

Universidad de las Ciencias Informáticas
Facultad 2



**“Métricas para Evaluar la Calidad en los Proyectos del
Polo de Telecomunicaciones.”**

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

Autoras:

Lisy Yera Muñoz

Carmen Rosa García Silva

Tutor:

Ing. Dariena Ramirez Luján

Co-tutor:

Ing. Ariagnis Yero Guevara

Ciudad de la Habana, Junio 2009



El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes, es la oportunidad.

Victor Hugo

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____. Autores:

Lisy Yera Muñoz

Carmen Rosa García Silva

Firma del Autor

Firma del Autor

Ing. Dariena Ramirez Luján

Ing. Ariagnis Yero Guevara

Firma del Tutor

Firma del Tutor

Datos de Contacto

Síntesis del Tutor: Ingeniero en Ciencias Informáticas y profesor de Ciencias Informáticas. Graduado por la Universidad de las Ciencias Informáticas en Julio del 2007. Actualmente profesor de la Universidad de las Ciencias Informáticas (UCI). Ha estado vinculado al trabajo productivo e investigativo en temas relacionados con la calidad del software al desempeñarse como Asesora de Calidad. Actualmente trabaja como Analista Principal en uno de los proyectos productivos de la Facultad 2.

Síntesis del Cotutor: Ingeniero en Ciencias Informáticas y profesor de Ciencias Informáticas. Graduado por la Universidad de las Ciencias Informáticas en Julio del 2007. Actualmente profesor de la Universidad de las Ciencias Informáticas (UCI). Ha estado vinculado al trabajo productivo en temas relacionados con las telecomunicaciones. Actualmente se encuentra cursando una Maestría en Ciencias de la Computación

Agradecimientos

A mis padres (María Isabel y Oneido) por el amor, comprensión y dedicación que me han dado. Por no faltar nunca cuando los necesité, por complacer todos mis caprichos y tener tanta paciencia. Por escucharme y aconsejarme. Son lo que más quiero...

A mi maravillosa hermana (Neydis), el mejor regalo que me dio la vida, parte de mis alegrías y preocupaciones. Gracias por los consejos que siempre me diste, más que una hermana eres mi segunda mamá. Te quiero infinitamente...

A (Yisenia y Yeisy) por ser unas primas especiales. Por preocuparse siempre por mí estando tal lejos...Las quiero mucho.

A mis abuelos por quererme siempre, en especial (Cesilia)...por ser tan alegre y preocuparte por mí en todo momento...Los quiero a todos.

A mis tíos y primos por su preocupación y cariño.

A mis mejores amigas de la universidad (Danay, Leoveydis y Naya), que aunque no estén en este momento conmigo, siempre se preocuparon por mí. Gracias por su cariño...

A mis amigas presentes (Mayrilis y Yaima) por compartir buenos y malos momentos, por estar siempre ahí para mí.

A mi compañera de tesis y mi amiga (Carmen Rosa), por tanto esfuerzo que hicimos juntas para realizar nuestro sueño y por estar siempre cuando lo necesité.

A (Elier), por estar a mi lado en este momento tan difícil, por ser partícipe del momento más importante de mi vida. Por apoyarme y darme tanto cariño. Te quiero mucho...

A mis amigos (Repa y Andrés) que me ayudaron y apoyaron siempre. Al Repa, gracias por tus consejos y las largas horas de charlas...siempre me sirvieron...Andrés, por ser tan divertido...

A mis tutoras (Daxiena y Ariganis) por la ayuda brindada en este tiempo.

A todos mis compañeros de cinco años, gracias por conocerlos...

...A todos Muchas Gracias...

Lisy

A mis padres Maidolis y Blanco, por ser la razón de mí existir, por su dedicación, amor, cariño, comprensión, por enseñarme siempre el buen camino y sobre todo por tener fe y confianza siempre en mí.

A mi abuelito Julio, por darme siempre su amor y consejos, te quiero mucho.

A mi abuelita Rosa que aunque solo pudo verme iniciar este sueño se que estaría orgullosa de verme alcanzarlo.

A mi tía Nani que aunque es un poco cascarrabias la quiero mucho.

A mi tía Rosi y tío Jorge, por ser mis ejemplos de persona a seguir al sobreponerse a todo en la vida.

A mis primos Pulgarcita, el chini y Jorge Julio, por ser como mis hermanos y apoyarme siempre, los quiero mucho.

A mi hermana Dayani, por quererme y creer en mí ciegamente.

A toda mi familia en general, por apoyarme siempre y confiar en mí.

A mi novio Andrés por apoyarme y guiarme en todo este tiempo que hemos estado juntos, por quererme y comprenderme, por todos los momentos felices que hemos pasado juntos, te quiero.

A la familia de mi novio, por quererme como un miembro más de la familia y apoyarme siempre en especial a mi suegra linda y a mi suegro y además a mis cuñadas que son como mis hermanas al igual que mis cuñados.

A mis vecinos a quienes he querido siempre como a mi familia.

A Disney y Lumy por haberse convertido en mis amigos incondicionalmente.

A mi compañera de tesis quien fue para mí la mejor sin discusión alguna, la que se encargó de darme ánimos cuando no tenía y sobre todo de hacerme reír cuando estaba triste.

A todos mis compañeros que me acompañaron en estos cinco años de universidad.

A todos los profesores que durante todos estos años contribuyeron con mi formación profesional.

A mis tutoras Dariena y Ariagnis, por brindarme su ayuda y dedicación, sin ellas no hubiera sido posible alcanzar esta meta.

A todas esas personas que de alguna forma ayudaron a que este día existiera...

Muchas Gracias por existir.

Carmen Rosa

Dedicatoria

*A mi Mamá, mi Papá y mi Hermana.
Por compartir mi sueño...*

Lisy

*A mis padres Maidolis y Blanco y a toda mi familia y seres
queridos, por ayudarme a realizar este sueño...*

Carmen Rosa

Resumen

Hoy en día las compañías del mundo industrializado reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general. La industria de desarrollo de software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de calidad en el ámbito del software. Una de las causas que provoca que el producto no tenga la calidad requerida es el control y evaluación del mismo.

El principal objetivo de este trabajo de diploma es realizar una Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones. Para lograr lo planteado anteriormente se realizó un estudio de las métricas existentes, así como, sus características y diferentes clasificaciones. Se realizaron también entrevistas a los diferentes Líderes de Proyectos de la Facultad 2 para determinar cómo se evalúa la calidad y conocer de esta forma, en cuáles se emplean Métricas de Software.

Como resultado de la investigación llevada a cabo, se concluyó una propuesta de solución que responde a las necesidades de los proyectos. Esta propuesta permitirá que los Líderes de proyectos tengan mayor control sobre el desarrollo del mismo, así como, mitigar y dar tratamiento a los errores que se van creando en el avance del proyecto, de forma tal que se realice la entrega en la fecha prevista del producto con una mejor calidad.

Palabras Claves: Métricas, calidad, software

Índice

Declaración de Autoría	I
Datos de Contacto	II
Agradecimientos.....	III
Dedicatoria	VI
Resumen	VII
Introducción	1
Capítulo 1: Fundamentación Teórica	4
1.1 Introducción	4
1.2 Conceptos fundamentales	4
1.2.1 Calidad	4
1.2.2 Calidad de Software	5
1.2.3 Control y Evaluación de la Calidad del Software.....	6
1.2.4 Métricas de Software	7
1.2.4.1 Características de las Métricas de Software	10
1.2.4.2 Clasificación de las Métricas de Software.....	11
1.3 Área de Aplicación.....	13
1.4 Técnicas de Validación de Resultados.....	14
1.4.1 Método DELPHI.....	15
1.4.2 Método Multicriterios.....	16
1.4.3 Test de Turing.....	17
1.4.4 Método de Validación Interna y Externa.....	17
1.4.5 Selección del Método de Validación	18
1.5 Conclusiones del Capítulo	18

Capítulo 2: Descripción de la Propuesta de Solución	19
2.1 Introducción	19
2.2 Resultados Obtenidos de la Entrevista.	19
2.3 Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones.	20
2.3.1 Métricas de Proyecto	20
2.3.1.1 Métrica para la Eficacia de la Eliminación de Defectos	20
2.3.1.2 Métrica de Calidad de la Especificación de Requisitos	22
2.3.1.3 Métrica para el Control de los Casos de Uso Diseñados	23
2.3.1.4 Métrica para la Agilidad en la Gestión de Cambios de los Requisitos	24
2.3.1.5 Métrica de Mantenimiento.....	25
2.3.1.6 Métrica de Esfuerzo de Módulo	26
2.3.1.7 Métrica de Esfuerzo de Proyecto	27
2.3.1.8 Métrica para Productividad del Proyecto.....	27
2.3.1.9 Métricas Orientadas a Puntos de Función	28
2.3.2 Guía para la Aplicación de la Propuesta	32
2.4 Conclusiones del Capítulo	34
Capítulo 3: Validación de la Propuesta de Solución	35
3.1 Introducción	35
3.2 Método de Validación de la Propuesta.....	35
3.2.1 Aplicación de la Propuesta en los Proyectos TIP y CUBACEL.....	36
3.2.2 Conclusión de la Aplicación de la Propuesta	51
3.3 Conclusiones del capítulo	52
Conclusiones Generales.....	53
Recomendaciones	54

Referencias Bibliográficas	55
Bibliografía	57
Anexos	58
Glosario de Términos	61

Índice de Figuras

Figura 1 Definición de Métricas de Software	8
Figura 2 Gráfica de valores alcanzados en la métrica EED.....	37
Figura 3 Gráfica de valores de la métrica calidad de especificación de requisitos.....	39
Figura 4 Gráfica de valores de la métrica CCUD.....	40
Figura 5 Gráfica de valores de la métrica AGCR.....	42
Figura 6 Gráfica de valores de la métrica de mantenimiento.....	43
Figura 7 Gráfica de valores de la métrica Esfuerzo por módulo	45
Figura 8 Gráfica de valores de la métrica esfuerzo del proyecto	47
Figura 9 Gráfica de valores de la métrica productividad orientada a los PF	51

Índice de Tablas

Tabla 1 Escala de evaluación del resultado de la Métrica Eficacia de la Eliminación de Defectos	22
Tabla 2 Escala de evaluación de los resultados de la Métrica Calidad de Especificación de Requisitos.	23
Tabla 3 Escala de evaluación del resultado de la Métrica para el Control de los Casos de Uso diseñados.	24
Tabla 4 Escala de evaluación del resultado de la Métrica para la Agilidad en la Gestión de Cambio de los Requisitos.	25
Tabla 5 Escala de evaluación del resultado de la Métrica de Mantenimiento.	26
Tabla 6 Puntos de Función no ajustados.	29
Tabla 7 Factores de Complejidad.	30
Tabla 8 Resumen de la Propuesta de Métricas.....	32
Tabla 9 Métrica para obtener la eficacia de la eliminación de defectos	36
Tabla 10 Métrica para evaluar la Calidad de la especificación de requisitos	38
Tabla 11 Métricas para el control de los casos de uso diseñados	39
Tabla 12 Métrica para la agilidad en la gestión de cambios de los requisitos	41
Tabla 13 Métrica de mantenimiento	42
Tabla 14 Métrica para el esfuerzo del módulo.....	44
Tabla 15 Métrica para el esfuerzo del proyecto.....	46
Tabla 16 Métrica para la obtención de los puntos de Función para el proyecto TIP	47
Tabla 17 Factores de complejidad para el proyecto TIP.....	48
Tabla 18 Métrica para la obtención de los puntos de Función para el proyecto CUBACEL	49
Tabla 19 Factores de complejidad para proyecto CUBACEL	49
Tabla 20 Métrica para el cálculo de la productividad del proyecto orientada hacia los puntos de función	50

Introducción

La industria del software, a diferencia de muchas otras industrias en el mundo, existe desde hace muy poco tiempo. Lo que ha llamado la atención del mercado hacia esta gran industria han sido dos factores esenciales: la velocidad con que ha crecido y su gran alcance. El interés por hacer uso de software especializado que permita automatizar procesos, ha estado presente siempre en personas que trabajan en distintos campos, que vieron el avance que representaba hacer uso de los mismos.

Al haber tanta demanda en cuanto al campo de la informática se iniciaron muchas investigaciones en la rama relacionada con el software y hardware. Con el tiempo los costos se redujeron y el software se convirtió en un negocio rentable, muchas personas empezaron a producir y así nacieron las primeras grandes empresas de software. Esto trajo consigo un problema natural en el proceso: al haber tantos desarrolladores en distintos países produciendo distintas aplicaciones, empezó a haber diversidad de estilos y a variar la calidad del producto final. En este marco se hizo necesario un estándar que permitiera a los consumidores de software decidir si el producto que estaban recibiendo contaba con la calidad necesaria y si cumplía ciertos requisitos de funcionalidad.

La calidad es un indicador clave, del que dependen la mayor parte de las organizaciones. Es un conjunto de características y propiedades que representan una ventaja estratégica permitiendo actuar sobre las tareas menos eficientes, mejorándolas. La calidad ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software. Para obtener un software con calidad es necesario la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, con el objetivo de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

Una manera eficaz para evaluar algunos parámetros como funcionalidad, complejidad y eficiencia durante el proceso de producción de software es la utilización de métricas, debido a que estas permiten evaluar la productividad de los desarrolladores, establecer una línea base para la estimación y de igual forma ayudan a justificar el uso de nuevas herramientas.

El uso de las métricas ha ganado popularidad entre las empresas desarrolladoras de software, reconociendo la importancia que tienen las mediciones para cuantificar y por consiguiente gestionar de forma más efectiva la calidad de los procesos y productos de software. En centros que se dedican

exclusivamente a la informática se tiene noción de la necesidad de formalizar los mecanismos de estimación, comprendiendo que los registros históricos de antiguos proyectos realizados pueden ayudar a estimar con mayor exactitud el esfuerzo, tiempo de desarrollo, costo, posibles errores, recursos y tamaño para los nuevos proyectos.

En la mayoría de los desafíos técnicos, las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. Son consideradas como un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo del software.

La Universidad de las Ciencias Informáticas (UCI) no es sólo una entidad educativa, sino también productiva, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo, destinada a la informatización de la sociedad cubana, así como, a la comercialización de estos productos. Cada una de las facultades de la UCI tiene asociado un perfil para producir. Particularmente la Facultad 2 se enfoca en la elaboración de productos relacionados con las telecomunicaciones, pero no existe conocimiento suficiente de las principales métricas posibles a aplicar en estos proyectos productivos con el objetivo de alcanzar una mayor calidad de los productos, ya que actualmente no se ha establecido un plan de métricas apropiadas para la aplicación en los proyectos de producción, puesto que este es un tema virgen, que se encuentra en investigación. Esto trae como consecuencia la desinformación de los Líderes de Proyectos respecto al desarrollo del mismo, la entrega de los productos sin la calidad requerida y con retraso en la fecha de culminación, por lo que se hace necesario contar con un repositorio de métricas en el Polo de Telecomunicaciones de manera que los Líderes de proyecto puedan acceder a ellas y aplicarlas en su proyecto.

