

Universidad de las Ciencias Informáticas

Facultad 3



**Título: Propuesta de métodos de medición para
determinar el tamaño de un producto
de Software.**

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

Autores: Lisset Roche Reyes

Tatiana Báez Hechavarría

Tutor: Msc. Michael González Jorrín

Co-tutor: Ing. Nadia Porro Lugo

17 junio de 2008



"Educar es depositar en cada hombre toda la obra humana que le ha antecedido: es hacer a cada hombre resumen del mundo viviente, hasta el día en que vive: es ponerlo a nivel de su tiempo, para que flote sobre él, y no dejarlo debajo de su tiempo, con lo que podrá salir a flote; es preparar al hombre para la vida."

José Martí

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 _____.

Firma del Autor

Tatiana Báez Hechavarría

Firma del Autor

Lisset Roche Reyes

Firma del Tutor

Msc. Michael González Jorrín

Firma del Co-tutor

Ing. Nadia Porro Lugo

DATOS DE CONTACTOS

Síntesis del tutor Msc. Michael González Jorrín

Profesión: Ingeniero Informático

Cargo: Especialista Superior de Calidad

Años de graduado: 8

Síntesis del co-tutor Ing. Nadia Porro Lugo

Profesión: Ingeniero de Ciencias Informáticas

Cargo: Asesora de Calidad

Años de graduado: Recién graduado.

AGRADECIMIENTOS

Agradezco antes que a nadie a mi mamá Arlenis Hechavarría, por ser tan maravillosa, por tener las palabras precisas para darme consejos cada vez que lo necesité, por alentarme en todo momento a seguir adelante hasta conseguir mis sueños, por ser la mejor amiga, madre y persona que he conocido, y por la confianza que siempre ha tenido en mí. Te quiero mamita, eres mi vida, y este triunfo, es tuyo también.

A mi papá Fidel, por siempre estar conmigo, darme tan buenos consejos, quererme como lo hace, y estar siempre que lo necesito. Quiero agradecerte por tu preocupación y esfuerzo durante estos cinco largos años, nunca me has dejado sola, y es por eso que quiero decirte que este no es solo mi éxito, sino tuyo también, sin tu ayuda no lo habría logrado.

A mami Sara, por ser la persona que me enseñó a dar mis primeros pasos, por siempre haberse sentido orgullosa de mí, por su amor, sus palabras, su ternura, y su deseo de que pudiera alcanzar este sueño.

A mis tías Alina y Carmencita, por su apoyo incondicional, sus consejos, por entenderme y ayudarme siempre, por estar siempre para mí, las adoro y siempre las tengo presente.

A mis mejores amigos, Teudis, Yunelis, Annia, Elifito, Ley sin ustedes no se que habría sido de mí, gracias por estar conmigo en las buenas y en las malas, por creer en mí, y por siempre tener las palabras precisas para esos momentos donde sentí que no podía continuar. Los adoro a todos y siempre serán mis hermanitos del alma.

A mis amigos del aula por haberme apoyado, y por haber compartido momentos de risas, tristeza, en fin por ser las personas tan especiales que son.

Le agradezco infinitamente a Michael González Jorrín, mi tutor, por su paciencia y entrega.

En general les agradezco a todas las personas que de una forma u otra hizo posible que lograré hacer mi sueño realidad.

Tatiana Báez Hechavarría.

Primeramente quiero agradecerle a las dos personas más importantes en mi vida y que si no fuera por ellos no estuviera aquí, a mi madre Dunia Bárbara Reyes y mi padre Jorge Luis Roche, muchas gracias a ellos por estar conmigo en todo momento, gracias por ayudarme a superar los baches por los que tuve que pasar en estos 5 años y siempre tener las palabras perfectas para yo poder seguir adelante, este no es solamente el fruto de mi trabajo, también es el de ustedes, los quiero con mi alma.

A mis abuelos y mis tías, por siempre consentirme, aconsejarme, siempre estar pendientes de mí.

A mis amigas Ary y Daimi por ayudarme tanto y aguantarme en estos años, y por siempre estar cuando más las necesitaba, Ary gracias por tus palabras tan reconfortantes para mí en estos últimos meses tan difíciles.

A mis amigos Mailén, Susana, Luis, Anaelis, Yúnior y Leisan por haberme apoyado siempre, compartir conmigo en las buenas y en las malas, y por confiar en mí.

A mi tutor Michael González Jorrín por su ayuda, y confiar en nosotros.

A la Revolución y a nuestro Fidel, por ponerme en las manos esta oportunidad tan maravillosa y permitirme hacer mis sueños realidad.

A todo aquel que a puesto su pedacito en el transcurso de mi carrera y de mi vida.

Muchas Gracias.

Lisset Roche Reyes

DEDICATORIA

Dedico este trabajo a mis padres Arlenis y Fidel, a mami Sara, a mi familia y amigos, por ser mi inspiración en la vida, y mi razón de ser.

Tatiana

Dedico este trabajo a mi mamita y a mi papito por ser mi inspiración, a mi hermanito por siempre tener su amor y cariño, a mi familia y amigos, por haberme apoyado durante toda mi vida y ayudarme a convertir mis sueños en realidad.

Lisset

RESUMEN

El presente trabajo tiene como objetivo fundamental proponer métodos de medición para determinar el tamaño de los diferentes productos resultantes del desarrollo de Software, ajustado a las necesidades de las pruebas y revisiones en la Facultad 3. Se presenta un estudio de los aspectos generales de las mediciones del Software, y se describen los métodos de medición de tamaño del Software, así como las actividades fundamentales que se llevan a cabo en el proceso de prueba y en el proceso de medición. El presente estudio se realiza sobre una selección de productos clasificados como entregables en la fase de pruebas y revisiones del software, para lograr la liberación de los mismos. Ajustadas a cada clase se proponen los métodos para determinar la dimensión de cada producto del desarrollo de software en base al esfuerzo necesario para comprobar su estado de calidad. Se realiza la validación de la propuesta y análisis sus resultados mediante el método de Estudio de Casos.

