



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**  
**Centro Nacional de Calidad Software**  
**CALISOFT**

**Catálogo automatizado de métricas de calidad para evaluar los  
productos en las pruebas**

Trabajo final presentado en opción al título de  
Máster en Calidad de Software

**Autor:** Ing. Asnier Enrique Góngora Rodríguez

**Tutor:** Dra. Ailyn Febles Estrada

**La Habana, Diciembre de 2011**

“Cuando pueda medir lo que está diciendo y expresarlo con números, ya conoce algo sobre ello; cuando no pueda medir, cuando no pueda expresar lo que dice con números, su conocimiento es precario y deficiente: puede ser el comienzo del conocimiento, pero en sus pensamientos, apenas está avanzando hacia el escenario de la ciencia”.

**Lord Kelvin**

**Lecturas y enfoques populares**

**1889**

## **AGRADECIMIENTOS**

A mis padres Isabel y Felipe por apoyarme en todo momento y siempre confiar en mi.

A mi hermano Felix por aconsejarme siempre que lo necesito.

A mis amigos Yoendris (El negro), Delvis (loco), Yoanis (Cabo) y Roig por brindarme siempre su ayuda.

A mi tutora Ailyn por su ayuda en el desarrollo de la tesis.

A mis compañeros de trabajo Yaneida, Tayché, Yadira, Heidy y a todos los demás por brindarme su ayuda.

A todos los que me han apoyado y dado ánimo, muchas gracias.

## DECLARACIÓN JURADA DE AUTORÍA

Declaro por este medio que yo Asnier Enrique Góngora Rodríguez, con carné de identidad 84020621486, soy el autor principal del trabajo final de maestría “Catálogo de métricas de calidad para evaluar los productos en las pruebas de liberación”, desarrollada como parte de la Maestría en Calidad del Software y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Ing. Asnier Enrique Góngora Rodríguez  
Autor

---

Dra. Ailyn Febles Estrada  
Tutora

## RESUMEN

En muchas áreas de la gestión, se proponen las mediciones como una herramienta eficaz para ayudar en la obtención del éxito de proyectos de software y sistemas a partir de cuatro razones básicas conocidas: caracterizar, evaluar, predecir y mejorar.

La medición es considerada como una eficaz herramienta en las pruebas a los software, es la base para: detectar las desviaciones del rendimiento aceptable en los procesos y producto de software, y las oportunidades de mejora, identificar y priorizar las principales preocupaciones, dar seguimiento a la solución y mejorar la calidad del producto.

En el Laboratorio Industrial de Pruebas de Software (LIPS) del Departamento de Pruebas de Software del Centro Nacional de Calidad Software (CALISOFT) no se evalúan los productos de software por las características de calidad debido a la carencia de mediciones realizadas a los mismo y la retroalimentación que existe del cumplimiento de las características de calidad hacia los directivos de CALISOFT no es la mejor, lo que influye negativamente en la opinión del cliente sobre el producto de software.

Esta tesis propone un sistema de métricas guiados por las características de calidad según el modelo de calidad que plantea la ISO 9126-1:2005 Modelo de Calidad para las pruebas desarrolladas en CALISOFT a los productos de los proyectos desarrollados en la Universidad de las Ciencias Informáticas (UCI). Se describe la vía de definición de los objetivos de medición, indicadores, medidas a recopilar y sus respectivos procedimientos de recolección, almacenaje y verificación, basándose en normas, estándares y modelos de calidad recomendados a nivel internacional.

**Palabras clave:** Medición, Métricas, pruebas de software.

## ÍNDICE

INTRODUCCIÓN .....	1
FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE .....	7
1.1 Introducción.....	7
1.2 Calidad del software.....	7
1.3 Métricas de software. ....	8
1.3.1 Diferentes enfoques de las métricas. ....	9
1.3.2 Clasificación de las métricas de software.....	10
1.4 Estándares, normas y modelos para el desarrollo de métricas.....	11
1.4.1 Modelos de calidad que definen métricas. ....	12
1.5 Metodología para el desarrollo de métricas de calidad.....	20
1.5.1 PSM: Medición Práctica de Software. ....	21
1.5.2 GQM: Objetivo-Pregunta-Métrica.....	22
1.5.3 IEEE 1061-1998 Estándar para metodología de Métricas de calidad de software.....	23
1.6 La medición en las pruebas de software.....	24
1.7 Herramientas para la gestión de las métricas.....	26
1.8 Principio de Pareto o Regla 80/20.....	29
1.9 Conclusiones del Capítulo.....	30
PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD .....	31
2.1 Introducción.....	31
2.2 Métodos y técnicas utilizados. ....	31
2.3 Establecer los requerimientos de calidad del software. ....	32
2.4 Identificar métricas de software. ....	36
2.5 Procedimiento para utilizar el catálogo de métricas en la pruebas del LIPS. ....	44
2.6 Información para la recolección de los datos, el cálculo y análisis de las métricas.....	47
2.7 Herramienta para la gestión de las métricas.....	49
2.8 Conclusiones del Capítulo.....	55
IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA .....	56
3.1 Introducción.....	56
3.2 Validación de la propuesta. ....	56
3.2.1 Implementación del catálogo de métricas. ....	56
3.2.2 Recolección de los datos en los proyectos.....	57
3.3 Cumplimiento de las características de calidad en las pruebas de liberación. ....	58
3.3.1 Cálculo de las Métricas de Calidad. ....	58
3.3.2 Resultados de las métricas. ....	58
3.3.3 Análisis de los resultados.....	60
3.4 Conclusiones del capítulo.....	65
CONCLUSIONES .....	66
RECOMENDACIONES .....	67
REFERENCIAS BIBLIOGRÁFICAS .....	68
BIBLIOGRAFÍA .....	70
ANEXOS.....	74
GLOSARIO DE TÉRMINOS Y SIGLAS.....	101

## ÍNDICE DE TABLAS

Tabla 1: Comparación entre herramientas para la gestión de las métricas.....	28
Tabla 2: Métricas por tipo de pruebas. ....	44
Tabla 3: Roles de la herramienta para la gestión de las métricas.....	49
Tabla 4: Variables de la métrica de idoneidad. ....	57
Tabla 5: Resultados del cálculo de las métricas por proyecto. ....	58
Tabla 6: Resultados de las métricas de Funcionalidad.....	59
Tabla 7: Resultados de las métricas de Confiabilidad. ....	59
Tabla 8: Resultados de las métricas de Usabilidad. ....	60
Tabla 9: Resultado de la métrica de Eficiencia. ....	60
Tabla 10: Métricas del proyecto A y B en la tercera iteración de las pruebas de liberación.....	62
Tabla 11: Métrica externa de la subcaracterística de calidad Idoneidad.....	78
Tabla 12: Métrica externa de la subcaracterística de calidad exactitud. ....	80
Tabla 13: Métrica externa de la subcaracterística de calidad Seguridad. ....	81
Tabla 14: Métrica externa de la subcaracterística de calidad Funcionalidad.....	83
Tabla 15: Métrica externa de la subcaracterística de calidad Madurez.....	84
Tabla 16: Métrica externa de la subcaracterística de calidad Tolerancia ante fallos....	87
Tabla 17: Métrica externa de la subcaracterística de calidad Recuperabilidad. ....	88
Tabla 18: Métrica externa de la subcaracterística de calidad Comprensibilidad. ....	90
Tabla 19: Métrica externa de la subcaracterística de calidad Atracción.....	91
Tabla 20: Métrica externa de la subcaracterística de calidad Instructibilidad. ....	92
Tabla 21: Métrica externa de la subcaracterística de calidad Eficiencia. ....	92
Tabla 22: Métrica externa de la subcaracterística de calidad Comportamiento en el tiempo. ....	93
Tabla 23: Métrica externa de la subcaracterística de calidad Rendimiento.....	94
Tabla 24: Métrica externa de la subcaracterística de calidad Adaptabilidad. ....	95
Tabla 25: Métrica externa de la subcaracterística de calidad Instalabilidad.....	96

## ÍNDICE DE FIGURAS

Figura 1: Métricas de software (Tomado de [7]).	8
Figura 2: La calidad en el ciclo de vida (Tomado de [8]).	13
Figura 3: Modelo para la calidad interna y externa (Tomado de [8]).	14
Figura 4 Representación del Modelo de Boehm (Tomado de [19]).	20
Figura 5: Estructura de PSM (Tomado de [5]).	21
Figura 6: Fases de GQM (Tomado de [20]).	22
Figura 7: Marco de trabajo de los requerimientos de calidad (Creación propia).	36
Figura 8: Interfaz de usuario del módulo Administración (Creación propia).	50
Figura 9: Interfaz de usuario del módulo Configuración. Vista general (Creación propia).	51
Figura 10: Interfaz de usuario del módulo Configuración. Visualizar artefacto (Creación propia).	52
Figura 11: Interfaz de usuario del módulo Gestión de proyectos. Crear proyecto (Creación propia).	53
Figura 12: Interfaz de usuario del módulo Gestión de proyectos. Asignar métricas (Creación propia).	53
Figura 13: Interfaz de usuario del módulo de Reportes (Creación propia).	54
Figura 14: Interfaz de usuario del módulo Mi Cuenta (Creación propia).	54
Figura 15: Diagrama de Despliegue (Creación propia).	55
Figura 16: Proyectos seleccionados para calcular las métricas (Creación propia).	57
Figura 17: Comportamiento de las métricas en la primera iteración de pruebas (Creación propia).	61
Figura 18: Comportamiento de las métricas en la segunda iteración de pruebas (Creación propia).	61
Figura 19: Comportamiento de las métricas en la tercera iteración de pruebas (Creación propia).	61
Figura 20: Comportamiento de las métricas en las pruebas (Creación propia).	62
Figura 21: Tabulación de resultados (Creación propia).	63
Figura 22: Resultados del nivel de calidad del proyecto C (Creación propia).	63
Figura 23: Resultados del nivel de calidad de Funcionalidad (Creación propia).	63
Figura 24: Resultados del nivel de calidad de Confiabilidad (Creación propia).	64
Figura 25: Resultados del nivel de calidad de Usabilidad (Creación propia).	64



## INTRODUCCIÓN

## INTRODUCCIÓN

En el mundo actual existe mucha demanda de software y al mismo tiempo aumentan los clientes que lo exigen con mayor calidad. Todo esto implica que tanto en el proceso de elaboración del software como en las pruebas que se le realicen al producto final deben de desarrollarse con la mayor calidad posible para que el usuario final quede totalmente satisfecho con el producto adquirido.

Desde los 90s en adelante se les puede ver como la era de la calidad. Con el estado del arte actual, en que la tecnología permite proveer abundante funcionalidad, los clientes demandan alta eficacia. La demanda de la calidad se intensifica dado la creciente dependencia de la sociedad actual del software. En esta era, la calidad ha sido llevada al centro del proceso de desarrollo del software. Desde el punto de vista de los comercializadores de software, la calidad ya no es un factor de ventaja en el mercado, se ha convertido en una condición necesaria si la compañía productora quiere competir con éxito.

Para lograr una mejor calidad en el software sin duda hay que tener en cuenta las normas, estándares y modelos de calidad, la mayoría de ellos tienen en cuenta las métricas ya que estas proporcionan gran ayuda para lograr un mejor desempeño en el desarrollo del software, como del producto final.

A partir de la importancia dada a los temas de calidad, en el mundo han surgidos varios laboratorios de pruebas que dentro de los servicios que prestan están las pruebas a los software. Dentro de ellos podemos mencionar al Centro de Ensayos de Software que se encuentra en Uruguay [1], Testhouse en España fundada en Febrero del año 2000 y ha conseguido un premio por lograr un 24% de crecimiento desde su creación [2], E-Quality, consorcio mexicano de empresas especializadas en pruebas de software y Firmware con una oferta integral de servicios dentro de los que se incluyen las pruebas de software [3].

Según la IEEE en su Glosario Estándar de Terminología de Ingeniería de Software (Standard Glossary of Software Engineering Terminology) una métrica es:

“una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [4].

En varias áreas de la gestión, se proponen las mediciones como una herramienta eficaz para ayudar en la obtención del éxito de proyectos de software y sistemas a partir de cuatro razones básicas conocidas para la medición en procesos de desarrollo

## INTRODUCCIÓN

de software, productos y recursos: caracterizar (o entender), evaluar (o controlar), predecir y mejorar [5].

La medición es utilizada en las actividades de la Gestión de Proceso, constituyendo una base para detectar desviaciones de un rendimiento aceptable y oportunidades de mejora de procesos [6]. Dentro de la Gestión de Proyecto es útil para evitar que los líderes asuman un riesgo significativo en el establecimiento y cumplimiento de los compromisos; a través de la identificación y corrección de problemas a tiempo, el seguimiento de los objetivos específicos del proyecto y la estimación y planeación adecuada, entre otros factores.

En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos [7].

En los laboratorios de pruebas antes mencionados se utilizan métricas en el proceso de pruebas que les ayuda a indicar y mejorar la calidad de los productos [2], y a establecer una línea base para la estimación en la planificación de las pruebas. Se identifican un conjunto de posibles riesgos a tener en cuenta en futuras pruebas a realizarse en el software. Utilizan atributos de calidad para medir la calidad de los artefactos que se liberen en las pruebas realizadas [3].

Las métrica de calidad en la pruebas ayuda en gran medida a mejorar la planificación de las mismas, a medir el cumplimiento de los requerimientos del software que se establecieron como indicadores en el proceso, a mejorar los tiempos de ejecución de las pruebas, a controlar y evaluar a los probadores y hasta el mismo producto final. En el proceso del cálculo de las métricas se pueden utilizar herramientas que ayuden a gestionar las métricas, estas agilizan la recogida de los datos para el cálculo de las métricas que en ocasiones se vuelven muy engorrosos y complicados para los especialistas que necesitan el resultado de las métricas.

En Cuba las empresas nacionales se empeñan cada vez más en obtener productos de buena calidad para incluirse en la competencia y cubrir las solicitudes crecientes y urgentes de los clientes. En el año 2002 se crea la Universidad de las Ciencias Informáticas (UCI) la cuál tiene la misión de ser una “universidad innovadora de excelencia científica, académica y productiva que forma de manera continua profesionales integrales comprometidos con la patria, soporte de la informatización del

## INTRODUCCIÓN

país y la competitividad internacional de la industria cubana del software”, en estos 9 años de creada ha alcanzado un papel importante en la Industria de Software Nacional, y va tomando un espacio en la internacional.

Dentro de la Universidad de las Ciencias Informáticas se encuentra el Centro Nacional de Calidad Software (CALISOFT) que dentro de sus funciones está brindar el servicio de pruebas a los productos que se confeccionen en la universidad, así como en el resto de país. Dentro de CALISOFT se encuentra el Laboratorio Industrial de Pruebas de Software (LIPS) donde se realizan las pruebas a todos los productos de los proyectos que soliciten este servicio de la Universidad y empresas del país.

Como resultados de las encuestas realizadas a especialistas de calidad del Departamento de pruebas de software se concluye que actualmente al realizarle las pruebas a los proyectos que las soliciten no se comprueba en que medida los artefactos liberados cumplen con las características de calidad establecidas en la organización esto puede traer consigo que los artefactos lleguen a manos del cliente sin tener en cuenta algunos de las características de calidad que deben de tener los software y a la vez puede influir en la opinión del cliente del proceso de pruebas y del producto final.

Mediante la observación que se realizó del proceso de pruebas que se lleva a cabo en el LIPS se puede concluir que en las pruebas no se evalúan los atributos de calidad según la ISO 9126 establecidos por la organización a tener en cuenta en la revisión de los diferentes productos que se encuentren en las pruebas de liberación. Solo se tienen en cuenta en la reunión de cierre de las pruebas la cantidad de No Conformidades encontradas al producto, dando una referencia del estado del producto al final de la liberación.

Además la retroalimentación que existe con los directivos de la universidad para saber cuan bien se está desarrollando los productos y como se están evaluando las características de calidad en el proceso de prueba de todos los productos que se elaboran en los proyectos productivos no es la mejor, ya que la información que se le proporciona relacionado con el cumplimiento de los atributos de calidad según la ISO 9126-1 Parte 1: Modelo de Calidad no es la más abarcadora según entrevistas realizadas.

## INTRODUCCIÓN

La situación planteada con anterioridad lleva al siguiente **Problema Científico**: ¿Cómo identificar la efectividad de las pruebas del Laboratorio Industrial de Pruebas de Software en CALISOFT?

Para resolver el problema de investigación planteado se propone la siguiente **Hipótesis**: Un catálogo automatizado de métricas de calidad basado en la norma ISO 9126-1 Parte 1: Modelo de Calidad para aplicar en el proceso de pruebas contribuye a obtener medidas de la efectividad del proceso efectuado.

Para demostrar la hipótesis y resolver el problema de investigación expuesto se trazó el siguiente **Objetivo General**: Desarrollar un catálogo automatizado de métricas de calidad basado en la norma ISO 9126-1 Parte 1: Modelo de Calidad que aplicando en el proceso de pruebas permita evaluar la efectividad de las mismas.

Para cumplir con este objetivo se definen los siguientes **Objetivos Específicos**:

1. Analizar las normas, estándares, modelos más significativos con métricas de calidad.
2. Seleccionar métricas de calidad del producto de software en el proceso de prueba de liberación, atendiendo al estado del arte y las condiciones propias del LIPS.
3. Elaborar un catálogo resumido de las métricas de calidad que permita una mejor interpretación de las mismas.
4. Desarrollar una herramienta de software que gestione las métricas de calidad.
5. Validar las métricas y la herramienta obtenida en proyectos que se encuentren en prueba de liberación.

El **Objeto de estudio** de esta investigación es: El proceso de pruebas en el Laboratorio Industrial de Pruebas de Software.

Para lograr el Objetivo general, la investigación se desarrolla concretamente sobre el siguiente **Campo de acción**: Las métricas de calidad basadas en las características y sub-características de la ISO 9126-1 Parte 1: Modelo de Calidad.

El **aporte práctico** en la presente investigación es un catálogo de mediciones y métricas de calidad basadas en las características y sub-características de calidad que plantea la ISO 9126-1 Parte 1: Modelo de Calidad que facilite la evaluación de los productos de software en las pruebas realizadas y una herramienta automatizada que permita gestionar las métricas de calidad para agilizar el proceso de las mismas;

además sirven para obtener indicadores para evaluar los productos de los proyectos que se liberen en el LIPS.

La investigación está orientada de manera general por los siguientes métodos científicos:

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

- Histórico-Lógico: para el estudio crítico de los trabajos anteriores, y para utilizar estos como punto de referencia y comprobación de los resultados alcanzados.
- Análisis y Síntesis: para descomponer el problema de investigación en elementos por separados y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta.
- Hipotético-Deductivo: para la elaboración de la hipótesis central de la investigación y para poner nuevas líneas de trabajo a partir de los resultados parciales.

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

- Entrevista: para precisar en que medida se retroalimentan los directivos de la universidad del resultado de las pruebas que se realizan en el departamento de pruebas de software.
- Encuesta: para detectar los problemas que existen en el departamento de pruebas de software relacionados con el cumplimiento de las características de calidad.
- Observación: para obtener información de las discusiones en secciones científicas y encuentros que se realicen en entornos que aporten información para la investigación.
- El grupo focal: para realizar discusiones con los especialistas del departamento de prueba de software con el objetivo de definir los requerimientos de calidad para la definición y selección de las métricas de calidad.

El presente documento está estructurado en tres capítulos:

## INTRODUCCIÓN

En el **Capítulo 1** se corresponde con la fundamentación teórica, donde se describe y conceptualiza el proceso de medición y calidad del software, y se presentan los enfoques fundamentales utilizados en la actualidad, así como su marco de integración. Se describe varias normas, estándares y modelos que tienen presente las métricas para medir la calidad del software, así como un análisis del estado actual de la medición en los ámbitos, internacional, nacional y en el marco de CALISOFT.

En el **Capítulo 2** se presentan las métricas seleccionadas para aplicar en las pruebas a los productos de software en revisión. Se definen un conjunto de métricas de calidad utilizando una metodología, de la misma forma se recolecta la información necesaria para el cálculo de las métricas. Así como una descripción de la herramienta para la gestión de las métricas confeccionada para apoyar este proceso.

En el **Capítulo 3** se describe la puesta en práctica y validación de la propuesta. Se detallan las actividades realizadas para el cálculo de las métricas definidas y seleccionadas en algunos proyectos que se encuentren en pruebas para de esta forma tener las medidas necesarias para evaluar la calidad de los mismos.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

### 1.1 Introducción.

En este capítulo se profundiza en los principales elementos de los procesos de medición descritos, sus características y definiciones necesarias para realizar y entender la propuesta en cuestión. Se realiza un recorrido comenzando con las normas y entidades que se dedican a la estandarización. Se lleva a cabo una descripción exhaustiva sobre las métricas de software y los distintos niveles organizacionales, así como estándares y modelos a utilizar en la definición e implementación de las métricas. Se hace un análisis del estado actual de las herramientas para la automatización de las métricas en las pruebas.

### 1.2 Calidad del software.

Afirmar que un producto de software tiene calidad pudiera resultar algo complicado teniendo en cuenta la ambigüedad que trae consigo la palabra “calidad”. A pesar de esto hay autores y entidades que brindan sus definiciones. La ISO/IEC 9126 la define como “La totalidad de características de un producto de software que tienen como habilidad, satisfacer necesidades explícitas o implícitas” [8]. Pressman, plantea que calidad del software es “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente” [9]. En la presente investigación se tendrá en cuenta la definición dada por Pressman ya que el autor considera que es la más completa porque tiene en cuenta las necesidades implícitas que debe tener todo producto de software y que el usuario final no tienen conocimiento del mismo, además establece el uso de normas, estándares y modelos establecidos internacionalmente.

Uno de los principales problemas a los que se enfrentan a la hora de hablar de la calidad del software es el siguiente: ¿Es realmente posible encontrar un conjunto de propiedades en un software que nos den un indicador de su calidad? Para dar respuesta a esta pregunta aparecen los Modelos de Calidad, en ellos la calidad se define de forma jerárquica y tienen como objetivo resolver la complejidad mediante la descomposición.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

La Calidad del Software debe implementarse en todo el ciclo de vida del mismo. Las distintas actividades para la implantación del control de calidad en el desarrollo de software son [10]:

1. Aplicación de metodología y técnicas de desarrollo.
2. Reutilización de procesos de revisión formales.
3. Prueba del software.
4. Ajustes a los estándares de desarrollo.
5. Control de cambios, mediciones y recopilación de información.
6. Gestión de informes sobre el control de calidad.

### 1.3 Métricas de software.

Aunque los términos medida, medición y métricas se utilizan a menudo indistintamente, existe gran confusión a la hora de referenciarse. Dentro del contexto de la ingeniería del software, una medida “proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto” [7]. La medición “es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas” [8], también es definida como el acto de determinar una medida.

El IEEE en su Glosario Estándar de Terminología de Ingeniería de Software (Standard Glossary of Software Engineering Terminology) define métrica como: “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [4].

Las métricas de software proveen la información necesaria para la toma de decisiones técnicas. En la figura 1 se ilustra una extensión de esta definición para incluir los servicios relacionados al software como la respuesta a los resultados del cliente.

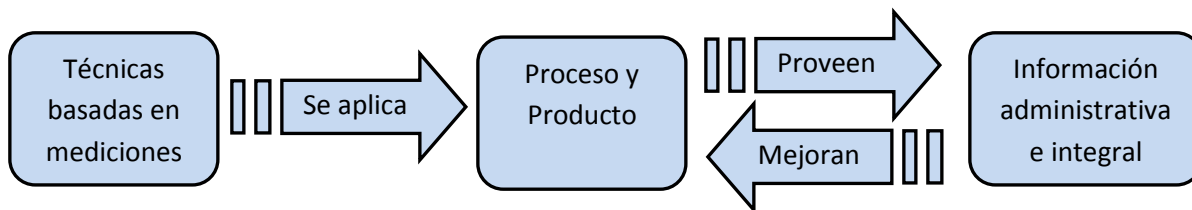


Figura 1: Métricas de software (Tomado de [7]).



## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

Las métricas son la maduración de una disciplina, ayudarán en el diseño de pruebas más efectivas. Es por eso que se propone un proceso de medición, el cual se puede caracterizar por cinco actividades [9]:

- **Formulación:** La obtención de medidas y métricas del software apropiadas para la representación de software en cuestión.
- **Colección:** El mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- **Análisis:** El cálculo de las métricas y la aplicación de herramientas matemáticas.
- **Interpretación:** La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
- **Realimentación:** Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

### 1.3.1 Diferentes enfoques de las métricas.

Muchos investigadores han intentado desarrollar una sola métrica que proporcione una medida completa de la complejidad del software. Aunque se han propuesto docenas de métricas o medidas, no todas proporcionan suficiente soporte práctico para su desarrollo. Algunas demandan mediciones que son demasiado complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas, y otras violan las nociones básicas intuitivas de lo que realmente es el software con buena calidad. Es por eso que se han definido una serie de atributos que deben acompañar a las métricas efectivas de software, por lo tanto la métrica obtenida y las medidas que conducen a ello deben cumplir con las siguientes características fundamentales [12]:

- **Simple y fácil de calcular:** debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no obligará a un esfuerzo o a una cantidad de tiempo inusual.
- **Empírica e intuitivamente persuasiva:** la métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en evaluación (por ejemplo: una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).
- **Consistente en el empleo de unidades y tamaños:** el cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

las variables del lenguaje de programación en el programa resulta una sospechosa mezcla de unidades que no son intuitivamente concluyentes.

- Independiente del lenguaje de programación: las métricas deberían apoyarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deben depender de la sintaxis o semántica del lenguaje de programación.
- Un mecanismo eficaz para la retroalimentación de calidad: la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.
- Consistentes y objetivas: La métrica debería siempre producir resultados sin ambigüedad. Un tercer equipo debería ser capaz de obtener el mismo valor de métrica usando la misma información del software.

### 1.3.2 Clasificación de las métricas de software.

Existen muchas formas de clasificar las métricas del software, distintas unas de otras según los autores. Las métricas del software pueden ser (1) Directas o Indirectas, (2) Primarias o Secundarias, (3) Internas o Externas, (4) Públicas o Privadas, (5) Simples o Complejas, (6) De Proceso, de Producto o de Proyecto, (7) Primitivas o Calculadas, etc.

Pressman clasifica el campo de las métricas en 6 categorías o grupos de métricas distintos:

- Métricas técnicas. Se centran en las características del software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo está hecho, es decir, están centradas en las características del software más que en su proceso de desarrollo.
- Métricas de calidad. Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.
- Métricas de productividad. Referidas al rendimiento del proceso de desarrollo como función del esfuerzo aplicado. Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- Métricas orientadas al tamaño. Es para saber en que tiempo voy a terminar el software y cuantas personas voy a necesitar. Son medidas directas al software y al proceso por el cual se desarrolla.
- Métricas orientadas a la función. Son medidas indirectas del software y del proceso por el cual se desarrolla. Las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.
- Métricas orientadas a la persona. Proporcionan medidas e información sobre la forma que la gente desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que voy a hacer de mi personal que hará el sistema.

En la presente investigación se tendrán en cuenta las métricas de calidad porque son utilizadas una vez que termine de desarrollarse el producto de software, las mismas serán calculadas en las pruebas que se le realicen a los productos de software en el LIPS una vez terminado el producto de software por los equipos de desarrollo de la UCI. Además sirven para evaluar el producto final ante de entregárselo al cliente, teniendo en cuenta los requisitos especificados por el mismo.

### **1.4 Estándares, normas y modelos para el desarrollo de métricas.**

Un estándar es el documento aprobado por consenso por un organismo reconocido, que proporciona reglas, pautas y/o características para uso común, con el objetivo de obtener un óptimo nivel de resultados en un contexto dado [8]. Una norma (tecnología) o estándar es toda actividad documentada que norma el comportamiento de un grupo de personas. Los estándares nos dan los medios para que todos los procesos se realicen siempre de la misma forma, mientras surjan ideas para mejorarlos.

Han sido muchos los departamentos de universidades, organismos de normalización o investigación nacionales o internacionales, sociedades de profesionales, departamentos de defensa, departamentos de calidad y procesos de empresas los que han ido generando normas y estándares. Se considera como entidades de mayor reconocimiento internacional, por sus trabajos y esfuerzos realizados para la normalización, y reconocimiento de la Ingeniería del software a: The International Organization for Standardization (ISO) [13], The Institute of Electrical and Electronics Engineers (IEEE) [14] y Software Engineering Institute (SEI) [15].

#### **1.4.1 Modelos de calidad que definen métricas.**

Los Modelos de Calidad no son más que documentos que integran en sí las mejores prácticas para un desarrollo de software eficiente. Proponen temas de administración en los que cada organización que los aplique debe hacer énfasis e integran diferentes prácticas dirigidas especialmente a procesos clave. De esta forma permite medir la evolución de la calidad durante todo el ciclo de vida del sistema informático.

La ventaja de los modelos de calidad es que la calidad se convierte en algo concreto, que se puede definir, que se puede medir y, sobre todo, que se puede planificar. Los modelos de calidad ayudan también a comprender las relaciones que existen entre diferentes características de un producto de software [14].

##### **➤ ISO/IEC 9126-1 Parte 1: Modelo de Calidad.**

El modelo de calidad que describe la calidad de los productos de software se divide en dos partes: a) calidad externa y calidad interna, b) calidad durante el uso. Se especifican seis características para la calidad interna y externa, que son además divididas en sub-características que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos o cualidades internos del software. Para la calidad durante el uso se definen cuatro características. La calidad durante el uso es el efecto combinado que percibe el usuario de la calidad interna y externa del software.

La calidad de cualquiera de los procesos del ciclo de vida, contribuye a mejorar la calidad del producto, y esta a su vez contribuye a mejorar la calidad en el uso. Por consiguiente, evaluar y mejorar un proceso es un medio para mejorar la calidad del producto; la evaluación y mejora de la calidad del producto son una vía para mejorar la calidad durante el uso (Ver Figura 2). De igual modo, la evaluación de la calidad durante el uso permite la retroalimentación para mejorar un producto, y cuando se produce la evaluación permite la retroalimentación para mejorar un proceso [8].

FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

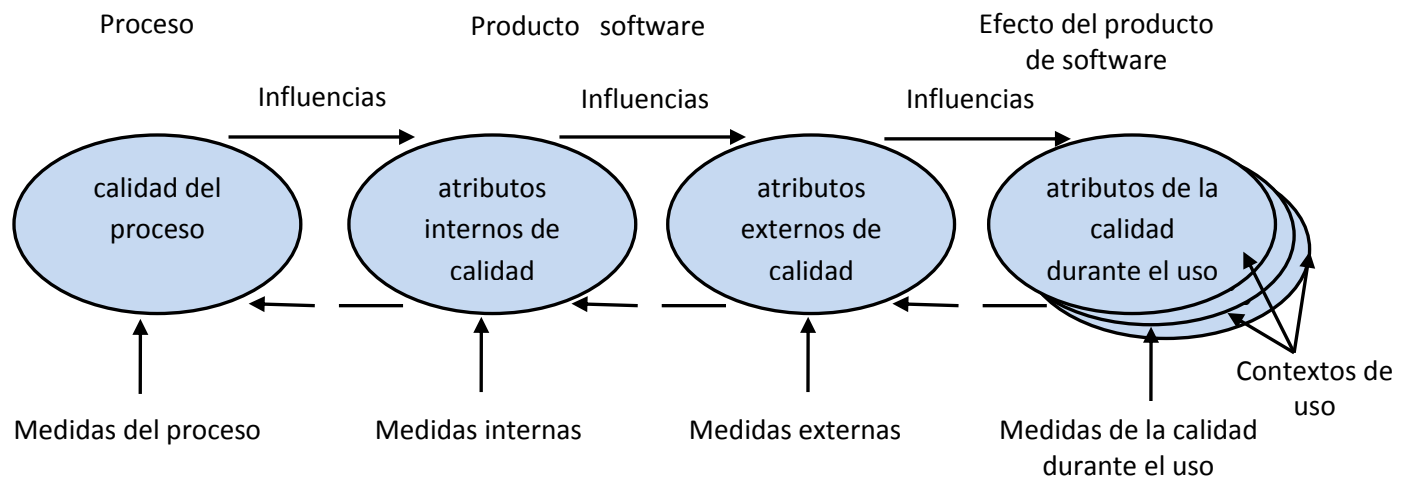


Figura 2: La calidad en el ciclo de vida (Tomado de [8]).

Como muestra la Figura 2, las métricas internas pueden ser aplicadas a los productos intermedios que se desarrollan a lo largo del ciclo de vida de desarrollo de un producto software, tales como solicitud de propuesta, especificación de requisitos, especificaciones de diseño o código fuente. Las métricas internas le proporcionan a los desarrolladores la habilidad de medir la calidad de estos productos intermedios, con lo cual se puede predecir la calidad del producto final. Esto le permite a los desarrolladores identificar los problemas que afecten la calidad e iniciar las acciones correctivas en las etapas tempranas del ciclo de vida de desarrollo del producto.

Por su lado, las métricas externas pueden ser usadas para medir la calidad del producto software a través de la medición del comportamiento del sistema del cual el software forma parte. Las métricas externas solo pueden ser usadas durante las etapas de pruebas del proceso ciclo de vida y durante cualquier otra etapa operacional. Por último, las métricas de calidad en uso (Figura 2) miden si un producto resuelve las necesidades de usuarios específicos para alcanzar metas específicas con eficacia, productividad, seguridad y satisfacción en un contexto específico de uso. Esto solo puede lograrse en un entorno real del sistema.

Este modelo se ha desarrollado en un intento de identificar los atributos más importantes para la calidad interna y externa en un producto software. El modelo identifica seis características claves de calidad donde cada una de ellas se descompone en un conjunto de sub-características como se muestra en la Figura 3 [16].

FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

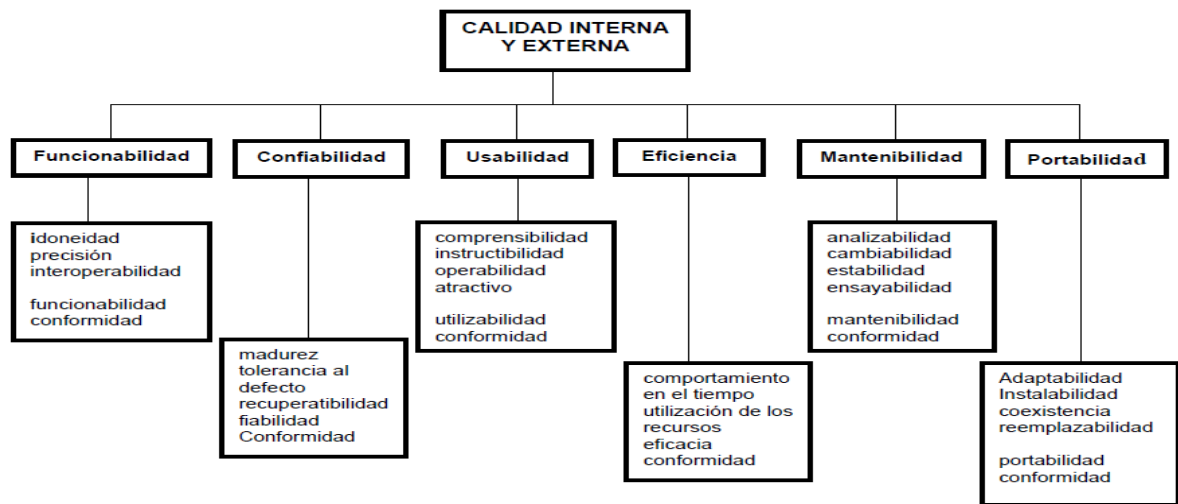


Figura 3: Modelo para la calidad interna y externa (Tomado de [8]).

A continuación se dan las definiciones para cada característica de la calidad y las sub-características del software que influyen en aquellas. Para cada característica y sub-característica, la capacidad del software es determinada por un conjunto de atributos internos que pueden medirse.

- ✓ **Funcionalidad:** es la capacidad del software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuándo el software se usa bajo las condiciones especificadas.
  - **Idoneidad:** capacidad del software para mantener un conjunto apropiado de funciones para las tareas y los objetivos del usuario especificados.
  - **Precisión:** capacidad del software para proporcionar efectos o resultados correctos o convenidos con el grado de exactitud necesario.
  - **Interoperabilidad:** capacidad del producto de software para interactuar recíprocamente con uno o más sistemas especificados.
  - **Seguridad (informática):** capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o pueden modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos.
  - **Conformidad con la funcionalidad:** capacidad del software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la funcionalidad.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- ✓ **Confiabilidad:** la capacidad del producto de software para mantener un nivel de ejecución especificado cuando se usa bajo las condiciones especificadas
  - **Madurez:** capacidad del producto de software de evitar un fallo total como resultado de haberse producido un fallo del software.
  - **Tolerancia ante fallos:** capacidad del producto de software de mantener un nivel de ejecución o desempeño especificado en caso de fallos del software o de infracción de su interfaz especificada.
  - **Recuperabilidad:** capacidad del producto de software de restablecer un nivel de ejecución especificado y recuperar los datos directamente afectados en caso de fallo total.
  - **Conformidad con la confiabilidad:** capacidad del producto de software para adherirse a las normas que se le apliquen, convenciones, regulaciones, leyes y las prescripciones similares relativas a la confiabilidad.
- ✓ **Usabilidad:** capacidad del producto de software de ser comprendido, aprendido, utilizado y de ser atractivo para el usuario, cuando se utilice bajo las condiciones especificadas.
  - **Comprensibilidad:** capacidad del producto de software para permitirle al usuario entender si el software es idóneo, y cómo puede usarse para las tareas y condiciones de uso particulares.
  - **Cognoscibilidad:** capacidad del producto del software para permitirle al usuario aprender su aplicación.
  - **Operabilidad:** capacidad del producto del software para permitirle al usuario operarlo y controlarlo.
  - **Atracción:** capacidad del producto del software de ser atractivo o amigable para el usuario.
  - **Conformidad con la usabilidad:** capacidad del producto de software para adherirse a las normas, convenciones, guías de estilo o regulaciones relativas a la usabilidad.
- ✓ **Eficiencia:** capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados, bajo condiciones establecidas.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- **Rendimiento:** capacidad del producto de software para proporcionar apropiados tiempos de respuesta y procesamiento, así como tasas de producción de resultados, al realizar su función bajo condiciones establecidas.
  - **Utilización de recursos:** capacidad del producto de software para utilizar la cantidad y el tipo apropiado de recursos cuando el software realiza su función bajo las condiciones establecidas.
  - **Conformidad de la eficiencia:** capacidad del producto de software de adherirse a las normas o convenciones que se relacionan con la eficiencia.
- ✓ **Mantenibilidad:** capacidad del producto de software de ser modificado. Las modificaciones pueden incluir las correcciones, mejoras o adaptaciones del software a cambios en el ambiente, así como en los requisitos y las especificaciones funcionales.
- **Diagnosticabilidad:** capacidad del producto del software de ser objeto de un diagnóstico para detectar deficiencias o causas de los fallos totales en el software, o para identificar las partes que van a ser modificadas.
  - **Flexibilidad:** capacidad del producto del software para permitir la aplicación de una modificación especificada.
  - **Estabilidad:** capacidad del producto de software para minimizar los efectos inesperados de las modificaciones realizadas al software.
  - **Contrastabilidad:** capacidad del producto del software para permitir la validación de un software modificado.
  - **Conformidad de la mantenibilidad:** capacidad del producto de software para adherirse a las normas o convenciones que se relacionan con la mantenibilidad.
- ✓ **Portabilidad:** capacidad de producto de software de ser transferido de un ambiente a otro.
- **Adaptabilidad:** capacidad del producto de software de ser adaptado a los ambientes especificados sin aplicar acciones o medios de otra manera que aquellos suministrados con el propósito de que el software cumpla sus fines.



## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- **Instalabilidad:** capacidad del producto de software de ser instalado en un ambiente especificado.
- **Coexistencia:** capacidad del producto de software de coexistir con otro software independiente en un ambiente común y compartir los recursos comunes.
- **Remplazabilidad:** capacidad del producto de software de ser usado en lugar de otro producto de software especificado para los mismos fines y en el mismo ambiente.
- **Conformidad con la portabilidad:** Capacidad del producto de software de adherirse a las normas o convenciones relativas a la portabilidad.

### ➤ ISO/IEC 9126-2 Parte 2: Métricas Externas.

La ISO 9126 parte 2 contiene un conjunto de métricas externas para el software, sin asignar rangos de valores a las métricas que propone ya que son específicas para cada producto, en dependencia de su categoría, nivel de integridad y necesidad del usuario final. Las métricas están distribuidas por las 6 características que recoge la parte 1 modelo de calidad.

Estas métricas pueden ser usadas por los siguientes usuarios [17].

- Comprador (una persona u organización que procura comprar un sistema, producto software o servicio de software de un proveedor)
- Evaluador (una persona u organización que realiza una evaluación, puede ser un laboratorio de pruebas, el departamento de calidad de una organización de desarrollo de software, una organización gubernamental o un usuario)
- Desarrollador (una persona u organización que realiza actividades de desarrollo, incluyendo el análisis de requerimientos, diseño y prueba a través de la aceptación durante el proceso de software de ciclo de vida)
- Mantenimiento (una persona u organización que realiza actividades de mantenimiento)
- Proveedor (una persona u organización que realiza un contrato con el comprador para el suministro de un sistema, servicio de producto de software según los términos del contrato) al validar la calidad del software en la prueba.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- **Usuario** (una persona u organización que utiliza el producto de software para realiza una función específica) en la evaluación de la calidad del producto de software en prueba de aceptación.

El uso de las métricas externas que contiene esta norma en el Laboratorio Industrial de Pruebas de Software no representará mucho esfuerzo para los especialistas que la utilicen ya que en su mayoría son fáciles de entender e interpretar. Las fórmulas utilizadas para el cálculo de las métricas no son complicadas ya que no contienen muchas operaciones en las mismas. El uso de las métricas no representan un obstáculo en el proceso de pruebas en al Laboratorio Industrial de Pruebas de Software, sino que ayudará en la evaluación de los productos y en la planificación y control de las pruebas.

### ➤ **McCall**

El modelo de Jim McCall, desarrollado inicialmente para la Fuerza Aérea de los EE.UU en 1977, es uno de los más renombrados actualmente. Este modelo busca reducir la brecha entre usuarios y desarrolladores enfocándose en un número de factores de calidad que reflejen las prioridades de ambos. El modelo establece una jerarquía de Perspectivas (3), Factores (11), Criterios de Calidad (23) y Métricas (41).

El modelo establece tres perspectivas para definir e identificar la calidad de un producto software [18]:

#### ✓ **Revisión del producto**

- **Mantenibilidad:** es el esfuerzo requerido para localizar y arreglar programas.
- **Flexibilidad:** es el esfuerzo requerido para modificar un sistema operativo.
- **Verificabilidad** (testability): es el esfuerzo requerido para probar un programa.

#### ✓ **Transición del producto**

- **Portabilidad:** es el esfuerzo requerido para transferir un software de un hardware o un entorno de sistemas a otro.
- **Reusabilidad** (en otro contexto): es el grado en que un programa (o partes de un programa) se puede reutilizar en otro.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- **Interoperabilidad:** es el esfuerzo requerido para asociar un programa a otro.

### ✓ Operaciones del producto

- **Corrección** (cumplimiento con la especificación): mide el grado en que un programa satisface sus especificaciones y consigue los objetivos del usuario.
- **Confiabilidad:** mide el grado en que se puede esperar que un programa lleve a cabo sus funciones esperada con la precisión requerida.
- **Eficiencia** (De ejecución y de almacenamiento): mide la cantidad de recursos de computadora y de código requerido por un programa para que lleve a cabo las funciones especificadas.
- **Integridad** (protección contra accesos no autorizados): es el grado en que puede controlarse el acceso al software o a los datos por personal no autorizado.
- **Usabilidad:** es el esfuerzo requerido para aprender un programa e interpretar la información de entrada y de salida.

Los 23 criterios de calidad del modelo son: Facilidad de operación, Facilidad de comunicación, Facilidad de aprendizaje, Control de accesos, Facilidad de auditoría, Eficiencia en ejecución, Eficiencia en almacenamiento, Precisión, Consistencia, Tolerancia a fallos, Modularidad, Simplicidad, Completitud, Trazabilidad, Auto descripción, Capacidad de expansión, Generalidad, Instrumentación, Independencia del S.O., Independencia del HW, Compatibilidad de comunicaciones, Compatibilidad de datos y Concisión.

### ➤ Modelo de Boehm

Barry Boehm, un conocido y además precursor ingeniero de software de los años 70 también realizó sus aportes propios al tema de la calidad del software. El modelo que propone es similar al de McCall al representar la estructura de características de calidad de forma jerárquica, contribuyendo cada una de ellas a la calidad total. En este modelo también están consideradas las necesidades del usuario, pero se encuentran características de calidad en el terreno del hardware que no están presentes en el otro modelo representado anteriormente [19]. En la Figura 4 se representa este modelo.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

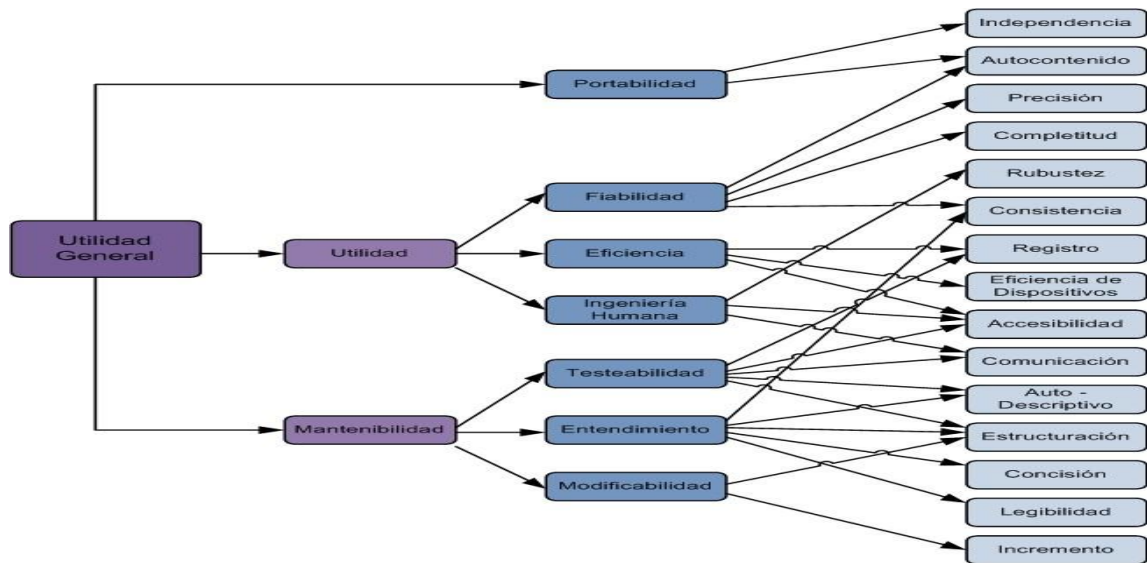


Figura 4 Representación del Modelo de Boehm (Tomado de [19]).

A consideración del autor se utiliza en la presente investigación la ISO/IEC 9126-1 Parte 1: Modelo de Calidad ya que es un modelo completo porque recoge la mayoría de las características de calidad de los demás modelos de calidad y es establecido por CALISOFT a utilizar en las pruebas de liberación. Además las características de calidad que plantea se ajustan a los tipos de pruebas que se aplican actualmente en el departamento de pruebas de software por los especialistas al frente de las pruebas de liberación. De esta forma se podrá definir o seleccionar las métricas de calidad para observar el cumplimiento de las características de calidad evaluando los productos de software que están en pruebas o que se encuentren liberados en el LIPS.

### 1.5 Metodología para el desarrollo de métricas de calidad.

La metodología para métricas de calidad del software es un acercamiento sistemático al establecimiento de requerimientos de calidad e identificación, implementación, análisis y validación de métricas de procesos y productos de software para un sistema de software.