Debido a la importancia que tiene el buen uso de las métricas de software se ha decidido plantear el siguiente problema: ¿Cómo garantizar la calidad en los proyectos del Polo de Telecomunicaciones?

El objeto de estudio del presente trabajo son las Métricas de Software. Reconociendo además como campo de acción las Métricas para Evaluar la Calidad del Software para Proyectos del Polo de Telecomunicaciones.

Con el fin de solucionar el problema planteado se define como objetivo general de este trabajo: Proponer un Conjunto de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones.

Con este trabajo se pretende aumentar y motivar el uso de las métricas más apropiadas en los proyectos productivos contribuyendo al desarrollo de software con mayor calidad.

Para darle solución a los objetivos trazados, se definen las siguientes tareas de investigación:

1. Realizar estudio de las métricas de software y su clasificación.
2. Analizar las características que presentan los Proyectos del Polo de Telecomunicaciones.
3. Describir la Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones.
4. Validar la Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones.

Esta investigación presenta la siguiente Idea a defender: La aplicación de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones garantizará la obtención de productos con mayor calidad.

El contenido de este trabajo está estructurado en tres capítulos, organizados de la siguiente manera:

Capítulo 1: Fundamentación Teórica. En este capítulo se hace un análisis de los diferentes conceptos y definiciones que se tendrán en cuenta durante todo el trabajo sobre el tema de Calidad de Software y Métricas del Software. Se describe el área de aplicación de las métricas que serán propuestas en el próximo capítulo, así como, las características que presentan los proyectos que se realizan en la Facultad 2. También se hace referencia a los diferentes métodos de validación de resultados, con el objetivo de seleccionar el que se utilizará en la validación de la propuesta de solución.

Capítulo 2: Descripción de la Propuesta de Solución. En este capítulo primeramente se describen los resultados obtenidos en las entrevistas realizadas a los diferentes Líderes de Proyectos de la Facultad 2 y los responsables de Evaluar la Calidad en estos proyectos, las cuales constituyen las bases para la Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones que se describe a continuación.

Capítulo 3: Validación de la propuesta. En este capítulo se describe la validación de la propuesta de métricas mediante el método de validación interna y externa, llamada también teórica y empírica.

Capítulo 1: Fundamentación Teórica

1.1 Introducción

En el presente capítulo se hace un análisis del concepto de métrica y todo lo referente a este tema, así como sus características y diferentes clasificaciones. Se abordan los términos que sirven de soporte teórico a la investigación desarrollada y que estarán presentes a lo largo de este trabajo de diploma.

1.2 Conceptos fundamentales

1.2.1 Calidad

La calidad es el término que más preocupa hoy en día a las empresas desarrolladoras de productos software, debido a la exigencia de los clientes para que se realicen aplicaciones cada vez más complejas en menor tiempo. Debido a esta situación la calidad se ha vuelto un reto para cualquier equipo de desarrollo y un factor determinante para alcanzar el éxito en esta industria, pues es fundamental para lograr la satisfacción de las necesidades y expectativas del cliente.

A continuación se exponen algunas definiciones de Calidad que sirven de preámbulo al hablar de Calidad de Software.

La calidad es la medida en que las propiedades de un bien o servicio cumplen con los requisitos establecidos en la norma o especificaciones técnicas, así como con las exigencias del usuario de dicho bien o servicio en cuanto a su funcionalidad, durabilidad y costo. [1]

El Diccionario de la Real Academia Española, conceptúa la Calidad como la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor, y es sinónimo de buena calidad la superioridad o excelencia. [2]

Por su parte, la Organización Internacional para la Estandarización del inglés International Organization for Standardization (ISO), agrega otros aspectos importantes a tener en cuenta cuando define la calidad como el conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer las necesidades explícitas o implícitas. [3]

De los conceptos de Calidad anteriormente citados se consideró el más abarcador el propuesto por la ISO ya que de manera general comprende el conjunto de parámetros que debe cumplir un software para alcanzar la calidad necesaria.

1.2.2 Calidad de Software

Uno de los problemas que se afrontan actualmente en la esfera de la computación es la Calidad del Software. Este tema ha sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software, los cuales han realizado gran cantidad de investigaciones al respecto con dos objetivos fundamentales:

1. ¿Cómo obtener un software con calidad?
2. ¿Cómo evaluar la calidad del software?

Ambas interrogantes conllevan amplias respuestas, pero están estrechamente ligadas con el concepto de la calidad del software.

La Calidad del Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La Calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. [4]

Pressman agrupa todos los aspectos que antes se mencionan para referirse a la Calidad del Software en un concepto único y realmente convincente que 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 espera de todo el software desarrollado profesionalmente”. [5]

De ahí la importancia que tiene:

- Mantener los productos bajo un constante control.
- Desarrollar mediciones que den como resultado sistemas de alta calidad.
- Contar con métodos y herramientas efectivas que evalúan la calidad con objetividad, no con subjetividad.

Para la ejecución de los tres aspectos anteriores los encargados del aseguramiento de la calidad, en un proyecto, deben basarse en medidas que garanticen la calidad de los productos, entre las que se pueden citar: [5]

- Corrección: La corrección es el grado en el que el software lleva a cabo su función requerida. Un programa debe operar correctamente o proporcionará poco valor a sus usuarios.

- **Facilidad de mantenimiento:** La facilidad de mantenimiento es la facilidad con la que se puede corregir un programa si se encuentra un error, se puede adaptar si su entorno cambia, o mejorar si el cliente desea un cambio de requisitos.
- **Integridad:** Este atributo mide la capacidad de un sistema para resistir ataques (tanto accidentales como intencionados) contra su seguridad. El ataque se puede realizar en cualquiera de los tres componentes del software: programas, datos y documentos.
- **Facilidad de uso:** La facilidad de uso de un sistema se basa en lo amigable que puede ser este con el usuario.
- **Eficacia de la Eliminación de Defectos (EED):** La EED es una medida de la habilidad de filtrar las actividades de la garantía de calidad y de control al aplicarse a todas las actividades del marco de trabajo del proceso.

Para esta investigación se considera que Calidad de Software, resume las características y los atributos propios de un producto. Determinando, sobre este, la no existencia de defectos y la conformidad de todo el personal que de una forma u otra se vinculan con él.

1.2.3 Control y Evaluación de la Calidad del Software

El control de calidad son técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centrados en dos objetivos fundamentales: [6]

- mantener bajo control un proceso
- eliminar las causas de los defectos en las diferentes fases del ciclo de vida

Para obtener un software con calidad es necesaria la utilización de metodologías o estándares que permitan equilibrar los procedimientos de trabajo, a favor de lograr una mayor confiabilidad, mantenibilidad y usabilidad, al mismo tiempo que eleven el rendimiento, tanto para el desarrollo del producto como para el control de la Calidad del Software.

La Calidad de un Software se debe controlar y evaluar para garantizar que los desarrolladores elaboren un producto que cumpla con las especificaciones del cliente. Para controlar la calidad es necesario evaluarla y esto implica una comparación entre los requisitos anteriormente establecidos y el producto finalmente desarrollado. [7]

Los ingenieros de software aplican mediciones e indicadores que le permitan evaluar cuantitativamente la calidad de las diferentes actividades de los Flujos de Trabajo que se han establecido al aplicar la ingeniería del software. Para obtener esta evaluación de calidad, el ingeniero debe utilizar medidas técnicas, que evalúan la calidad con objetividad, no con subjetividad. [7]

El software posee determinados índices medibles que son las bases para la calidad, el control y el perfeccionamiento de la productividad. Una vez seleccionados los índices de calidad, se debe establecer el proceso de control, que requiere los siguientes pasos: [8]

- Definir el software que va a ser controlado: clasificación por tipo, esfera de aplicación, complejidad, de acuerdo con los estándares establecidos para el desarrollo del software.
- Seleccionar una medida que pueda ser aplicada al objeto de control. Para cada clase de software es necesario definir los indicadores y sus magnitudes.
- Crear o determinar los métodos de valoración de los indicadores: métodos manuales como cuestionarios o encuestas estándares para la medición de criterios periciales y herramientas automatizadas para medir los criterios de cálculo.
- Definir las regulaciones organizativas para realizar el control: ¿quiénes participan en el control de la calidad?, ¿cuándo se realiza?, ¿qué documentos deben ser revisados y elaborados?

1.2.4 Métricas de Software

En la actualidad se ha hecho imprescindible el uso de las mediciones como parte de las actividades cotidianas de todas las organizaciones de software, pues estas brindan la información objetiva necesaria para la toma de decisiones, teniendo un impacto efectivo en el negocio y desempeño en la ingeniería.

Se comenzó por definir los posibles términos que se encuentran encerrados en la palabra métrica, porque es muy común asociarla con las palabras medición y medida, aunque estas tres son distintas. 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. Una medida proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto y la métrica definida como una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.

Las métricas proporcionan una indicación de la efectividad de las actividades de control y de la garantía de calidad en grupos o en particulares. Las mismas tienen dos objetivos bien definidos, primero establecer un conjunto abierto de procedimientos e indicadores adecuados de calidad y segundo definir métodos para medir los indicadores de calidad del software, este contexto es verdaderamente significativo, que las características de calidad puedan ser cuantificables y medibles. [5]

Las métricas del software se refieren a un amplio elenco de mediciones para el software de computadora. La medición se puede aplicar al proceso del software con el intento de mejorarlo sobre una base continua. Se puede utilizar en el proyecto del software para ayudar en la estimación, el control de calidad, la evaluación de la productividad y el control de proyectos. Finalmente, el ingeniero de software puede utilizar la medición para ayudar a evaluar la calidad de los resultados de trabajos técnicos y para ayudar en la toma de decisiones tácticas a medida que el proyecto evoluciona. [5]

Métricas de software no es más que la aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos. Las métricas de software proveen la información necesaria para la toma de decisiones técnicas. [9]

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:

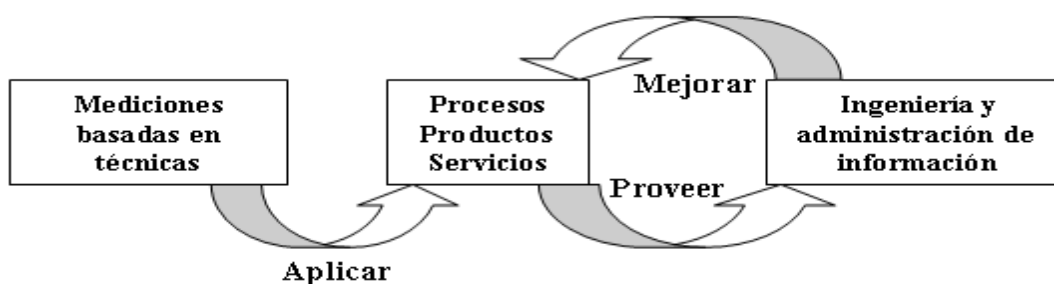


Figura 1 Definición de Métricas de Software

Las métricas del software pueden ser utilizadas para que los profesionales e investigadores tomen las mejores decisiones. Proporcionan información objetiva que contribuye al mejoramiento de los procesos

y productos de software lo cual, evidentemente, favorece al logro de la calidad, siempre que se haga un uso adecuado de las mismas. [5]

Las métricas son la maduración de una disciplina, que, según Pressman van a ayudar a la (1) evaluación de los modelos de análisis y de diseño, (2) en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente, y (3) ayudarán en el diseño de pruebas más efectivas. Es por eso que propone un proceso de medición, el cual se puede caracterizar por cinco actividades:

- **Formulación:** La obtención de medidas y métricas apropiadas para la representación del 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.

Las métricas de software deben medir el proceso, el proyecto y el producto partiendo del hecho de que: [10]

- proceso: es la secuencia o las actividades invocadas para producir el producto de software (y otros artefactos).
- producto: son los artefactos del proceso, incluyendo el software, los documentos y modelos.
- proyecto: son todos los recursos del proyecto, actividades y artefactos.

Los principios fundamentales que deben seguir las métricas son [10]:

- Las métricas deben ser simples, objetivas, fáciles de coleccionar, fáciles de interpretar y difíciles de malinterpretar.

- La colección de las métricas debe ser automática o sea, no interferir en las actividades de los desarrolladores.
- Las métricas deben contribuir a la evaluación de la calidad temprana en el ciclo de vida, cuando los esfuerzos por mejorar la calidad del software son efectivos.
- Los valores absolutos y las tendencias de las métricas, deben ser usados activamente por el personal administrativo y el personal de ingeniería, para comunicar progreso y calidad en un formato coherente.
- La selección de un mínimo o más extensivo conjunto de métricas, dependerá de las características y contexto del proyecto: Si es muy grande o si tiene restricciones de seguridad o de confiabilidad de los requerimientos; y si el equipo de desarrollo y de valoración (evaluación) es conocedor de las métricas, lo cual hará muy útil coleccionar y analizar las métricas técnicas.

En resumen, una métrica de software es el término que describe muchos y muy variados casos de medición. Siendo una medida estadística que se aplica a todos los aspectos de Calidad de Software, los cuales deben ser medidos desde diferentes puntos de vista como el análisis, construcción, funcionalidad, documentación, métodos, proceso, usuario, entre otros.

1.2.4.1 Características de las Métricas de Software

Existen numerosas métricas propuestas para el empleo en los proyectos, aunque no todas proporcionan un soporte práctico para los desarrolladores. Muchas exigen mediciones que son complejas y otras

Muchas han sido las métricas propuestas para el software, pero no todas proporcionan un soporte práctico para el desarrollador. Algunas demandan mediciones que son demasiado complejas y otras violan las nociones básicas intuitivas de lo que realmente es el software de alta calidad.

Las características de las métricas de software se resumen en: [5]

- **Simple y fáciles 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.
- **Empíricas e intuitivas:** la métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión (por ejemplo: una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).

- **Consistentes 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 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 deberían depender de los caprichos de la sintaxis o semántica del lenguaje de programación.
- **Eficaz para aumentar la calidad del software:** la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.

Resumiendo estas características se puede decir que es importante que las métricas sean directas de manera que su empleo no obstaculice el trabajo del desarrollador. Los cálculos deben ser simples y a la vez objetivos y su empleo no debe ser específicamente para un lenguaje de programación.

1.2.4.2 Clasificación de las Métricas de Software

Existen innumerables métricas con propósitos diferentes que reflejan o describen la conducta del software, estas pueden medir entre otros aspectos la competencia, calidad, desempeño y la complejidad del software contribuyendo a establecer de una manera sistemática y objetiva una visión interna del trabajo mejorando así la calidad del producto.

Las Métricas de Software han sido clasificadas de innumerables formas por diferentes personalidades que han estudiado este tema:

Pressman clasifica el campo de las métricas en seis categorías o grupos de métricas distintos: [5]

- **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 qué tan productivo va a ser el software que voy a diseñar.
- **Métricas orientadas al tamaño.** Es para saber en qué tiempo voy a terminar el software y cuántas 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 aplicaciones 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.