PALABRAS CLAVE

Medición, pruebas de software, proceso de prueba, proceso de medición.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	8
1.1 Introducción.....	8
1.2 Situación actual de la industria del software respecto a la medición del Software y la Calidad de los proyectos.....	8
1.2.1 En Cuba.....	10
1.3 Entrevista.....	11
1.3.1. Evaluación de las entrevistas.....	11
1.4 ¿Qué es la medición?.....	12
1.4.1. Medida.....	13
1.4.1.1. Clasificación de las medidas.....	13
1.4.2. Proceso de medición.....	14
1.4.3. Tamaño del software.....	15
1.5 Producto de Software.....	17
1.6 Productos de Prueba.....	17
1.6.1. Modelo Entidad-Relación.....	17
1.6.2. Diagramas de Casos de Uso.....	19
1.6.3. Diagrama de Despliegue.....	19
1.6.4. Modelo de Análisis.....	19
1.6.5 Modelo del Diseño.....	21
1.7 Calidad del software.....	21
1.8 Gestión de la Calidad.....	23
1.9 Gestión de Proyecto.....	23
1.10 Estimación.....	24
1.11 Método.....	24
1.12 Métricas de software.....	25
1.12.1. Propiedades de las métricas.....	27
1.12.2. Métricas sobre el producto.....	27
1.13 Métricas para la estimación.....	29
1.13.1 Documentación.....	29
1.13.3 Tamaño medio de operación (TOMedio).....	29
1.13.4 Basados en Casos de Uso.....	30
1.13.5. Método de análisis de puntos de función IFPUG.....	33
1.13.6. Método de análisis puntos de función MK II – FPA.....	34
1.13.7. Método de análisis puntos de función FPA.....	34
1.13.8. Líneas de Código.....	35
1.14 Proceso de Pruebas.....	36
1.14.1 Artefactos.....	37

1.15 Normas de Calidad.....	37
1.15.1 ISO/IEC ISO 12119:1994. Paquetes de Software.....	37
1.15.2 ISO/IEC 12207:1995 Procesos del ciclo de vida del software.....	38
1.16 Conclusiones.....	39
CAPÍTULO II. PROPUESTA DE SOLUCIÓN.....	40
2.1 Introducción.....	40
2.2 ¿Por qué es importante medir el software?.....	40
2.3 Pasos para la obtención de los métodos de medición de los productos de prueba.....	41
2.3.1 Identificación y Decantación.....	41
2.3.2 Clasificación.....	41
2.3.3 Estudio y Recopilación.....	41
2.3.4 Definición de características generales.....	41
2.3.5 Elaboración de las fórmulas.....	42
2.3.5.1 Obtención de variables.....	42
2.3.5.2 Aplicación piloto.....	42
2.3.5.3 Validación.....	42
2.3.6 Formalizar la propuesta.....	43
2.3.7 Aplicar los métodos propuestos.....	43
2.4 Aplicación de los pasos propuestos a los productos de prueba de los proyectos productivos de la Facultad 3.....	43
2.4.1 Identificación y Decantación.....	43
2.4.2 Clasificación.....	48
2.4.3 Estudio y Recopilación.....	50
2.4.4 Definición de características generales.....	54
2.4.4.1 Modelo Entidad Relación.....	54
2.4.4.2 Diagrama de Despliegue.....	55
2.4.4.3 Modelo de Análisis.....	55
2.4.4.4 Modelo de Diseño.....	57
2.4.4.5 Diagrama de Caso de Uso del Negocio.....	57
2.4.4.6 Casos de Uso del Sistema.....	58
2.4.4.7 Documentos.....	58
2.4.5 Elaboración de las fórmulas.....	59
2.4.5.1 Obtención de las variables.....	59
2.4.5.2 Aplicación piloto.....	60
2.4.5.3 Validación.....	61
2.4.6 Formalizar la propuesta.....	61
2.4.7 Aplicar los métodos propuestos.....	61
2.5 Métodos propuestos.....	62
2.5.1 Método para la medición de código del Software.....	62
2.5.2 Método para la medición de los gráficos del Software.....	63

2.5.2.1.	Medición del Modelo Entidad-Relación	64
2.5.2.2.	Medición del Diagrama de Despliegue.	65
2.5.2.3.	Medición del Modelo del Análisis.	65
2.5.2.4.	Medición del Modelo del Diseño.	67
2.5.2.5.	Medición de los Diagramas de Casos de Uso del Negocio.	67
2.5.2.6.	Medición de los Casos de Uso del Sistema.	68
2.5.3	Medición de los Documentos del Software.	69
2.6	Conclusiones.	70
CAPITULO III. EVALUACIÓN DE LA PROPUESTA.		71
3.1.	Introducción.	71
3.2.	Método de Estudio de Caso.	71
3.3.	Selección de Conceptos.	72
3.4.	Selección de Variables.	72
3.5.	Clasificación de artefactos y construcción de casos de estudio.	76
3.6.	¿Cómo se realiza la validación de la propuesta?	78
3.7.	Descripción de los estudios de casos.	81
3.7.1.	Caso 1.	81
3.7.2.	Caso 2.	89
3.8.	Análisis de resultado de los Estudios de Casos.	91
3.8.1.	Estudio del Caso 1.	91
3.8.2.	Estudio del Caso 2.	95
3.9.	Conclusiones.	98
CONCLUSIONES		99
RECOMENDACIONES		100
BIBLIOGRAFÍA		101
ANEXOS		104
GLOSARIO		105

INTRODUCCIÓN

El desarrollo tecnológico mundial de las últimas décadas del pasado siglo, y los inicios del presente han estado marcados por el nivel alcanzado en la rama de la informática, la invención de hardware y la producción de programas para computadoras. El software se ha convertido en un tema crítico en la sociedad moderna mundial, y un factor fundamental para el desarrollo exitoso de estos productos, está dado por la calidad de los mismos, juega un papel fundamental en el nivel de satisfacción de los clientes.

La demanda de la calidad se intensifica dado la creciente dependencia de la sociedad actual hacia los procesos informatizados. En el presente, la calidad ha sido llevada al centro del proceso de desarrollo del software. La calidad ya no es un factor de ventaja en el mercado, sino que se ha convertido en una condición necesaria si se quiere producir y competir con éxito.

Mirándolo desde el punto de vista técnico para la calidad de los productos, es necesario identificar las mediciones y los criterios que serán utilizados para identificar el nivel deseado de la misma y determinar si se está alcanzando. Las mediciones describen el método para capturar los datos que serán utilizados para evaluar la calidad, mientras que los criterios definen el nivel o el punto en el cual el producto logra la calidad aceptable (o inaceptable) (RUP 2003).

