa. Metodología de desarrollo de modelos de calidad orientados a dominio y su aplicación al dominio de los productos finales de seguridad de tecnologías de la información. Alcalá de Henares, 2009.

## Anexos

### Anexo 1: Fases del ciclo de vida del proyecto.

- Estudio preliminar
- Modelación del negocio
- Requisitos
- Análisis y diseño
- Implementación
- Pruebas internas
- Pruebas de liberación
- Despliegue
- Soporte

### Anexo 2: Subsistema DIPP



Anexo3: Subsistema Pozo

Operaciones				
Inventarios (4)	Obligatorias (10)	Opcionales (12)	Geólogo (3)	Otras (4)
Barrenas	Reportes Operativos	Aditivos Añadidos	Topes y Bases	Cambio de Intervalo
Motores de Fondo	Metraje	Pérdidas Fluido Perforación	Descripción Litológica	Cambio de BHA
Jar	Actividades	Torque y Arrastre	Última Descripción	Costo Inicial
Productos Químicos	Costos	Caudal Reducido		Capacidad Combustible
	Consumo de Combustible	Lectura de Gas		
	Propiedades del Fluido	Inclinometría		
	Control de Sólidos	Velocidad Ascensional		
	Horas Acumuladas del Jar	Código de Desgaste		
	Horas Acumuladas del Motor de Fondo	Tiempos Perdidos		
	Pronósticos y Comentarios	Necesidades		
		Abastecimiento Combustible		
		Transferencia Combustible		

Sistema de Manejo Integral de Perforación de Pozos Petroleros (SIPP)