Otras formas de clasificaciones de Métricas de Software son:

- **Directas o Indirectas [11]**
 - Directa: Una métrica de la cual se pueden realizar mediciones sin depender de ninguna otra métrica y cuya forma de medir es un método de medición.
 - Indirecta: Una métrica cuya forma de medir es una función de cálculo, es decir, las mediciones de dicha métrica utilizan las medidas obtenidas en mediciones de otras métricas directas o indirectas.
- **Internas o Externas [12]**
 - Interna: Puede ser aplicada a un producto de software no ejecutable (como una especificación o código fuente) durante el diseño y la codificación.
 - Externa: Usa medidas de un producto de software, derivadas del comportamiento del mismo, a través de la prueba, operación y observación del software.
- **De Proyecto, del Producto o de Procesos**

- Producto: Los productos son artefactos que pueden ser documentos, componentes, modelos, diagramas, módulos, a los cuales se les aplican métricas para obtener mediciones de cada uno de estos productos. Las métricas del producto describen características como el tamaño, complejidad, rasgos del diseño, rendimiento y nivel de calidad. [13]
- Proceso: Las Métricas del Proceso se recopilan de todos los proyectos y durante un largo período de tiempo. Su intento es proporcionar indicadores que lleven a mejoras de los procesos de software a largo plazo. Se utilizan para evaluar la eficiencia de un proceso o si este ha mejorado con los cambios realizados. [14]
- Proyecto: Las Métricas del Proyecto se utilizan para minimizar la planificación de desarrollo haciendo los ajustes necesarios para evitar retrasos o riesgos potenciales, minimizar los defectos, y por tanto la cantidad de trabajo que ha de rehacerse, lo que ocasiona una reducción del coste global del proyecto. Son manejadas para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificando el enfoque técnico que mejore la calidad. [5]

Dado que el proyecto engloba todos los recursos, actividades y artefactos, que se organizan para lograr un producto de software es de vital importancia definir algunas mediciones que ayuden al mejoramiento del mismo. La propuesta a realizar en el presente trabajo de diploma contará con métricas destinadas a evaluar la calidad en el proyecto del Polo de Telecomunicaciones por lo que se incluirán mediciones para aspectos relacionados con el proceso y el producto, evaluando de esta forma también la calidad de los mismos.

1.3 Área de Aplicación

Debido a la gran importancia que posee en el mundo, y en el país la rama de las Telecomunicaciones y en particular la relacionada con la Informática, se ha decidido enfocar la presente investigación hacia los proyectos del Polo de Telecomunicaciones para lograr que los productos que salen de esta facultad cuenten con la mayor calidad posible. Estos proyectos que se realizan actualmente, no presentan características diferentes a los proyectos de gestión que se fabrican en el resto de la Universidad, por lo cual, es de vital importancia el conocimiento de las características que presentan dichas aplicaciones, las cuales servirán como guía para la propuesta de solución del presente trabajo de diploma.

Entre las características que presentan los proyectos que se producen en el Polo de Telecomunicaciones se encuentran:

- Presentan gran tamaño al poseer decenas y hasta centenares líneas de código fuente. Trayendo como consecuencia que el diseño tenga que ser abarcado por un numeroso grupo de personas.
- Poseen complejidad al estar constituidos por un número elevado de módulos y exhibir algún grado de concurrencia. Trayendo como consecuencia necesidades técnicas de análisis y especificación formal en el proceso de descomposición modular, y a su vez que surjan necesidades como el manejo de diferentes niveles de abstracción, jerarquización con el objetivo de disminuir la complejidad observable, independencia entre los módulos y la reutilización, siendo esta el punto de partida de un desarrollo existente.
- Se caracterizan además por poseer un gran tiempo de explotación generalmente entre 15 a 25 años, lo que les garantiza una larga vida. Lo que trae como consecuencia la necesidad de controlar la evolución de los mismos y que a su vez surjan necesidades como una buena mantenibilidad, modularidad y documentación y además, la necesidad de que el entorno cambie durante períodos de tiempo debiendo el sistema adaptarse a dichos cambios.
- Manejo de un gran volumen de datos.

1.4 Técnicas de Validación de Resultados

La validación es el acto documentado de probar que cualquier procedimiento, proceso, equipo, material, actividad o sistema conduce realmente al resultado esperado. Es el mecanismo o sistema usado por el fabricante para planear, obtener datos, registrar datos, e interpretar datos. Está definida también como la evidencia documentada que provee un alto grado de seguridad de que un proceso específico o un equipo producirán en forma homogénea y reproducible un producto que cumplirá con especificaciones predeterminadas y sus atributos de calidad.

Existen en el mundo diversos Métodos de Validación, que se basan en procedimientos y estándares diferentes. A continuación se explican brevemente algunos de los que se estudiaron para utilizar en este Trabajo de Diploma.

1.4.1 Método DELPHI

El Método Delphi se utiliza cuando no se tienen datos o se dispone de muy pocos, acerca del sistema que se está considerando. En este método, se selecciona un grupo de expertos los cuales deben llegar a un consenso en las respuestas que den acerca de una serie de preguntas que se les plantean. En un entorno de simulación los expertos pueden ser los administradores y usuarios del sistema y las cuestiones son acerca del comportamiento del sistema bajo ciertas condiciones de operación. El Método Delphi excluye las discusiones cara a cara entre los miembros del grupo. [15]

En este método se les plantea al grupo una serie de cuestiones:

- Se envía un cuestionario a cada miembro del grupo. Para la validación de un modelo de simulación, las cuestiones podrían tratar sobre las respuestas del sistema real ante ciertas entradas o cambios en su estructura.
- Basándose en las respuestas dadas a las cuestiones planteadas, se elaboran nuevos cuestionarios que van centrándose en temas más específicos.
- Los nuevos cuestionarios se envían al grupo junto con las respuestas obtenidas a las cuestiones en rondas anteriores.

Estos tres pasos se repiten hasta que el analista consiga de los expertos una predicción de la respuesta de sistema.

Una crítica de este método es que consume mucho tiempo. No necesariamente debe suponer un tiempo adicional en el proyecto de simulación, ya que se puede ir realizando paralelamente al desarrollo del modelo. Su costo puede resultar caro, pero no mucho más que otros métodos de validación. En general, aunque se critique que los métodos de validación son caros y consumen tiempo, más costoso es construir un modelo que no sea válido.

Otra crítica que se hace al método es que si el grupo de expertos lo que va a hacer es predecir el comportamiento del sistema ¿por qué no usar el método en lugar del modelo de simulación? En algunas situaciones este método puede ser utilizado, sin embargo, normalmente no es práctico mantener el grupo de expertos para predecir el comportamiento del sistema ante los posibles cambios que se planteen. Incluso si se pudiera mantener el tiempo de respuesta podría ser muy largo.

1.4.2 Método Multicriterios

Los Métodos de Evaluación y Decisión Multicriterios comprenden la selección entre un conjunto de alternativas factibles, la optimización con varias funciones objetivos simultáneamente, un agente decisor y procedimientos de evaluación racionales y consistentes. [16]

Los principios de estos métodos se derivan de la Teoría de Matrices, Teoría de Grafos, Teorías de las Organizaciones, Teoría de la Media, Investigación de Operaciones, Teoría de las Decisiones Colectivas entre otras.

En la Decisión Multicriterio un elemento clasificador es el número de alternativas a tener en cuenta en la decisión, que puede ser finito o infinito. Dependiendo de esta situación existen diferentes métodos. Cuando las alternativas tienen un número infinito de valores posibles del problema se llama **Decisión Multiobjetivo**. Por el contrario cuando el número de alternativas es finito se denominan **Decisión Multicriterio Discreta**. Estos problemas son más comunes en la realidad y se utilizan para realizar una evaluación y decisión respecto a problemas que por naturaleza o diseño, admiten un número finito de alternativas de solución a través de:

1. Una familia de criterios de evaluación (atributos, objetivos) que permiten evaluar cada una de las alternativas (analizar sus consecuencias), conforme a los pesos (o ponderaciones) asignadas por el agente decisor y que reflejan la importancia (preferida) relativa de cada criterio.
2. Un conjunto de alternativas estables, generalmente finito (soluciones factibles que cumplen con las restricciones posibles o previsibles), se asume que cada una de ellas es perfectamente identificada, aunque no son necesariamente conocidas en forma exacta y completa de todas sus consecuencias cuantitativas y cualitativas.
3. Una Matriz de decisión o de impactos que resume la evaluación de cada alternativa conforme a cada criterio, una valoración (precisa o subjetiva) de cada una de las soluciones a la luz de cada uno de los criterios, la escala de medida de las evaluaciones puede ser cuantitativa o cualitativa, y los medios pueden expresarse en escala cardinal (razón o intervalo), ordinal, nominal y probabilísticas.
4. Una metodología o modelo de agregación de preferencias en una síntesis global, ordenación, clasificación, participación, jerarquización de dichos juicios para determinar la solución que globalmente recibe las mejores evaluaciones.

5. Un proceso de toma de decisiones (contexto de análisis) en el cual se lleva a cabo una negociación consensual entre los actores o interesados (analista-“experto”-, decisor y usuario).

1.4.3 Test de Turing

Alan Turing sugirió este método como un test de inteligencia artificial. En este test, a un experto, o grupo de expertos, se le presentan resúmenes o informes de resultados de ejecución del sistema y del modelo, a los que se les ha dado el mismo formato. Estos informes se reparten aleatoriamente a los ingenieros y administradores del sistema, para ver si son capaces de discernir cuáles son los reales del sistema y cuáles la imitación resultado de la simulación. Si los expertos no son capaces de distinguir entre ambos, se puede concluir que no hay evidencias para considerar inadecuado al modelo. Si descubren diferencias las respuestas sobre lo que encuentran inconsistente se puede utilizar para realizar mejoras en el modelo. [15]

Se puede considerar que este método es el inverso al Método de Delphi. En el Test de Turing se consulta a los expertos para ver si son capaces de identificar las respuestas del sistema, mientras que en el de Delphi se pregunta a los expertos para que predigan las respuestas del sistema.

Aunque este test parece muy intuitivo, hay muy pocos informes de su uso, ya que requiere un esfuerzo considerable para formatear las medidas de ejecución del sistema a la hora de crear el informe que se da a los expertos. Otra dificultad está en ajustar las medias del sistema real ya que en ellas intervienen elementos que no se han considerado en el modelo. Por último este test requiere un análisis estadístico por parte del grupo de expertos para determinar si hay diferencias significativas entre el informe real y el simulado.

1.4.4 Método de Validación Interna y Externa

La Validación Interna es un ejercicio teórico que asegura que la métrica tiene una caracterización numérica apropiada de la propiedad que pretende medir. Esto es por supuesto un pre-requisito para demostrar la utilidad de dicha métrica (validación empírica).

La Validación Externa se lleva a cabo para demostrar con evidencia real que una métrica es útil en el sentido de que está asociada con alguna característica externa del software tal como la usabilidad, la mantenibilidad, etc.[17]

El proceso que se realiza para el empleo de estos métodos de validaciones consta de tres pasos:

- Primeramente el análisis de la veracidad de las métricas a emplear.

- La recolección de los datos reales de la muestra en cuestión.
- El cálculo de las métricas con los valores obtenidos del proceso de recolección y la realización de la valoración correspondiente a los resultados obtenidos.

De esta manera se obtendrá una idea real de la situación que presente la muestra en cuestión en cuanto a los parámetros que se pretenden medir en cada una de las métricas a emplear para el desarrollo del proceso de validación.

1.4.5 Selección del Método de Validación

Luego de realizar un análisis de toda la información anteriormente planteada, se concluyó que el método Delphi se utiliza cuando no se tienen datos del sistema donde se propone realizar la validación o se dispone de muy pocos, es por ello que no se considera su selección debido a que en el caso de la propuesta realizada sí se cuenta con datos suficientes para realizar la validación. Este método requiere gran cantidad de tiempo para su realización, sucediendo lo mismo con el método Multicriterios el cual tampoco es considerado para su empleo, debido a este factor tiempo. El método de Test de Turing no se empleó debido a la dificultad que presentan todos los pasos a realizar y el tiempo que se debe emplear es bastante largo. Además, por no contar con suficientes expertos que en este caso debían tener un amplio conocimiento del contenido de la propuesta y a la vez dominar términos estadísticos para realizar comparaciones, iteraciones, etc.

Se decidió emplear los métodos de Validación Interna y Externa ya que estos permiten demostrar la utilidad de la propuesta realizada con datos reales obtenidos de una fuente perteneciente a los proyectos hacia donde se encuentra enfocada la propuesta. Este método permite que a partir del valor matemático que se obtiene de la métrica se lleguen a conclusiones y soluciones en el proyecto mediante comparaciones que determinen los factores de riesgos. El empleo de este método no requiere gran cantidad de tiempo y además se adecúa a los objetivos propuestos.

1.5 Conclusiones del Capítulo

En este capítulo se fundamentaron diferentes temas que son importantes para el buen entendimiento de la investigación. Se abordó el tema de la calidad del software y el control y evaluación de la misma. Se realizó un estudio de los conceptos teóricos relacionados con las métricas del software, características y sus diferentes clasificaciones, destacando la importancia que tiene la aplicación de métricas en la evaluación de la calidad del software.

Capítulo 2: Descripción de la Propuesta de Solución

2.1 Introducción

En el presente capítulo se describe la Propuesta de Métricas para evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones. Esta propuesta se basa en un conjunto de métricas seleccionadas para aplicarlas en los proyectos, específicamente orientadas a lograr una mejor calidad en los productos, así como, el estado en que se encuentra el mismo permitiendo a su vez evaluar el proceso. Estas métricas ayudarán a predecir, evaluar y caracterizar los productos que se obtienen en el Polo de Telecomunicaciones y sobre todo contribuirán a mejorar la calidad de los mismos.

2.2 Resultados Obtenidos de la Entrevista

Por lo amplio que es el campo de las Métricas de Calidad de Software se realizó una entrevista a los Líderes de proyectos y a los Responsables de Calidad de los mismos. Se entrevistó además a varios compañeros miembros de la Dirección de Calidad de Software (DCS) de la UCI.

La entrevista constó de 3 preguntas previamente elaboradas, en la primera se aborda acerca del conocimiento existente de las Métricas de Calidad, así como, si las emplean en el proyecto al cual pertenecen. En la segunda y tercera pregunta se hace alusión a cuáles métricas se aplican y el por qué de su selección, logrando de esta manera conocer la importancia que le conciernen al empleo de las Métricas de Calidad de Software en los proyectos a los que pertenecen.

Las entrevistas realizadas a los diferentes Líderes de Proyectos del Polo de Telecomunicaciones arrojaron como resultado que no existe un estándar de métricas definido para evaluar la calidad a los Proyectos Productivos. No planean ni registran el trabajo realizado, así como, tampoco miden y administran la calidad de los productos. En la minoría de los proyectos de la facultad los responsables de la calidad aplican algunas métricas que ellos consideran necesarias y eficientes, y emplean las Pruebas de Caja Blanca, Caja Negra y Pruebas de Aceptación, pero nunca se ha establecido un procedimiento estándar para todos los proyectos por igual.