En la actualidad, la comunidad de Ingeniería del Software reconoce que los diferentes software poseen características que los diferencian. Estas características se deben al tamaño y complejidad de las aplicaciones, el carácter multidisciplinar del equipo de desarrollo, la tasa apresurada de entrega del proyecto, etc. Éstas características hacen que los procesos, modelos y métricas existentes para evaluar la calidad tengan que ser adaptados para considerar los cambios impuestos por las nuevas tecnologías.

Todo proyecto debe partir con un buen plan, la planificación es una tarea a la cual no se le da la importancia requerida. Uno de los aspectos que dificulta la labor de los jefes de proyecto en torno a ésta, es la difícil tarea de realizar una estimación de costos y plazos realista. El manejador de costo principal para un proyecto de desarrollo de software es sin duda el tamaño del producto. La medida del tamaño debe ser tal que esté en relación directa con el esfuerzo de desarrollo, por lo que las métricas de tamaño tratan de considerar todos los aspectos que influyen en el costo, como tecnología, tipos de recursos y complejidad.

En el informe presentado por el Ministerio de Relaciones Exteriores de Cuba en la Cumbre Mundial para la Sociedad de la Información en el 2004 plantea:

“La Industria Cubana del Software (InCuSoft) está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. La promoción de la industria cubana del software en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. El continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación”. (GONZALEZ 2004).

La creación del Ministerio de la Informática y las Comunicaciones, la creación de la Universidad de las Ciencias Informáticas (UCI), la revitalización y potenciación de los tecnológicos de informática, la creación de los IPI, las Mini-UCI y la Universalización de la Carrera de Ingeniería Informática, son ejemplos ilustrativos del avance significativo que está alcanzando Cuba en la industria del software. Para lograr el avance en esta rama tan importante de la economía, como lo constituye actualmente la informática, es importante desarrollar la calidad en los trabajos que se realizan, es por eso que es fundamental tomar en cuenta lo que significa el término medición del Software, y las ventajas que el mismo plantea.

Las mediciones permiten cuantificar tanto el proceso como el producto. Proporcionan la visión del desempeño del proceso permitiendo: desarrollar perfiles de los datos de los proyectos anteriores que se pueden utilizar para la planificación y mejora del proceso; analizar un proceso para determinar como mejorarlo y determinar la eficacia de modificaciones en el proceso.

El área de las mediciones de software, es una de las áreas en la ingeniería de software donde se ha investigado desde hace más de 30 años, esta área se conoce también como métricas de software. Existe una confusión en la utilización de los términos métricas de software y mediciones de software, en la literatura los términos métrica, medida o medición son usados como sinónimos. (Zuse 1998)

Las mediciones de software son usadas para medir atributos específicos de un producto de software o del proceso de desarrollo de software.

Se usan fundamentalmente para:

- Seguir el progreso de los proyectos.
- Ayudar a comprender cuando se ha alcanzado un estado deseado de calidad.
- Analizar los defectos.
- Validar experimentalmente las mejores prácticas.
- Obtener las bases para la estimación.
- Determinar la complejidad (relativa).

Esta investigación se centrará en la medición del tamaño de un producto de Software, ya que no existe en la Universidad un método para determinar el tamaño de los diferentes productos resultantes del desarrollo de Software, permitirá además obtener las bases para la estimación del esfuerzo en las pruebas y revisiones de los mismos así como determinar su complejidad relativa.

Primeramente se debe conocer como aplicar los conceptos de Medición del tamaño de un producto a los procesos de Pruebas de Software en la Universidad, de tal manera que permita mejorarlos a partir de las experiencias acumuladas, para esto se realiza un estudio de los principales métodos que se utilizan en esta etapa del desarrollo del software, que ayudan a comprender la problemática real en la UCI.

Es importante investigar como se llevan a cabo los procesos de Pruebas de Software en la Facultad de donde llega a lo siguiente:

- Situación problemática:

A pesar de que hasta ahora el trabajo del laboratorio de Calidad de la Facultad 3 de la Universidad de Ciencias Informáticas (UCI) ha generado buenos resultados, existen numerosos puntos que atentan contra la eficiencia del mismo:

- El hecho de que no se mide el tamaño de los diferentes productos resultantes del desarrollo del Software.
- Como consecuencia, las estimaciones de costes, esfuerzos, tiempos de desarrollo y recursos para los proyectos no son efectivas o no se realizan.

- No están clasificados en composición o tamaño los productos que reciben los probadores para ser liberados.
- Presentan problemas a la hora de capacitar al personal encargado del proceso de pruebas, pues se utilizan los métodos tradicionales de formación, como son los cursos presénciales, conferencias. En general todos los métodos utilizados presentan los siguientes inconvenientes:
 - No se cuenta con profesores capacitados en la materia o disponibles en el momento requerido, lo que afecta significativamente los cronogramas, trayendo consigo el retraso en la entrega de los productos finales.
 - La mala capacitación del personal incide en que los probadores no cuenten con la experiencia suficiente para enfrentarse a las revisiones de la fase de Prueba, lo que conlleva a que los productos muchas veces no sean liberados con la calidad requerida.

Estos últimos aspectos se ven influido por el desconocimiento de los aspectos a revisar dentro de cada artefacto, dependiente de las propiedades de los elementos contenidos en ellos así como los aspectos contenidos en reglas y lineamientos que son revisados o tenidos en cuenta mediante el uso de listas de comprobación o chequeo.

Es de vital importancia lograr una mayor agilidad y veracidad en el proceso de revisión, además de una correcta clasificación y medición del tamaño de los productos a probar dado el origen y estructura de los mismos.

Diseño teórico:

- Problema en cuestión:

¿Cómo contribuir al desarrollo de las métricas de esfuerzo del Software mejorando los métodos de medición del tamaño de un producto de Software en la Facultad 3?

- Objeto de Estudio:

Métricas de esfuerzo del desarrollo de Software.

- Campo de Acción:

Métodos de medición del tamaño de un producto del desarrollo de Software.

- Objetivo General:

Contribuir al desarrollo de las métricas de esfuerzo del Software mejorando los métodos de medición del tamaño de un producto de Software en la Facultad 3.