De forma general para definir una métrica el primer paso a seguir es la documentación del proceso de desarrollo mediante la recopilación de datos previamente identificados. Debe definirse además un procedimiento para la recolección de los mismos. Deben establecerse metas y las métricas necesarias para alcanzarlas. Han de implantarse las herramientas necesarias para el análisis de las métricas. Se crea una base de datos para archivar toda la información recolectada además de las propias métricas.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

Finalmente se define un mecanismo de retroalimentación. Para efectuar todo este proceso debe ser conformado un equipo que se encargue únicamente de él según la opinión del autor.

### 1.5.1 PSM: Medición Práctica de Software.

Este marco de referencia es patrocinado por el D.o.D y el Ejército de las EE.UU y describe un proceso de medición dirigido por la información que se orienta a las metas técnicas y de negocio de una organización. PSM está definido por un conjunto de 9 prácticas denominadas principios de medición que ayudan en la medición a nivel del proyecto proporcionando los datos necesarios para dirigirlo con el objetivo de completar costos, cronogramas y objetivos técnicos exitosamente. Estas definen un enfoque de análisis dirigido cuyo fin es brindar la información cuantitativa necesaria que ayude al Gerente del Proyecto en la toma de decisiones relacionadas al software y al sistema. Además da soporte a los requisitos de medición a nivel de organización y provee la información requerida para la identificación de diversos aspectos fuera del alcance de los proyectos. Las funciones de administración de software en sí se agrupan de la siguiente forma [5]:

**Administración de proyecto:** permite la planificación y control de los productos.

**Administración del proceso:** asegura que los procesos se desarrollan de forma correcta y son mejorados.

**Ingeniería del producto:** asegura la satisfacción del cliente y por ende la aceptación del producto.

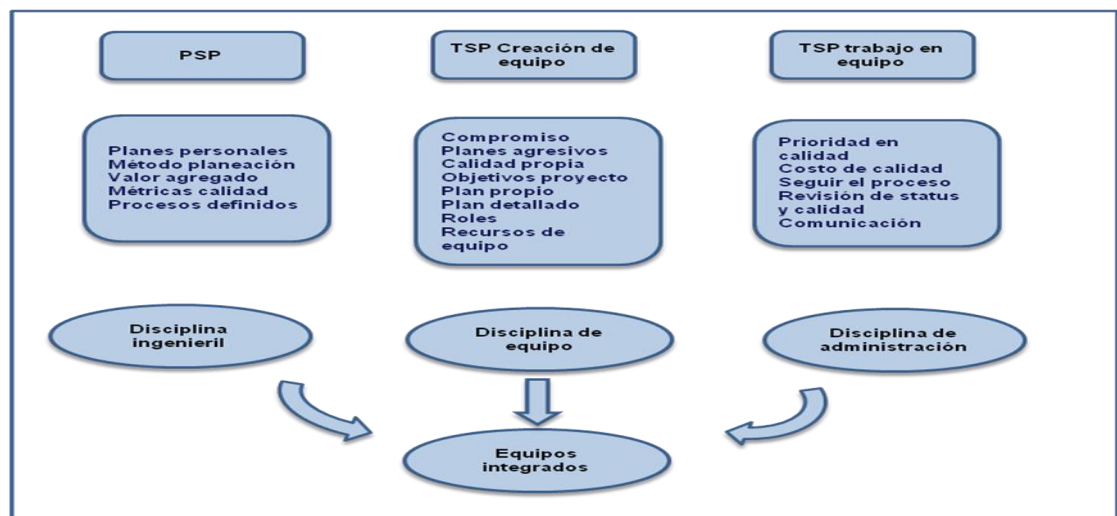


Figura 5: Estructura de PSM (Tomado de [5]).

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

Esta metodología representa las mejores prácticas empleadas por los profesionales en la medición del software. El proceso es flexible adaptándose a los objetivos y necesidades específicas de cada programa.

### 1.5.2 GQM: Objetivo-Pregunta-Métrica.

Desarrollada originalmente por Baili y Weiss en 1984 y extendida en 1990 por Rombach como resultado de muchos años de experiencia e investigación. Se basa en que la medición debe ser realizada siempre orientada a un objetivo. Con GQM este objetivo es definido y refinado mediante preguntas potencialmente medibles que a su vez son respondidas por métricas que aportan toda la información necesaria. Presenta 4 fases: Planificación, Definición, Recopilación de Datos e Interpretación y 3 niveles: Conceptual, Operacional y Cuantitativo (Ver Figura 6). En éstas fases primero se emplean las metas del negocio para llegar a las métricas, luego se recopilan los datos y se analizan con el objetivo de mejorar el proceso de la toma de decisiones [20].

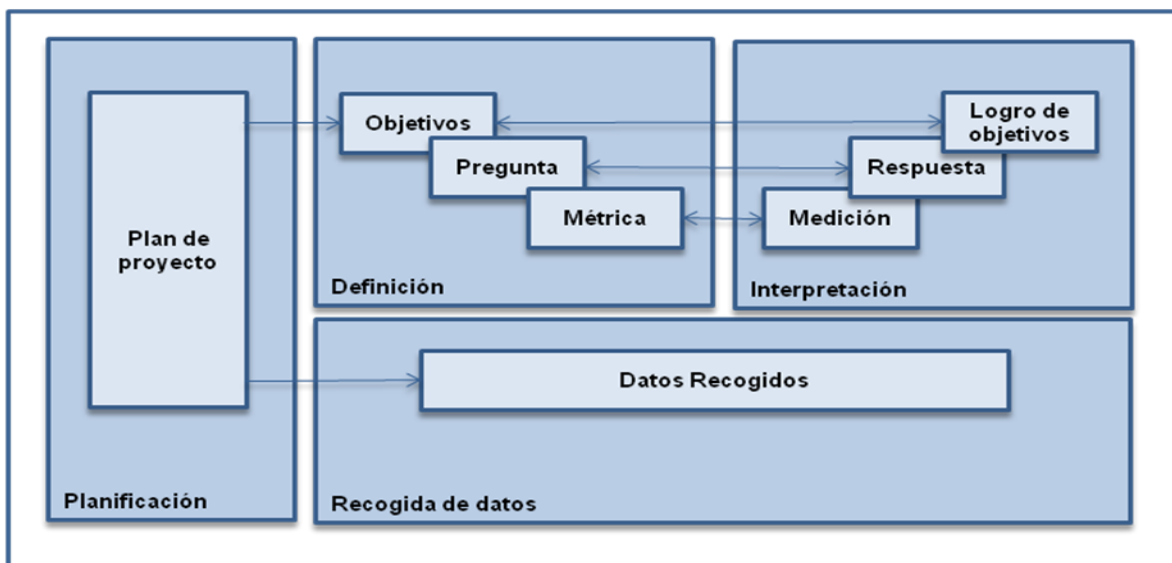


Figura 6: Fases de GQM (Tomado de [20]).

Para su aplicación se crea un equipo independiente al equipo que desarrolla el proyecto y se selecciona el área que se desea mejorar, ya sea del producto o el proceso. En éste el Gerente de Proyecto es el responsable de la continuidad del proceso de medición.

Esta metodología es aplicable a todo el ciclo de vida de software, procesos y recursos. Se alinea fácilmente con el ambiente organizacional y puede ser empleada individualmente por los miembros del equipo de desarrollo.

### **1.5.3 IEEE 1061-1998 Estándar para metodología de Métricas de calidad de software.**

Este estándar se divide en 4 cláusulas: Cláusula 1 Proporciona el alcance de esta norma, Cláusula 2 Proporciona un conjunto de definiciones, Cláusula 3 ofrece una visión general del marco de las métricas de calidad del software y Cláusula 4 proporciona una metodología para la métrica de calidad del software. También en esta norma se incluyen tres apéndices que son para fines de referencia e ilustrativos.

Estándar que provee una metodología para el establecimiento de requerimientos de calidad e identificar, implementar, analizar y validar métricas de calidad del software para procesos y productos. Es aplicable a cualquier software en todas las fases de su ciclo de vida. Este estándar está dirigido a aquellos que se encuentran asociados a la adquisición, desarrollo, uso, soporte, mantenimiento y auditoría de software. Puede ser empleado por: gerentes de proyectos, desarrolladores, auditores y usuarios.

Específicamente el uso de esta metodología para medir la calidad permite a la organización:

- Evaluar el logro de las metas de calidad.
- Establecer los requerimientos de calidad para un sistema a su salida.
- Establecer estándares y criterios de aceptación.
- Evaluar el nivel de calidad alcanzado contra los requerimientos requeridos.
- Detectar anomalías o puntos de problemas potenciales en el sistema.
- Predecir el nivel de calidad que será alcanzado en el futuro.
- Monitorear los cambios en la calidad del software si este es modificado.
- Evaluar la facilidad de cambio en el sistema durante la evolución del producto.

Además establece una metodología para el establecimiento de requisitos de calidad y la identificación, aplicación, análisis y validación de procesos y métricas de producto de software de calidad. Esta metodología se aplica a todo el software en todas las fases de un ciclo de vida del software. La norma está especialmente dirigida a aquellos que se dedican a medir o evaluar la calidad del software.

Este estándar puede ser utilizado por los siguientes roles:

- Director del proyecto para identificar, definir y priorización los requisitos de calidad de un sistema.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

- Un desarrollador de sistemas para identificar las características específicas que deben ser incorporados en el software a fin de satisfacer los requisitos de calidad.
- Administrador de la calidad / auditor y un desarrollador de sistemas le brinda la posibilidad de evaluar los requerimientos de calidad que son el ser de ellos.
- Un encargado del sistema para ayudar a implementar los cambios durante la evolución de los productos.
- Un usuario para ayudar a especificar los requisitos de calidad de un sistema.

Esta metodología es un acercamiento sistemático al establecimiento de requerimientos de calidad e identificación, implementación, análisis y validación de métricas de procesos y productos de software. Comprende 5 pasos:

- Establecer los requerimientos de calidad del software.
- Identificar métricas de software.
- Implementar el sistema de métricas de software.
- Analizar los resultados de las métricas de software.
- Validar las métricas de software.

Estos pasos pueden ser aplicados iterativamente por la visión ganada de la aplicación de un paso que puede mostrar la necesidad de evaluación de los resultados de pasos previos. Cada paso establece las actividades necesarias para completar los resultados indicados [11].

A consideración del autor se utiliza en la presente investigación la IEEE 1061-1998 en su Estándar para metodología de Métricas de calidad de software (Standard for a Software Quality Metrics Methodology) como metodología para la definición de métricas de software por ser una de las más generales, sencillas y explícitas. Además es fácil de implementar para todos los involucrados y la forma de aplicación que propone involucra un menor número de personas, ya que no es específica para determinados roles del equipo de desarrollo del proyecto. El estándar resulta de mucha utilidad en la fase de prueba del producto de software ya que ayuda en gran medida a medir el cumplimiento de las características de de calidad.

### **1.6 La medición en las pruebas de software.**

Varias bibliografías, entre ellas Pressman, mencionan los mecanismos de medición dentro de las actividades de pruebas, y plantean la necesidad de utilizar mediciones



## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

para: supervisar resultados del proyecto, determinar adherencia a normas, identificar causas de un rendimiento insatisfactorio, y de manera general utilizarlas como retroalimentación al aseguramiento de calidad para desarrollar estrategias que mejoren el proceso del software y como consecuencia, la calidad del producto final. Las medidas no sólo se emplean para entender mejor los atributos de los modelos que se crean sino, fundamentalmente, para valorar la calidad de los productos de ingeniería o de los sistemas que se construyen [7], pues "...el uso de métricas reduce la subjetividad en la evaluación de la calidad del software, al proveer una base cuantitativa para tomar decisiones acerca de esta..." [21]. Por tanto, se puede decir que proporcionan una visión interna en el acto, en vez de a posteriori, evitando así que los problemas potenciales se conviertan en defectos catastróficos.

Primeramente se establecieron métricas para identificar contratistas potenciales en la rama del desarrollo de Software. Unido a este objetivo y orientado a la mejora de proceso surgen modelos que soportan la medición entre sus elementos propuestos.

Se crearon múltiples espacios para orientar sobre Calidad, Mejora de Procesos de Producción de Software y gerencia de Proyectos de Software que utilizan las métricas, así como importantes entes de la estandarización y normalización que le han dedicado un espacio a pautar el tema de las mediciones los casos más significativos.

En Cuba, en el orden del desarrollo de la Industria de Software, se desarrollan acciones que involucran las métricas de Software como elemento importante para lograr la calidad de los productos. En este sentido diversos especialistas de la rama y con experiencia en el tema, imparten conferencias y cursos que incluyen en su contenido elementos referentes a estos tópicos [22].

Existe una Red de Investigación: "Red Iberoamericana de Tecnologías de Software para la década del 2000", que integra Grupos de Investigación a nivel internacional con representantes, en universidades cubanas ("Marta Abreu" de Las Villas y "Universidad de La Habana"), y cuyo primer objetivo es potenciar el desarrollo de las tecnologías y su transferencia a la industria y administración. Entre sus prioridades de investigación se encuentran: los Procesos de software y Métricas de software [23].

Por todo lo anteriormente expuesto, se puede decir que en el mundo, y en Cuba en particular, se sigue trabajando intensamente en investigaciones buscando perfeccionar el uso exitoso de las mediciones y métricas en los Productos de Software y la necesidad de su aplicación.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

En CALISOFT se tiene presente en las pruebas que se realizan a los proyectos que solicitan este servicio varias métricas relacionadas principalmente con la estimación de tiempo para la ejecución de los casos de pruebas en las revisiones de las aplicaciones clasificadas por diferentes niveles de complejidad (alto, medio y bajo), al igual que la documentación que se revise en las liberaciones. Estas métricas son utilizadas generalmente para la planificación de las pruebas determinando el tiempo aproximado que deben durar las mismas.

### **1.7 Herramientas para la gestión de las métricas.**

En el mundo las empresas que brindan servicios de calidad cuentan con herramientas y sitios web que le permiten dar a conocer los productos y servicios que ofrecen, además de brindarlos con mayor calidad.

Entre las compañías líderes en los servicios de calidad del software y pruebas, se encuentra la SQS S.A, empresa que cuenta con un portal web [24]. Esta empresa ha alcanzado su objetivo a través de servicios de consultoría, programas de formación y el desarrollo de nuevas herramientas. SQS enfoca sus servicios de consultoría de calidad del software en mejorar y asegurar la calidad de los desarrollos de sus clientes. Entre los servicios que brindan se encuentran los siguientes:

- ✓ Evaluación.
- ✓ Validación y verificación.
- ✓ Laboratorio de pruebas.

SQS desarrolla herramientas para la automatización de procesos de validación y la gestión de requisitos, como [24]:

- ✓ SQS TestWORKFLOW.
- ✓ SQS Q-Val.
- ✓ SQS interCENELEC.

Por otra parte IBM, empresa especializada en el desarrollo de herramientas software, desarrolló el Rational Quality Manager Express Edition. Herramienta que se destaca en la gestión de pruebas y gestión de la calidad para ofrecer software orientado a la calidad.

Rational Quality Manager Express Edition se diseñó para la colaboración entre pequeñas y medianas empresas, ya que les permite compartir información, usar la automatización para agilizar las programaciones de proyectos y generar informes de mediciones sobre los proyectos con el fin de tomar decisiones bien fundamentadas.

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

Además contribuye a que tengan un mejor control de sus proyectos mediante la provisión de métricas confiables y oportunas. También soporta gran cantidad de funciones de usuario como: gerente de pruebas, arquitecto de pruebas, líder de pruebas y gerente de laboratorio.

El Rational Quality Manager es utilizado fundamentalmente para [25]:

- ✓ La identificación de defectos duplicados analiza los defectos existentes asociados con una gran variedad de artefactos relacionados y presenta una lista de posibles duplicados para minimizar los esfuerzos del equipo de duplicados.
- ✓ Informe métrico integrado para laboratorios de pruebas y activos de pruebas conforme al MCIF (marco de mejora de capacidad medido).
- ✓ Plan de pruebas del ciclo de vida que define los roles, procesos y propiedad entregable, y automatiza el flujo de trabajo y activos.
- ✓ Un repositorio central almacena el historial con versión de activos y plantillas de pruebas para su reutilización.
- ✓ Recopilación de datos automatizada mediante generación de informes con una amplia aplicación de filtros.

Existen otras herramientas para gestionar las métricas desarrolladas por medianas empresas o desarrolladores individuales entre ellas se encuentran:

- SourceMonitor es una herramienta para el cálculo de las métricas, desarrollado por Campwood Software en Burlington, es una empresa privada, se estableció en 1997 e incorporada en Vermont calificados en Servicios de programación de computadoras. Empresas como Software Campwood por lo general ofrecen: Programación Avanzada, Programa del Ejército de equipos pequeños, Programación básica [26].
- TestWorks METRIC es una herramienta que ayuda a las especialistas en los diferentes procesos de las métrica desarrollado por Advanced User Systems creado en 1992 para distribuir productos de software que ayudan a las empresas operar con mayor eficiencia a través de la tecnología de automatización, gestión de la información, y el diseño de la interfaz de usuario avanzada [27].

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

A continuación se muestra una tabla donde se realiza una comparación de algunas herramientas para la gestión de las métricas donde se tienen en cuenta 4 aspectos: (1) se adapta la herramienta al proceso actual que existe en el LIPS, (2) beneficios o funcionalidades importantes, (3) medio de adquisición de la herramienta y (4) facilidad de uso de la herramienta para los Especialistas de calidad del LIPS.

Tabla 1: Comparación entre herramientas para la gestión de las métricas.

Herramienta	Se adapta al LIPS	Beneficios	Adquisición	Facilidad de uso
TestWorks METRIC	No	<ul style="list-style-type: none"> <li>• Después de procesar un archivo de código fuente, calcula automáticamente las mediciones de software.</li> <li>• Es configurable y fácil de leer los informes y gráficos.</li> <li>• El informe completo ofrece un conjunto de métricas para cada uno de los módulos en un fichero fuente.</li> </ul>	Propietaria	Fácil
TychoMetrics	Sí	<ul style="list-style-type: none"> <li>• Es un sistema basado en la web, abierto.</li> <li>• Automatiza la recogida de datos desde cualquier lugar del mundo a través de Internet.</li> <li>• Utiliza la tecnología de medición de modelado para asegurar la integridad de los datos y la repetibilidad de la medida.</li> </ul>	Propietaria	Normal
SourceMonitor	No	<ul style="list-style-type: none"> <li>• Recopila las métricas en un pase rápido, solo a través de archivos de código fuente.</li> <li>• Métricas de medidas para el código fuente escrito en C ++, C, C #, VB.NET, Java, Delphi, Visual Basic (VB6) o HTML.</li> <li>• Ofrece la opción de modificación de complejidad métrica.</li> </ul>	Gratis	Fácil

FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

		•Exportar las métricas para XML o CSV (valores separados por comas) para su procesamiento posterior con otras herramientas.		
--	--	---	--	--

Es importante tener una herramienta para automatizar todo el proceso para gestionar las métricas, ya que estas ayudan agilizar el proceso, a que sea menos engorroso y más fácil para los que recolectan los datos de las métricas, está claro que hace falta una herramienta que nos ayude en las gestión de las métricas. Las herramientas mencionadas contienen varias funcionalidades específicas para el proceso que se desarrolla en cada una de las organizaciones, además son muy costosas por esta razón en la presente investigación se desarrolla una herramienta que se adecue al proceso del LIPS y personalizada para las métricas definidas o seleccionadas para el mismo.

### 1.8 Principio de Pareto o Regla 80/20.

Pareto enunció el principio basándose en el denominado conocimiento empírico. Observó que la gente en su sociedad se dividía naturalmente entre los «pocos de mucho» y los «muchos de poco»; se establecían así dos grupos de proporciones 80-20 tales que el grupo minoritario, formado por un 20% de población, ostentaba el 80% de algo y el grupo mayoritario, formado por un 80% de población, el 20% de ese mismo algo. El Principio de Pareto dice que el 20% de cualquier cosa producirá el 80% de los efectos, mientras que el 80% restante sólo cuenta para el 20% de los efectos [28].

No obstante, el principio de Pareto permite utilizar herramientas de gestión, como el diagrama de Pareto, que se usa ampliamente en temas de control de calidad (el 80% de los defectos radican en el 20% de los procesos). Así, de forma relativamente sencilla, aparecen los distintos elementos que participan en un fallo y se pueden identificar los problemas realmente relevantes, que acarreen el mayor porcentaje de errores.

De la misma manera, en el mundo de la ingeniería del software el principio de Pareto puede ser enunciado de diferentes formas [29]:

- Así por ejemplo cuando hablamos de los costes de desarrollo podríamos decir que "el 80% del esfuerzo de desarrollo (en tiempo y recursos) produce el 20%

## FUNDAMENTACIÓN TEÓRICA Y ESTADO DEL ARTE

del código, mientras que el 80% restante es producido con tan sólo un 20% del esfuerzo".

- Si hablamos de pruebas de software, el principio nos dice que "el 80% de los fallos de un software es generado por un 20% del código de dicho software, mientras que el otro 80% genera tan solo un 20% de los fallos".

### 1.9 Conclusiones del Capítulo.

- La medición es considerada como una eficaz herramienta en las pruebas a los software, es la base para: detectar las desviaciones del rendimiento aceptable en los procesos y producto de software, y las oportunidades de mejora, identificar y priorizar las principales preocupaciones, dar seguimiento a la solución y mejorar la calidad del producto.
- En la propuesta presentada a continuación, se utiliza como metodología para definir e implementar las métricas seleccionadas la IEEE en su Estándar para metodología de Métricas de calidad de software (Standard for a Software Quality Metrics Methodology) y como modelo de calidad a tener en cuenta las características de calidad la ISO/IEC 9126-1 del 2005 Parte 1: Modelo de Calidad, dado que los mismos facilitan al proceso de prueba de un software, un conjunto de elementos complementarios a considerar y contienen terminologías y actividades consistentes.
- Las herramientas para la gestión de las métricas en las pruebas son la fuerza más importante para agilizar el proceso, ayudan en el cálculo de las métricas, en la recogida de información y en los reportes de las métricas para evaluar los indicadores establecidos para el proceso de pruebas.

## **PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD**

### **2.1 Introducción.**

En este capítulo se describe el catálogo de métricas de calidad según las características de calidad que plantea la ISO /IEC 9126-1 Parte 1: Modelo de Calidad. Estas métricas ayudarán a predecir, evaluar y caracterizar el software que se encuentra en las pruebas y sobre todo ayudará a mejorar la calidad de los mismos. Además se propone un sistema para gestionar las métricas donde se da una breve descripción de los roles que intervienen en el mismo. La aplicación de este sistema ayuda agilizar la recogida y cálculo de las métricas de calidad definidas en el presente capítulo.

### **2.2 Métodos y técnicas utilizados.**

Los métodos empíricos permiten describir y explicar las características del fenómeno en estudio. Los métodos dentro de los empíricos se aplicaron con el objetivo de recolectar los datos necesarios para identificar la problemática e identificar las causas de estas, así como determinar la magnitud de su influencia.

Las encuestas y las entrevistas fueron vitales para el diagnóstico del departamento de prueba de software, avalar los conceptos que se manejan en la investigación, medir el alcance y la importancia que tiene la temática. Captar la información cualitativa y cuantitativa del fenómeno, conocer los criterios sobre la forma en que se organiza y se lleva a cabo las métricas de calidad a los productos en las liberaciones, así como las posibles soluciones que se proponen en la investigación, para ello se entrevistaron y encuestaron personas involucradas e interesadas en las pruebas del LIPS.

#### **Entrevista**

La población a estudiar fue el personal interesado en los resultados de las pruebas a los productos de software de la UCI. La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional para poder obtener la mayor representatividad e información posible, de acuerdo con los intereses de la investigación que fue entrevistar personas interesadas en la calidad de los productos de software que se liberan en el LIPS.

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

Se seleccionaron 5 directivos institucionales, 2 vicedecanos de producción y 12 Líderes de proyectos, teniendo en cuenta la cantidad de proyectos en pruebas en el LIPS la muestra para la investigación es mayor del 20%.

Se realizó con el objetivo de identificar cuantitativamente los problemas que se habían identificado, el grado de conocimiento de los involucrados de la situación problemática y del problema y recopilar elementos a tener en cuenta en la solución. En el Anexo 1 se puede ver el diseño de la entrevista.