Otra entrevista realizada a la Ing. Irina Napal Torres, Jefe del Departamento de Métricas del DCS de la UCI, arrojó como resultado que en la Universidad no existe un Plan de Métricas definido para aplicar a los proyectos, es por ello que la DCS se encuentra en el desarrollo de una investigación donde se obtendrá como resultado un plan a nivel de Universidad de métricas posibles a aplicar en todos los proyectos, aunque se mantiene la propuesta de que cada Líder de Proyecto o Responsable de Calidad de cada uno de ellos puedan utilizar además de las establecidas, otras que crean que son

indispensable en su uso. Este departamento ha realizado entrevistas a los diferentes proyectos de la UCI incluyendo los que pertenecen a la Facultad 2 hacia la cual se encuentra dirigida la propuesta de este Trabajo de Diploma. Como resultado de estas entrevistas se obtuvo que la mayoría de los desarrolladores de software en la UCI todavía no miden, a pesar de que las mediciones pueden ayudarlos a lograr que su trabajo cada día sea mejor y con una mayor calidad. El desconocimiento de las métricas por parte del equipo de trabajo hace que establezcan un rechazo hacia ellas.

2.3 Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones

Teniendo en cuenta todo lo anteriormente planteado, se procederá a realizar la Propuesta de Métricas para Evaluar la Calidad en los Proyectos del Polo de Telecomunicaciones, que se espera uniformen el trabajo que realizan los responsables de calidad de los proyectos. La selección tuvo como punto de partida principalmente las características de los Proyectos de Telecomunicaciones realizados en la Facultad 2, así como, los resultados obtenidos tanto en las entrevistas realizadas como en la investigación en general. Se concluyó que la mayoría de los proyectos se encuentran divididos en módulos para realizar el trabajo, por lo que se cree necesario y de gran utilidad realizar el cálculo de ciertas métricas que permiten analizar cada uno de estos módulos por separado. La propuesta se encuentra compuesta por Métricas de Proyecto.

2.3.1 Métricas de Proyecto

2.3.1.1 Métrica para la Eficacia de la Eliminación de Defectos

La Eficacia de la Eliminación de Defectos (EED) es una métrica de la calidad que proporciona beneficios tanto a nivel del proyecto como del proceso. Debido al empleo de pruebas de aceptación en los proyectos del Polo de Telecomunicaciones para evaluar la calidad, esta métrica permitirá disminuir los defectos posibles a detectar por el cliente final en los procesos de despliegue. En esencia, es una medida de la habilidad de filtrar las actividades de la garantía de calidad y de control al aplicarse a todas las actividades del marco de trabajo del proceso. Cuando un proyecto se toma en consideración globalmente, EED se define de la forma siguiente:

$$EED= E/ (E+D)$$

Donde:

- E: es el número de errores encontrados antes de la entrega del software al usuario final.

- D: es el número de defectos encontrados después de la entrega.

El valor ideal de EED es 1 que simboliza que no se han encontrado defectos en el software. De forma realista, D será mayor que 0, pero el valor de EED todavía se puede aproximar a 1. Cuando E aumenta (para un valor de D dado), el valor total de EED empieza a aproximarse a 1. De hecho, a medida que E aumenta, es probable que el valor final de D disminuya (los errores se filtran antes de que se conviertan en defectos). Si se utilizan como una métrica que proporciona un indicador de la habilidad de filtrar las actividades de la garantía de la calidad y del control, EED anima a que el equipo del proyecto instituya técnicas para encontrar todos los errores posibles antes de su entrega.

Del mismo modo EED también se puede utilizar dentro del proyecto para evaluar la habilidad de un equipo de encontrar errores antes de que pasen a la siguiente actividad estructural o tarea de ingeniería del software. Por ejemplo, la tarea del análisis de los requisitos produce un modelo de análisis que se puede revisar para encontrar y corregir errores. Esos errores que no se encuentren durante la revisión del modelo de análisis se pasan a la tarea de diseño (en donde se pueden encontrar o no). Cuando se utilizan en este contexto, EED se vuelve a definir como: [5].

$$EED_i = E_i / (E_i + E_{i+1})$$

Donde:

- E_i : es el número de errores encontrado durante la actividad de Ingeniería de Software i .
- E_{i+1} : es el número de errores encontrado durante la actividad de Ingeniería de Software $i+1$ que se puede seguir para llegar a errores que no se detectaron en la actividad i .

Un objetivo de calidad de un equipo de Ingeniería de Software es alcanzar un EED que se aproxime a 1, que significa que los errores se deberían filtrar antes de pasar a la actividad siguiente. Esto también puede ser utilizado dentro del proyecto para evaluar la habilidad de un equipo, con el objetivo de encontrar deficiencias que harán que se atrase el proyecto.

Esta métrica se aplica antes de cada una de las versiones desplegadas con el cliente final y el responsable es el Evaluador de Calidad.

Tabla 1 Escala de evaluación del resultado de la Métrica Eficacia de la Eliminación de Defectos

Eficacia de la eliminación de Defectos	Valores	Evaluación
0 < EED ≤ 1 Valor Ideal 0.75	0 ≤ EED < 0.5	Mal
	0.5 ≤ EED < 0.75	Regular
	0.75 ≤ EED ≤ 1	Bien

2.3.1.2 Métrica de Calidad de la Especificación de Requisitos

La Calidad de Especificación de Requisitos (CER) definida por Pressman, proporciona una indicación de la especificidad o el grado en que se ha completado la especificación de los requisitos. Permite al jefe de proyecto y la alta dirección, evaluar el estado de un producto que se está realizando o se terminó, así como, seguir la pista de los riesgos potenciales de la no satisfacción de los requisitos acordados con el cliente. Por la importancia que posee la fase de levantamiento de requisitos, esta métrica permitirá que este proceso se realice con la calidad y las especificaciones requeridas. Esta métrica está referida a la capacidad de entender el significado de los requisitos, o sea que no exista ambigüedad, que cada requisito tenga una sola interpretación. [5]. Es aplicada en la Fase de Inicio en el Flujo de Requerimientos por el Evaluador de Calidad.

Fórmula para calcular la cantidad de requerimientos:

$$N_r = N_f + N_{nf}$$

Donde:

- N_r = Número de requerimientos.
- N_f = Número de requerimientos funcionales.
- N_{nf} = Número de requerimientos no funcionales.

Para determinar la especificidad de los requisitos, se sugiere una métrica basada en la consistencia de la interpretación de los revisores para cada requisito: [5]

Esta métrica tiene como propósito obtener el porcentaje de requerimientos que han sido igualmente interpretados por los revisores.

$$Q = N_{ui} / N_r$$

Donde:

- Q= Especificidad
- Nui= Número de requerimientos donde todos los revisores tuvieron interpretaciones idénticas.

$$0 \leq Q \leq 1$$

Cuanto más cerca de 1 esté el valor de Q menor será la ambigüedad de la especificación.

Tabla 2 Escala de evaluación de los resultados de la Métrica Calidad de Especificación de Requisitos.

Calidad de Especificación de Requisitos	Valores	Evaluación
$0 \leq Q \leq 1$ Valor Ideal 0.75	$0 \leq Q \leq 0.5$	Mal
	$0.5 \leq Q \leq 0.75$	Regular
	$0.75 \leq Q \leq 1$	Bien

2.3.1.3 Métrica para el Control de los Casos de Uso Diseñados

En el Flujo de Análisis y Diseño deben quedar diseñados cada uno de los Casos de Uso (CU) que se definieron en el proyecto, cumpliendo con las funcionalidades descritas en el proceso de Gestión de Requisitos.

La Métrica para el Control de los Casos de Uso Diseñados (CCUD) definida en el trabajo de diploma Métricas para la evaluación del proceso de desarrollo de Software Educativo proporciona información sobre la cantidad de CU que se diseñaron con respecto a los que se precisaron para el proyecto. [18] Es aplicada en la Fase Inicio. En el Flujo de Requerimiento se encuentran los CU definidos para el proyecto y en el Flujo de Análisis y Diseño se encuentran los CU diseñados. El responsable de la aplicación de esta métrica es el Evaluador de Calidad.

Se calcula como:

$$CCUD = CUD / CUDef$$

Donde:

- CUD: número de CU que fueron diseñados.

- CUDef: número de CU que fueron definidos.
- CCUD: control de los casos de uso diseñados.

El único valor para el que la métrica CCUD evalúa el proceso de Bien es 1. Mientras más cerca se encuentre de 1 en resultado, significa que se han diseñado la mayor cantidad de casos de uso definidos en el proyecto.

Tabla 3 Escala de evaluación del resultado de la Métrica para el Control de los Casos de Uso diseñados.

Control de Casos de Uso Diseñados	Valores	Evaluación
$0 \leq CCUD \leq 1$	$0 \leq CCUD \leq 0.75$	Mal
Valor Ideal 1	$0.75 \leq CCUD \leq 1$	Regular
	$CCUD = 1$	Bien

2.3.1.4 Métrica para la Agilidad en la Gestión de Cambios de los Requisitos

Dentro del proceso de Gestión de Requisitos se realizan actividades relacionadas con la gestión de cambio, que permiten resolver todos los problemas que ocasionan los cambios en los requisitos. Estas actividades se deben realizar con la mayor agilidad posible para no afectar el tiempo planificado para el proceso en general.

La Agilidad en la Gestión de Cambio de los Requisitos (AGCR) definida en el trabajo de diploma Métricas para la evaluación del proceso de desarrollo de Software Educativo proporciona información sobre la relación entre el tiempo dedicado a la a la Gestión de Cambio de los Requisitos y el utilizado en el proceso en general. [18]. Es aplicada en la Fase de Inicio en el Flujo de por el Evaluador de Calidad.

Se puede calcular AGCR:

$$AGCR = 1 - (TGCR / TPGR)$$

Donde:

- TGCR: tiempo dedicado a la Gestión de Cambio de los Requisitos.
- TPGR: tiempo total dedicado al proceso de Gestión de Requisitos.

Si el valor obtenido al calcular AGCR se acerca a 1, la Gestión de Cambio de los Requisitos se habrá realizado de la forma más rápida posible.

Tabla 4 Escala de evaluación del resultado de la Métrica para la Agilidad en la Gestión de Cambio de los Requisitos.

Agilidad en la Gestión de Cambios en los Requisitos	Valores	Evaluación
$0 \leq AGCR \leq 1$ Valor Ideal 0.75	$0 \leq AGCR \leq 0.5$	Mal
	$0.5 \leq AGCR \leq 0.75$	Regular
	$0.75 \leq AGCR \leq 1$	Bien

2.3.1.5 Métrica de Mantenimiento

El Mantenimiento del software cuenta con más esfuerzo que cualquier otra actividad de Ingeniería de Software. En la mayoría de las aplicaciones surgen cambios durante el proceso de desarrollo de las mismas por lo que se ha propuesto la siguiente métrica que está diseñada explícitamente para actividades de mantenimiento enfocada a estos cambios. El estándar IEEE 982.1-1988 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto de software basada en los cambios que ocurren con cada versión del producto. Se determina la siguiente información: [5]. Esta métrica se aplica en cada una de las fases por las que atraviesa el producto y el responsable es el Evaluador de Calidad y el Líder de Proyecto.

- M_r = Número de módulos en la versión actual.
- F_c = Número de módulos en la versión actual que se han cambiado.
- F_a = Número de módulos en la versión actual que se han añadido.
- F_d = Número de módulos de la versión anterior que se han borrado en la versión actual.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = [M_r - (F_a + F_c + F_d)] / M_r$$

A medida que el IMS se aproxima a 1 el producto se empieza a estabilizar. El IMS puede emplearse también como métrica para la planificación de las actividades de mantenimiento del software. El tiempo medio para producir una versión de un producto software puede correlacionarse con el IMS desarrollándose modelos empíricos para el mantenimiento. [5]

Tabla 5 Escala de evaluación del resultado de la Métrica de Mantenimiento.

Índice de madurez del software	Valores	Evaluación
$0 \leq \text{IMS} \leq 1$ Valor Ideal 0.75	$0 \leq \text{IMS} \leq 0.5$	Mal
	$0.5 \leq \text{IMS} \leq 0.75$	Regular
	$0.75 \leq \text{IMS} \leq 1$	Bien

2.3.1.6 Métrica de Esfuerzo de Módulo

Debido a la alta complejidad que poseen los proyectos del Polo de Telecomunicaciones al estar constituidos por un elevado número de módulos es necesario el cálculo del esfuerzo realizado en cada uno de ellos, por lo que la métrica Esfuerzo de Módulo definida en el Trabajo de Diploma Propuesta de un Estándar y Métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela, permitirá conocer el esfuerzo de las personas asignadas para realizar cierta tarea, constituyendo una base del conocimiento para valorar el proceso de asignación de tareas en el módulo. Teniendo como propósito calcular el esfuerzo de los módulos para poder obtener el esfuerzo medio del proyecto. Esta métrica se aplica en cada una de las fases de desarrollo y el responsable es el Evaluador de Calidad y el Líder de Proyecto. [17]

$$E_m = P / t$$

Donde:

- P: cantidad de personas asignadas a realizar la tarea.
- t: tiempo planificado para el cumplimiento de la tarea.
- E_m (esfuerzo del módulo que se obtiene en personas-semanas).

Esta métrica tiene como propósito calcular el esfuerzo de los módulos para poder obtener el esfuerzo medio del proyecto.

Siguiendo la idea planteada, el resultado debe ser la productividad del proyecto, por tanto, con el esfuerzo de todos los módulos y teniendo en cuenta que lo que se evalúa es la productividad en esa semana, se puede obtener entonces el esfuerzo del proyecto.

2.3.1.7 Métrica de Esfuerzo de Proyecto

Esta métrica permite calcular el esfuerzo medio del proyecto en determinado período de tiempo. Se aplica en cada una de las fases de desarrollo y el responsable es el Evaluador de Calidad y el Líder de Proyecto.

$$E_p = \sum E_m (i) / C_m$$

Donde:

- E_m : esfuerzo del módulo
- M : módulo del proyecto
- i : desde 1 hasta C_m
- C_m : cantidad de módulos del proyecto.
- E_p (esfuerzo del proyecto) se obtiene en persona- semana.

Esta claro que para obtener la productividad de la producción se debe conocer el tamaño o complejidad del producto a producir, por lo que se define que la medida básica a utilizar en la métrica de productividad para el control de la misma será puntos de función, con los puntos de función y el esfuerzo del proyecto quedan creadas las bases para obtener la productividad del proyecto.

2.3.1.8 Métrica para Productividad del Proyecto

Esta métrica tiene como propósito evaluar los puntos de función según el esfuerzo del proyecto para obtener la productividad del proyecto.

$$PP = PF / EP \text{ (p/día, mes, año)}$$

Donde:

- PF : puntos de función del proyecto calculados según el procedimiento que se describe en el epígrafe 2.3.1.10.
- EP : esfuerzo del proyecto

- PP (productividad del proyecto) se obtiene en PF/ persona-semana (puntos de función por persona - semana)

Esta métrica se aplica durante diferentes etapas del desarrollo del software, y el responsable es el Evaluador de Calidad y el Líder de Proyecto. El resultado que se obtiene en esta métrica no es un valor predeterminado para todos los proyectos, por el contrario facilitará el control de la productividad para arrojar conclusiones que posteriormente se podrán emplear para conocer el avance en que se encuentra el proyecto mediante la realización de comparaciones.