- Tareas:

1. Selección y revisión de la bibliografía correspondiente para actualizar los logros y limitaciones existentes sobre la medición del tamaño de un producto de pruebas.
2. Identificación de los principales autores y tendencias, enfatizando en los que más relación tienen con el tema.
3. Evaluación del contenido de la información obtenida.
4. Estudio, identificación y clasificación los diferentes productos de trabajo de un probador en la UCI.
5. Elaboración de un diagnóstico de las tendencias actuales que se llevan a cabo en los proyectos sobre la medición del tamaño de un producto de software en la Universidad.
6. Aplicación de entrevistas a directivos y especialistas en el tema.
7. Estudio de las propiedades que definen a las métricas y a los métodos, para la medición de los productos.
8. Formulación de una propuesta de los métodos más efectivos para la medición del tamaño de un producto de pruebas.
9. Evaluación de los resultados obtenidos aplicando los métodos a un grupo de productos de prueba de estudio.

Estrategia de Investigación:

Será utilizada como estrategia de investigación, la Investigación Descriptiva, pues los conocimientos procedentes del problema son suficientes para definir y plantear una hipótesis a nivel descriptivo. Se puede además representar el problema de una forma clara y caracterizar el fenómeno que se estudia en

sus aspectos externos con el objetivo de establecer lo esencial y más significativo del mismo. En este tipo de investigación es de principal importancia la profundidad teórica del planteamiento investigativo, pues ayuda a comprender el valor científico de los resultados obtenidos.

- Métodos de trabajo científico a utilizar en la tesis:

Método teórico:

-Histórico- Lógico (Hipotético-deductivo).

Se utiliza este método pues se requiere un análisis de la trayectoria completa del fenómeno, develando en cada etapa su desenvolvimiento. Además, el método expresa en forma teórica la esencia del objeto de estudio, explica la historia de su desarrollo, reproduce el objeto en su forma superior y permite unir el estudio de la estructura del objeto de investigación con su concepción histórica.

Método hipotético deductivo.

Este método será utilizado pues partiendo de la hipótesis planteada, se seguirán reglas de deducción para llegar a nuevos conocimientos que posteriormente se someterán a verificaciones empíricas.

Método sistémico.

Con la utilización de este método se estudia el objeto mediante la determinación de sus componentes, así como la relación entre ellos. Es decir, el objeto son las métricas del desarrollo del Software y para poder darle solución al problema, se deben determinar métodos de medición que permitan determinar el tamaño de los productos que constituyen entregables y que son generados durante toda la etapa del desarrollo del Software.

-Métodos particulares:

-Entrevista.

-Encuesta.

De los métodos empíricos que existen fue utilizado para el desarrollo y entendimiento del problema, la entrevista. Su uso permite obtener conocimiento cualitativo del tema que se abarca en la investigación.

El documento tesis está compuesto por 3 capítulos. En el primer capítulo se hace un análisis para ver como se comporta el tema de la medición del Software en el mundo, Cuba, y la UCI. Se tratan además conceptos importantes relacionados con la calidad de los productos, el proceso de prueba, métodos para determinar la medida del Software, entre otros. En el segundo capítulo se desarrollan los métodos existentes actualmente en el mundo para medir el Software, se seleccionan los productos a medir, se clasifican los mismos, y se hace la propuesta de solución para determinar el tamaño de los productos anteriormente seleccionados. Y en el tercer capítulo se evalúa la propuesta de solución planteada en el capítulo 2, mediante la metodología escogida que en este caso fue la de Casos de Estudio, se toman 2 casos de estudio para la evaluación de la propuesta y se analiza el resultado del estudio de los casos.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

“En la carrera por la calidad no hay línea de meta” (Kearns)

En el desarrollo del presente capítulo se abordarán una serie de conceptos como son: la calidad del software, Pruebas de Software, medición, proceso de medición, métodos que se llevan a cabo para la estimación del tamaño de un producto de pruebas y Métricas, así como su utilidad y aplicabilidad para la mejora de los procesos de prueba, con lo cual también se verá la garantía que estos ofrecen para la medición del tamaño de un producto. Además se hará un análisis de la situación actual de la medición, y en el caso de la UCI estará respaldada por entrevistas realizadas al proyecto de Calidad de la Facultad 3.

1.2 Situación actual de la industria del software respecto a la medición del Software y la Calidad de los proyectos.

La medición es una herramienta primaria para los administradores de software y de sistemas para asegurar que los productos cumplan los objetivos definidos. Son un elemento clave en toda disciplina de ingeniería bien establecida. Cuando se integran en la totalidad de los procesos de administración de proyectos, ayudan al administrador del mismo a:

- Identificar los riesgos; llevar el seguimiento de problemas específicos; evaluar el impacto de dichos problemas en el costo del proyecto y en el desempeño de objetivos técnicos; desarrollar alternativas de solución y seleccionar el mejor enfoque para corregir los problemas.
- Las mediciones brindan la visión que el administrador del proyecto necesita para la toma de decisiones críticas para el éxito del mismo.(Labdelaoui 2005).

La medición de software se presenta en nuestros días como un medio esencial para realizar las estimaciones oportunas del esfuerzo, tiempo y coste necesarios, para el desarrollo de productos software. Además, tiene el objetivo de permitir llevar a cabo diversos estudios relativos a la productividad y calidad

de software. A lo largo de los últimos veinte años, se han desarrollado varios métodos de medición: los puntos de función, puntos de función extendidos, features points, etc.

Durante muchos años el método de estimación de puntos de función introducido por primera vez por Albrecht, ha sido utilizado ampliamente por un gran número de instituciones y organizaciones, para medir el tamaño del producto software en diversos ámbitos de aplicación.

Siempre se han expuesto las limitaciones de los métodos de medición de tamaño funcional, sobre todo aquellos de primera generación debido a que no producían resultados satisfactorios y adecuados para software del tipo tiempo real, orientado a objetos o científicos. Para remediar esas limitaciones y mejorar el proceso de medición de esos tipos de software, se han desarrollado diversas extensiones así como nuevos métodos.