### **Encuesta**

La población a estudiar fueron los especialistas que se encuentran involucrados en las pruebas de varios productos de software en el LIPS. La selección se realizó con la técnica de muestreo no probabilística, muestreo intencional para poder obtener la mayor representatividad e información posible, de acuerdo con los intereses de la investigación que fue encuestar a los especialistas al frente de las pruebas del departamento de prueba de software.

Se seleccionaron 20 especialistas que teniendo en cuenta la cantidad de personal involucrado en las prueba de liberación, la muestra para la investigación es mayor del 60%.

Se realizó con el objetivo de identificar cuantitativamente los problemas que se habían identificado, el grado de conocimiento de los involucrados de la situación problemática y del problema, así como su percepción.

A la hora de elaborar la encuesta se combinaron los tipos de preguntas. La mayoría fueron semicerradas pues se tiene el interés de conocer la información cuantitativa pero también de saber la opinión del tema, así como involucrar y motivar a los encuestados en la solución. Se utilizaron además preguntas cerradas, directas e indirectas. En el Anexo 2 se puede ver el modelo de la encuesta aplicada.

### **2.3 Establecer los requerimientos de calidad del software.**

#### **a) Identificar una lista de posibles requerimientos de calidad.**

Para identificar los requerimientos de calidad (RC) que son aplicables al sistema de software, se emplea la experiencia organizacional, estándares, regulaciones o leyes. Adicionalmente, se listan otros requerimientos del sistema que puedan afectar la viabilidad de los requerimientos de calidad. Se consideran requerimientos contractuales e implicaciones, como costo o planificación, garantías, requerimientos de



## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

métricas del cliente e intereses propios de la organización. No deben regularse mutuamente requerimientos exclusivos en esta etapa. Es mejor enfocarse en la calidad de combinaciones factores/métricas directas en vez de métricas predictivas.

Para seleccionar la posible lista de requerimientos de calidad a tener en cuenta para definir o seleccionar las métricas de calidad que formarán parte del catálogo de métricas, se realiza una reunión de trabajo. En la misma participan todos los especialistas de calidad que se involucran en las pruebas del Departamento de Pruebas de Software y se utiliza el método grupo focal. Después de haber escuchado las diferentes opiniones y criterios de los especialistas de calidad se obtiene como resultado la posible lista de requerimientos de calidad a tener en cuenta para definir o seleccionar las métricas de calidad, ver Anexo 3.

### **b) Determinar la lista de requerimientos de calidad.**

Una vez que se obtiene la lista de posibles requerimientos de calidad se procede a seleccionar los que se tendrán en cuenta para la selección o definición de las métricas, para esto se realizó otra reunión de trabajo con los especialistas involucrados en las pruebas y se tuvieron en cuenta las siguientes consideraciones para la selección de los mismos.

- Las prioridades relativas de los requerimientos se discuten con todos los involucrados.
- Se asegura que todos los puntos de vista son considerados.
- Los resultados del estudio son resumidos en una lista simple de requerimientos de calidad.
- Los factores de calidad propuestos por esta lista deben tener relaciones conflictivas o cooperativas.
- Hay que tener cuidado al elegir la lista deseada para asegurar que los requerimientos son técnicamente viables, razonables, complementarios, alcanzables y verificables.
- Cualquier conflicto relacionado con los requerimientos ha de ser resuelto.

A continuación se listan los requerimientos de calidad que quedaron propuestos a tener en cuenta para la definición o selección de las métricas de calidad que conformarán el catálogo de métricas de calidad para evaluar los productos en las pruebas de liberación.

PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- RC 1: Las funciones deben estar correctamente implementadas.
- RC 2: Cumplir con la especificación de requisitos.
- RC 3: No implementar más funciones que las que se encuentran en la especificación de requisitos.
- RC 4: Las funcionalidades no deben ser cambiadas en la revisión del software.
- RC 5: Las funcionalidad de las aplicaciones deben de devolver el resultado esperado.
- RC 6: Las funcionalidades implementadas deben tener un alto nivel de precisión en sus respuestas.
- RC 7: La aplicación no debe permitir la corrupción de los datos.
- RC 8: La aplicación no debe permitir que usuarios o personas no autorizadas accedan a la información que se encuentra en la aplicación.
- RC 9: Todas las funcionalidades especificadas deben ser probadas en las pruebas de liberación.
- RC 10: Las interfaces de la aplicación deben estar diseñadas e implementadas de acuerdo a lo especificado.
- RC 11: Correcta solución de todos los defectos encontrados en las prueba de liberación.
- RC 12: Ejecución del 100 % de los casos de pruebas diseñados para las pruebas.
- RC 13: Evitar la ocurrencia de fallos en la aplicación por acciones o componentes externos.
- RC 14: Lograr el menor tiempo posible para recuperarse la aplicación ante cualquier fallo ocurrido.
- RC 15: Lograr el menor tiempo de inactividad de la aplicación ante cualquier fallo ocurrido.
- RC 16: Lograr el mayor número posible de puntos de restauración ante cualquier fallo ocurrido.
- RC 17: Confeccionar demos, manuales de usuarios y ayudas tanto online como en formato duro.
- RC 18: Lograr una buena descripción de las funcionalidad de la aplicación en demos, manuales de usuarios y ayudas tanto online como en formato duro.
- RC 19: Lograr la mayor personalización de la aplicación para el usuario.

PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

RC 20: Lograr un fácil acceso a todas las funciones implementadas en la aplicación.

RC 21: Cumplimiento con los requisitos de eficiencias establecidas con el usuario final.

RC 22: Lograr menor tiempo posible en las pruebas de liberación.

RC 23: Lograr el menor tiempo posible de respuesta de la aplicación.

RC 24: La aplicación debe de funcionar correctamente en las condiciones especificadas por los usuarios.

RC 25: Los instaladores de las aplicaciones deben permitir una buena personalización de la misma.

**c) Cuantificar cada subfactor de calidad.**

Los subfactores de calidad que se tendrán en cuenta para la definición y selección de las métricas de calidad son las subcaracterísticas de calidad que plantea la ISO/IEC 9126-1 Parte 1: Modelo de calidad. Los requerimientos de calidad son asociados a cada uno de los subfactores de calidad que se van a tener en cuenta para la definición y selección de las métricas de calidad. Para cada subfactor de calidad se asigna una o más métricas directas que lo representen y valores de métricas directas que sirvan de requerimientos cuantitativos a dicho subfactor de calidad. Se emplean métricas directas para verificar el logro de los requerimientos de calidad. En la Figura 7 se muestra el marco de trabajo de los requerimientos de calidad donde se puede ver la asociación de cada uno de los requerimientos de calidad a un subfactor de calidad y a la vez las métricas que mide este subfactor de calidad.

PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

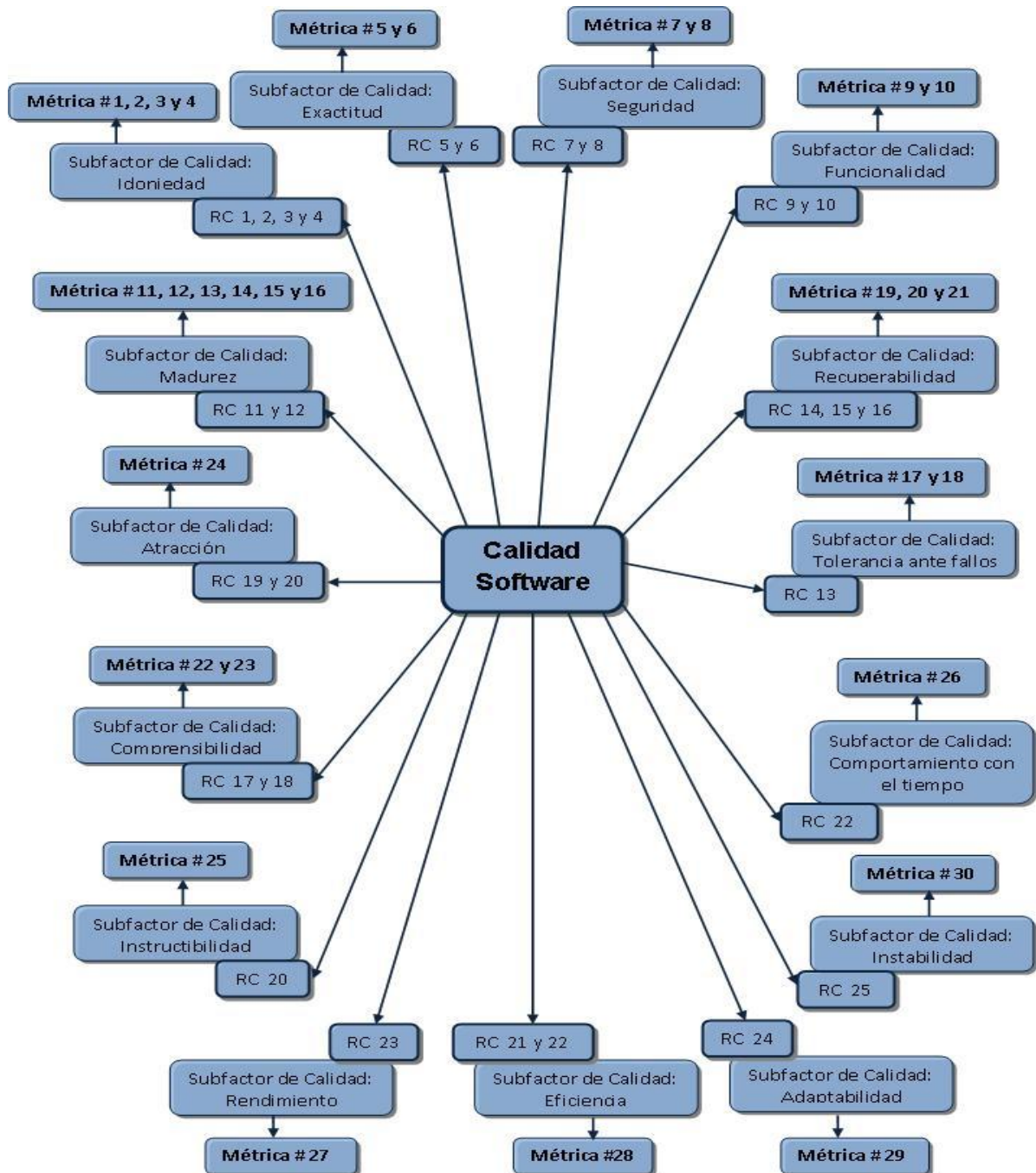


Figura 7: Marco de trabajo de los requerimientos de calidad (Creación propia).

## 2.4 Identificar métricas de software.

### Aplicar el marco de trabajo de las métricas de calidad.

Empleando el marco de trabajo de los requerimientos de calidad, se descomponen los subfactores de calidad en métricas. Se emplean solo métricas que están asociadas con los requerimientos de calidad del proyecto de software. En la propuesta de métricas se

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

definen y seleccionan las métricas externas de calidad ya que las mismas están destinadas para medir la calidad del producto software a través de la medición del comportamiento del sistema del cual el software forma parte. Las métricas externas solo pueden ser usadas durante las etapas de pruebas del proceso ciclo de vida y durante cualquier otra etapa operacional. En el anexo 4 se muestran las métricas externas definidas y seleccionadas según la característica de calidad a la que pertenezcan, guiado por la ISO 9126-1 parte 1: Modelo de Calidad con el método de aplicación y el tipo de medida de cada una de las variables que intervienen en la fórmula para el cálculo de la misma. A continuación se nombran las 30 métricas que conforman al catálogo de métricas según las características de calidad a la que pertenecen.

### **Métricas de Funcionalidad.**

Las métricas externas de funcionalidad deben ser capaces de medir de un atributo como es el comportamiento funcional del sistema en el cual el software está presente. Estas son:

#### ➤ *Métricas de idoneidad:*

Las métricas externas de idoneidad deben ser capaces de medir de un atributo como es la ocurrencia de un funcionamiento insatisfactorio o la ocurrencia de una operación insatisfactoria.

Un funcionamiento u operación insatisfactoria puede ser:

- Funcionamiento u operación que no se desempeña de la forma especificada en el Manual de usuario o la especificación de requisitos.
- Funcionamiento u operación que no provee una salida aceptable o razonable al tomar en consideración un objetivo específico de las tareas del usuario.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística idoneidad son:

- Métrica # 1: Adecuación funcional (Propone medir: ¿Cuán adecuada es la función evaluada?)
- Métrica # 2: Completitud de la implementación funcional (Propone medir: ¿Cuán completa ha sido la implementación y su conformidad con la especificación de requisitos?)

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- Métrica # 3: Cobertura de la implementación funcional (Propone medir: ¿Cuán correcta ha sido la implementación funcional?)
- Métrica # 4: Estabilidad en las especificaciones funcionales (Propone medir: ¿Cuan estable son las especificaciones funcionales después de entrar a funcionar?)

### ➤ *Métricas de exactitud*

Las métricas externas de precisión deben ser capaces de medir de un atributo como es la frecuencia con que los usuarios se encuentren con la ocurrencia de una falta de exactitud o de precisión, como pueden ser:

- Resultados incorrecto o imprecisos causados por datos inadecuados; por ejemplo, un dato con pocos dígitos significativos para un cálculo de precisión.
- Inconsistencia entre el procedimiento operación actual y el descrito en el manual de operación.
- Diferencias entre el resultado actual y el razonablemente esperado producto de una tarea ejecutada durante la operación.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística exactitud son:

- Métrica # 5: Exactitud esperada (Propone medir: ¿Existen diferencias entre los resultados actuales y los razonablemente esperados?)
- Métrica # 6: Precisión (Propone medir: ¿Con qué frecuencia los usuarios finales encuentran resultados con la precisión adecuada?)

### ➤ *Métricas de Seguridad*

Una métrica de seguridad externa debe ser capaz de medir de un atributo como el número de funciones, u ocurrencias de problemas de seguridad, que son:

- Impedir fugas de información o datos de salida segura.
- Evitar la pérdida de datos importantes.
- Defender contra el acceso ilegal u operaciones ilegales.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística seguridad son:

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- Métrica # 7: Prevención de la corrupción de los datos (Propone medir: ¿Cuál es la frecuencia de eventos de corrupción de los datos?)
- Métrica # 8: Capacidad de control de acceso (Propone medir: ¿Cómo controlar el acceso al sistema?)

### ➤ *Métricas de Funcionalidad*

Una métrica de funcionalidad externa debe ser capaz de medir el cumplimiento de un atributo con el número de funciones, o por otros acontecimientos de los problemas de cumplimiento como que el producto de software no cumpla con las normas, convenios, contratos u otros requisitos reglamentarios.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística seguridad son:

- Métrica # 9: Cumplimiento funcional (Propone medir: ¿Cómo es compatible con la funcionalidad del producto a las normas aplicables, los estándares y regulaciones?)
- Métrica # 10: Cumplimiento de estándar de interfaz (Propone medir: ¿Cómo se cumplen las interfaces con los reglamentos aplicables, las normas y regulaciones?)

### **Métricas de Confiabilidad.**

Las métricas externas de confiabilidad deben ser capaces de medir atributos relacionados con el comportamiento del sistema del cual el software forma parte durante la ejecución de las pruebas para indicar la magnitud de la confiabilidad, o sea, seguridad de funcionamiento del software durante la operación del sistema, con las que en la mayor parte de los casos no se distingue entre el software y el sistema. Ellas son:

### ➤ *Métricas de madurez*

Las métricas externas de madurez deben ser capaces de medir de un atributo como la exención de fallas en el software, causados por la ocurrencia de fallos existentes en el propio software.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística madurez son:

#### PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- Métrica # 11: Densidad de fallos totales contra casos de prueba (Propone medir: ¿Cuántos fallos totales fueron detectados durante un período de pruebas definido?)
- Métrica # 12: Grado de solución ante fallos totales (Propone medir: ¿Cuántas condiciones de fallo total están resueltas?)
- Métrica # 13: Densidad de fallos (Propone medir: ¿Cuántos fallos fueron detectados durante un período de pruebas definido?)
- Métrica # 14: Erradicación de fallos (Propone medir: ¿Cuántos fallos han sido corregidos?)
- Métrica # 15: Cobertura de las pruebas (Propone medir: ¿Cuántos casos de pruebas requeridos han sido ejecutados detectados durante las pruebas?)
- Métrica # 16: Madurez de las pruebas (Propone medir: ¿Está bien probado el producto?)

##### ➤ *Métricas de tolerancia ante fallos*

Las métricas externas de tolerancia ante fallos deben estar relacionadas con la capacidad del software de mantener un nivel de ejecución específico en casos de fallos de operación, o se infrinjan las interfaces especificadas.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística tolerancia ante fallos son:

- Métrica # 17: Evitación de operaciones incorrectas (Propone medir: ¿Cuántas funciones están implementadas con capacidad de evitación de operaciones incorrectas?)
- Métrica # 18: Evitar el fracaso (Propone medir: ¿Cómo los patrones de muchas fallas fueron puestos bajo control para evitar fallos críticos y graves?)

##### ➤ *Métricas de recuperabilidad*

Las métricas externas de recuperabilidad deben ser capaces de medir de un atributo como son los softwares y sistemas capaces de restablecer su nivel adecuado de ejecución y recuperar los datos directamente afectados en casos de fallos totales.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística recuperabilidad son:



## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- Métrica # 19: Grado de disponibilidad (Propone medir: ¿Cuán disponible está el sistema para su uso durante un período de tiempo especificado?)
- Métrica # 20: Tiempo medio de inactividad (Propone medir: ¿Cuál es el tiempo promedio en que el sistema se mantiene no disponible cuando ocurre un fallo total y antes de la arrancada gradual?)
- Métrica # 21: Restaurabilidad (Propone medir: ¿Cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud?)

### **Métricas de Usabilidad.**

Las métricas externas de usabilidad miden la dimensión con que el software puede ser comprendido, estudiado, operado, atractivo y acorde con las regulaciones y guías relativas a la usabilidad.

Resulta recomendable que la evaluación de estas métricas se haga por un grupo (7 u 8 aunque menores pueden obtener información de utilidad) de usuarios o evaluadores, usuarios simulados o clonados (pero representativos de un rango de usuarios) sin que reciban asistencia externa alguna.

#### ➤ *Métricas de comprensibilidad*

Las métricas externas de comprensibilidad deben ser capaces de valorar cómo un nuevo usuario podría comprender:

- Si el software es idóneo para la aplicación a la cual lo destina.
- Cómo el software puede ser usado para una tarea en particular.

Las métricas de calidad que conforman al catálogo de métricas de la subcaracterística comprensibilidad son:

- Métrica # 22: Accesibilidad a demos (Propone medir: ¿A qué proporción de demos/tutoriales pueden acceder los usuarios?)
- Métrica # 23: Comprensibilidad de la función (Propone medir: ¿Qué proporción de las funciones de producto el usuario podrá entender correctamente?)

#### ➤ *Métricas de atracción*

Las métricas externas de atracción deben ser capaces de evaluar la apariencia del software, y van a estar influenciadas por factores tales como el color en la pantalla y su diseño.

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística atracción es:

- Métrica # 24 Adaptabilidad de la apariencia de la interfaz (Propone medir: ¿Qué proporción de los elementos de la interfaz puede ser, por su apariencia, adaptado por el usuario para la satisfacción del mismo?)

### ➤ *Métricas de instructibilidad*

Una métrica externa de instructibilidad debe ser capaz de evaluar cómo los usuarios a larga distancia le llevará aprender a utilizar funciones específicas y la eficiencia de los sistemas de ayuda y documentación.

La instructibilidad está muy relacionada con claridad, comprensibilidad y medidas pueden ser indicadores de la potencial capacidad de aprendizaje del software.

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística instructibilidad es:

- Métrica # 25: Eficiencia de la documentación del usuario y/o sistemas de ayuda en el uso (Propone medir: ¿Qué proporción de las funciones pueden ser utilizados correctamente después de leer la documentación o el uso de sistemas de ayuda?)

## **Métricas de Eficiencia.**

Una métrica de eficiencia externa debe ser capaz de medir atributos tales como el consumo de tiempo y el comportamiento de utilización de recursos de sistemas informáticos, incluyendo el software durante las pruebas o las operaciones.

### ➤ *Métricas de comportamiento en el tiempo*

Una métrica externa de comportamiento en el tiempo debe ser capaz de medir atributos tales como el comportamiento temporal del sistema informático incluyendo el software durante las pruebas o las operaciones.

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística comportamiento en el tiempo es:

- Métrica # 26: Tiempo de respuesta (Propone medir: ¿Cuál es el tiempo necesario para completar una tarea específica? ¿Cuánto tiempo se tarda antes de que la respuesta del sistema a una operación especificada?)

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

### ➤ *Métricas de rendimiento*

Una métrica externa de rendimiento debe ser capaz de medir atributos tales como el comportamiento utilizado los recursos del sistema informático incluyendo software durante las pruebas o las operaciones.

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística rendimiento es:

- Métrica # 27 Rendimiento (Propone medir: ¿Cómo muchas tareas pueden realizarse con éxito en un período determinado de tiempo?)

### ➤ *Métricas de eficacia*

Una métrica de eficiencia debe ser capaz de medir el cumplimiento de atributos tales como el número de funciones, o por otros acontecimientos de los problemas de cumplimiento, que puede ser el producto de software que no cumplan con las normas, convenciones o regulaciones relativas a la eficiencia.

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística eficiencia es:

- Métrica # 28: Cumplimiento de la eficiencia (Propone medir: ¿Cómo es compatible la eficacia del producto a las normas, los estándares y regulaciones?)

### **Métricas de Portabilidad.**

Una métrica externa de portabilidad debe ser capaz de medir atributos tales como el comportamiento del operador o del sistema durante la actividad de portabilidad.

### ➤ *Métricas de adaptabilidad*

Una métrica externa de adaptación debe ser capaz de medir atributos tales como el comportamiento del sistema o el usuario que está tratando de adaptar el software a diferentes entornos especificados. Cuando un usuario tiene que aplicar un procedimiento de adaptación que no ha sido proporcionado por el software para una necesidad de adaptación específica, el esfuerzo necesario del usuario para la adaptación debe ser medido.

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística adaptabilidad es:

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- Métrica # 29: Capacidad de adaptación al entorno del software (Propone medir: ¿Puede el usuario o desarrollador de software adaptarse fácilmente al entorno?)

### ➤ Métricas de instalabilidad

Una métrica externa de instalabilidad debe ser capaz de medir atributos tales como el comportamiento del sistema o el usuario que está tratando de instalar el software en un entorno específico del usuario.

La métrica de calidad que conforma al catálogo de métricas de la subcaracterística instalabilidad es:

- Métrica # 30: Facilidad de instalación (Propone medir: ¿El usuario puede fácilmente instalar en el entorno de funcionamiento?)

Fueron seleccionadas métricas externas de 5 características de calidad (Funcionalidad, Confiabilidad, Usabilidad, Eficiencia y Portabilidad) de las 6 que contiene la ISO 9126 ya que actualmente en el Laboratorio Industrial de Pruebas de Software solo se realizan pruebas enfocada a estas características. Las mismas fueron seleccionadas según los requerimientos de calidad expuestos en el epígrafe 2.3 ya que cada una tributa a evaluar uno de estos requerimientos.

## 2.5 Procedimiento para utilizar el catálogo de métricas en la pruebas del LIPS.

Las pruebas en el Laboratorio Industrial de Pruebas de Software comienzan con una reunión de inicio donde participan los miembros del equipo de desarrollo, los especialistas de calidad que guiarán las pruebas y el jefe del Departamento de Pruebas de Software quien dirige la reunión. En esta reunión se definen los tipos de pruebas que se realizarán en las pruebas enfocadas a las características de calidad, ver anexo 5.

Al quedar definido los tipos de pruebas que se le realizarán al software, se decide las métricas de calidad que se tendrán en cuenta en las pruebas para la evaluación final del producto de software. A continuación se muestran las métricas según el tipo de prueba a realizar.

Tabla 2: Métricas por tipo de pruebas.