2.3.1.9 Métricas Orientadas a Puntos de Función

Debido a la presencia de un gran tamaño en las aplicaciones producidas en los proyectos del Polo de Telecomunicaciones se decidió emplear la métrica de Puntos de Función para calcular el tamaño y la complejidad de las mismas. Las Métricas Orientadas a los Puntos de Función se centran en la funcionalidad o utilidad del programa, por eso muchos la definen como métrica funcional. Los Puntos de Función proporcionan una medida objetiva, cuantitativa y auditable del tamaño de las aplicaciones, desde el punto de vista de los requisitos especificados por el usuario final de la aplicación. Las métricas de productividad orientadas a puntos de función, son medidas indirectas del software y del proceso por el cual se desarrolla a diferencia de las orientadas al tamaño (LDC), que son medidas directas. También son un medio de entendimiento entre lo que el usuario quiere y lo que al final se le suministra. Su valoración se deriva a partir de los requisitos funcionales que la aplicación debe satisfacer, modelos de datos, definición de pantallas e interfaces gráficas y diagramas de análisis. [19]. Esta métrica se aplica en cada una de las fases de desarrollo. El responsable es el Evaluador de Calidad.

Los Puntos de Función constituyen una técnica de medida del software, simple de obtener pero muy potente en sus resultados. Esta potencia radica en que del valor de la medida en Puntos de Función se derivan un conjunto de métricas esenciales para la gestión de la productividad, la calidad y el coste del software. Con estas medidas registradas en distintas fases del ciclo de vida, se puede llevar a cabo un análisis exhaustivo de su evolución y, por tanto, del control de la productividad, la calidad y los costes asociados a lo largo del tiempo. De esta forma, y almacenando en un registro histórico de datos el valor en Puntos de Función de cada uno de los proyectos realizados, se podrá disponer de una sólida base para futuras estimaciones del costo y duración de los proyectos, información altamente valiosa para la dirección de las organizaciones.

Este método no se basa en las LDC (líneas de código) sino en una métrica que cuantifica la funcionalidad que hay que entregar al usuario al construir una aplicación. Dicha métricas se denomina Puntos de Función. La propuesta inicial de esta métrica fue realizada por A. J. Albrecht. Todas las variedades de puntos de función se apoyan en datos que implican, preferentemente, la existencia de una especificación más o menos formalizada.

Las métricas orientadas a Puntos de Función se caracterizan por:

Tener un componente empírico, basado en la experiencia de muchos proyectos.

Tener en cuenta la complejidad, aunque es muy difícil de determinar en un proyecto.

Ser independientes del entorno tecnológico y de las metodologías aplicadas.

Utilizar medidas indirectas, que se caracterizan por ser subjetivas y difíciles de calcular, sin embargo el resultado obtenido es fácilmente comparable.

Para el cálculo de los Puntos de Función es necesario seguir los siguientes pasos:

❖ Identificación de los componentes necesarios para el cálculo.

En esta etapa se identifican los elementos a tener en cuenta para el cálculo de los puntos función. Primeramente se enumeran todos los componentes de cada tipo (entradas de usuario, salidas de usuario, peticiones del usuario, archivos e interfaces); seguidamente, se evalúa individualmente la complejidad de cada uno de ellos, utilizando unas tablas ya establecidas que proporcionan el factor de complejidad de cada componente individual, siendo estos factores: COMPLEJO, MEDIO o SENCILLO.

❖ Cálculo de los Puntos de Función no ajustados (PFNA).

Tabla 6 Puntos de Función no ajustados.

Parámetros	Cantidad	Complejidad			PF
		simple	media	alta	
Número de entradas de usuario					
Número de salidas de usuario					

Número de peticiones de usuario					
Archivos					
Interfaces Externas					

❖ Ajuste de los Puntos de Función.

- Una vez calculado el valor que se obtuvo en los (PFNA) este se debe ajustar a las características del proyecto mediante un factor complejidad (FA).
- Existen 14 factores que contribuyen a la complejidad de una aplicación, estos se muestran a continuación.

Tabla 7 Factores de Complejidad.

Factores de Complejidad (FC)	(0-5)	Factores de Complejidad (FC)	(0-5)
Comunicación de Datos		Funciones Distribuidas	
Rendimiento		Gran carga de trabajo	
Frecuencia de transacciones		Entrada on-line de datos	
Requisitos de manejo del usuario final		Actualizaciones on-line	
Procesos Complejos		Utilización con otros sistemas	
Facilidad de mantenimiento		Facilidad de operación	
Instalación en múltiples lugares		Facilidad de cambio	

- A cada atributo se le asignará un valor entre 0 y 5, dependiendo del grado de influencia de éstos. Los posibles valores son:

- Sin influencia (0): El sistema no contempla este atributo.
- Influencia mínima (1): La influencia de este atributo es muy poco significativa.
- Influencia moderada (2): El sistema contempla este atributo y su influencia, aunque pequeña, ha de ser considerada.
- Influencia apreciable (3): La importancia de este atributo debe ser tenida en cuenta, aunque no es fundamental.
- Influencia significativa (4): Este atributo tiene una gran importancia para el Sistema.
- Influencia muy fuerte (5): Este atributo es esencial para el Sistema y ha de ser tenido en cuenta a la hora del diseño.

- El factor de ajuste de complejidad (FA) se calcula, a partir de la suma de los valores de los 14 factores de complejidad (FC), según la siguiente fórmula:

$$FA = (0.01 \times \sum FC) + 0.65$$

- El factor de ajuste de complejidad (FA) puede oscilar entre 0.65 y 1.35. Es decir permite una variación máxima de $\pm 35\%$ sobre el valor de los PFNA.

Una vez obtenida la información requerida para calcular los puntos de función, se obtiene el valor de los puntos de función a través de la siguiente fórmula:

$$PFA = PFNA \times FA$$

Obteniendo el valor de los puntos de función, se pueden aplicar métricas que aporten información sobre aspectos como costo, productividad y calidad.

$$\text{Productividad} = PF / \text{persona mes}$$

Se obtiene como resultado la productividad del equipo de desarrollo en función del esfuerzo.

(persona-mes) = cantidad de personas que realizan las actividades en un mes.

$$\text{Calidad} = \text{Errores}/PF$$

Con la aplicación de esta métrica se obtiene información que permite el control de que el cálculo de puntos de función se hizo correctamente.

Costo = Efectivo empleado para la elaboración del producto/ PF

Esta métrica permite una estimación del coste del producto que se produce. Con el valor de los puntos de función, se estima el posible coste que puede demandar el producto.

Documentación = Páginas de documentación / PF

Es una manera de controlar la documentación del software, lo que permite de paso estimar el avance del proyecto.

2.3.2 Guía para la Aplicación de la Propuesta

Luego de la realización de la propuesta se elaboró un resumen de los principales aspectos que contemplan cada una de las métricas, de forma que sirva como guía para la aplicación de las mismas en los proyectos del Polo de Telecomunicaciones. Se propone además, el empleo de un documento donde se registrarán los resultados de la aplicación de las métricas (Ver Anexo 3). Este documento será la base para la confección de un registro de datos donde cada proyecto podrá publicar los resultados de la evaluación de la calidad.

Tabla 8 Resumen de la Propuesta de Métricas

Nombre de la Métrica	La métrica se propone	Fórmula	Frecuencia de aplicación	Responsable	Fuentes	Recepción de la información
Eficacia de la eliminación de defectos. EED	Detectar errores antes de la entrega del producto al cliente final.	$EED = E / (E + D)$	Antes de cada una de las versiones desplegadas con el cliente final.	Evaluador de la Calidad.	Modelo de Despliegue.	Documento Resultados de Aplicación de Métricas de Software. (RAM) Ver Anexo 3.

Capítulo 2: Descripción de la Propuesta de Solución

Calidad de la especificación de requisitos. Q	Detectar que no exista ambigüedad en los requisitos.	$Q = N_{ui} / N_r$ $N_r = N_f + N_{nf}$	Fase de Inicio, Flujo Requerimientos	Evaluador de la Calidad.	Especificación de Requisitos.	RAM
Control de casos de uso diseñados. CCUD	Detectar el número de CU definidos que no fueron diseñados.	$CCUD = CUD / CU_{Def}$	Fase de Inicio, Flujo de Análisis y Diseño.	Evaluador de la Calidad.	Modelo de Casos de uso del sistema.	RAM
Agilidad en la gestión de cambios de requisitos. AGCR	Detectar la agilidad con que se realiza el proceso de Gestión de Cambios en los Requisitos.	$AGCR = 1 - (TGCR / TPGR)$	Fase de Inicio, Flujo de Requerimientos	Evaluador de la Calidad.	Cronograma del Proyecto.	RAM
Mantenimiento IMS	Detectar la estabilidad del producto.	$IMS = [M_r - (F_a + F_c + F_d)] / M_r$	Todas las fases de desarrollo.	Evaluador de la Calidad.	Registro de Proyecto.	RAM
Esfuerzo del módulo. EM	Conocer el esfuerzo de las personas asignadas a realizar cierta tarea dentro del módulo.	$E_m = P / t$	Todas las fases de desarrollo.	Evaluador de la Calidad y Líder de Proyecto.	Registro de Proyecto.	RAM
Esfuerzo del proyecto.	Conocer el esfuerzo total realizado	$E_p = \sum E_m(i) / C_m$	Todas las fases de desarrollo.	Evaluador de la Calidad y Líder de	Registro de Proyecto.	RAM

EP	en el proyecto.			Proyecto.		
Puntos de Función. PF	Medir el tamaño o la complejidad del proyecto.	$PFA = PFNA \times FA$ $FA = (0.01 \times \sum FC) + 0.65$	Todas las fases de desarrollo.	Evaluador de la Calidad.	Expediente del Proyecto.	RAM
Productividad del proyecto. PP	Medir el estado de la productividad del proyecto.	$PP = PF / EP$	Todas las fases de desarrollo.	Evaluador de la Calidad y Líder de Proyecto.	Registro de Proyecto y Expediente de Proyecto.	RAM

2.4 Conclusiones del Capítulo

En el desarrollo del presente capítulo se describió la entrevista realizada como uno de los métodos que ayudaron a la selección de la propuesta de solución. Además se describió la propuesta de métricas a utilizar en los proyectos. Para todas las métricas se hizo una breve descripción de la actividad o característica que se pretende medir, se expresó la fórmula explicando cada una de las variables que la componen y mediante una tabla se delimitó el rango de valores que puede tomar y el valor ideal de la métrica.

Capítulo 3: Validación de la Propuesta de Solución

3.1 Introducción

En este capítulo se realizará la evaluación técnica de la propuesta descrita en el capítulo anterior. Se usará el Método de Validación Interna y Externa, también llamada teórica y empírica. Este capítulo aporta elementos que permiten validar la solución propuesta en la investigación realizada. Posteriormente se presentarán los resultados obtenidos de la aplicación de las métricas propuestas en los proyectos Teleidentificador Personal (TIP) y CUBACEL de la Facultad 2.

3.2 Método de Validación de la Propuesta

En ocasiones se ha definido la medición como el proceso por el cual se asignan números o símbolos a atributos de entidades del mundo real. De esta manera, se disminuye cierto grado de subjetividad en un proceso de medición, mediante la utilización de métricas. Sin embargo, no sólo interesa contar con una diversidad de métricas, sino también, saber si dichas métricas son válidas. Es decir, si miden lo que realmente debieran medir. De esta forma, la validación constituye un aspecto crucial en la medición del software, ya que asegura que las medidas representen con precisión los atributos del software que se pretende cuantificar.

En la literatura se reconocen dos tipos de validación: la interna y la externa, comúnmente referidas como teórica y empírica respectivamente.

Este tipo de validación, descrita en el epígrafe 1.4.4 permite que a partir de los datos numéricamente reales, recogidos de los proyectos, se realicen los cálculos correspondientes a cada una de las métricas propuestas. Con la obtención de estos resultados, se podrán hacer comparaciones con resultados anteriormente obtenidos en las diferentes fases de desarrollo por las que atraviesa el software. También a partir del valor obtenido en cada métrica y siendo este valorado en el rango que se define en cada una, se llegarán a conclusiones y soluciones determinando los factores que impiden la calidad del producto y el buen funcionamiento del mismo. De esta manera queda plasmada la evidencia de que la propuesta realizada cumple con el objetivo que se plantea, evaluar aún más la calidad de los proyectos en la Facultad 2, logrando así la mejora continua de la misma.

Se propone validar la propuesta de solución en función de esta información, demostrando que la propuesta es realmente útil y que matemáticamente se puede obtener un valor con el que se pueda obtener información para el desarrollo de los proyectos así como para futuros trabajos a realizar.

3.2.1 Aplicación de la Propuesta en los Proyectos TIP y CUBACEL

Para el proceso de aplicación de las métricas propuestas se seleccionó de una población de 8 proyectos que integran el Polo de Telecomunicaciones una muestra de 2 proyectos (TIP y CUBACEL), considerados los más apropiados debido al avance en el desarrollo del producto final y por poseer una amplia organización de todo el trabajo realizado.

A continuación se describe el proceso llevado a cabo para la validación de las métricas descritas en la propuesta de solución en el capítulo 2 de forma que se demuestre que la selección es realmente útil. Este proceso se realizó con una muestra real de datos de los proyectos TIP y CUBACEL, permitiendo la obtención de resultados que ayudarán a la mejora de la calidad de los mismos.