De manera general, los métodos de medición han sido diseñados para medir un determinado tipo de software. Por tanto, la aplicación de cada método depende particularmente del dominio software o del tipo de desarrollo. Existen por un lado, los sistemas tiempo real, científicos, ingeniería, etc., donde no sólo se deben tomar en cuenta los parámetros convencionales del software sino también parámetros adicionales propios a cada tipo. Por otro lado, existen los sistemas basados en un análisis y diseño orientados a objetos, cuya medición depende mucho de las características de ese tipo de software así como en su caso, de la utilización de la notación UML en todas las etapas de desarrollo.

Por otra parte, la publicación de la norma ISO de medición de tamaño funcional ha permitido la definición de reglas importantes con el objetivo de realizar mediciones correctas únicamente de la funcionalidad del sistema. Esta norma ha influido considerablemente sobre la creación de métodos de medición como COSMIC-FFP, permitiendo así realizar estimaciones simplificadas y genéricas aplicables a un gran número de tipos de software.

En general, la medición persigue tres objetivos fundamentales:

- Ayudar a entender qué ocurre durante el desarrollo y el mantenimiento de los productos de software.
- Permitir controlar lo que ocurre en los proyectos.
- Mejorar los procesos y productos del software.

1.2.1 En Cuba

Aunque en el pasado se reconocía la necesidad de crear software de calidad, no se había hecho un esfuerzo formal para que industria generara productos que diera la oportunidad de competir en el mercado internacional. Afortunadamente, dicha situación ha cambiado; el gobierno en conjunto con la industria, ha iniciado un gran esfuerzo para impulsar la industria del software y no sólo la producción sino además la creación con calidad, para lo que se ha fundado Calisoft (Centro de calidad del software en Cuba). Con este nuevo proyecto se pretende establecer una industria de software competitiva internacionalmente y asegurar su crecimiento a largo plazo.

El país tiene un gran potencial de personal calificado en el área de la informática. Recientemente se ha producido una concentración de dichas entidades en empresas especializadas, sin embargo, a pesar de la potencialidad que poseen, aun existen índices de exportaciones irrisorios. No obstante se posee un mercado interno relativamente fuerte, debido a la identificación de numerosas áreas de posible informatización.

Uno de los aspectos que atentan contra la calidad de los diferentes proyectos desarrollados en el país, es el tema de las estimaciones, actualmente no existen métodos efectivos para determinar el tamaño del Software, ni la cantidad de personas y tiempo que se deben utilizar para llevar a cabo el proceso de pruebas. Este problema está dado en gran medida porque no se conocen métodos que determinen el tamaño de los diferentes productos de pruebas, y como se conoce, en el presente este es el primer paso para realizar una estimación más certera del tiempo de revisión del Software en general.

A pesar de todos los avances e intentos por mejorar, en las empresas productoras de software del país no se han aplicado procesos de certificación ni tampoco de auto-evaluación que permitan valorar el estado de la organización en cuanto a la madurez de sus procesos. Por lo que el reto es trabajar para lograr Organizaciones de Software maduras, caracterizadas por tener una gran capacidad para administrar sus procesos, con personal comprometido, y que cumplan con el tiempo establecido para alcanzar todos los requisitos del proyecto. Si estas condiciones se alcanzan permitiría satisfacer el mercado interno, las empresas pudieran certificarse formalmente para aumentar la capacidad de adquisición de clientes provenientes del mercado internacional, con lo que se incrementarán las exportaciones con concepto de

venta de software. Además, cuando se implante un sistema de calidad bien diseñado, la empresa estará en condiciones de obtener una certificación internacional, pues el sistema permitirá concretar los principios exigidos por los modelos, al mismo tiempo que logrará hacer explícito el conocimiento acumulado por los especialistas.

De manera general se puede decir que las empresas aún hoy no cuentan con métodos que le permitan obtener la medición del tamaño de productos, trayendo esto consigo el incumplimiento de las entregas de los proyectos a los usuarios finales, pues no se conoce una estimación del tiempo ni el esfuerzo que deben utilizar en el proceso de prueba.

1.3 Entrevista.

Para poder adentrarse en la problemática de las mediciones en la UCI específicamente en la Facultad 3, se realiza una entrevista a algunos de los que trabajan en el Laboratorio de Calidad de la dicha facultad. Ver (Anexo 1)

1.3.1. Evaluación de las entrevistas.

La Universidad de las Ciencias Informática (UCI) no está aislada del problema de las mediciones. Hoy la mayoría de los desarrolladores de software en la UCI: líderes de proyectos, programadores, diseñadores, entre otros, no planean ni registran su trabajo y no miden y administran la calidad de los productos. El desconocimiento de las métricas es un factor que pesa en la evaluación del software. Entrevistas realizadas al proyecto de Calidad de la Facultad 3 demuestra que “No se utilizan los métodos de medición de los productos del Software porque el nivel de conocimiento y desarrollo no lo permite, debido a la inestabilidad existente”. Las estimaciones que se realiza en la mayoría de los casos no utilizan los métodos reconocidos a nivel mundial para determinar el tamaño del software, sino que se basan en la experiencia de los expertos, que en este caso es el jefe del proyecto.

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 los métodos de medición por parte del equipo de trabajo hace que establezcan un rechazo hacia ellos.

En un intento por esclarecer la importancia de los métodos de medición muchos desarrolladores oponen resistencia, no entienden para que sirven y se preguntan porque es tan necesario medir el proceso de desarrollo de software y el producto que produce. Los rigores del trabajo diario de un proyecto del software no dejan mucho tiempo para pensar en estrategias. Los líderes del proyecto de software están más preocupados por otros aspectos (aunque igualmente importantes): desarrollo de estimaciones significativas del proyecto, terminar el producto a tiempo y dejan a un lado la calidad del producto, que está dada en gran medida por el tiempo y el esfuerzo que se dedique a revisar cada uno de los productos resultantes del desarrollo del software. Mediante el uso de la medición para establecer una línea base del proyecto, cada uno de estos asuntos se hace más fácil de manejar. La línea base sirve como un lineamiento para la estimación. Además, la recopilación de métodos de medición permite a una entidad organizar mejor su trabajo, ya que la medición será la base para realizar estimaciones de tiempo y esfuerzo más exactas, lo que contribuye a que los productos sean revisados de acuerdo al tamaño que realmente representan dentro del Software desarrollado.

1.4 ¿Qué es la medición?