Tipo de pruebas	Nombre de la métrica
Funcionales	Métrica # 1: Adecuación funcional

PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

	<p>Métrica # 2: Completitud de la implementación funcional</p> <p>Métrica # 3: Cobertura de la implementación funcional</p> <p>Métrica # 4: Estabilidad en las especificaciones funcionales</p> <p>Métrica # 5: Exactitud esperada</p> <p>Métrica # 6: Precisión</p> <p>Métrica # 9: Cumplimiento funcional</p> <p>Métrica # 10: Cumplimiento de estándar de interfaz</p> <p>Métrica # 11: Densidad de fallos totales contra casos de prueba</p> <p>Métrica # 12: Grado de solución ante fallos totales</p> <p>Métrica # 13: Densidad de fallos</p> <p>Métrica # 14: Erradicación de fallos</p> <p>Métrica # 15: Cobertura de las pruebas</p> <p>Métrica # 16: Madurez de las pruebas</p>
Seguridad	<p>Métrica # 7: Prevención de la corrupción de los datos</p> <p>Métrica # 8: Capacidad de control de acceso</p>
Recuperación y tolerancia a fallas	<p>Métrica # 17: Evitación de operaciones incorrectas</p> <p>Métrica # 18: Evitar el fracaso</p> <p>Métrica # 19: Grado de disponibilidad</p> <p>Métrica # 20: Tiempo medio de inactividad</p> <p>Métrica # 21: Restaurabilidad</p>
Usabilidad	<p>Métrica # 22: Accesibilidad a demos</p> <p>Métrica # 23: Comprensibilidad de la función</p> <p>Métrica # 24: Adaptabilidad de la apariencia de la interfaz</p> <p>Métrica # 25: Eficiencia de la documentación del usuario y / o sistemas de ayuda en el uso</p>
Carga y Estrés	<p>Métrica # 26: Tiempo de respuesta</p> <p>Métrica # 27: Rendimiento</p> <p>Métrica # 28: Cumplimiento de la eficiencia</p>
Instalación y Configuración	<p>Métrica # 29: Capacidad de adaptación al entorno del software</p> <p>Métrica # 30: Facilidad de instalación</p>

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

Al quedar acordado en la reunión de inicio los tipos de pruebas y las métricas a calcular en el proceso de pruebas en el Laboratorio Industrial de Pruebas de Software se procede a comenzar la primera iteración de las pruebas de liberación.

Durante la ejecución de las pruebas en su primera iteración y al realizarse las pruebas de regresión en la segunda iteración se recolectan los datos necesarios para el cálculo de las métricas, en el epígrafe 2.6 se muestran cuales son estos datos.

Las métricas que se mencionan a continuación son las que pueden ser calculadas al terminar una iteración de las pruebas:

- Métrica # 1: Adecuación funcional
- Métrica # 2: Completitud de la implementación funcional
- Métrica # 3: Cobertura de la implementación funcional
- Métrica # 5: Exactitud esperada
- Métrica # 6: Precisión
- Métrica # 7: Prevención de la corrupción de los datos
- Métrica # 8: Capacidad de control de acceso
- Métrica # 9: Cumplimiento funcional
- Métrica # 10: Cumplimiento de estándar de interfaz
- Métrica # 11: Densidad de fallos totales contra casos de prueba
- Métrica # 13: Densidad de fallos
- Métrica # 15: Cobertura de las pruebas
- Métrica # 16: Madurez de las pruebas
- Métrica # 17: Evitación de operaciones incorrectas
- Métrica # 18: Evitar el fracaso
- Métrica # 19: Grado de disponibilidad
- Métrica # 20: Tiempo medio de inactividad
- Métrica # 21: Restaurabilidad
- Métrica # 22: Accesibilidad a demos
- Métrica # 23: Comprensibilidad de la función
- Métrica # 24: Adaptabilidad de la apariencia de la interfaz
- Métrica # 25: Eficiencia de la documentación del usuario y / o sistemas de ayuda en el uso
- Métrica # 26: Tiempo de respuesta

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

- Métrica # 27: Rendimiento
- Métrica # 28: Cumplimiento de la eficiencia
- Métrica # 29: Capacidad de adaptación al entorno del software
- Métrica # 30: Facilidad de instalación

Las métricas que se mencionan a continuación son las que pueden ser calculadas al terminar las pruebas de regresión de una iteración:

- Métrica # 4: Estabilidad en las especificaciones funcionales
- Métrica # 12: Grado de solución ante fallos totales
- Métrica # 14: Erradicación de fallos

Al terminar cada iteración de las pruebas se puede calcular las métricas seleccionadas según el tipo de prueba realizado al producto de software y se puede ir observando el porcentaje de cumplimiento de las características de calidad en cada iteración y comprobar si aumentan o disminuyen las mismas. El cálculo de las mismas se realiza mediante la herramienta para la gestión de las métricas, la cual se explica en el epígrafe 2.7.

### **2.6 Información para la recolección de los datos, el cálculo y análisis de las métricas.**

La recolección de los datos es uno de los momentos más importantes en el proceso para obtener la métrica final. Los datos deben ser recogidos con la mayor precisión posible ya que de ellos depende que las métricas calculadas sean lo más real posible. A continuación se muestra una lista de algunos de los datos que se recogieron mediante la herramienta para la gestión de las métricas de cada uno de los proyectos seleccionados.

1. Número de problemas detectados en las funcionalidades del sistema.
2. Número de especificaciones de requisitos evaluadas.
3. Número de funcionalidades actualizadas después de entrar en funcionamiento.
4. Número de interfaces con problemas detectados.
5. Número total de Casos de pruebas utilizados.
6. Número total de fallos detectados.
7. Número de fallos totales solucionados.

PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

8. Número total de problemas reales detectados.
9. Número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas.
10. Cantidad de no conformidades que fueron detectadas por utilizar los casos de pruebas.
11. Número de demos/tutoriales a los que pueden acceder los usuarios exitosamente.
12. Número total de demos/tutoriales a los que se puede acceder.
13. Número de funciones que se pueden utilizar.
14. Total de número de funciones que ofrece la aplicación.
15. Número de tarea completada.
16. Período de tiempo de observación de las pruebas.

Estos son los datos necesarios para el cálculo de las métricas ya que tributan a las variables que se necesitan para calcular las métricas según las fórmulas establecidas para cada una de ellas. Además cada uno de ellos tributa a una o varias de los requerimientos de calidad especificados en epígrafes anteriores.

Una vez definidas las métricas se procede al cálculo de las mismas para cada uno de los proyectos seleccionados. Al obtener las métricas por proyecto se calcula la métrica requerida en cada una de las métricas calculadas por subcaracterísticas de calidad en los proyectos, para ello se utiliza la siguiente formula.

$$M = \frac{\sum_{k=1}^n P_k}{n}$$

Donde:

M- Métrica requerida.

n- Cantidad de proyectos.

P- Métrica individual de cada proyecto.

Los resultados del análisis de las métricas son cambios en la organización y en el proceso de desarrollo que son indicados por la interpretación y el empleo de los datos de medición. Analizar las diferencias entre los datos recolectados de las métricas y los valores objetivos e investigar las diferencias significativas son actividades a realizar en esta investigación.



## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

Identificar y revisar los valores de las métricas de calidad en las pruebas de software. Identificar valores de las métricas que están fuera de los intervalos anticipados de tolerancia (baja o inesperadamente alta calidad) para estudios posteriores. La calidad inaceptable puede ser manifestada como una complejidad excesiva, documentación inadecuada, poca traceabilidad, u otros atributos indeseables. La existencia de condiciones como éstas, indican que el software puede no satisfacer los requerimientos de calidad cuando se vuelva operacional. Muchas de las métricas que usualmente son de interés, no pueden ser recolectadas durante el desarrollo del software.

Se hace necesario también comparar los valores de las métricas con los valores críticos de las métricas. Analizar en detalle los componentes de software cuyos valores son diferentes a los críticos. En dependencia del resultado del análisis, rediseñar (la calidad aceptable es alcanzada mediante el rediseño), cancelar (la calidad es tan pobre que rediseñar no es factible), o no hacer cambios (desviaciones de valores críticos de métricas resultan insignificantes) en los componentes de software.

### 2.7 Herramienta para la gestión de las métricas.

Para agilizar la gestión de las métricas se implementó una herramienta que interviene en el proceso de recolección de los datos de las métricas ayudando a los especialistas de calidad al frente de las pruebas hacer este proceso lo más rápido y menos engorroso posible. Además realiza los cálculos de todas las métricas definidas ya que en ocasiones el cálculo de las formulas de las métricas son complicadas y retrasan los diferentes procesos que se estén realizando. Además informatiza algunas funciones básicas del proceso de gestión de las métricas. En el anexo 6 están los requisitos funcionales de la herramienta los cuales fueron la base para el desarrollo de la herramienta, teniendo en cuenta que varios de estos no están es las herramientas estudiadas para la gestión de las métricas. En la tabla 3 se describen los diferentes roles que intervienen en la herramienta para la gestión de las métricas.

Tabla 3: Roles de la herramienta para la gestión de las métricas.

Actor	Descripción
Jefe del Laboratorio	Es el encargado de asignar los proyectos que están en pruebas a los especialistas de calidad. Elaborar los reportes de métricas de todos los proyectos que se están revisando,

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

	para luego enviárselos a los directivos de la universidad para que ayude en las toma de decisiones.
Especialista de Calidad	Encargado de dirigir las pruebas que se le realicen a los proyectos, así como asignar las métricas seleccionadas, recoger los datos y calcular las métricas, además de realizar los reportes por proyectos.
Administrador	Encargado del mantenimiento del sistema así como de gestionar todo el proceso de permisos a los usuarios que acceden al mismo.

La herramienta está formada por 5 módulos que a continuación se explica cuales son sus funcionalidades.

- Módulo Administración

En el módulo de Administración el administrador de la herramienta será el encargado de añadir los usuarios del sistema que posteriormente se autenticarán en la aplicación. Será el encargado de otorgarle los diferentes roles (Especialista de Calidad y Jefe del Laboratorio) y gestionar los permisos en la aplicación según el rol que vayan a desempeñar. En la figura 8 se muestra la interfaz de usuario para listar y agregar nuevos usuarios.

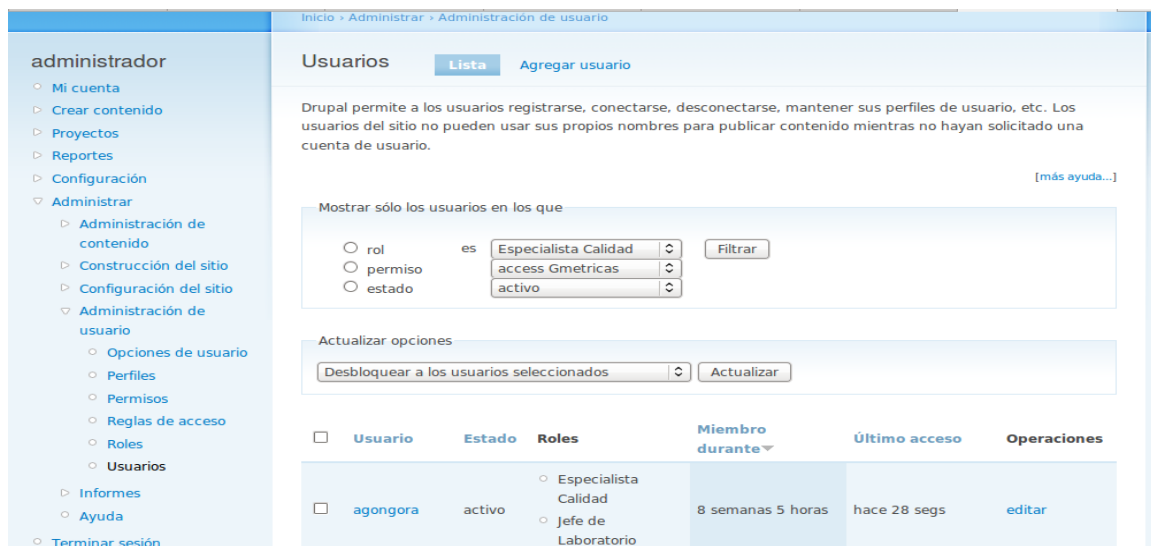


Figura 8: Interfaz de usuario del módulo Administración (Creación propia).

- Módulo Configuración

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

En el módulo de Configuración el Jefe del Laboratorio será el encargado de todos los nomencladores que se encuentran en la aplicación. Estos nomencladores serán utilizados por los demás módulos del sistema, entre ellos están los diferentes centros productivos que se encuentran en la universidad, el Jefe del Laboratorio podrá Añadir, modificar o eliminar los centros que soliciten el servicio de las pruebas a los productos. También se podrán gestionar las etapas por la que puede pasar un artefacto en las pruebas (Pruebas Exploratorias, Primera Iteración, Segunda Iteración, Tercera Iteración y Pruebas Finales). Además se configuran los tipos de proyectos (Nocionales y Exportación) y artefactos que se desarrollan en la universidad ya que es necesario tener en cuenta para los reportes estos nomencladores y realizarlos en base a esta selección. Las posibles evaluaciones que se les darán a los artefactos y al proyecto que entran en pruebas serán otros de los nomencladores a poder gestionar en la herramienta. En la figura 9 se muestra la interfaz de usuario de la vista general para todas las configuraciones y en la figura 10 se muestra la interfaz de usuario para visualizar y agregar nuevos artefactos.



Figura 9: Interfaz de usuario del módulo Configuración. Vista general (Creación propia).

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD



Figura 10: Interfaz de usuario del módulo Configuración. Visualizar artefacto (Creación propia).

- Módulo Gestión de proyecto

En el módulo de Gestión de proyecto el Jefe del Laboratorio es el encargado en la reunión de inicio de las pruebas añadir el proyecto y asignárselo al Especialista de Calidad que estará al frente de las pruebas de liberación, así como seleccionar el tipo de proyecto y centro al que pertenece, ver figura 11. Una vez definido los tipos de pruebas a realizar el Especialista de Calidad desde su área de trabajo en la herramienta podrá seleccionar las métricas que se calcularán según la característica de calidad a la que pertenece y la iteración en que se encuentre en el momento de la pruebas, ver figura 12, estas métricas son las 30 que conforman el catálogo expuesto en el epígrafe 2.4. Al terminar las iteraciones de las pruebas el Especialista de Calidad será el encargado de insertar los datos necesarios en el sistema para el cálculo de las métricas y el sistema será el encargado de calcular las métricas y mostrar los resultados de cada una de ellas según la característica de calidad a la que pertenezca y posteriormente evaluar el proyecto por el cumplimiento de estas.

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD

The screenshot displays the 'Gestión de Métricas' module interface. On the left is a sidebar with the 'agongora' logo and a navigation menu including 'Mi cuenta', 'Proyectos', 'Reportes', 'Configuración', and 'Terminar sesión'. The 'Proyectos' section is expanded, showing options like 'Gestión de Proyectos', 'Crear Proyecto', 'Ver Proyectos', 'Asignar Métricas', 'Evaluar Artefacto', and 'Evaluar Proyecto'. The main content area is titled 'Gestión de Métricas' and shows a breadcrumb trail 'Inicio > Proyectos > Gestión de Proyectos'. Below this is the 'Crear Proyecto' form, which includes fields for 'Nombre de proyecto' (with a text input and a search icon), 'Centro' (a dropdown menu with 'CEDIN' selected), 'Tipo de Proyecto' (a dropdown menu with 'Nacional' selected), and 'Asignar Especialista al Proyecto' (a dropdown menu with 'Yonnis Costilla' selected). A 'Continuar' button is at the bottom of the form.

Figura 11: Interfaz de usuario del módulo Gestión de proyectos. Crear proyecto (Creación propia).

The screenshot displays the 'Gestión de Métricas' module interface, specifically the 'Asignar Métricas' form. The sidebar and navigation menu are identical to the previous figure. The main content area shows a breadcrumb trail 'Inicio > Proyectos > Gestión de Proyectos' and the title 'Asignar Métricas'. Below this is the 'Asignar Métricas a los artefactos' form, which includes fields for 'Proyecto' (a dropdown menu with 'RHODA' selected), 'Artefacto' (a dropdown menu with 'Aplicación Web' selected), 'Tipo de Métrica' (a dropdown menu with 'Subcaracterísticas de madurez' selected), and 'Iteración' (a dropdown menu with '1ra Iteración' selected). A 'Continuar' button is at the bottom of the form. A 'Buscar' button is located at the bottom left of the main content area.

Figura 12: Interfaz de usuario del módulo Gestión de proyectos. Asignar métricas (Creación propia).

- Módulo de Reportes

En el módulo de Reportes los Especialistas de Calidad y el Jefe del Laboratorio se puede obtener diferentes reportes de los proyectos que se encuentran en pruebas ya sea por tipo de proyecto, por la métricas calculas o por la evaluación de los mismos. En la figura 13 se muestra la interfaz de usuario para mostrar los reportes de los proyectos.

## PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD



Figura 13: Interfaz de usuario del módulo de Reportes (Creación propia).

- Módulo Mi Cuenta

En el módulo Mi Cuenta el Especialista de Calidad autenticado puede cambiar el nombre de su usuario, su contraseña y su correo electrónico. Además puede cambiar el idioma de la aplicación y la zona horaria en la que desea trabajar. En la figura 14 se muestra la interfaz de este módulo.



Figura 14: Interfaz de usuario del módulo Mi Cuenta (Creación propia).

La herramienta para la gestión de las métricas de calidad estará montada en dos servidores, un servidor para la aplicación web y otro para la base de datos (PostgreSQL), las máquinas clientes de los Especialistas de Calidad se conectarán a estos servidores mediante el protocolo http y a la vez la máquina cliente estará

PROPUESTA DEL CATÁLOGO AUTOMATIZADO DE MÉTRICAS DE CALIDAD conectada a una impresora para imprimir los diferentes reportes de las métricas en caso de ser necesario, como se muestra en el diagrama de despliegue de la figura 15.

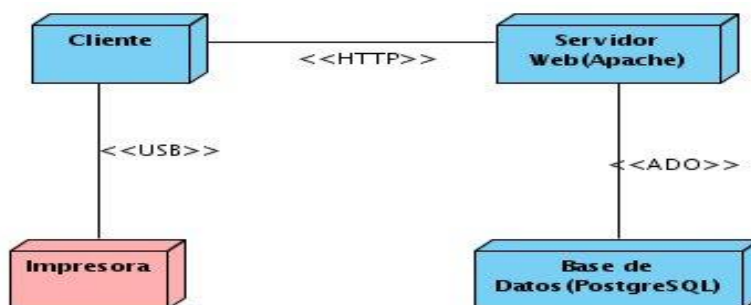


Figura 15: Diagrama de Despliegue (Creación propia).

## 2.8 Conclusiones del Capítulo.

- El ambiente actual en el Laboratorio Industrial de Pruebas de Software en CALISOFT y las características actuales son el marco propicio para implementar un proceso de medición y análisis basado en el modelo de calidad que plantea la ISO 9126-1 Parte 1: Modelo de Calidad.
- Las métricas externas de calidad definidas y seleccionadas teniendo en cuenta las características de calidad según la ISO 9126-1 Parte 1: Modelo de Calidad, permite tener una referencia de que porcentaje de cumplimiento tienen los software de las características de calidad, ayudando a los directivos a la toma de decisiones y a mejorar la calidad del proceso de pruebas y del producto final.
- La herramienta propuesta para gestionar las métricas en las pruebas ayuda a que el cálculo de las mismas sea más ágil y menos engorroso para los especialistas en las pruebas de liberación.

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

### 3.1 Introducción.

En este capítulo se abordan temas referentes a la implementación de la propuesta realizada en un marco de pilotaje del departamento de pruebas de software, los resultados percibidos y los análisis realizados a estos resultados. A partir de estos análisis es posible evaluar y verificar la calidez de la propuesta. Para la implementación del catálogo de métricas se sigue la metodología propuesta por la IEEE Standard for a Software Quality Metrics Methodology, así como las actividades que deben ejecutarse para evaluar la propuesta. Orientado a tal fin, se presentan los resultados de las verificaciones realizadas durante las pruebas a los proyectos seleccionados, logrando medidas de la efectividad del proceso de pruebas a partir de la evaluación de los productos liberados teniendo en cuenta el cumplimiento de las características de calidad de la ISO 9126-1 Parte 1: Modelo de Calidad.

### 3.2 Validación de la propuesta.

Para validar la propuesta se tendrán en cuenta la variable definida en la hipótesis que es: El cumplimiento de las características de calidad en las pruebas de liberación.

#### 3.2.1 Implementación del catálogo de métricas.

En la implementación del catálogo de métricas de calidad se utilizan 12 proyectos, de ellos algunos se encuentran en pruebas en el Laboratorio Industrial de Pruebas de Software y otros ya fueron liberados por especialista del mismo departamento. Estos proyectos fueron seleccionados ya que tienen toda la documentación y los datos necesarios para obtener las métricas de calidad.

Para la selección de estos 12 proyectos se utilizó el Principio de Pareto que es también conocida como la Regla de 80/20 y es uno de los conceptos más útiles para la productividad personal. Si lo vinculamos a las métricas en las pruebas podríamos decir que: “Las métricas de calidad calculadas al 20 % de los proyectos en pruebas tendrá el mismo efecto en el 80 % de los restantes proyectos que se encuentran en pruebas de liberación.”

Actualmente hay 62 Proyectos en el Departamento de Pruebas de Software que cumplen con las condiciones y datos necesarios para obtener las métricas de calidad, estos proyectos se encuentran en pruebas o ya fueron liberados. Aplicando la Regla



## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

80/20 podemos observar como se muestra en la figura 16, que el 20% más significativo de los 62 proyectos lo conforman 12, y el otro 80% serán los proyectos a evaluar con las métricas calculadas para ver el cumplimiento de las características de calidad de la ISO 9126-1 Parte 1: Modelo de Calidad y así obtener medidas de la efectividad del proceso de liberación.

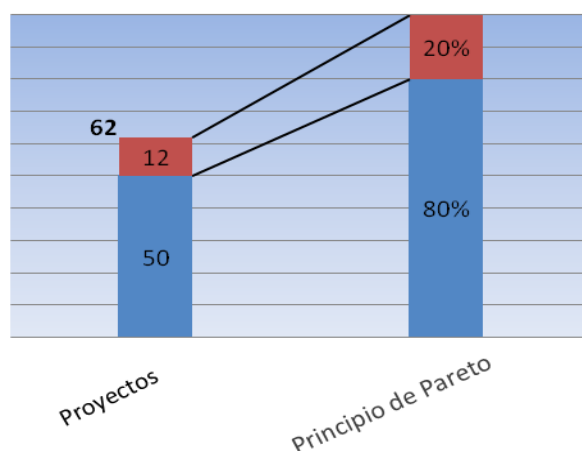


Figura 16: Proyectos seleccionados para calcular las métricas (Creación propia).

### 3.2.2 Recolección de los datos en los proyectos.

Los datos son recogidos mediante la herramienta para la gestión de la métricas propuesto en el capítulo 2, en la Tabla 4 se muestra un ejemplo de la plantilla utilizada para la recogida de los datos. Cada especialista al frente del proyecto que se encuentre en pruebas será el encargado de registrar los datos para posteriormente proceder al cálculo de la métrica de calidad. Los datos son verificados en cuanto a exactitud y a la unidad de medida apropiada especificada en las métricas descritas en el capítulo 2, así como su uniformidad en caso de ser recogidos por más de una persona en las prueba de liberación.

Tabla 4: Variables de la métrica de idoneidad.

Nombre de la métrica	Variables	Datos
Adecuación funcional	A - Número de problemas detectados en las funcionalidades del sistema.	A = 5
	B - Número de especificaciones de requisitos evaluadas.	B = 8

### 3.3 Cumplimiento de las características de calidad en las pruebas de liberación.

#### 3.3.1 Cálculo de las Métricas de Calidad.

Una vez recopilados todos los datos necesarios de los proyectos seleccionados se procede a calcular las métricas. Cada métrica de calidad tiene una fórmula y una interpretación diferente de los resultados que arroja. El cálculo de las métricas se realiza mediante la herramienta para la gestión de las métricas propuesta en el capítulo 2, la misma utiliza una plantilla con los datos que se muestran en la tabla 5 por cada proyecto.

Tabla 5: Resultados del cálculo de las métricas por proyecto.