Tabla 9 Métrica para obtener la eficacia de la eliminación de defectos

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Número de errores encontrados antes de la entrega del software al usuario final	0	0
Número de defectos encontrados después de la entrega	5	6
Eficacia en la eliminación de defectos	0	0

Fórmula

$$EED = E / (E + D)$$

Sustitución para el proyecto TIP

$$EED = 0 / 0 + 5$$

$$EED = 0 / 5$$

$$EED = 0$$

Sustitución para el proyecto CUBACEL

$$EED = 0 / 0 + 6$$

$$EED = 0 / 6$$

$$EED = 0$$

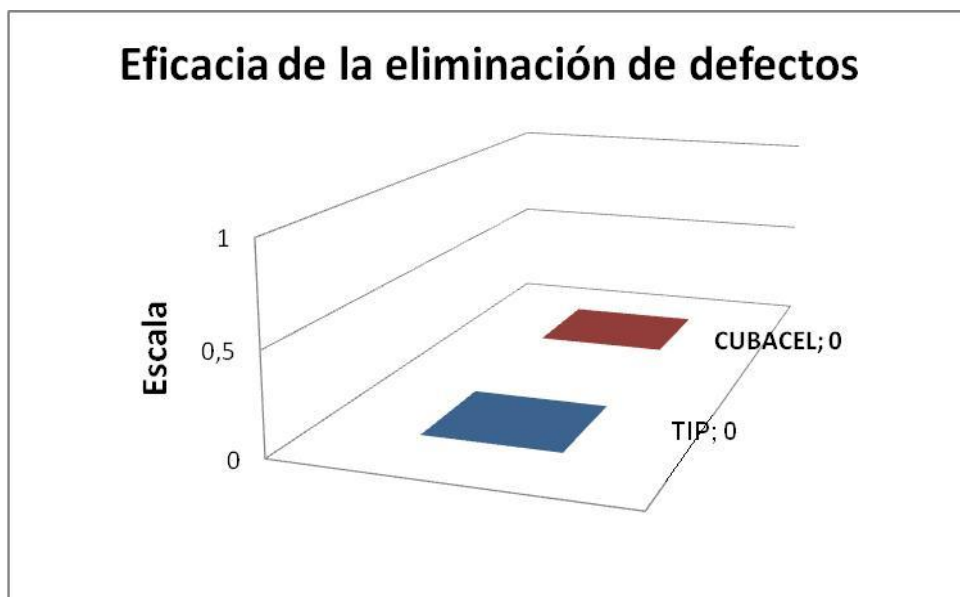


Figura 2 Gráfica de valores alcanzados en la métrica EED

Esta métrica fue calculada con datos obtenidos de una de las versiones entregadas al usuario en el proceso de despliegue del proyecto TIP y CUBACEL como procedimiento de pruebas de aceptación, donde el usuario es capaz de encontrar todos los defectos que presenta el producto antes de la entrega final del mismo. Este procedimiento es llevado a cabo en este proyecto para evaluar la calidad que tiene el producto que se está fabricando. Se debe destacar que el proyecto TIP no emplea ningún procedimiento o métrica para evaluar la capacidad que tiene el equipo de trabajo en detectar los errores antes que el usuario final y en el caso del proyecto CUBACEL realizan pruebas de Caja negra y Caja blanca antes del proceso de despliegue. Es recomendable seleccionar un grupo de estudiantes que sean los encargados de hacer las pruebas necesarias antes de la entrega de cada versión al usuario, y de esta manera filtrar la mayor cantidad de errores que tenga el software. Es importante destacar que aunque los defectos encontrados en las pruebas realizadas por el usuario no dependían totalmente del equipo de trabajo sino del sistema operativo en el caso de TIP y de algunos errores de estética en el caso de CUBACEL, pueden estar en riesgo que nuevamente se encuentren otros defectos diferentes. Con el empleo de esta métrica y la inclusión de un grupo de revisores, se logrará que los resultados que se obtengan en los próximos despliegues hasta la entrega final del producto sean cada vez mejores, disminuyendo la cantidad de defectos encontrados por el usuario y documentando estos errores para después hacer comparaciones que sirvan de guía para evaluar el avance que va teniendo el proyecto en cuanto a la eliminación de defectos encontrados por el usuario final.

Tabla 10 Métrica para evaluar la Calidad de la especificación de requisitos

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Número de requerimientos	42	57
Número de requerimientos funcionales	17	31
Número de requerimientos no funcionales	25	26
Número de requerimientos donde todos los revisores tuvieron interpretaciones idénticas.	42	57
Especificidad	1	1

Fórmula

$$Nr = Nf + Nnf$$

$$\text{Especificidad (Q)} = Nui / Nr$$

Sustitución para TIP

$$Nr = 17 + 25$$

$$Nr = 42$$

$$Q = 42 / 42$$

$$Q = 1$$

Sustitución para CUBACEL

$$Nr = 31 + 16$$

$$Nr = 47$$

$$Q = 47 / 47$$

$$Q = 1$$

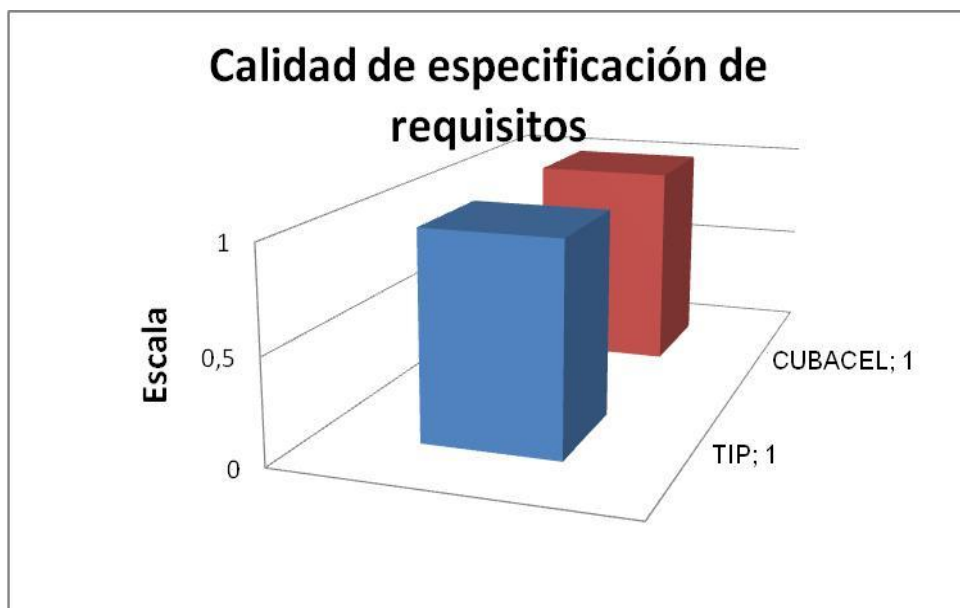


Figura 3 Gráfica de valores de la métrica calidad de especificación de requisitos

Con el cálculo realizado se puede obtener un valor que permitirá conocer el nivel de ambigüedad de la especificación de los requisitos. En el caso de los resultados obtenidos tanto en el proyecto TIP como CUBACEL se pudo apreciar que en ambos proyectos se obtiene el valor ideal de esta métrica, lo que resalta que no existe ambigüedad ninguna entre los requisitos obtenidos durante toda la fase de levantamiento de requisitos en el proyecto, permitiendo de esta manera almacenar un dato que podrá ser de utilidad para comparaciones posteriores a lo largo de todo el proceso de producción desarrollado en la facultad donde aparezca un proyecto con características similares a los analizados. Se concluyó que ambos proyectos realizaron el proceso de levantamiento de requisitos de forma correcta, evidenciando de esta forma, el excelente trabajo llevado a cabo por todo el equipo. Es necesario destacar la importancia del empleo de esta métrica en el momento del Levantamiento de Requisitos es primordial el conocimiento de los errores que se cometen en esta fase.

Tabla 11 Métricas para el control de los casos de uso diseñados

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Número de CU que fueron diseñados.	18	16
Número de CU que fueron definidos.	18	20

Fórmula

$$CCUD = CUD / CUDef$$

Sustitución para TIP

$$CCUD = 18/18$$

$$CCUD = 1$$

Sustitución para CUBACEL

$$CCUD = 16/20$$

$$CCUD = 0.8$$

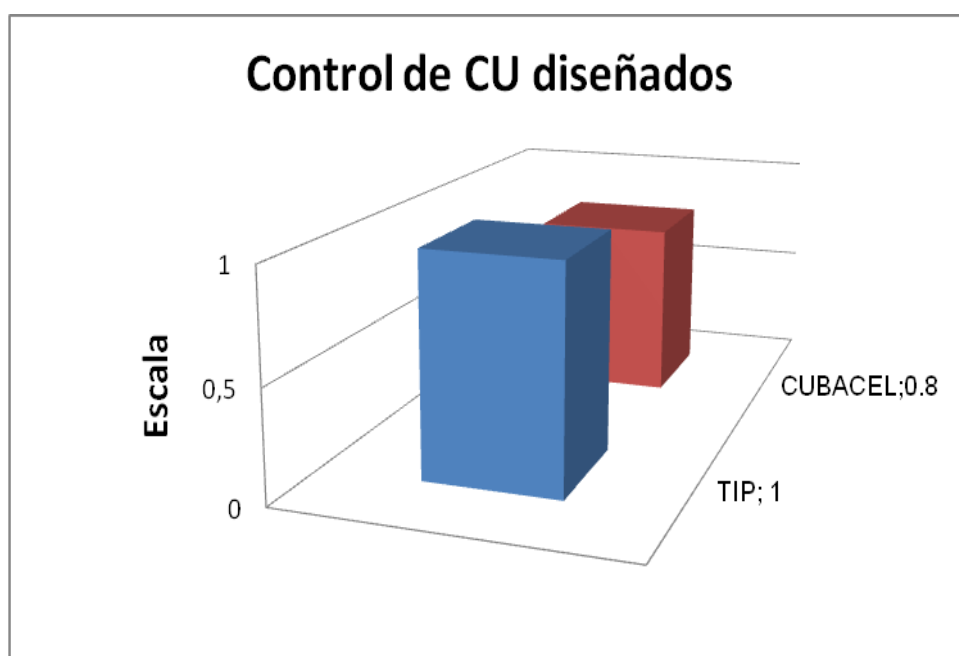


Figura 4 Gráfica de valores de la métrica CCUD

Realizando la observación del resultado obtenido en el cálculo de la métrica planteada en los proyectos TIP y CUBACEL se puede apreciar que el mismo permite tener constancia que todos los casos de usos definidos fueron en su totalidad bien diseñados en el proyecto TIP, por lo que evaluándolo en la tabla de rango se obtiene que el trabajo realizado está categorizado de bien. Esto facilita el control de la evolución de los casos de usos del proyecto por parte del Líder del proyecto. En el caso del proyecto CUBACEL se aprecia que no fueron diseñados todos los casos de uso que se definieron primeramente, evaluándolo en la tabla de rango de regular, debido a que no se cumplió con el diseño completo de los CU que se definieron para el proyecto. La aplicación de esta métrica en el

proceso de Análisis y Diseño servirá como base para que el Líder de proyecto pueda medir el trabajo que están realizando los diseñadores de CU y de esta manera tomar medidas para mejorar su desempeño.

Tabla 12 Métrica para la agilidad en la gestión de cambios de los requisitos

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Tiempo dedicado a la Gestión de Cambio de los Requisitos.	3 días	3 días
Tiempo total dedicado al proceso de Gestión de Requisitos.	7 días	60 días

Fórmula

$$AGCR = 1 - (TGCR / TPGR)$$

Sustitución para TIP

$$AGCR = 1 - (3/7)$$

$$AGCR = 1 - 0.42$$

$$AGCR = 0.58$$

Sustitución para CUBACEL

$$AGCR = 1 - (3/60)$$

$$AGCR = 1 - 0.05$$

$$AGCR = 0.95$$

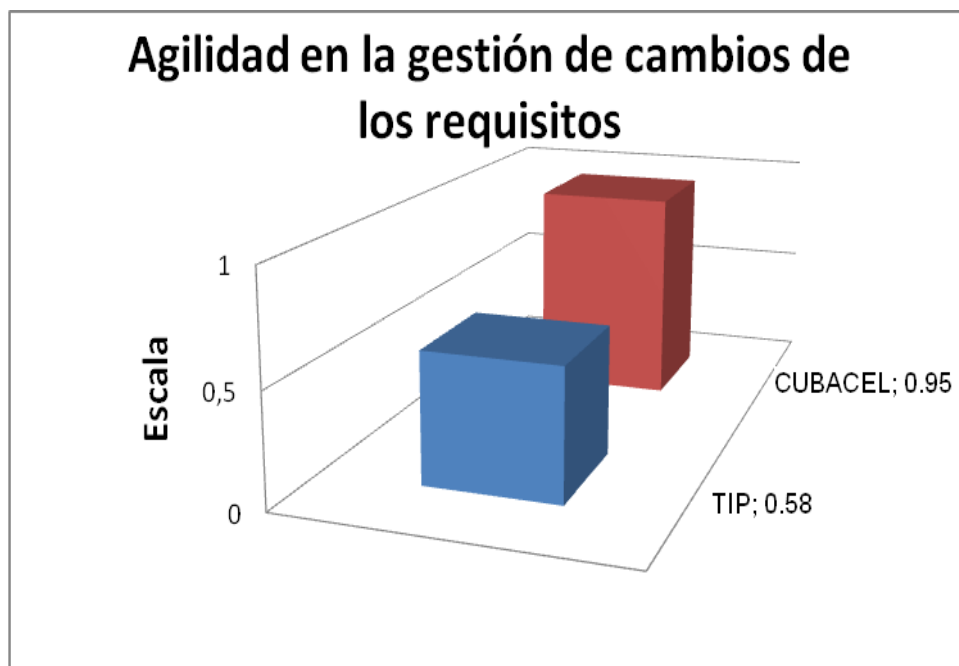


Figura 5 Gráfica de valores de la métrica AGCR

El resultado obtenido del cálculo de la métrica para la agilidad en la gestión de cambios de los requisitos aplicada en los proyectos TIP y CUBACEL permite conocer que este proceso no se ha desarrollado con la agilidad total requerida en el proyecto TIP, sino, que se encuentra en un proceso medio lo que facilita al líder del proyecto el conocimiento necesario para enfatizar aún más en esta área de trabajo y así lograr que el proceso llegue a su máximo rendimiento. Sin embargo en el proyecto CUBACEL se pudo apreciar que el proyecto se encuentra en un desarrollo plenamente ágil lo que le facilita al líder del proyecto el conocimiento que el proceso de gestión de cambios de requisitos se desarrolla con bastante agilidad obteniéndose los mejores resultados. Es indispensable que el resultado de esta métrica se evalúe en el proceso de la Gestión de Cambios de Requisitos, debido a que si los resultados obtenidos son perjudiciales para la entrega del producto final, se deben aplicar las medidas necesarias en este momento.

Tabla 13 Métrica de mantenimiento

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Número de módulos en la versión actual.	7	9

Número de módulos en la versión actual que se han cambiado.	-	-
Número de módulos en la versión actual que se han añadido.	-	-
Número de módulos de la versión anterior que se han borrado en la versión.	-	-

Fórmula

$$IMS = [Mr - (Fa + Fc + Fd)] / Mr$$

Sustitución para TIP

$$IMS = [7 - (0 + 0 + 0)] / 7$$

$$IMS = 1$$

Sustitución para CUBACEL

$$IMS = [9 - (0 + 0 + 0)] / 9$$

$$IMS = 1$$

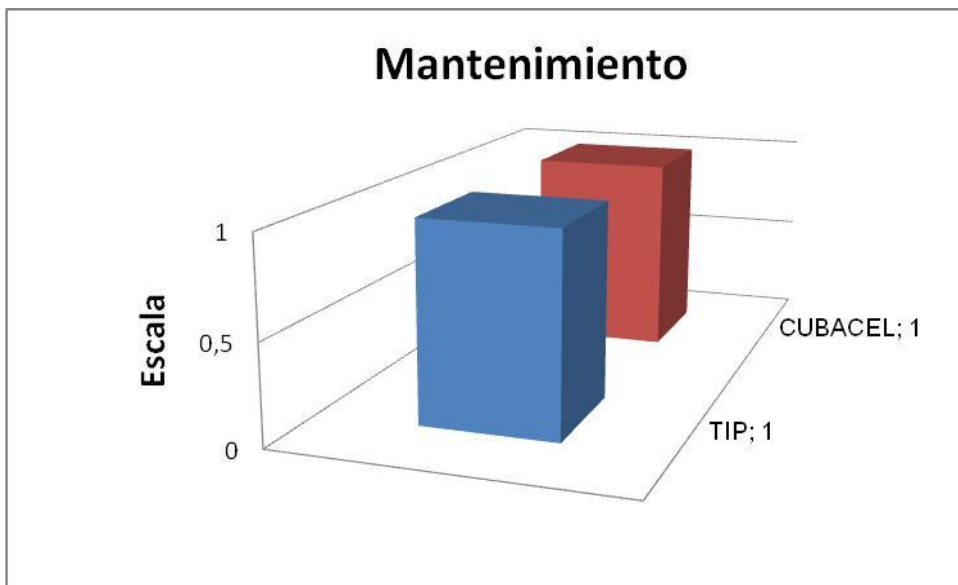


Figura 6 Gráfica de valores de la métrica de mantenimiento

El valor obtenido de la métrica de mantenimiento aplicada en los proyectos TIP y CUBACEL, da como resultado, conocer que el producto se encuentra estabilizado totalmente ya que permite controlar que no se han realizado cambios de ningún tipo en los módulos, lo que garantiza que el trabajo se lleva a cabo con una organización adecuada y sin modificaciones algunas en la composición del área de trabajo destinada al desarrollo del producto que se obtiene.