Es el resultado de la comparación cuantitativa de una variable de un fenómeno o situación con un patrón pre-establecido, el cual debe ser estable, reproducible y universalmente conocido y aceptado.

La operación de la medición consiste en dar valores a los fenómenos que interesan en el marco de un planteamiento geográfico. Por una parte, sirve inicialmente para caracterizar los atributos de los objetos que se estudian y entonces se emplea al comienzo de todo tratamiento. Por otra parte, interviene al final para caracterizar formas espaciales, describir la naturaleza y la intensidad de relaciones, calificar semejanzas, etc. En el primer caso la medición está inserta en la fase de adquisición de los datos e informaciones necesarios para tratar la cuestión estudiada. Estas informaciones provienen, ya sea de fuentes estadísticas de organismos públicos o privados, ya sea de encuestas (relevamientos de terreno, sondeos, etc.). En el segundo caso, se trata de la ayuda de mediciones apropiadas, de rendir cuenta de resultados originados en tratamientos, estadísticos u otros. (Sanders 2004)

La medición es considerada como el "proceso por el cual se asignan números o símbolos a atributos de entidades del mundo real de tal forma que los describa de acuerdo con reglas claramente definidas" (Fenton 1997)

La medición es el acto de determinar una medida. (PRESSMAN 1998)

En general la medición es una comparación cuantitativa de variables, que se utiliza para comparar una cantidad con su respectiva unidad, con el fin de saber cuantas veces la segunda está contenida en la primera

1.4.1. Medida.

La medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. (PRESSMAN 1998)

1.4.1.1. Clasificación de las medidas.

Medidas Directas

Las medidas directas son aquellas en donde el instrumento de medida mide directamente la magnitud desconocida. Como ejemplo de medida directa se tiene la medida de una distancia concreta con un instrumento de estación total. Otros ejemplos son: la medida de la longitud de un lápiz concreto con una regla graduada, o la medida de un ángulo con un teodolito o con un instrumento de estación total.

Ejemplo de Medidas Directas:

- Coste.
- Esfuerzo humano.
- Líneas de código.
- Velocidad de ejecución.
- Tamaño de memoria.
- Número de defectos.

Medidas Indirectas:

Las medidas indirectas son aquellas que se obtienen al aplicar a unas determinadas medidas directas, una función matemática que relaciona la cantidad de interés con la magnitud desconocida. Un ejemplo sencillo es la determinación de las coordenadas de una estación a partir de la medida de unos ángulos y distancias. Después a partir de estas coordenadas se pueden medir otros ángulos y distancias que no fueron medidos directamente. Durante este procedimiento los errores de las medidas directas se transmiten a las cantidades indirectas medidas. (Topografía 2005)

Ejemplos de Medidas Indirectas:

- Calidad.
- Complejidad.
- Eficiencia.
- Fiabilidad.
- Facilidad de uso.

1.4.2. Proceso de medición.

Todo proceso de medición del software tiene como objetivo fundamental satisfacer necesidades de información a partir de las cuales se deben identificar las entidades y los atributos que deben ser medidos.

El proceso de medición, se caracteriza en cinco actividades (PRESSMAN 2002)

- **Formulación:** Obtención de medidas y métricas del software apropiadas para la presentación del software en cuestión.
- **Colección:** Mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- **Análisis:** 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 presentación.
- Retroalimentación: Recomendaciones obtenidas de la interpretación de métricas y técnicas transmitidas al equipo de desarrollo de software.

1.4.3. Tamaño del software

Una de las medidas fundamentales en el ámbito del software es la del tamaño, pues constituye una medida base para la estimación del costo, esfuerzo y productividad. Por lo general se habla de tamaño de un sistema basado en líneas de código o en funcionalidad. La medida del tamaño en líneas de código (LOC) o miles de líneas de código (KLOC) es una medida directa mientras que el tamaño en puntos de función es una medida indirecta.

Para una mayor comprensión:

Al calcular las líneas de código se tiene en cuenta entre otras cosas contabilizar cada línea nueva o modificada, nunca los comentarios o pseudos-código, además el código generado por macros o includes solo se considera una vez. (PRESSMAN 1998)

Los puntos de función de un sistema software se calculan teniendo en cuenta:

- Entradas al sistema.
- Salidas del sistema.
- Consultas.
- Archivos lógicos del sistema.
- Archivos de interfaz.

Debe computarse cuántas ocurrencias de cada parámetro contiene un sistema, calificándolos según su complejidad en alta, media y baja. Cada parámetro, para una complejidad dada tiene un determinado peso, ese peso son los puntos de función asignados a ese parámetro. Por ejemplo, una entrada de

complejidad alta equivale a 6 puntos de función. Luego de este proceso los puntos de función se ajustan de acuerdo a las características generales del sistema. La decisión es un tema muy polémico.

Las estimaciones y métricas basadas en líneas de código son a menudo el blanco de muchas críticas, pues se plantea que es dependiente del lenguaje de programación utilizado para llevar a cabo el proyecto y de los programadores involucrados en el mismo (pues cada funcionalidad podría ser programada por distintos programadores de formas diferentes y empleando un número mayor o menor de líneas de código), también se alega que no es una buena opción para una estimación temprana puesto que en las primeras etapas del proyecto aun no existe código escrito, de ahí que muchos prefieren la utilización de puntos de función, estos son independientes del lenguaje, herramientas o metodologías que se vaya a utilizar.

Sin embargo esta técnica también tiene sus limitaciones, pues aún cuando se trabaja sobre la base de una misma especificación de requisitos y un mismo método de cálculo de puntos de función la asignación de complejidad que el método exige para cada ítem (entrada, salida, consulta, archivo lógico, archivo de interfaz), es un tanto subjetiva y normalmente difiere de un desarrollador a otro, también puede ocurrir que no se tenga claro cuándo se está en presencia de cada uno de estos elementos. Por estos motivos el método de cálculo por puntos de función puede arrojar resultados diversos dependiendo de quién lo ejecute.