Proyecto	Artefacto	Iteración	Sub característica	Métricas	Nivel requerido	Valor
Proyecto A	Aplicación	1	Idoneidad (Adecuación funcional)	$X = 1-A/B$	1 ( $0 \leq X \leq 1$ )	0.37
			Funcionalidad (Cumplimiento de estándar de interfaz)	$X = A/B$	1 ( $0 \leq X \leq 1$ )	0.66
			Seguridad (Prevención de la corrupción de los datos)	$X = 1-A/N$	1 ( $0 \leq X \leq 1$ )	0.87

#### 3.3.2 Resultados de las métricas.

Las tablas que aparecen a continuación muestran los resultados obtenidos de cada una de las métricas en las 3 iteraciones y en el proceso de pruebas de liberación. Los porcentajes representan el cumplimiento adecuado de la subcaracterística de calidad según la métrica calculada, así tenemos la medida requerida de los productos para velar por el cumplimiento de las características de calidad.

#### Funcionalidad:

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

Tabla 6: Resultados de las métricas de Funcionalidad.

Subcaracterística de calidad	Nombre de la métrica	Métricas 1 Iteración	Métricas 2 Iteración	Métricas 3 Iteración	Métrica en el proceso de prueba
Idoneidad	Adecuación funcional	62,2 %	75,0 %	97,8 %	78,8 %
Idoneidad	Estabilidad en la especificaciones funcionales	64,1%	75,0 %	97,8 %	78,9 %
Funcionalidad	Cumplimiento de estándar de interfaz	67,0 %	81,8 %	98,0 %	82,3 %

### Confiabilidad:

Tabla 7: Resultados de las métricas de Confiabilidad.

Subcaracterística de calidad	Nombre de la métrica	Métricas 1 Iteración	Métricas 2 Iteración	Métricas 3 Iteración	Métrica en el proceso de prueba
Madurez	Densidad de fallos totales contra casos de prueba	6,8 %	48,9 %	93,3 %	48,4 %
Madurez	Grado de solución ante fallos totales	95,3 %	97,2 %	93,2 %	95,2 %
Madurez	Cobertura de las pruebas	99,0 %	98,8 %	100 %	99,3 %
Madurez	Madurez de las pruebas	45,4 %	61,0 %	95,4 %	67,9 %

### Usabilidad:

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

Tabla 8: Resultados de las métricas de Usabilidad.

Subcaracterística de calidad	Nombre de la métrica	Métricas 1 Iteración	Métricas 2 Iteración	Métricas 3 Iteración	Métrica en el proceso de prueba
Comprensibilidad	Accesibilidad a demos	50,0 %	54,2 %	54,2 %	52,8 %
Instructibilidad	Eficiencia de la documentación del usuario y / o sistemas de ayuda en el uso	83,7 %	88,5 %	98,5 %	90,2 %

### Eficiencia:

Tabla 9: Resultado de la métrica de Eficiencia.

Subcaracterística de calidad	Nombre de la métrica	Métricas 1 Iteración	Métricas 2 Iteración	Métricas 3 Iteración	Métrica en el proceso de prueba
Rendimiento	Rendimiento	2,1	3,5	7,5	4,4

### 3.3.3 Análisis de los resultados.

Una vez obtenidas las métricas de calidad se hace necesario comparar los resultados de las métricas obtenidas en cada proyecto y la métrica general de cada subcaracterísticas de calidad, para así ver el cumplimiento de las subcaracterísticas de calidad en el producto liberado. Como se puede ver en las figuras que se muestran a continuación, el porcentaje de cumplimiento de las subcaracterísticas de calidad va aumentando a medida que pasan las iteraciones, ya que las pruebas están cumpliendo al objetivo principal de ella, se puede comprobar cuan efectivas puede ser el proceso de prueba de liberación. En los dos proyectos seleccionados se puede comprobar como la métrica de las subcaracterísticas de calidad se encuentra entre el porcentaje de cumplimiento de los dos proyectos, de la misma forma es el comportamiento en los demás proyectos lo que demuestra la validez de las métricas calculadas.

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

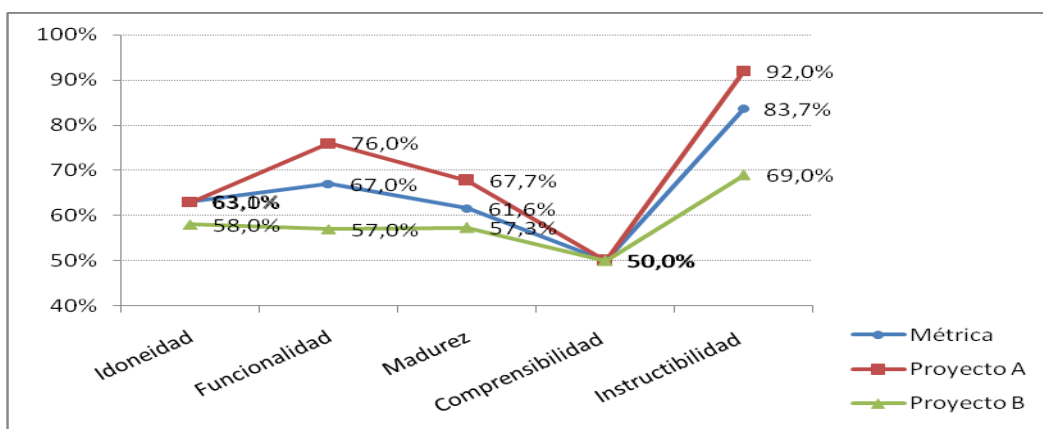


Figura 17: Comportamiento de las métricas en la primera iteración de pruebas (Creación propia).

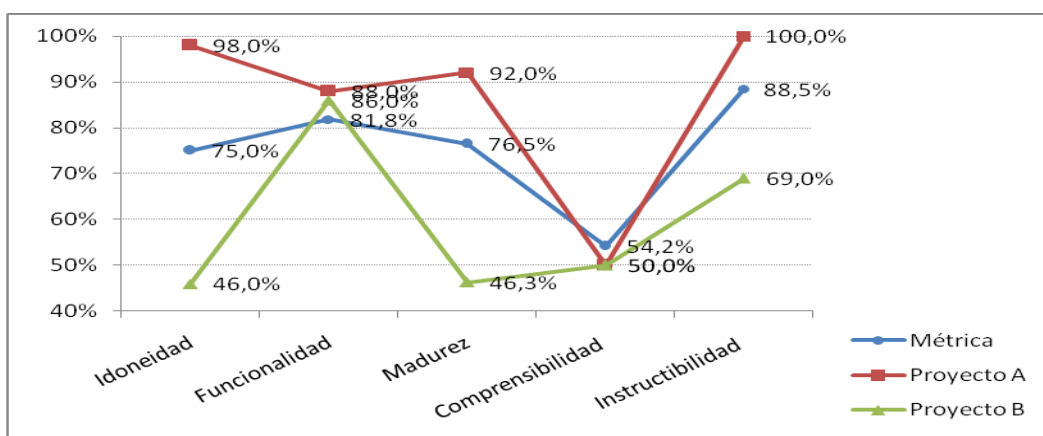


Figura 18: Comportamiento de las métricas en la segunda iteración de pruebas (Creación propia).

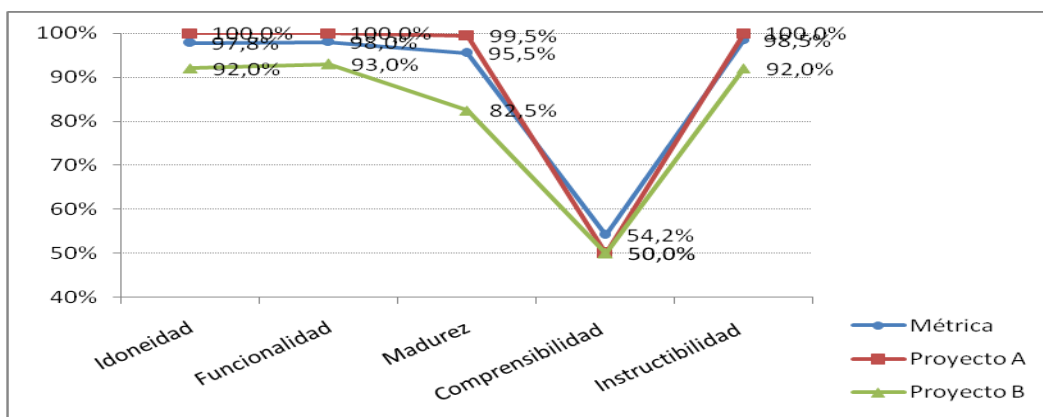


Figura 19: Comportamiento de las métricas en la tercera iteración de pruebas (Creación propia).

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

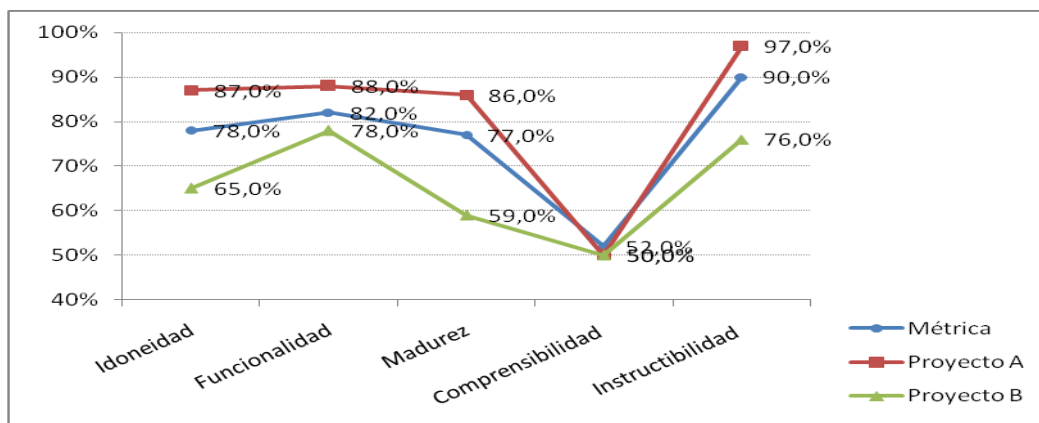


Figura 20: Comportamiento de las métricas en las pruebas (Creación propia).

Del resultado mostrado en las figuras anteriores podemos deducir que el proyecto A mantiene todo el tiempo un buen porcentaje de cumplimiento de cada una de las subcaracterísticas de calidad y termina en su tercera iteración de pruebas con los resultados que se muestran en la tabla 10, lo que no pasa con el proyecto B porque este proyecto incumple con varias de los requerimientos de calidad que se establecieron en el capítulo 2, por ejemplo podemos citar que la cantidad de NC de tipo Funcionalidad, Validación, Excepciones y Opciones que no funcionan de este proyecto exceden por el doble la cantidad de Casos de pruebas y aún en la tercera iteración quedan NC que el equipo de desarrollo no puede resolver.

Tabla 10: Métricas del proyecto A y B en la tercera iteración de las pruebas de liberación.

Subcaracterísticas de calidad	Métrica del proyecto A (% cumplimiento del proyecto)	Métrica del proyecto B (% cumplimiento del proyecto)	Métrica (% cumplimiento esperado)
Idoneidad	100 %	92,0 %	97,0 %
Funcionalidad	100 %	93,0 %	98,0 %
Madurez	95,5 %	82,0 %	95,5 %
Comprensibilidad	50,0 %	50,0 %	54,2 %
Instructibilidad	100 %	92,0 %	98,5 %

El cumplimiento de las características de calidad de la ISO 9126-1 Parte 1: Modelo de Calidad se pueden obtener mediante las métricas de calidad aplicadas a los proyectos. En las pruebas realizadas al proyecto C se aplicaron las métricas anteriormente

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

mencionadas, en la figura 21 se muestra la tabulación de los datos obtenidos. En el gráfico se puede observar que las subcaracterísticas poseen valores aceptables de calidad en la tercera iteración de las pruebas de liberación.

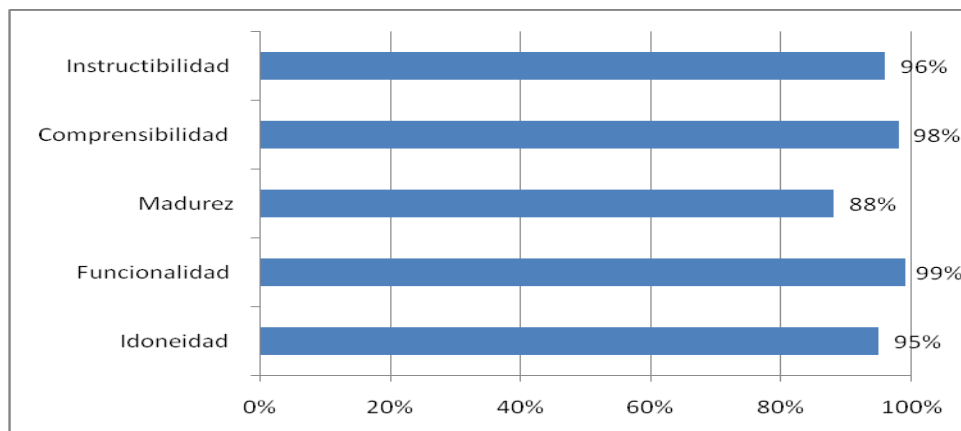


Figura 21: Tabulación de resultados (Creación propia).

En la Figura 22 se muestra los resultados del nivel de calidad obtenido por el Proyecto C, en donde se puede ver que cumple con el 95% de las características de calidad seleccionadas encontrándose dentro del nivel de aceptabilidad, por lo tanto satisface los requerimientos de calidad.

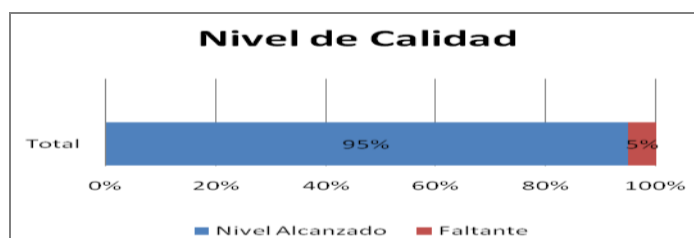


Figura 22: Resultados del nivel de calidad del proyecto C (Creación propia).

De los resultados obtenidos en el proceso de medición en las pruebas del Proyecto C se obtiene que el nivel de Funcionalidad es Alto, como se puede apreciar en la Figura 23, por lo tanto cumple las expectativas del personal responsable del sistema.

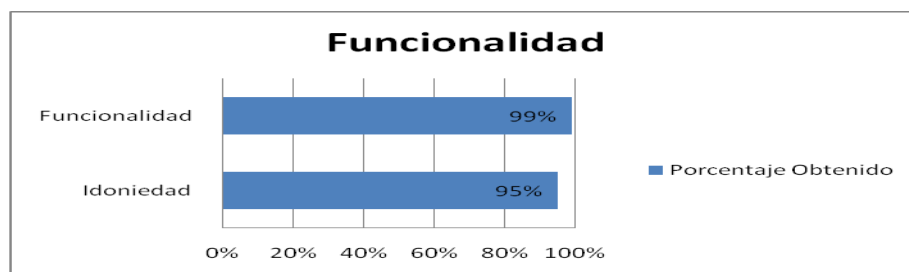


Figura 23: Resultados del nivel de calidad de Funcionalidad (Creación propia).

## IMPLEMENTACIÓN Y EVALUACIÓN DE LA PROPUESTA

El nivel de Confiabilidad del sistema es medio, como se puede apreciar en la Figura 24, por tanto se necesita mejorar el procedimiento de recuperación del sistema y la corrección de fallos detectados, ya que no satisface las expectativas del personal responsable del mismo.

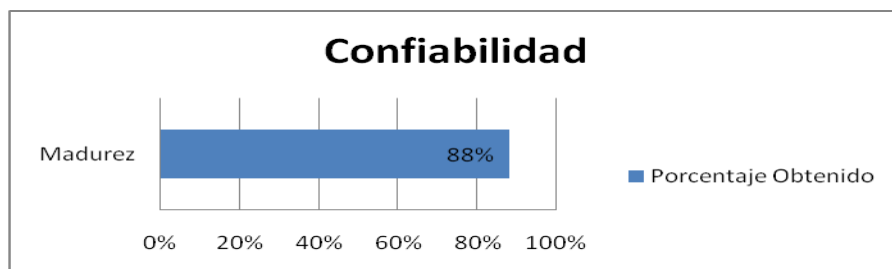


Figura 24: Resultados del nivel de calidad de Confiabilidad (Creación propia).

El nivel de Usabilidad es Alto, como se puede apreciar en la Figura 25, por tanto cumple las expectativas del personal responsable del sistema. Una mejora para el sistema, en cuanto a subcaracterística Comprensibilidad podría ser que la ayuda del sistema se encuentre accesible desde cualquiera interfaz del sistema.

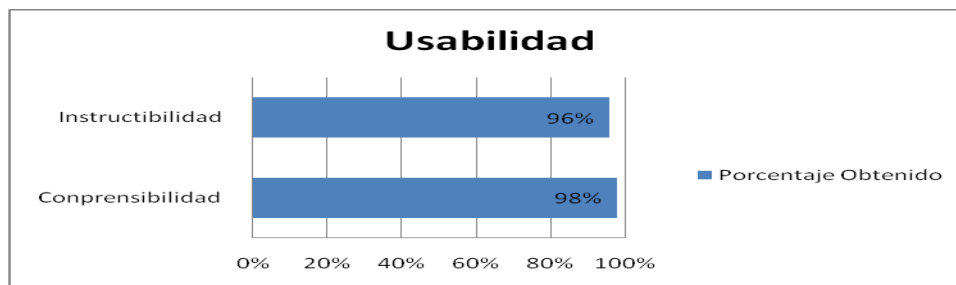


Figura 25: Resultados del nivel de calidad de Usabilidad (Creación propia).

Con las métrica de calidad obtenidas del catálogo que se presenta en la propuesta de la investigación, se logra obtener medidas para conocer el cumplimiento de las características de calidad que exige la ISO 9126-1 Parte 1: Modelo de Calidad y de esta forma se logra evaluar los productos que se encuentran liberados por especialistas del departamento de pruebas de software en el LIPS, obteniendo medidas de la efectividad en las pruebas de liberación, no así antes de la presente investigación ya que no existían medidas que de una forma u otra ayudaran a los especialista del departamento de pruebas de software a evaluar los productos de software.



### **3.4 Conclusiones del capítulo.**

- La ejecución de la propuesta fue realizada a través de una selección de 12 proyectos que se encuentran en prueba o ya se encuentran liberados y que contienen los datos necesarios para el cálculo de las métricas.
- La recolección de los datos se realizó mediante la herramienta para la gestión de las métricas, al igual que el cálculo de las mismas, obteniéndose las métricas de calidad para cada una de las características de calidad en las tres iteraciones de las pruebas de liberación.
- Se analizaron los datos obtenidos de las métricas y su aplicación en proyectos seleccionados, viéndose el cumplimiento de cada una de las subcaracterísticas de calidad seleccionadas, para así evaluar los productos y contribuir a obtener medidas de la efectividad del proceso de pruebas de liberación.

## CONCLUSIONES

1. Atendiendo al estado del arte y las condiciones propias del Laboratorio Industrial de Pruebas de Software y guiadas por las características de calidad que plantea el modelo de calidad ISO Parte 1: Modelo de calidad se definió y seleccionó un conjunto de métricas de calidad para el proceso de pruebas de liberación.
2. Se logró evaluar los productos liberados en el Laboratorio Industrial de Pruebas de Software, comprobando el cumplimiento de las características de calidad, gracias a las métricas de calidad obtenidas con los datos de proyectos liberados.
3. Se analizaron las métricas de calidad obtenidas con las resultantes en los proyectos que se encuentran en pruebas de liberación, validando un conjunto de métricas de calidad en proyectos que se encuentran liberados o están en pruebas de liberación.
4. Se logró agilizar la recogida de los datos, haciendo más preciso el cálculo de las métricas y teniendo reportes específicos para el análisis de las métricas de calidad gracias a la implementación de la herramienta para la gestión de las métricas en el Laboratorio Industrial de Pruebas de Software.

## RECOMENDACIONES

Se pueden realizar algunas recomendaciones como seguimiento para esta investigación a partir de la experiencia percibida en la aplicación de la propuesta en el Laboratorio Industrial de Pruebas de Software, las lecciones aprendidas y el criterio de los especialistas de calidad del departamento de pruebas de software:

- Obtener las métricas del catálogo de métricas de calidad que no se llegaron a calcular para evaluar los productos en las pruebas para así tener más elementos en la evaluación los productos según el cumplimiento de las características de calidad de la ISO 9126-1 Parte 1: Modelo de Calidad.
- Agregar a la herramienta para la gestión de las métricas una nueva funcionalidad que permita adicionar nuevas métricas de calidad definidas para luego calcularlas y tener más mediciones en el proceso de prueba de liberación.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] CES Centro de Ensayos de software, Disponible en: <http://www.ces.com.uy/>.
- [2] Testhouse, Disponible en: <http://www.es.testhouse.net/>.
- [3] Centro del Software e-Quality, Disponible en <http://www.centrodelsoftware.com.mx/>.
- [4] IEEE, 610.12-1990, (R2002), "IEEE Standard Glossary of Software Engineering Terminology".
- [5] D.o.D. and U. Army, Eds. Practical Software and Systems Measurement (PSM), 2003.
- [6] Florac, W. A., R. E. Park, et al. Practical Software Measurement: Measuring for Process Management and Improvement, Software Engineering Institute. Carnegie Mellon University, 1997.
- [7] Pfleeger, S.L. y Fenton, N.E. Software metrics. A rigorous and practical approach, PWS Pub, 1997.
- [8] ISO/IEC 9126-1: 2005 Parte 1: Modelo de Calidad, Ingeniería de Software – Calidad del Producto.
- [9] Pressman, R.S, Mc Graw Hill, Ingeniería del Software. Un enfoque práctico, 6ta Edición, 2009.
- [10] Fenton, N. Software quality assurance & Measurement. A worldwide perspective, Chapman&Hall, 1997.
- [11] IEEE, (1998), "IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology".
- [12] Vega L.C., Rivera P.L., García S.A., "Mejores prácticas para el establecimiento y aseguramiento de la calidad del software", 1ra Edición, 2008.
- [13] ISO, "Official Site of International Organization for Standardization (ISO)", 2011, Disponible en: <http://www.iso.org/iso/home.htm>.
- [14] IEEE, "Official site of the Institute of Electrical and Electronics Engineers", 2011, Disponible en: <http://www.ieee.org/index.html>.
- [15] SEI, S. E. I. and C. M. U. CMU. "Official Web Site of Software Engineering Institute", 2009, Disponible en: <http://www.sei.cmu.edu/about/>.
- [16] ITEC DaVinci, Leonardo, "Gestión, Control y Garantía de la Calidad del Software", 2006.

REFERENCIAS BIBLIOGRÁFICAS

- [17] ISO/IEC 9126-2: 2003 Parte 2: Métricas Externas, Ingeniería de Software – Calidad del Producto.
- [18] McCall, J. A, and P. K Richards. Factors in Software Quality. RADC TR77369, US Rome Air Development Center Reports NTIS AD/A049 1977.
- [19] Boehm, B, J. R Brown, and M Lipow. Quantitative Evaluation of Software Quality. Proceed 2nd Int'l Conference on Software Engineering. 1978, 592-605.
- [20] Basili V., Gianluigi C., Dieter R, "The Goal Question Metric Approach", 1994 Disponible en: <http://www.cs.toronto.edu/~sme/CSC444F/handouts/GQM-paper.pdf>.
- [21] Condori-Fernández, N., J. B. Faguás, et al. "Modelo de Agregación Basado en un Sistema Neurodifuso para un Proceso de Evaluación de Calidad del Software", 2002.
- [22] Cárdenas, D. S. A., D. A. H. González, et al. "Diplomado en Ingeniería del Software", 2006. Disponible en:  
[http://www.icda.edu.do/english/cg/cursos/diplomado\\_ingenieria\\_software.asp](http://www.icda.edu.do/english/cg/cursos/diplomado_ingenieria_software.asp).
- [23] BRISABOA, D. N. R. "Red Iberoamericana de Tecnologías de Software para la Década del 2000 (RITOS2)", 2006. Disponible en: <http://www.cytel.org>.
- [24] SQS 2011: "Software Quality Systems S.A." Disponible en: <http://www.sqs.es>.
- [25] IBM 2011: Official site of IBM Disponible en: <http://www.ibm.com>.
- [26] Campwood Software: Official site Disponible en: <http://www.campwoodsw.com/>
- [27] Advanced User Systems (AUS) 2006: Official site Disponible en:  
<http://www.aus.com.au/sr/testworks>.
- [28] Ferris, Timothy, "La semana Laboral de 4 hora", 1ra Edición 2010.
- [29] Besterfield, Dale, "Control de la Calidad", 1ra Edición, 2000.