Tabla 14 Métrica para el esfuerzo del módulo

Parámetro	Valor Obtenido TIP							Valor Obtenido CUBACEL								
	Módulos							Módulos								
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	8	9
Cantidad de personas asignadas a realizar la tarea.	2	2	2	6	2	2	2	1	1	1	1	1	1	1	1	1
Tiempo planificado para el cumplimiento de la tarea.	36	36	36	36	36	36	36	8	8	8	8	8	8	8	8	8

Fórmula

$$E_m = P / t$$

t → en semanas

Sustitución para proyecto TIP

<u>Módulo #1</u>	<u>Módulo #2</u>	<u>Módulo #3</u>	<u>Módulo #4</u>	<u>Módulo #5</u>
$E_m = 2/36$	$E_m = 2/36$	$E_m = 2/36$	$E_m = 6/36$	$E_m = 2/36$
$E_m = 0.05$	$E_m = 0.05$	$E_m = 0.05$	$E_m = 0.16$	$E_m = 0.05$
<u>Módulo #6</u>	<u>Módulo #7</u>			

$E_m = 2/36$ $E_m = 2/36$

$E_m = 0.05$ $E_m = 0.05$

Sumatoria de los esfuerzos de los módulos: 0.46

Sustitución para proyecto CUBACEL

Módulo #1 Módulo #2 Módulo #3 Módulo #4 Módulo #5

$E_m = 1/8$ $E_m = 1/8$ $E_m = 1/8$ $E_m = 1/8$ $E_m = 1/8$

$E_m = 0.12$ $E_m = 0.12$ $E_m = 0.12$ $E_m = 0.12$ $E_m = 0.12$

Módulo #6 Módulo #7 Módulo #8 Módulo #9

$E_m = 1/8$ $E_m = 1/8$ $E_m = 1/8$ $E_m = 1/8$

$E_m = 0.12$ $E_m = 0.12$ $E_m = 0.12$ $E_m = 0.12$

Sumatoria de los esfuerzos de los módulos: 1.08

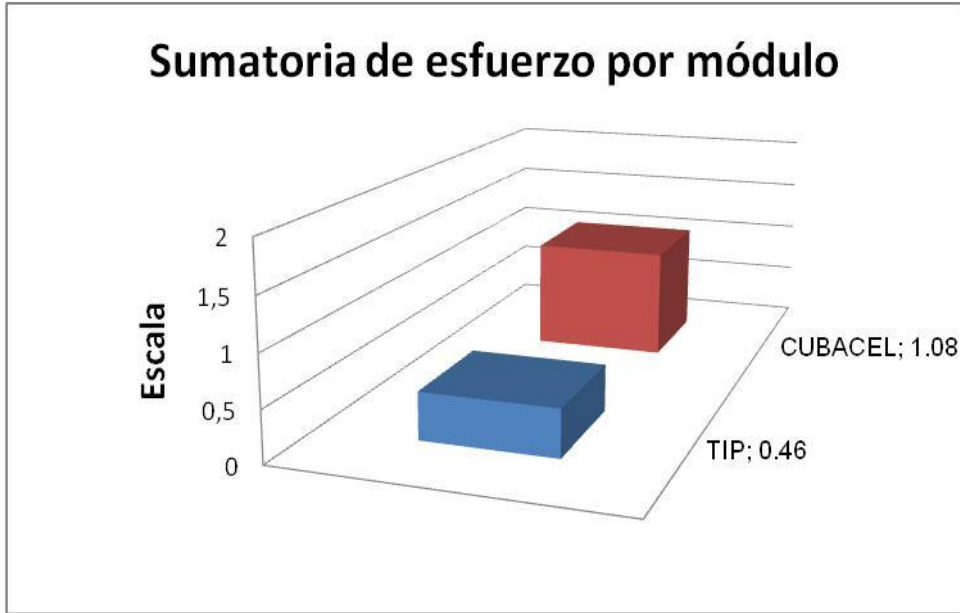


Figura 7 Gráfica de valores de la métrica Esfuerzo por módulo

El resultado obtenido luego de la evaluación de la métrica para el cálculo del esfuerzo de los módulos en los proyectos TIP y CUBACEL, permitirá conocer el esfuerzo que se está realizando en cada uno

de los módulos de estos proyectos. De esta forma se podrán realizar comparaciones de dichos esfuerzos en distintas etapas de la producción para así conocer el estado del avance en que se encuentra el proyecto con el cálculo del esfuerzo del proyecto en general.

Tabla 15 Métrica para el esfuerzo del proyecto

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Sumatoria del Esfuerzo de los módulos.	0.46	1.08
Cantidad de módulos del proyecto.	7	9
Esfuerzo del Proyecto	0.65	0.12

Fórmula

$$E_p = \sum E_m (i) / C_m$$

Sustitución para TIP

$$\sum E_m = E_{m1} + E_{m2} + E_{m3} + E_{m4} + E_{m5} + E_{m6} + E_{m7}$$

$$\sum E_m = 0.05 + 0.05 + 0.05 + 0.16 + 0.05 + 0.05 + 0.05$$

$$\sum E_m = 0.46$$

$$E_p = 0.46 / 7$$

$$E_p = 0.65$$

Sustitución para CUBACEL

$$\sum E_m = E_{m1} + E_{m2} + E_{m3} + E_{m4} + E_{m5} + E_{m6} + E_{m7} + E_{m8} + E_{m9}$$

$$\sum E_m = 0.12 + 0.12 + 0.12 + 0.12 + 0.12 + 0.12 + 0.12 + 0.12 + 0.12$$

$$\sum E_m = 1.08$$

$$E_p = 1.08/9$$

Ep = 0.12

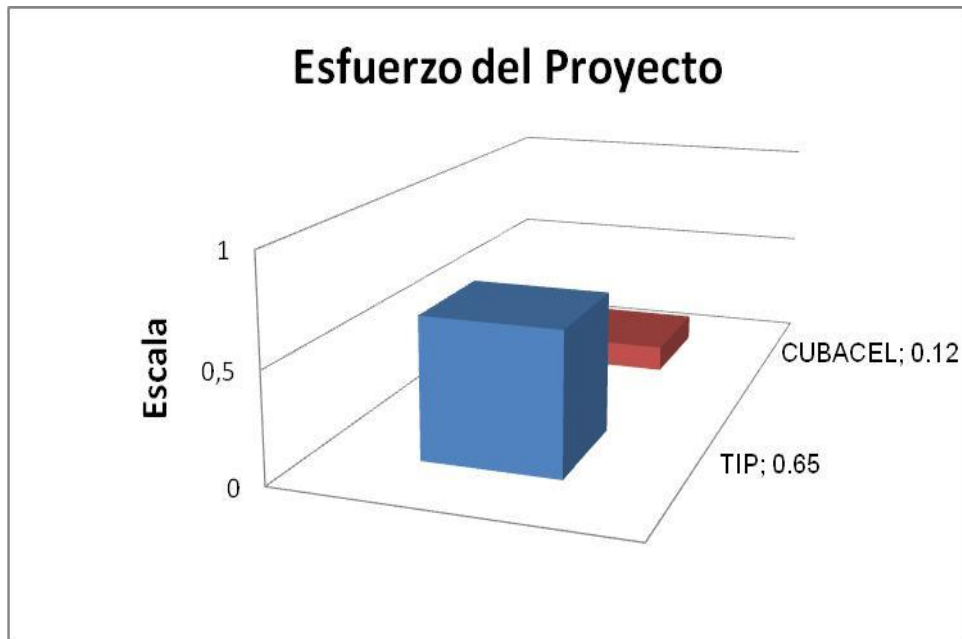


Figura 8 Gráfica de valores de la métrica esfuerzo del proyecto

Con el resultado obtenido del empleo de la métrica para obtener el esfuerzo en los proyectos TIP y CUBACEL, se podrá obtener la productividad de los mismos y además le permitirá al líder del proyecto tener un conocimiento real del esfuerzo que se está realizando, lo que facilitará la toma de decisiones en cuanto a este parámetro para mejorar o mantener la calidad del mismo.

Tabla 16 Métrica para la obtención de los puntos de Función para el proyecto TIP

Parámetros	Cantidad	Complejidad			PF
		simple	media	alta	
Número de entradas de usuario	11	8x3	2x4	1x6	38
Número de salidas de usuario	6	6x4	x5	x7	24
Número de peticiones de usuario	4	2x3	2x4	x6	14
Archivos	3	3x7	x10	x15	21

Interfaces Externas	2	2x5	x7	x10	10
PFNA					107

Tabla 17 Factores de complejidad para el proyecto TIP

Factores de Complejidad (FC)	(0-5)	Factores de Complejidad (FC)	(0-5)
Comunicación de Datos	5	Funciones Distribuidas	5
Rendimiento	3	Gran carga de trabajo	2
Frecuencia de transacciones	3	Entrada on-line de datos	5
Requisitos de manejo del usuario final	1	Actualizaciones on-line	5
Procesos Complejos	2	Utilización con otros sistemas	1
Facilidad de mantenimiento	2	Facilidad de operación	4
Instalación en múltiples lugares	1	Facilidad de cambio	2

Fórmula

$$FA = (0.01 \times \sum FC) + 0.65$$

$$PFA = PFNA \times FA$$

Sustitución para el proyecto TIP

$$FA = (0.01 \times 41) + 0.65$$

$$FA = 0.41 + 0.65$$

$$FA = 1.06$$

$$PFA = 107 \times 1.06$$

$$PFA = 113.42$$

Capítulo 3: Validación de la Propuesta de Solución

Por lo tanto el tamaño del software es de 113.42 puntos de función.

Tabla 18 Métrica para la obtención de los puntos de Función para el proyecto CUBACEL

Parámetros	Cantidad	Complejidad			PF
		simple	media	alta	
Número de entradas de usuario	7	4x3	2x4	1x6	26
Número de salidas de usuario	4	4x4	x5	x7	16
Número de peticiones de usuario	3	3x3	x4	x6	9
Archivos	1	x7	x10	1x15	15
Interfaces Externas	1	x5	x7	1x10	10
PFNA					76

Tabla 19 Factores de complejidad para proyecto CUBACEL

Factores de Complejidad (FC)	(0-5)	Factores de Complejidad (FC)	(0-5)
Comunicación de Datos	4	Funciones Distribuidas	1
Rendimiento	5	Gran carga de trabajo	2
Frecuencia de transacciones	1	Entrada on-line de datos	3
Requisitos de manejo del usuario final	4	Actualizaciones on-line	4
Procesos Complejos	2	Utilización con otros sistemas	5
Facilidad de mantenimiento	3	Facilidad de operación	3

Fórmula

$$FA = (0.01 \times \sum FC) + 0.65$$

$$PFA = PFNA \times FA$$

Sustitución para el proyecto TIP

$$FA = (0.01 \times 37) + 0.65$$

$$FA = 0.37 + 0.65$$

$$FA = 1.02$$

$$PFA = 76 \times 1.02$$

$$PFA = 77.52$$

Por lo tanto el tamaño del software es de 77.52 puntos de función.

Tabla 20 Métrica para el cálculo de la productividad del proyecto orientada hacia los puntos de función

Parámetro	Valor Obtenido TIP	Valor Obtenido CUBACEL
Esfuerzo del proyecto	0.65	0.12
Puntos de función	113.42	77.52
Productividad	174.49	646

Fórmula

$$PP = PF / EP \text{ (p/día, mes, año)}$$

Sustitución para TIP

$$PP = 113.42 / 0.65$$

$$PP = 174.49$$

Sustitución para CUBACEL

$$PP = 77.52 / 0.12$$

$$PP = 646$$

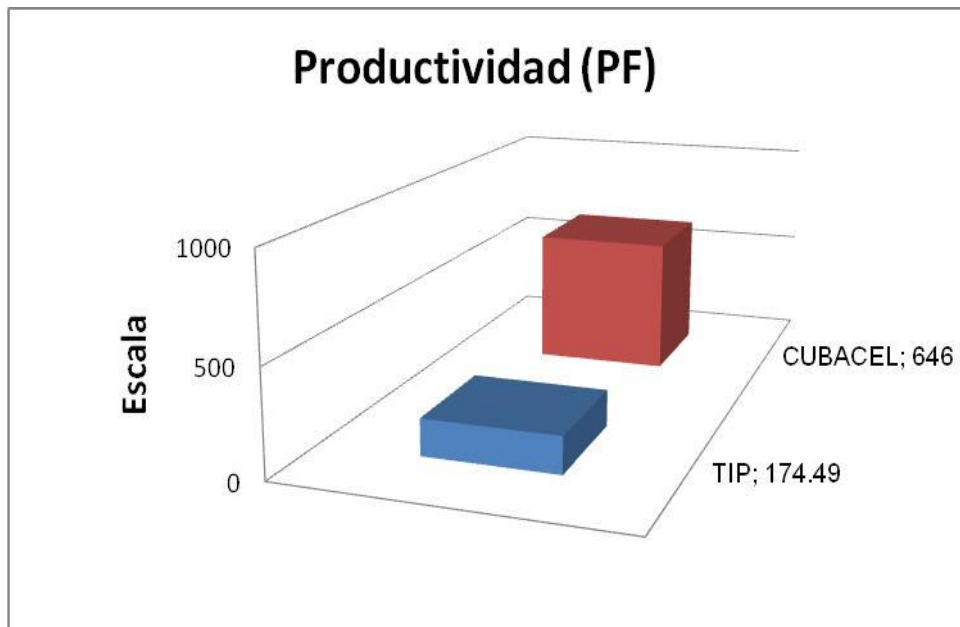


Figura 9 Gráfica de valores de la métrica productividad orientada a los PF

El valor de la productividad obtenida al evaluar los valores en esta métrica, se puede obtener información como por ejemplo: si existe mucha diferencia entre las productividades de los proyectos indica que no se ha seguido un procedimiento estándar, si se compara la productividad del proceso general con la productividad en etapas anteriores, se puede concluir el estado de la productividad, si ha avanzado, o no.