También puede mencionarse los puntos de características que utilizan la lógica de los Puntos de Función para estimar el tamaño del software, actualmente es usado con mucho éxito en software del tipo CAD, sistemas embebidos y sistemas en tiempo real, que añaden un componente adicional a los cinco ítems ya mencionados, este componente se denomina algoritmos complejos. Las diferencias entre los puntos de función y los puntos de características, son básicamente que cuando el número de algoritmos de una aplicación es mucho mayor que el de los archivos, los puntos de características generan estimaciones más precisas que los puntos de función. A la inversa, si existe un número de archivos considerablemente mayor al de algoritmos (situación muy común en los sistemas de gestión), la cuenta de puntos de características será mucho menos confiable que la de puntos de función.

Los métodos o modelos para la estimación del tamaño en cada proyecto, dependerá del criterio y conocimiento del líder de proyecto, la experiencia en trabajos anteriores y la naturaleza del proyecto y del producto que se intenta medir. (HURTADO 2007)

1.5 Producto de Software.

Conjunto de programas de computadora, procedimientos y posiblemente documentación y datos asociados. (C Génova 1995)

1.6 Productos de Prueba.

Son todos los entregables, generados después de finalizada la fase de Análisis & Diseño e Implementación de un proyecto, lo que permite pasar a la fase de las pruebas de liberación. Algunos de estos productos se verán en los acápites a continuación.

1.6.1. Modelo Entidad-Relación.

El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas. (Andrés 2002)

Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Las entidades se representan gráficamente mediante rectángulos y su nombre aparece en el interior. Un nombre de entidad sólo puede aparecer una vez en el esquema conceptual.

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil. (Andrés 2002)

Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las relaciones se representan gráficamente mediante rombos y su nombre aparece en el interior. (Andrés 2002)

Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Toda la información extensiva es portada por los atributos. Gráficamente, se representan mediante bolitas que cuelgan de las entidades o relaciones a las que pertenecen.

Cada atributo tiene un conjunto de valores asociados denominado dominio. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio.

Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

- Se parte de una descripción textual del problema o sistema de información a automatizar (los requisitos).
- Se hace una lista de los sustantivos y verbos que aparecen.
- Los sustantivos son posibles entidades o atributos.
- Los verbos son posibles relaciones.
- Analizando las frases se determina la cardinalidad de las relaciones y otros detalles.
- Se elabora el diagrama (o diagramas) entidad-relación.
- Se completa el modelo con listas de atributos y una descripción de otras restricciones que no se pueden reflejar en el diagrama. (Andrés 2002)

1.6.2. Diagramas de Casos de Uso

Un diagrama de casos de uso es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un diagrama *de Casos de Uso*, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

Un actor es una entidad que utiliza alguno de los casos de uso del sistema, representa un rol que alguien o algo podría desempeñar y no un alguien o algo específico. (Vilas 2001)

1.6.3. Diagrama de Despliegue

Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos). Estarán formados por instancias de los componentes software que representan manifestaciones del código en tiempo de ejecución (los componentes que sólo sean utilizados en tiempo de compilación deben mostrarse en el diagrama de componentes).

Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo puede contener instancias de componentes *software*, objetos, procesos (caso particular de un objeto). En general un nodo será una unidad de computación de algún tipo, desde un sensor a un *mainframe*. Las instancias de componentes *software* pueden estar unidas por relaciones de dependencia, posiblemente a interfaces (ya que un componente puede tener más de una interfaz). (Vilas 2001)

1.6.4. Modelo de Análisis

En la construcción del modelo de análisis se tienen que identificar las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellos. Con esta información se construye el

Diagrama de clases del análisis, que por lo general se descompone para agrupar las clases en paquetes. Esta descomposición tiene impacto por lo general en el diseño e implementación de la solución.

Clases de análisis:

Se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP propone clasificar a las clases en: Control, Interfaz y Entidad.(PRESSMAN 2002)

- Clase Entidad: Modelan información que posee larga vida y que es a menudo persistente.
- Clase Control: Coordinan la realización de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan funcionalidad del caso de uso.
- Clase Interfaz: Modelan la interacción entre el sistema y sus actores.

Pasos para construir el Diagrama de clases del análisis

1. Crear una lista de conceptos significativos e interesantes del dominio

- Utilice los nombres existentes en el dominio.
- Excluya las características irrelevantes.
- No agregue cosas que no existan

2. Identificar las asociaciones.

- Asociación: Relación entre dos conceptos que indican alguna conexión significativa e interesante entre ellos.
- Las asociaciones que vale la pena describir son aquellas que incluyen el conocimiento de una relación que ha de preservarse durante algún tiempo.

3. Identificar atributos.

Incluir solo los atributos que, según los requerimientos, son necesarios recordar para que se ejecute el caso de uso.(UCI)

Realización de los casos de uso del análisis:

La realización de los casos de uso en el análisis es una colaboración que describe cómo se lleva a cabo y ejecuta un caso de uso determinado en término de las clases del análisis y de sus objetos en interacción por lo tanto, se centra en los requerimientos funcionales.

1.6.5 Modelo del Diseño

Es un modelo de objetos que describe la realización de los Casos de Uso y al mismo tiempo constituye una abstracción del modelo de implementación y del código fuente, constituye una entrada esencial a las actividades de implementación y prueba. El Modelo del Diseño constituye un vehiculo de análisis durante la fase de elaboración, pero se refina con un diseño detallado durante la fase de construcción. (Software 2007)

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. De hecho, cuando la precisión del diseño es muy grande, la implementación puede ser hecha por un generador automático de código.(UCI)

1.7 Calidad del software.

La calidad de un software es el grado en que se satisfacen los requerimientos funcionales y no funcionales definidos por el cliente, y que está fuertemente vinculado a las cualidades que caracterizan y determinan la utilidad y existencia del software con los estándares de desarrollos explícitos y documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.

La calidad de software se fundamenta en la utilización de métodos de desarrollo adecuados sobre herramientas que den soporte a estos. “El proceso se basa en la práctica de revisiones de los productos obtenidos para garantizar la adecuación de la construcción, y en la realización de pruebas tendentes a la búsqueda y depuración de errores. El resultado de dichas revisiones y pruebas es la elaboración de un informe que indique el grado de calidad del producto obtenido, los problemas detectados y las soluciones propuestas” (PRESSMAN 2005)

La calidad de Software es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.(PRESSMAN 1993)

Para controlar la calidad del *software* es necesario, ante todo, definir los parámetros, indicadores o criterios de medición. Las cualidades para medir la calidad del *software* son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. La Metodología para la evaluación de la calidad del Software define indicadores de calidad estructurados en cuatro niveles jerárquicos: factor, criterio, métrica, elemento de evaluación, donde cada nivel inferior contiene los indicadores que conforman el nivel precedente. Otros autores identifican la calidad con el nivel de complejidad del software y definen dos categorías de métricas: de complejidad del programa o código, y de complejidad de sistema o estructuras.