## BIBLIOGRAFÍA

- Addison Wesley, "Metric and Model in software Quality Engineering", 2da Edición 2002.
- BRISABOA, D. N. R. "Red Iberoamericana de Tecnologías de Software para la Década del 2000 (RITOS2)", 2006, Disponible en: <http://www.cyted.org>.
- Bizcaia. La calidad del software y los procesos de testing. 2010. Disponible en: <http://www.euskadinnova.net/documentos/363.aspx>.
- BugHuntress QA Lab. Outsourcing Software Testing Services. Company profile. 2011. Disponible en: <http://www.bughuntress.com/software-testing-services/independent-verification-validation.html>
- Cárdenas, D. S. A., D. A. H. González, et al. "Diplomado en Ingeniería del Software", 2006. Disponible en: [http://www.icda.edu.do/english/cg/cursos/diplomado\\_ingenieria\\_software.asp](http://www.icda.edu.do/english/cg/cursos/diplomado_ingenieria_software.asp).
- Carrasco, O. M. F, D. G León, and A. B Benavides. Un enfoque actual sobre la calidad del software. Revista ACI 3, 2004.
- CHAOS. THE STANDING GROUP: "New Chaos number show. StartLing results." 2011. Disponible en: <http://www.marketwire.com/pressrelease/newstandishgroupreportsnowsmoreprojectsaresuccessfullessprojectsailing1405513.htm>.
- CMMI. Capability Maturity Model Integration (CMMI). 2002.
- Cusmai, Marcelo. Calidad del Software desde Latinoamérica. Infotesting. Un espacio para testers profesionales. Revista digital. Abril 2011. Disponible en: <http://www.infotesting.com>.
- Condori-Fernández, N., J. B. Faguás, et al. "Modelo de Agregación Basado en un Sistema Neurodifuso para un Proceso de Evaluación de Calidad del Software", 2002.
- D.o.D. and U. Army. Practical Software and Systems Measurement (PSM), 2003.
- D.o.D. and U. Army. "Official Practical Software and Systems Measurement Web Site", 2007. Disponible en: <http://www.psmc.com/>.
- Delgado M.D, Rosete S.A. (2005) "Una propuesta de introducción de las revisiones en el proceso de desarrollo de software", 2005, Disponible en: <http://rev-inv-ope.univ-paris1.fr/files/26205/IO-26205-1.pdf>.

- Diccionario de la Lengua Española-Vigésima segunda edición. Disponible en: [http://buscon.rae.es/draeI/SrvltConsulta?TIPO\\_BUS=3&LEMA=calidad](http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=calidad).
- Durán, I. M. R. PSM: Una Propuesta para la Medición de Software en la Universidad de las Ciencias Informáticas. Habana, Universidad de las Ciencias Informáticas, 2007.
- Durán, M. R. "Mediciones prácticas de software y sistemas (PSM): Informática 2007.
- E-Quallity. Disponible en: <http://www.equallity.net>.
- Electrónica., C. S. d. A. "MÉTRICA. VERSIÓN 3. Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información", 2007, Disponible en: <http://www.csi.map.es/csi/metrica3/>.
- Fenton, N. Software quality assurance & Measurement. A worldwide perspective, Chapman&Hall, 1997.
- Febles, Ailyn et al. Una experiencia novedosa para el testing desarrollada por un Departamento de Pruebas de Software. V Taller de Calidad en las Tecnologías de la Información y las Comunicaciones. Informática 2011 Febrero 2011.
- Finquelievich, Susana. Las TIC en la cooperación Sur-Sur: el acuerdo de librecomercio entre la India y el Mercosur. Revista CTS 1, Septiembre 2004.
- Florac, W. A., R. E. Park, et al. Practical Software Measurement: Measuring for Process Management and Improvement, Software Engineering Institute. Carnegie Mellon University, 1997.
- García, A. de Amescua, M Velasco, and A Sanz. Ten Factors that Impede Improvement of Verification and Validation Processes in Software Intensive Organizations. EuroSPI 2007, Postdam, Alemania Septiembre 2007.
- Garcia, F. El Proceso de Medición Software. Proceso Software y Gestión del Conocimiento, Universidad de Castilla-La Mancha, 2008.
- García, M. N. M. Medición de la calidad del software en el ámbito de la especificación de requisitos, Universidad de Salamanca, Departamento de Informática y Automática, 2004 Disponible en: <http://www.sc.ehu.es/jiwdocoj/remis/docs/mmg-2000.ppt>.
- González, Santiago. La calidad del Software en modo Factoría de Pruebas. Noviembre 2010. Disponible en: <http://www.mkmpi.com/byteti/lacalidaddelsoftwareenmodofactoriadepuebas/>.

- Humphrey, W. S. A Discipline for Software Engineering, Carnegie Mellon University, 1995.
- IEEE, 610.12-1990, (R2002), "IEEE Standard Glossary of Software Engineering Terminology".
- IEEE, (1998), "IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology".
- INTECO. Instituto Nacional de Tecnologías de la Información. Laboratorio Nacional de Calidad del Software, INTECO. Disponible en: <http://www.inteco.es>.
- ISO, "Official Site of International Organization for Standardization (ISO)", 2011, Disponible en: <http://www.iso.org/iso/home.htm>.
- ISO 9000:2005. Quality Management Systems – Fundamentals and Vocabulary.
- ISO/IEC. ISO/IEC 91261.2. Parte 1: Modelo de Calidad. 2005.
- Kan, Stephen H. Metrics and Models in Software Quality Engineering. 2000.
- León, R. A. H. and S. C. González, El paradigma cuantitativo de la Investigación Científica. Ciudad de la Habana, EDUNIV. Editorial Universitaria, 2002.
- López, C. Mejoramiento continuo principio de Gestión de la Calidad, 2007, Disponible en: <http://www.gestiopolis.com/canales5/ger/gksa/136.htm>.
- McCall, J. A, and P. K Richards. Factors in Software Quality. RADC TR77369, US Rome Air Development Center Reports NTIS AD/A049 1977.
- Mehta, D. La Industria de la Tecnología de Información en la India, 2006, Disponible en: <http://www.gobernabilidad.cl/modules.php?name=News&file=print&sid=1066>.
- METRICS, T. "Official Web Site of TOTAL METRIC", 2008, Disponible en: <http://www.totalmetrics.com/>.
- MIC Ministerio de la Informática y las Comunicaciones, 2011 Disponible en: <http://www.informaticahabana.com/>.
- Myers, G. The art of software testing, 2da edición. John Wiley & Sons Inc. 2004.
- Park, R. E., W. B. Goethert, et al. Goal-Driven Software Measurement - A Guidebook, Software Engineering Measurement and Analysis. Software Engineering Institute. Carnegie Mellon University, 1996.
- Pichaco, A. M. "Ingeniería de Software. El estándar ISO 15504", 2007, Disponible en: [http://dmi.uib.es/~dmiamp/ESIII/0708\\_ESIII\\_SPI\\_Tema3.pdf](http://dmi.uib.es/~dmiamp/ESIII/0708_ESIII_SPI_Tema3.pdf).



- Pressman, R. S. Ingeniería de Software. Un Enfoque práctico, 6ta Edición, 2009.
- Pfleeger, S.L. y Fenton, N.E., Software metrics. A rigorous and practical approach, PWS Pub, 1997.
- QSM, Q. S. M. I. "Quantitative Software Management", 2009, Disponible en: <http://www.qsm.com/>.
- Riascos, Sandra Critina. Modelo para la evaluación de la efectividad de la tecnología informática en el entorno empresarial. Ingeniería e Investigación 28, 2008. Disponible en: [http://www.scielo.unal.edu.co/scielo.php?pid=S012056092008000200019&script=sci\\_arttext](http://www.scielo.unal.edu.co/scielo.php?pid=S012056092008000200019&script=sci_arttext).
- Rodríguez, Moisés. Calidad de Procesos y Productos de Software. Calidad de productos de software ISO/IEC 25000. Julio 2010. Disponible en: <http://www.alarcos.infcr.uclm.es/per/fruiz/.../santander/mrodrigueziso25000update.pdf>.
- SEI, S. E. I. and C. M. U. CMU. "Official Web Site of Software Engineering Institute", 2009, Disponible en: <http://www.sei.cmu.edu/about/>.
- SEMA and C. M. U. SEI. "Official Web Site of Software Engineering Measurement and Analysis (SEMA)", 2009, Disponible en: <http://www.sei.cmu.edu/sema/>.
- SQRL. Software Quality Research Laboratory. Computer Science and Information Systems Dept. University of Limerick. Disponible en: <http://www.sqrl.ul.ie>.
- Stephen H. Kan, Addison- Wesley "Metrics and Models in Software Quality Engineering", 2000.
- Capote, Tayché. Conceptualización e implementación de un Laboratorio Industrial de Pruebas de Software, Junio, 2011.
- Verdecia, E, R Portuondo, J Lavandero, and J Arencibia. Proposal of a methodology for carrying out the processes of roles certification during the training of an engineer on informatics sciences. International Technology, Education and Development Conference. Valencia, España, 2011.

## ANEXOS

**Anexo 1:** Diseño de la Entrevista a interesados de la evaluación de los productos en las pruebas de liberación.

1. ¿Considera importante evaluar un producto de software en las pruebas de liberación?
2. ¿Los datos que se brindan actualmente de la evaluación de los productos son suficientes?
3. ¿Considera que utilizando las métricas de calidad en las prueba se podrían tener más argumentos para evaluar un producto de software?
4. ¿Qué datos considera importante para conocer de la calidad de los productos de software que se liberen en las pruebas de liberación?
5. ¿Considera necesario evaluar los productos de software por las características de calidad que plantea la ISO 9126 Modelo de calidad?
6. ¿Que otras cosas considera usted necesario a tener en cuenta en la evaluación de un producto de software?

**Anexo 2:** Diseño de la Encuesta a especialistas del departamento de pruebas de software.

### Cuestionario

Con este cuestionario pretendemos identificar potencialidades y deficiencias en la evaluación de los productos de software en las pruebas de liberación. Le pedimos sinceridad a la hora de responder las preguntas, le aseguramos confidencialidad y anonimato a su respuesta.

Responde las siguientes preguntas y marcar con una x en el caso que haga falta:

1. ¿Se tienen en cuenta las características de calidad de las ISO 9126 Modelo de calidad en las pruebas de liberación?

Sí \_\_\_\_\_ No \_\_\_\_\_

En caso de ser afirmativo ¿Qué características de calidad se tienen en cuenta?

---

---

2. ¿Se miden de alguna forma las características de calidad que se tienen en cuenta en las pruebas de liberación?

Sí \_\_\_\_\_ No \_\_\_\_\_

En caso de ser afirmativo ¿Qué método, técnica o artefacto utilizan para la medición?

---

---

3. ¿Actualmente en las prueba se evalúan los productos de software?

Sí \_\_\_\_\_ No \_\_\_\_\_ A veces \_\_\_\_\_

En caso de ser afirmativo ¿Cómo evalúan los productos?

---

---

4. ¿Se realiza algún tipo de medición de las características de calidad en las pruebas de liberación?

Sí \_\_\_\_\_ No \_\_\_\_\_ A veces \_\_\_\_\_

5. ¿Utilizan alguna métrica en las pruebas de liberación?

Sí \_\_\_\_\_ No \_\_\_\_\_ A veces \_\_\_\_\_

En caso que sea afirmativo responde las siguientes preguntas:

- ¿Qué métricas utilizan para el producto de software?

---

---

- ¿Consideran suficiente las métricas que utilizan?

---

---

- ¿Utilizan algún método para la recolección de los datos y el cálculo de las métricas?

---

---

6. ¿Considera necesario las métricas para evaluar los productos de software en las prueba de liberación?

**Anexo 3:** Lista de posibles requerimientos de calidad.

RC 1: Las funciones deben estar correctamente implementadas.

RC 2: Cumplir con la especificación de requisitos.

RC 3: No implementar más funciones que las que se encuentran en la especificación de requisitos.

RC 4: Las funcionalidades no deben ser cambiadas en la revisión del software.

RC 5: Las funcionalidad de las aplicaciones deben de devolver el resultado esperado.

RC 6: Las funcionalidades implementadas deben tener un alto nivel de precisión en sus respuestas.

RC 7: La aplicación no debe permitir la corrupción de los datos.

RC 8: La aplicación no debe permitir que usuarios o personas no autorizadas accedan a la información que se encuentra en la aplicación.

RC 9: Todas las funcionalidades especificadas deben ser probadas en las pruebas de liberación.

RC 10: Las interfaces de la aplicación deben estar diseñadas e implementadas de acuerdo a lo especificado.

RC 11: Correcta solución de todos los defectos encontrados en las prueba de liberación.

RC 12: Ejecución del 100 % de los casos de pruebas diseñados para las pruebas.

RC 13: Evitar la ocurrencia de fallos en la aplicación por acciones o componentes externos.

RC 14: Lograr el menor tiempo posible para recuperarse la aplicación ante cualquier fallo ocurrido.

RC 15: Lograr el menor tiempo de inactividad de la aplicación ante cualquier fallo ocurrido.

RC 16: Lograr el mayor número posible de puntos de restauración ante cualquier fallo ocurrido.

RC 17: Confeccionar demos, manuales de usuarios y ayudas tanto online como en formato duro.

RC 18: Lograr una buena descripción de las funcionalidad de la aplicación en demos, manuales de usuarios y ayudas tanto online como en formato duro.

RC 19: Lograr la mayor personalización de la aplicación para el usuario.

RC 20: Lograr un fácil acceso a todas las funciones implementadas en la aplicación.

RC 21: Cumplimiento con los requisitos de eficiencias establecidas con el usuario final.

RC 22: Lograr menor tiempo posible en las pruebas de liberación.

RC 23: Lograr el menor tiempo posible de respuesta de la aplicación.

RC 24: La aplicación debe de funcionar correctamente en las condiciones especificadas por los usuarios.

RC 25: Los instaladores de las aplicaciones deben permitir una buena personalización de la misma.

RC 26: Las funciones implementadas deben devolver los resultados esperados.

RC 27: Los datos entre aplicaciones o módulo deben de intercambiarse correctamente en pruebas de integración.

RC 28: Solucionar el 100 % de las fallas del sistema que ocurran durante las pruebas.

RC 29: Lograr 100 % de recarga de los datos ante el reinicio o recarga en las pruebas.

RC 30: Lograr que las interfaces de las aplicaciones se adapten a la personalización del usuario.

#### **Anexo 4:** Descripción de las métricas de calidad.

Métricas para la medición de la característica funcionalidad.

- Métrica Idoneidad.

Tabla 11: Métrica externa de la subcaracterística de calidad Idoneidad.

Nombre de la métrica	Lo que se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Adecuación funcional	¿Cuán adecuada es la función evaluada?	Número de funciones idóneas para ejecutar funciones específicas en comparación con el número de funciones evaluadas.	$X = 1 - A/B$  A - Número de problemas detectados en las funcionalidades del sistema.  B - Número de especificaciones de requisitos evaluadas.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará más adecuada	X = Contable/ Contable  A = Contable  B = Contable
Compleitud de la implementación funcional	¿Cuán completa ha sido la implementación y su conformidad con la especificación de requisitos?	Ejecutar las pruebas (de caja negra) funcionales de acuerdo con la especificación de requisitos.  Cuenta el número de funciones perdidas detectadas y compare el resultado con el número de funciones descritas en la especificación de requisitos.	$X = 1 - A/B$  A - Número de funciones perdidas detectadas en la evaluación.  B - Número de funciones descritas en especificación de requisitos.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X = contable/ Contable  A = Contable  B = Contable
Cobertura de la	¿Cuán correcta	Ejecutar las pruebas	$X = 1 - A/B$	$0 \leq X \leq 1$	X = Contable/

implementación funcional	ha sido la implementación funcional?	funcionales (de caja negra) de acuerdo con la especificación de requisitos. Cuente el número de funciones incorrectamente implementadas o funciones perdidas detectadas y compare el resultado con el número total de funciones descritas en la especificación de requisitos. Cuente el número de funciones que están completas en relación con las que no lo están.	<p>A - Número de funciones incorrectamente implementadas o funciones perdidas detectadas.</p> <p>B - Número de funciones descritas en la especificación de requisitos.</p>	A mayor cercanía al 1 resultará mejor	<p>Contable</p> <p>A = Contable</p> <p>B = Contable</p>
Estabilidad en las especificaciones funcionales	¿Cuan estable son las especificaciones funcionales después de entrar a funcionar?	Cuenta el número de funciones que se describen en las especificaciones funcionales que tuvo que ser cambiado después de que el sistema se pone en funcionamiento y compararlo con el número total de funciones que se describen en las	<p><math>X=1-A/B</math></p> <p>A- Número de funcionalidades actualizadas después de entrar en funcionamiento.</p> <p>B- Número de funcionalidades descritas en la especificación de requisitos.</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p>	<p><math>X = \text{Contable} / \text{Contable}</math></p> <p>A=Contable</p> <p>B=Contable</p>

		especificaciones de requisitos.			
--	--	---------------------------------	--	--	--

- Métricas de Exactitud.

Tabla 12: Métrica externa de la subcaracterística de calidad exactitud.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Exactitud esperada	¿Existen diferencias entre los resultados actuales y los razonablemente esperados?	Ejecutar los casos de pruebas de entrada versus salida y comparar los resultados actuales y los razonablemente esperados. Cuento el número de casos encontrados con diferencias inaceptables en relación con los resultados razonablemente esperados.	$X = A/T$  A - Número de casos encontrados con diferencias entre los resultados razonablemente esperados y aquellos resultantes más allá de lo permisible.  T - Tiempo de operación.	$0 \leq X$  A mayor cercanía al 0 resultará mejor	X = Contable/ Tiempo  A = Contable  T =Tiempo
Precisión	¿Con qué frecuencia los usuarios finales encontrar resultados con la precisión	Registrar el número de resultados con la precisión adecuada.	$X=A/T$  A- Número de resultados encontrados por los usuarios con un	$0 \leq X$  A mayor cercanía al 0 resultará mejor	X=Contable/ Tiempo  A= Contable  T=Tiempo



	adecuada?		nivel de precisión diferente de la necesaria.  T- Tiempo de Operación.		
--	-----------	--	--	--	--

- Métrica de Seguridad.

Tabla 13: Métrica externa de la subcaracterística de calidad Seguridad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Prevención de la corrupción de los datos	¿Cuál es la frecuencia de eventos de corrupción de los datos?	Contar las apariciones de eventos de datos mayor y menor corrupción.	<p>b) <math>X=1-A/N</math></p> <p>A- Número de veces que un evento de corrupción de los datos importantes se produce.</p> <p>N- Número de casos de prueba que provocó los eventos de corrupción de datos</p> <p>b) <math>Y=1-B/N</math></p> <p>B- Número de veces que un</p>	<p>a) <math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p> <p>b) <math>0 \leq Y \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p> <p>c) <math>0 \leq Z</math></p> <p>A mayor cercanía al 1 resultará mejor</p>	<p>X=Contable/ Contable</p> <p>A=Contable</p> <p>B=Contable</p> <p>N=Contable</p> <p>Y=Contable/ Contable</p> <p>T=Tiempo</p> <p>Z=Contable/ Contable</p>

			<p>evento de corrupción de datos de menor importancia se produjeron.</p> <p>c) <math>Z = A/T</math> ó <math>B/T</math></p> <p>T- Período de tiempo de funcionamiento.</p>		
Capacidad de control de acceso	¿Cómo controlar el acceso al sistema?	Contar el número de detectar operaciones ilegales en comparación con el número de operaciones ilegales que se encuentran en las condiciones.	<p><math>X = A/B</math></p> <p>A- Número de detectar diferentes tipos de operaciones ilegales.</p> <p>B- Número de tipos de operaciones ilegales que están en las condiciones.</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p>	<p><math>X = \text{Contable} / \text{Contable}</math></p> <p><math>A = \text{Contable}</math></p> <p><math>B = \text{Contable}</math></p>

- Métricas de Funcionalidad.

Tabla 14: Métrica externa de la subcaracterística de calidad Funcionalidad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Cumplimiento funcional	¿Cómo es compatible con la funcionalidad del producto a las normas aplicables, los estándares y regulaciones?	Cuenta el número de elementos que requieren de cumplimiento que se han cumplido y comparar con el número de elementos que requieran el cumplimiento en las condiciones especificadas.	$X=1-A/B$  A- Número de elementos cumplidos de las funcionalidades especificadas que no se han aplicado durante las pruebas.  B- Número total de elementos cumplidos con funcionalidad especificada.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X=Contable/ Contable  A=Contable  B=Contable
Cumplimiento de estándar de interfaz	¿Cómo se cumplen las interfaces con los reglamentos aplicables, las normas y regulaciones?	Cuenta el número de interfaces que se cumplieron y la compara con el número de interfaces a exigir el cumplimiento de las especificaciones.	$X=A/B$  A- Número de interfaces con problemas detectados.  B- Número total de Casos de pruebas utilizados.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X=Contable/ Contable  A=Contable  B=Contable

Métricas para la medición de la característica confiabilidad.

- Métricas Madurez.

Tabla 15: Métrica externa de la subcaracterística de calidad Madurez.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Densidad de fallos totales contra casos de prueba	¿Cuántos fallos totales fueron detectados durante un período de pruebas definido?	Cuente el número de fallos totales detectados y el número de casos de pruebas.	$X = 1 - A1 / A2$ A1 - Número total de fallos totales detectados. A2 - Número de casos de pruebas ejecutados.	$0 \leq X$ En dependencia del estadio de las pruebas. En etapas más avanzadas, mientras más pequeño, mejor.	X = Contable/ Contable A1 = Contable A2 = Contable
Grado de solución ante fallos totales	¿Cuántas condiciones de fallo total están resueltas?	Cuente el número de fallos totales que no se repitieron en determinado período de pruebas bajo condiciones similares. Mantenga un reporte de solución de problemas describiendo la situación de todos los fallos totales.	$X = A1 / A2$ A1 - Número de fallos totales solucionados. A2 - Número total de problemas reales detectados.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor, cuantos más fallos totales estén resueltos.	X = Contable/ Contable A1 = Contable A2 = Contable
Densidad de fallos	¿Cuántos fallos fueron detectados	Cuente el número de fallos detectados y compute su Densidad.	$X = A / B$ A - Número total	$0 \leq X$ Depende del	X = Contable/ Tamaño

	durante un período de pruebas definido?		de fallos detectados.  B - Tamaño del producto.	estadio de las pruebas. En etapas más avanzadas, mientras más pequeño, mejor.	A = Contable  B = Tamaño
Erradicación de fallos	¿Cuántos fallos han sido corregidos?	Cuente el número de fallos resueltos durante el período de pruebas y compárelos con el número total de fallos detectados y el número total de fallos pronosticados.	1) $X = A1 / A2$  A1 - Número de fallos solucionados.  A2 - Número total de fallos reales detectados.  2) $Y = A1 / A3$  A3 - Número total de fallos latentes pronosticados.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor  (cuanto menos fallos queden)  $0 \leq Y$  A mayor cercanía al 0 resultará mejor (cuanto menos fallos queden)	1) X = Contable/ Contable  A1 = Contable  A2 = Contable  2) Y = Contable/ Contable  A3 = Contable
Cobertura de las pruebas	¿Cuántos casos de pruebas requeridos han sido ejecutados detectados durante las pruebas?	Cuente el número de casos de pruebas que han sido ejecutados detectados durante las pruebas y compárelo con el número de casos de pruebas requeridos para obtener una adecuada cobertura de pruebas.	$X = A / B$  A –Número de casos de pruebas que han sido realmente ejecutados, y que representan el escenario de operación durante las pruebas.	$0 \leq X \leq 1$  Mientras más cercano al 1, mejor cobertura.	X = Contable/ Contable  A = Contable  B = Contable

			B – Número de casos de pruebas a ejecutar requeridos para cubrir los requisitos.		
Madurez de las pruebas	¿Está bien probado el producto?	Cuenta el número de casos de pruebas que han obtenido un resultado satisfactorio de los casos realmente ejecutados y compárelo con el número total de casos de pruebas requeridos para cubrir los requisitos.	$X = 1 - A / B$ A –Cantidad de NC que fueron detectadas por utilizar los casos de pruebas. B – Número de casos de pruebas a ejecutar para cubrir los requisitos.	$0 \leq X \leq 1$ A mayor cercanía al 1 resultará mejor	$X = \text{Contable} / \text{Contable}$ A = Contable B = Contable

- Métricas de Tolerancia ante fallos.