3.2.2 Conclusión de la Aplicación de la Propuesta

Luego de la realización de la aplicación de las métricas propuestas se concluyó que:

- No detectar los errores antes de pasar a fases posteriores.
- Emplear tiempo rectificando el diseño de los casos de uso.
- Falta de agilidad en el proceso de gestión de cambios de requisitos.

Son factores que impiden que en los proyectos analizados no se produzcan los productos con la mejor calidad posible. Se concluyó además, que el Evaluador de la Calidad realiza junto a sus funciones el trabajo del Revisor, debido a la ausencia de este rol en los proyectos. Esta función se hace imprescindible para lograr una mayor organización y un mejor resultado del trabajo.

Además de estos factores negativos que arrojaron los resultados de la aplicación de las métricas en los proyectos TIP y CUBACEL puede ser posible que en los restantes proyectos del Polo de Telecomunicaciones se obtengan igualmente factores como:

- La presencia de ambigüedad en los requisitos.
- La realización de cambios en el área de los módulos de trabajo.
- La diferencia entre las productividades obtenidas en las diferentes fases de desarrollo.

Estos factores al igual que los anteriormente mencionados influyen en que se necesite emplear un mayor tiempo para la obtención de la aplicación, ocasionando de esta forma la entrega tardía del producto final.

3.3 Conclusiones del Capítulo

Para la validación de la propuesta se utilizó el Método de Validación Interna y Externa. Al aplicar este método y analizar los resultados se obtuvo una alta probabilidad de éxito por lo que la aplicación de la propuesta debe brindar resultados favorables para lograr la calidad de los proyectos que se realizan en la Facultad 2. Con la aplicación de esta propuesta se demostró que lo planteado hasta el momento se adapta valiosamente a las condiciones existentes y aporta novedosos elementos que resultan imprescindibles para adquirir un producto con la calidad requerida.

Conclusiones Generales

Con el trabajo realizado se le dio cumplimiento al objetivo trazado donde:

- Se obtuvo una propuesta, la cual recoge la interpretación de varias métricas que permiten evaluar la calidad de los proyectos que se producen en el Polo de Telecomunicaciones.
- Se investigó sobre la calidad, su control y evaluación y si en los proyectos de la Facultad 2 se estaban aplicando métricas para que los productos tuvieran la calidad que se requiere. Toda la investigación llevada a cabo, arrojó como resultado la propuesta de solución, la cual se describió detalladamente y con una fácil interpretación.
- El empleo de esta propuesta arrojó beneficios para los proyectos seleccionados, así como, para el resto de los proyectos pertenecientes al Polo de Telecomunicaciones. Los resultados obtenidos sirvieron de indicadores para conocer el estado de la calidad en que se encontraban estos productos.
- Se validó la propuesta realizada a partir de los métodos de validación interna y externa en los proyectos seleccionados lo que evidenció resultados satisfactorios que demuestran la eficiencia de las métricas a la hora de evaluar la calidad del proyecto y da un grado de confianza y certeza sobre la futura implantación de la propuesta en todos los proyectos que se realizan en el Polo de Telecomunicaciones de la Facultad 2.

Recomendaciones

Este trabajo de diploma propuso un sistema de Métricas para Evaluar la Calidad al Software desarrollado en los proyectos del Polo de Telecomunicaciones se recomienda:

- Publicar los resultados de este trabajo de diploma para poner a disposición de la facultad el sistema de métricas propuesto.
- Publicar este trabajo en foros nacionales e internacionales para obtener valoraciones de personas interesadas en el tema de la propuesta.
- Actualizar la propuesta de forma periódica debido a la entrada de nuevos proyectos al Polo de Telecomunicaciones que pueden presentar características nuevas que ocasionen la necesidad del empleo de nuevas métricas.

Referencias Bibliográficas

1. SEI. Capability Maturity Model Integration (CMMISM). Carnegie Mellon University, 2002. p.
2. Infocalidad 2005. Disponible en: http://www.infocalidad.net/gest_calidad_def/definicion.asp
3. Organización Internacional para la Estandarización. Concepts and Terminology. Revision of ISO 9000 Quality management systems- Fundamentals and vocabulary Preliminary working draft. 2007. 45 p.
4. Fernández, O. M. Un enfoque actual sobre la calidad del software. Disponible en: http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm
5. PRESSMAN, R. S. Ingeniería de Software. Un enfoque práctico. Quinta Edición. 2002. p.
6. Cueva, J. M. Calidad de Software. Departamento de Informática. Universidad de Oviedo, España.
7. González, H.D., Las Métricas de Software y su Uso en la Región, Desarrollado en: Departamento de Ingeniería en Sistemas Computacionales. Mayo, 2001, Universidad de las Américas Puebla.
8. Oscar M. Fernández, Delba García, Alfa Beltrán Un enfoque actual sobre la calidad del software. 1995. Disponible en: http://www.bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm
9. González, D. Las Métricas de Software y su Uso en la Región. México, Universidad de las Américas, Puebla, 2001. p.
10. Estévez, I.P., Métricas para el control de Proyectos de Software. Junio 2002, Instituto Superior Politécnico “José Antonio Echeverría” Facultad de Ingeniería Industrial: Cuba.
11. MUÑOZ, C. C. Métricas del Software: Conceptos básicos, definición y formalización, [Artículo]. Universidad de Castilla-La Mancha, 2006. [2009-01-19]. Disponible en: http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software

12. ALONSO, E. B. Medición y métricas del software, [Artículo]. Universidad de Vigo, 2004. [2009-02-05]. Disponible en: <http://trevinca.ei.uvigo.es/~ebalonso/asignaturas/esx/guiones>
13. ESTRADA, A. F. and I. P. ESTÉVEZ. Medir el proceso de control de configuración, ¿una utopía para la Industria Nacional de Software?, Instituto Superior Politécnico “José Antonio Echeverría”, 2003. [2009-01-14]. Disponible en: <http://www.inf.udec.cl/revista/ediciones/edicion9/febles.pdf>
14. ILINA, L. T. and R. P. GONZÁLEZ. Propuesta de un sistema de métricas para la evaluación de los proyectos de gestión de la UCI Universidad de las Ciencias Informáticas, 2007. p.
15. Verificación y Validación. Disponible en: <http://www.di.ujaen.es/asignaturas/computacionestadistica/pdfs/tema6.pdf>
16. Curbelo, M. H. Los métodos de evaluación y decisión Multicriterio, Universidad de Cienfuegos 2006. Disponible en: <http://www.monografias.com/trabajos40/decision-multicriterio/decision-multicriterio.shtml>
17. Cordovi, N. C. Propuesta de un Estándar y Métricas para el control del Proyecto Productivo Informatización del Convenio Integral Cuba-Venezuela. La Habana, Universidad de las Ciencias Informáticas, 2008. p.
18. Oliva, A. D and Esther, C. O. Métricas para la evaluación del proceso de desarrollo del Software Educativo. La Habana, Universidad de las Ciencias Informáticas, 2008. p.
19. Albrecht 1979. Puntos de Función. Disponible en: <http://www.sc.ehu.es/jiwdocoj/mmis/fpa.htm>

Bibliografía

- C, M. P. V. "Modelo de Gestión de Proyectos Software: Estimación del Esfuerzo de Desarrollo".
- CERRILLO, D. "ESTIMACIÓN DEL SOFTWARE", 1999.
- Estrada, A. F. (2003). Aseguramiento de Calidad de Software. La Habana, Cuba.
- Estrada, D. A. (2005). Gestión de Proyectos. Métricas de Software. La Habana, Cuba.
- Giraldo, O.P. (2006) Métricas, Estimación y Planificación en Proyectos de Software. Volumen, IEEE, Software Engineering Standards. 1993. p. 47-48.
- ISO/IEC (2001). 9126-1:2001 Software engineering—Product quality—Part 1: Quality Model.
- Oliva, A. D and Esther, C. O. Métricas para la evaluación del proceso de desarrollo del Software Educativo. La Habana, Universidad de las Ciencias Informáticas, 2008. p.
- PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico. Quinta Edición. 2002. p.
- RUP. "Rational Unified Process". 2003.
- VERGAS, E. "PLAN DE MÉTRICAS EN OCHO PASOS".

Anexos

Anexo #1: Documentos empleados para obtener la información necesaria para realizar la validación.

Documento de Autorizo

FACULTAD 2 TELECOMUNICACIONES Y SEGURIDAD INFORMATICA.

CURSO 2008-2009

Ciudad Habana, 15 de Marzo de 2009.

“Año 51 de la Revolución”

Autorizo para el empleo de métricas para evaluar la calidad.

Yo _____ Ingeniero en Ciencias Informáticas y líder del proyecto _____, autorizo a las estudiantes Carmen Rosa García Silva y Lisy Yera Muñoz que optan por el título de Ingenieros en Ciencias Informáticas a realizar la evaluación de la calidad en el proyecto que dirijo y a utilizar el resultado obtenido en la investigación de su trabajo de diploma.

Documento de Solicitud

FACULTAD 2 TELECOMUNICACIONES Y SEGURIDAD INFORMATICA.

CURSO 2008-2009

Ciudad Habana, 15 de Marzo de 2009.

“Año 51 de la Revolución”

Solicitud para el empleo de métricas para evaluar la calidad.

Por medio de este documento se pide que se autorice a las estudiantes Carmen rosa García Silva y Lisy Yera Muñoz de la Facultad#2 a realizar la evaluación de la calidad a el proyecto productivo _____, con el objetivo de que el resultado de estas pruebas formen parte de la investigación de su trabajo de diploma para optar por el título de Ingeniero Informático.

Anexo#2: Entrevista realizada a los Líderes de proyecto.

Universidad de las Ciencias Informáticas

Encuesta a realizar al Líder de Proyecto

Polo Telecomunicaciones

Nombre del Proyecto del cual es líder:

¿Qué tipo de proyecto (Aplicación) es la que desarrollan?:

¿Qué tipo de producto crean?:

¿Qué métricas utilizan en el proyecto?:

¿Cómo miden o evalúan el trabajo en el proyecto, o la calidad de los procesos, o la calidad del producto?:

¿Qué aspectos, indicadores, utilizan o tienen en cuenta para determinar la calidad de las tres p de arriba?:

¿Qué aspectos, indicadores, son los que consideran se deben tener en cuenta para garantizar la calidad de las 3 p?:

Anexo #3: Documento para el registro de los resultados de la aplicación de las métricas (RAM).

RAM

Resultados de Aplicación de Métricas de Software.

Nombre del Proyecto: _____

Nombre del Responsable de la aplicación: _____

Resultados

Nombre de la Métrica: _____

Fecha de aplicación: ___ / ___ / _____

Valor Obtenido: _____ Bien: _____ Regular: _____ Mal: _____

Análisis de las consecuencias del valor obtenido:

Firma del Líder de Proyecto

Firma del Responsable de aplicación

Glosario de Términos

A continuación se presenta por orden alfabético algunos de los términos utilizados en la realización de este documento y que puedan crear confusión en la comprensión del mismo.

Calidad de software: Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Control de Calidad: Actividades para evaluar la calidad de los productos desarrollados.

Criterios periciales: Se refiere a criterios con certeza y base, criterios razonables y con un índice de aceptación.

Defecto: Indicadores de que un artefacto no funciona como ha sido especificado, o cualquier otra característica indeseable. Cualquier requerimiento, elemento de diseño o de implementación que si no es cambiado, causará un diseño, implementación, prueba, uso, o mantenimiento inapropiado del producto.

Error: Fisura en un producto de trabajo de la ingeniería de software o en la entrega descubierta por los ingenieros antes de que el software sea entregado al usuario final.

Evaluación: Es un proceso de información, interpretación y valoración para la toma de decisiones y para la mejora.

Indicador: Magnitud utilizada para medir o comparar los resultados efectivamente obtenidos en la ejecución de un trabajo. Es, por tanto, un resultado cuantitativo que se mide en porcentaje, tasas y razones para permitir comparaciones.

Módulo: Un módulo es un componente autocontrolado de un sistema, el cual posee una interfaz bien definida hacia otros componentes.

Métrica: Asignación de valor a un atributo de una entidad propia del software, ya sea un producto o un proceso.

Métricas del software: Es una medida cuantitativa del grado en el que un sistema, un componente o un proceso poseen un determinado atributo.

Norma o estándar: Documento aprobado por consenso por un organismo reconocido, que proporciona reglas, pautas y/o características para uso común, con el objeto de obtener un óptimo nivel de resultados en un contexto dado.

Producto: Resultado concreto, observable y medible que surge como consecuencia del proceso, proyecto o experiencia desarrollada.

Proceso: Conjunto de actividades, realizadas en forma secuencial, que realiza una organización, para crear, producir y entregar productos, de tal manera que satisfagan las necesidades de sus clientes.

Proyecto: Es un elemento organizativo a través del cual se gestiona el desarrollo de software, su resultado es un producto.

Pruebas de Aceptación: Es el proceso de comparar el producto final con las necesidades actuales de los usuarios finales. Generalmente es realizada por los usuarios finales o clientes e involucra correr y operar el software a modo de producción por un período de tiempo especificado previamente. Si el software se desarrolla bajo contrato, las pruebas de aceptación las realiza el cliente y el criterio de aceptación se define en el contrato.

Pruebas de Caja Blanca: Es un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo. Consiste en diseñar un plan de pruebas en las que se vaya ejecutando sistemáticamente el código hasta que haya corrido todo o la gran mayoría de él.

Pruebas de Caja Negra: Las pruebas de caja negra son aquellas que se enfocan directamente en el exterior del módulo, sin importar el código, son pruebas funcionales en las que se trata de encontrar fallas en las que este no se atiene a su especificación, como ser interfaz con el usuario, apariencia de los menús, control de las teclas.

Requisitos: Son las funciones, servicios y restricciones operativas del sistema.

Requisitos funcionales: Son aquellos que describen lo que debe hacer el sistema.

Requisitos no funcionales: Son aquellos que describen las facilidades que debe proporcionar el sistema.

Software: Se refiere a los programas y datos almacenados en un ordenador.

Usuario: Persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público o privado, empresarial o profesional.

Validación: Confirmación mediante el suministro de evidencia objetiva de que se han cumplido los requisitos para una utilización o aplicación específica prevista.

Siglas

AGCR: Agilidad en la Gestión de Cambio de los Requisitos

CU: Casos de Uso

CCUD: Control de los casos de uso diseñados.

CER: Calidad de la especificación de requisitos.

EED: Eficacia de la Eliminación de Defectos

Em: Esfuerzo del modulo

EP: Esfuerzo del proyecto

ISO: International Organization for Standardization – Organización Internacional para la Estandarización

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos o The Institute of Electrical and Electronics Engineer.

IMS: índice de madurez del software

KLOC: 1000 líneas de código

LDC: Líneas de código fuente

PP: Productividad del proyecto

PFNA: Puntos de función no ajustados

Q: Especificidad

RFA: Factor de ajuste

UCI: Universidad de las Ciencias Informáticas