Tabla 16: Métrica externa de la subcaracterística de calidad Tolerancia ante fallos.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Evitación de operaciones incorrectas	¿Cuántas funciones están implementadas con capacidad de evitación de operaciones incorrectas?	Cuenta el número de casos de prueba de operaciones incorrectas que fueron evitadas para que no causaran fallos totales críticos o serios, y compárelo con el número de casos de pruebas ejecutados a los patrones de operaciones incorrectas considerados período de pruebas bajo condiciones similares.	$X = A / B$  A - Número de ocurrencia de fallos totales críticos o serios evitada.  A - Número de casos de pruebas ejecutados a los patrones de operaciones incorrectas (casi causantes de fallos) durante las pruebas.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor, cuantas más operaciones incorrectas del usuario sean evitadas.	X = Contable/ Contable  A1 = Contable  A2 = Contable
Evitar el fracaso	¿Cómo los patrones de muchas fallas fueron puestos bajo control para evitar fallos críticos y graves?	Cuenta el número de patrones de fallos a evitar y compara con el número de patrones de fallas para ser considerado.	$X=A/B$  A- Número de ocurrencias a evitar fallos críticos y graves contra los casos de prueba del patrón de culpa.  B- Número de	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor, el usuario puede evitar el fracaso más a menudo críticos o graves.	X=Contable/ Contable  A=Contable  B=Contable

			casos de prueba ejecutados del patrón de culpa (casi causando insuficiencia) durante la prueba.		
--	--	--	---	--	--

- Métricas de Recuperabilidad.

Tabla 17: Métrica externa de la subcaracterística de calidad Recuperabilidad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Grado de disponibilidad	¿Cuán disponible está el sistema para su uso durante u periodo de tiempo especificado?	<p>Pruebe el sistema en la producción como ambiente durante un período de tiempo especificado ejecutando todas las operaciones del usuario.</p> <p>Medir el período de tiempo de reparación cada vez que el sistema no estaba disponible durante las pruebas.</p>	<p>1) <math>X = (T_o / T_o + T_r)</math></p> <p>2) <math>Y = A1 / A2</math></p> <p><math>T_o</math> – Tiempo de operación.</p> <p><math>T_r</math> – Tiempo de reparación.</p> <p><math>A1</math> – Total de casos de uso del software disponibles exitosamente, cuando se han intentado utilizar.</p> <p><math>A2</math> - Total de casos de uso del</p>	<p><math>0 \leq X \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p> <p><math>0 \leq Y \leq 1</math></p> <p>A mayor cercanía al 1 resultará mejor</p>	<p><math>X = \text{Tiempo} / \text{Tiempo}</math></p> <p><math>T_o - \text{Tiempo}</math></p> <p><math>T_r - \text{Tiempo}</math></p> <p><math>Y = \text{Contable} / \text{Contable}</math></p> <p><math>A1 = \text{Contable}</math></p> <p><math>A2 = \text{Contable}</math></p>



			software que se han intentado utilizar durante el tiempo de observación.		
Tiempo medio de inactividad	¿Cuál es el tiempo promedio en que el sistema se mantiene no disponible cuando ocurre un fallo total y antes de la arrancada gradual?	Mida el tiempo de inactividad cada vez que el sistema se encuentre no disponible durante un período de prueba especificado y compute el tiempo medio.	$X = T / N$  T - Tiempo total de inactividad.  N - Número de desastres observados.  El peor caso o la distribución del tiempo de inactividad deben ser medidos.	$0 < X$  Cuanto menor sea mejor, el sistema estará inactivo por menos tiempo.	$X = \text{Tiempo} / \text{Contable}$  T = Tiempo N = Contable
Restaurabilidad	¿Cuán capaz es el producto de auto restaurarse luego de un evento anormal o una solicitud?	Cuenta el número de de restauraciones exitosas y compárelo con el número de restauraciones probadas requeridas por las especificaciones.  Ejemplos de requisitos de restauración son: - función deshacer - función rehacer	$X = A / B$  A – Número de casos de restauración exitosos.  B – Número de casos de restauración probados por los requisitos.	$0 \leq X \leq 1$  A mayor cercanía al 1 mejor y el producto es más capaz de de restaurarse en casos definidos.	$X = \text{Contable} / \text{Contable}$  A = Contable B = Contable

Métricas para la medición de la característica usabilidad.

- Métricas de Comprensibilidad.

Tabla 18: Métrica externa de la subcaracterística de calidad Comprensibilidad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Accesibilidad a demos	¿A qué proporción de demos/tutoriales pueden acceder los usuarios?	Conduzca las pruebas de usuario. Observe el comportamiento del usuario.	$X = A / B$  A - Número de demos/tutoriales a los que pueden acceder los usuarios exitosamente.  B - Número total de demos/tutoriales a los que se puede acceder.	$0 < X \leq 1$  A mayor cercanía al 1 mejor	X =Contable/ Contable  A = Contable B = Contable
Comprensibilidad de la función	¿Qué proporción de las funciones de producto el usuario podrá entender correctamente?	Llevar a cabo pruebas de usuario y el usuario entrevista con cuestionarios. Cuente el número de la función de interfaz de usuario cuando el destino sean fácilmente	$X=A/B$  A- Número de funciones de la interfaz, cuya finalidad es correctamente descrito por el	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X= Contable/ Contable  A= Contable B=Contable

		comprensibles para el usuario y compararlo con el número de funciones disponibles para el usuario.	usuario.  B- Número de funciones disponibles en la interfaz.		
--	--	--	--	--	--

- Métricas de Atracción.

Tabla 19: Métrica externa de la subcaracterística de calidad Atracción.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Adaptabilidad de la apariencia de la interfaz	¿Qué proporción de los elementos de la interfaz puede ser, por su apariencia, adaptado por el usuario para la satisfacción del mismo?	Conduzca las pruebas de usabilidad. Observe el comportamiento del usuario.	$X = A / B$  A - Número de elementos de la interfaz del sistema cuya apariencia puede ser adaptada por el usuario.  B - Número de elementos de la interfaz del sistema cuya apariencia querría adaptar el usuario.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X = Contable/ Contable  A = Contable  A = Contable

- Métricas de Instructibilidad.

Tabla 20: Métrica externa de la subcaracterística de calidad Instructibilidad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Eficacia de la documentación del usuario y / o sistemas de ayuda en el uso	¿Qué proporción de las funciones pueden ser utilizados correctamente después de leer la documentación o el uso de sistemas de ayuda?	Cuente el número de la función que se usa correctamente después de leer la documentación o el uso de sistemas de ayuda y comparar con el número total de funciones.	$X=A/B$  A- Número de funciones que se pueden utilizar.  B- Total de número de funciones que ofrece.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X= Contable/ Contable  A=Countable  B=Contable

Métricas para la medición de la característica Eficiencia.

- Métricas de Eficiencia.

Tabla 21: Métrica externa de la subcaracterística de calidad Eficiencia.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Cumplimiento de la eficiencia	¿Cómo es compatible la eficacia del producto a las normas, los estándares y	Cuente el número de elemento que requieren de cumplimiento que se han cumplido y comparar con el número de elementos que requieran el cumplimiento según	$X=1-A/B$  (X=relación de elementos cumplimiento satisfechos en relación a la	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X= Contable/ Contable  A=Contable  B=Contable

	regulaciones?	las especificaciones.	<p>eficiencia)</p> <p>A- Número de elementos cumplidos de la eficiencia que se especifica que no se han aplicado durante las pruebas.</p> <p>B- Número total de elementos cumplidos de la eficiencia especificado.</p>		
--	---------------	-----------------------	--	--	--

- Métricas de Comportamiento en el tiempo.

Tabla 22: Métrica externa de la subcaracterística de calidad Comportamiento en el tiempo.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Tiempo de respuesta	<p>¿Cuál es el tiempo necesario para completar una tarea específica?</p> <p>¿Cuánto tiempo se tarda antes</p>	Iniciar una tarea específica. Medir el tiempo que tarda la muestra para completar su operación. Mantener un registro de cada intento.	$T = (\text{Tiempo al obtener el resultado}) - (\text{Tiempo al entrar los datos})$	<p><math>0 &lt; T</math></p> <p>Cuanto antes es mejor</p>	$T = \text{Tiempo}$

	de que la respuesta del sistema a una operación especificada?				
--	---	--	--	--	--

- Métricas de Rendimiento.

Tabla 23: Métrica externa de la subcaracterística de calidad Rendimiento.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Rendimiento	¿Cómo muchas tareas pueden realizarse con éxito en un período determinado de tiempo?	Calibrar cada tarea de acuerdo a la prioridad dada por objeto. Iniciar las tareas de varios puestos de trabajo. Medir el tiempo que tarda la tarea para completar su operación. Llevar un registro electrónico de cada intento.	$X=A/T$ A- Número de tarea completada. T= Período de tiempo de observación.	$0 < X$ El más grande es el mejor	X= Contable/ Tiempo A=Contable T=Tiempo

Métricas para la medición de la característica Portabilidad.

- Métricas de Adaptabilidad.

Tabla 24: Métrica externa de la subcaracterística de calidad Adaptabilidad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Capacidad de adaptación al entorno del software	<p>¿Puede el usuario o desarrollador de software adaptarse fácilmente al entorno?</p> <p>El software es lo suficientemente capaz de adaptarse al entorno de funcionamiento.</p>	Observar el comportamiento del usuario o mantenedor cuando el usuario está tratando de adaptar el software al entorno de operación.	$X = 1 - A/B$ <p>A- Número de funciones operativas de las tareas que no se completa o no dio suficiente para satisfacer el nivel adecuado durante la prueba de funcionamiento combinado el software con el sistema operativo o el software con aplicaciones simultánea.</p> <p>B- Cantidad total de la función que se probaron.</p>	$0 \leq X \leq 1$ <p>A mayor cercanía al 1 resultará mejor</p>	<p>X= Contable/ Contable</p> <p>A=Contable</p> <p>B=Contable</p>

- Métricas de Inestabilidad.

Tabla 25: Métrica externa de la subcaracterística de calidad Instalabilidad.

Nombre de la métrica	La métrica se propone medir	Método de aplicación	Medición (fórmula)	Interpretación del valor obtenido	Tipo de medida
Facilidad de instalación	¿El usuario puede fácilmente instalar en el entorno de funcionamiento?	Observar el comportamiento del usuario cuando está tratando de instalar el software al entorno de funcional.	$X=A/B$  A- Número de casos que un usuario tuvo éxito en cambiar el funcionamiento de instalación para su conveniencia.  B- Número total de casos que un usuario ha intentado cambiar la operación de instalación para su conveniencia.	$0 \leq X \leq 1$  A mayor cercanía al 1 resultará mejor	X= Contable/ Contable  A= Contable  B= Contable



**Anexo 5:** Tipos de prueba definidos según las características de calidad de la ISO/IEC 9126.

Características de calidad	Tipos de pruebas
Funcionalidad	<p><b>Funcionales:</b> Consisten en la revisión de los requisitos aceptados por el cliente contra las funcionalidades presentes en la aplicación.</p> <p><b>Seguridad:</b> Capacidad del producto de software para proteger información y los datos, para que personas o sistemas desautorizados no puedan leer o puedan modificar los mismos, y las personas o sistemas autorizados tenga el acceso a ellos. Se han definido 3 niveles de seguridad.</p> <p><b>Volumen (*):</b> Enfocada en verificar las habilidades de los programas para manejar grandes cantidades de datos, tanto como entrada, salida o residente en la BD.</p>
Confiabilidad	<p><b>Recuperación y tolerancia a fallas:</b> Verificar que los procesos de recuperación (manual o automática) restauran apropiadamente la base de datos, aplicaciones y sistemas, y los llevan a un estado conocido o deseado.</p> <p><b>Benchmark (Comparativa) (*):</b> Es un tipo de prueba que compara el rendimiento de un elemento nuevo o desconocido a uno de carga de trabajo de referencia conocido. Esta en estudio de forma general, pero se ha comenzado a aplicar el tipo de prueba <b>Comparativa de código</b>, donde se compara el código nuevo entregado por el proyecto, con el que se liberó por el LIPS, de manera que se pueda comprobar que este no ha tenido cambios o que estos se han realizado solo en la interfaz del producto.</p>
Eficiencia	<p><b>Contención (*):</b> Enfocada a la validación de las habilidades del elemento a probar para manejar aceptablemente la demanda de múltiples actores sobre un mismo recurso (registro de recursos, memoria, etc.).</p> <p><b>Carga:</b> Usada para validar y valorar la aceptabilidad de los</p>

	<p>límites operacionales de un sistema bajo carga de trabajo variable, mientras el sistema bajo prueba permanece constante. La variación en carga es simular la carga de trabajo promedio y con picos que ocurre dentro de tolerancias operacionales normales.</p> <p><b>Estrés:</b> Enfocada a evaluar cómo el sistema responde bajo condiciones anormales. (extrema sobrecarga, insuficiente memoria, servicios y hardware no disponible, recursos compartidos no disponible).</p> <p><b>Rendimiento (*):</b> Enfocadas a monitorear el tiempo en flujo de ejecución, acceso a datos, en llamada a funciones y sistema para identificar y direccionar los cuellos de botellas y los procesos ineficientes.</p> <p><b>Picos (*):</b> Se trata de observar el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga. Esta prueba se recomienda que sea realizada con un software automatizado que permita realizar cambios en el número de usuarios mientras que los administradores llevan un registro de los valores a ser monitoreados.</p>
Mantenibilidad	<p>No se han definido tipos de prueba realizar durante la liberación, pues esta característica tiene condiciones particulares. Se está estudiando a fondo de qué forma se puede comprobar si las aplicaciones informáticas que se prueben, cumplen con los elementos esenciales que lo puedan hacer mantenible en el tiempo.</p>
Portabilidad	<p><b>Configuración:</b> Enfocada a asegurar que funciona en diferentes configuraciones de hardware y software. Esta prueba es implementada también como prueba de rendimiento del sistema.</p> <p><b>Instalación:</b> Enfocada a asegurar la instalación en diferentes configuraciones de hardware y software bajo diferentes condiciones (insuficiente espacio en disco, etc.).</p>

Los tipos de prueba marcados con (\*) los casos que están en estudio para su implementación en el LIPS. El resto de los casos se están ejecutando, algunos con los probadores de 2do año y otros directamente con especialistas por el nivel de complejidad y profundidad.

Se han definido dos tipos de prueba, no asociados a características de calidad específicas, pero que son importantes y se realizan sistemáticamente:

- Evaluación estática: Las técnicas de evaluación estática de artefactos del desarrollo se les conoce de modo genérico por Revisiones, las cuales pretenden detectar manualmente defectos en cualquier producto del desarrollo, o sea, el artefacto se evalúa mediante la lectura del mismo, pues no es ejecutable o no se ejecuta en ese momento.
- Pruebas de Regresión: Prueba enfocada a comprobar que las No Conformidades detectadas en una iteración fueron resueltas correctamente por el equipo de proyecto, para poder pasar a la siguiente iteración, en la que se comprobará que no se introdujeron errores al corregir los encontrados.

#### **Anexo 6:** Requisitos funcionales de la herramienta para la gestión de las métricas.

##### **RF 1: Gestionar Proyecto**

- 1.1 Adicionar proyecto
  - 1.1.1 Nombre del proyecto
  - 1.1.2 Centro
  - 1.1.3 Tipo de proyecto
- 1.2 Modificar proyecto
  - 1.2.1 Nombre del proyecto
  - 1.2.2 Centro
  - 1.2.3 Tipo de proyecto
- 1.3 Eliminar proyecto
- 1.4 Visualizar proyecto
- 1.5 Buscar Proyecto

##### **RF 2: Asignar métricas**

- 2.1 Asignar métricas a un artefacto

##### 2.1.1 Proyecto

##### 2.1.2 Artefacto

##### 2.1.3 Tipo de Métrica

##### 2.1.4 Iteración

##### 2.1.5 Métrica

##### 2.2 Eliminar asignación de métricas a un artefacto

##### 2.3 Visualizar asignación de métricas a un artefacto

##### 2.4 Seleccionar métricas asignar

##### 2.5 Buscar métrica asignada

##### **RF 3: Calcular métricas**

##### 3.1 Insertar datos de las métricas

##### 3.2 Calcular métricas

3.3 Visualizar cálculo de las métricas

**RF 4: Imprimir Reportes de las métricas calculadas de los proyectos por tipo de características de calidad**

**RF 5: Evaluar Proyectos**

5.1 Seleccionar proyectos

5.2 Seleccionar artefactos a tener en cuenta en la evaluación

5.3 Visualizar resultado de la evaluación

5.4 Modificar la evaluación

5.5 Buscar Proyecto evaluado

**RF 6: Generar reportes de las métricas por proyecto**

**RF 7: Generar reporte General**

**RF 8: Configuración**

8.1 Adicionar tipo de proyecto

8.2 Modificar tipo de proyecto

8.3 Eliminar tipo de proyecto

8.4 Visualizar tipo de proyecto

8.5 Buscar tipo de proyecto

8.6 Adicionar Centro

8.6.1 Nombre del centro

8.6.2 Especialidad del centro

8.7 Modificar Centro

8.7.1 Nombre del centro

8.7.2 Especialidad del centro

8.8 Eliminar Centro

8.9 Visualizar Centro

8.10 Buscar Centro

8.11 Adicionar tipo de artefacto

8.12 Modificar tipo de artefacto

8.13 Eliminar tipo de artefacto

8.14 Visualizar tipo de artefacto

8.15 Buscar tipo de artefacto

8.13 Adicionar artefacto.

8.13.1 Nombre del artefacto

8.13.2 Tipo de artefacto

8.14 Modificar artefacto

8.13.1 Nombre del artefacto

8.13.2 Tipo de artefacto

8.15 Eliminar artefacto

8.16 Visualizar artefacto

8.17 Buscar artefacto

8.18 Adicionar evaluación

8.19 Modificar evaluación

8.20 Eliminar evaluación

8.21 Visualizar evaluación

8.22 Buscar evaluación

8.23 Adicionar iteración

8.24 Modificar iteración

8.25 Eliminar iteración

8.26 Visualizar iteración

8.27 Buscar iteración

**RF 9: Autenticar Usuario**

9.1 Usuario

9.2 Contraseña

**RF 10: Gestionar Roles**

10.1 Adicionar roles

10.2 Modificar roles

10.3 Eliminar roles

10.4 Visualizar roles

**RF 11: Gestionar Usuario**

11.1 Adicionar usuario

11.2 Modificar usuario

11.3 Eliminar usuario

11.4 Visualizar Usuario

**RF 12: Evaluar Artefacto**

12.1 Evaluar artefacto

12.1.1 Proyecto

12.1.2 Artefacto

12.1.3 Iteración

12.1.4 Evaluación

12.2 Seleccionar métricas a tener en cuenta en la evaluación

12.3 Visualizar resultado de la evaluación

12.4 Modificar la evaluación

12.5 Buscar artefacto evaluado

## GLOSARIO DE TÉRMINOS Y SIGLAS

### Términos:

**Actividad:** Conjunto de operaciones o tareas propias de una persona o entidad que permite que el trabajo ha realizar sea descrito y entendido de manera precisa por aquellos que tienen que ejecutarlo.

**Artefacto:** Es un término general, para cualquier tipo de información creada, producida, cambiada o utilizada por los trabajadores en el desarrollo del sistema.

**Calidad:** Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas.

**Calidad del software:** la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

**Equipo de desarrollo:** Es un grupo de trabajo constituido por una serie de profesores, investigadores, colaboradores y alumnos unidos en la ilusión de acometer un determinado proyecto o avanzar en el conocimiento y en la investigación teórica y aplicada.

**Estándar:** Lo que es establecido por la autoridad, la costumbre o el consentimiento general. En este sentido se utiliza como sinónimo de norma.

**Efectividad:** Capacidad de lograr el efecto que se desea o se espera.

**Gestión:** Gestión es la acción y efecto de gestionar o la acción o efecto de administrar. Comprende todas las actividades de una organización que implican el establecimiento de metas u objetivos, así como la evaluación de su desempeño y cumplimiento; además del desarrollo de una estrategia operativa que garantice la supervivencia de la misma, según al sistema social correspondiente.

**Gestión del conocimiento:** Es la gestión de los activos intangibles que generan valor para la organización. La mayoría de estos intangibles tienen que ver con procesos relacionados de una u otra forma con la captación, estructuración y transmisión de conocimiento. Por lo tanto, la Gestión del Conocimiento tiene en el aprendizaje organizacional su principal herramienta.

**Gestión de proyecto:** La Gestión de Proyectos tiene como finalidad principal la planificación, el seguimiento y control de las actividades y de los recursos humanos y

## GLOSARIO DE TÉRMINOS Y SIGLAS

materiales que intervienen en el desarrollo de un Sistema de Información o en la vida de un proyecto.

**Herramientas:** Utensilios o provisiones necesarias para poder emprender un proyecto de software. Soportan los procesos de desarrollo de software modernos.

**Indicador:** Un indicador es una medida que ofrece una estimación o evaluación de determinados atributos derivados de un modelo definido con respecto a las necesidades de información. Los indicadores son la base para el análisis y la toma de decisiones. La medición se basa siempre en una información imperfecta, por tanto cuantificar la incertidumbre, la precisión, o la importancia de los indicadores es un componente esencial de la presentación del valor real del indicador.

**Medida:** proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto.

**Medición:** es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas.

**Métrica:** una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.

**Norma:** La norma es una regla a seguir para alcanzar un fin determinado. Las normas se crean en Comisiones Técnicas de Normalización. Una vez elaborada la norma, esta es sometida durante seis meses a la opinión pública. Transcurrido este tiempo y analizadas las observaciones se procede a su redacción definitiva, con las posibles correcciones que se estimen, publicándose luego. Todas las normas son sometidas a revisiones periódicas con el fin de ser actualizadas.

**Proceso:** Es un conjunto de actividades y resultados asociados que producen un resultado.

**Producto:** Conjunto de artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables y documentación.

**Proyecto:** Combinación de recursos humanos y no humanos reunidos en una organización temporal para conseguir un propósito, tiene un punto de de comienzo definido y con objetivos definidos mediante los que se identifican.

**Proyecto de Software:** El elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

**Requerimiento:** Son capacidades o características que debe tener el sistema o

## GLOSARIO DE TÉRMINOS Y SIGLAS

modelo desarrollo para satisfacer la demanda y/o necesidad del cliente.

**Técnicas:** Sucesión ordenada de acciones que se dirigen a un fin concreto, conocido y que conduce a unos resultados precisos.

**Unidad de Medida:** Una cantidad, definida y adoptada por convención, con la cual otras cantidades del mismo tipo se comparan con el fin de expresar su magnitud con relación a esa cantidad. Sólo las cantidades expresadas en las mismas unidades de medida son directamente comparables.

### Siglas:

**GQM:** Goal Question Metric.

**IEEE:** Corresponde a las siglas de The Institute of Electrical and Electronics Engineers, el Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas.

**ISO:** Organización Internacional de Estándares.

**IBM:** Es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) sus siglas en inglés, (International Business Machines).

**LIPS:** Laboratorio Industrial de Pruebas de Software.

**PSP:** Proceso de Software Personal.

**RC:** Requerimientos de Calidad.

**SEI:** Instituto de Ingeniería de Software. El SEI es un centro de investigación y desarrollo patrocinado por el departamento de defensa de los Estados Unidos y gestionado por la Carnegie-Mellon.

**SQS:** Software Quality Systems.

**TSP:** Proceso de Software en Equipo.

**UCI:** Universidad de Ciencias Informáticas.