Todos los autores coinciden en que el software posee determinados índices medibles que son la base 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:

- Definir el software que va a ser controlado: clasificación por tipo, esfera de aplicación, complejidad, etc., 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 del 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 parciales 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, etc. (Carrasco, León et al. 1995)

1.8 Gestión de la Calidad.

La gestión de calidad es una estrategia organizativa y un método de gestión que hace participar a todos los empleados y pretende mejorar continuamente la eficacia de una organización en satisfacer el cliente. Es actualmente una alternativa empresarial indispensable para la supervivencia y la competitividad de la propia empresa en los mercados en los que actúa. A través de ella, se busca la optimización de recursos, la reducción de fallos y costes y la satisfacción propia y del cliente.

La Gestión de la Calidad está medida por una serie de normas aplicables genéricamente a todas las organizaciones, sin importar su tipo, tamaño o su personalidad jurídica. Se trata de que la Gestión de la Calidad sea compatible con otros sistemas como el de Política Medioambiental de la Empresa.

La implantación de un Sistema de Gestión de Calidad se puede hacer a través de la Normalización, la Certificación y la Acreditación. (CEDEFOP 1998) (Jamaica)

1.9 Gestión de Proyecto.

La Gestión de Proyectos es un conjunto de técnicas, conocimientos, habilidades y herramientas encaminadas a planificar tareas que conduzcan a alcanzar los requisitos del proyecto. (investigativo 2006)

Un proyecto consiste en colocar/utilizar los recursos para lograr un objetivo específico siguiendo un esquema planificado y organizado. La Gestión de Proyectos se enfoca en planificar las actividades, programarlas y luego mientras se está en el proceso de ejecución controlar dichas actividades.

Pero específicamente el control consiste en mantener tres factores bajo control, ellos son: tiempo, costo y alcance. Diferentes áreas de la empresa pueden entrar en conflicto respecto a los factores anteriores por los intereses particulares de cada departamento. (Dominguez 2007)

Tiempo

El tiempo se descompone para propósitos analíticos en el tiempo requerido para completar los componentes del proyecto que es, a su vez, descompuesto en el tiempo requerido para completar cada tarea que contribuye a la finalización de cada componente. Cuando se realizan tareas utilizando gestión de proyectos, es importante partir el trabajo en pedazos menores para que sean fáciles de seguir.

Costo

El costo de desarrollar un proyecto depende de múltiples variables incluyendo costes de mano de obra, costes de materiales, administración de riesgo, infraestructura (edificios, máquinas, etc.), equipo y utilidades. Cuando se contrata a un consultor independiente para un proyecto, el coste típicamente será determinado por la tarifa de la empresa consultora multiplicada por un estimado del avance del proyecto.

Alcance

Requerimientos especificados para el resultado final. La definición global de lo que se supone que el proyecto debe alcanzar y una descripción específica de lo que el resultado final debe ser o debe realizar. Un componente principal del alcance es la calidad del producto final. La cantidad de tiempo dedicado a las tareas individuales determina la calidad global del proyecto. Algunas tareas pueden requerir una cantidad dada de tiempo para ser completadas adecuadamente, pero con más tiempo podrían ser completadas excepcionalmente. A lo largo de un proyecto grande, la calidad puede tener un impacto significativo en el tiempo y en el costo (o viceversa). (Dominguez 2007)

1.10 Estimación.

Consiste en determinar el valor de una variable desconocida a partir de otras conocidas, o de una pequeña cantidad de valores conocidos de esa misma variable. La estimación forma parte de la inferencia estadística. El razonamiento por inferencia va de lo particular a lo general, de lo conocido a lo desconocido, y puede decirse que es lo inverso a la deducción, que va de lo general a lo particular. (Andrés 2000)

1.11 Método.

Presenta diversas definiciones debido a la complejidad de una exactitud en su conceptualización: "Conjunto de pasos fijados de antemano por una disciplina con el fin de alcanzar conocimientos válidos mediante instrumentos confiables", "secuencia estándar para formular y responder a una pregunta", "pauta que permite a los investigadores ir desde el punto A hasta el punto Z con la confianza de obtener

un conocimiento válido". Así el método es un conjunto de pasos que trata de proteger la subjetividad en el conocimiento. (Andrés 2000)

1.12 Métricas de software.

Todas las organizaciones de software exitosas implementan mediciones como parte de sus actividades cotidianas pues estas brindan la información objetiva necesaria para la toma de decisiones y que tendrá un impacto efectivo en el negocio y desempeño en la ingeniería.

Para poder asegurar que un proceso o sus productos resultantes son de calidad o poder compararlos, es necesario asignar valores, descriptores, indicadores o algún otro mecanismo mediante el cual se pueda llevar a cabo dicha comparación. (GARCÍA 2007)

Para entender mejor el concepto de métrica es necesario aclarar que los términos, métricas, medición y medida no tienen el mismo significado. Visto que es la medición y medida, ahora se verá el término de métrica.

Métrica: Es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. (PRESSMAN 1998)

Una métrica es una medida efectuada sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite, previa comparación con unos valores (medidas) de referencia, obtener conclusiones sobre el aspecto medido con el fin de adoptar las decisiones necesarias. Con esta definición, la definición y aplicación de una métrica no es un objetivo en sí mismo sino un medio para controlar el desarrollo de un sistema de software. (Durán 2007) (Muñoz 2006)

Se definen las métricas de software como "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", como se muestra en la Fig 1.(GONZÁLEZ 2001)



Fig 1. Definición de las métricas de estimación.

La funcionalidad principal de las métricas, es medir aspectos o particularidades específicas de los productos obtenidos. Las métricas del software responden a dos objetivos fundamentales como se muestra en la Fig 2, la valoración y la estimación. Las principales magnitudes dentro de la valoración son la calidad, fiabilidad y la productividad; mientras que las magnitudes de la estimación corresponden al esfuerzo y al tiempo. (GONZÁLEZ 2001)

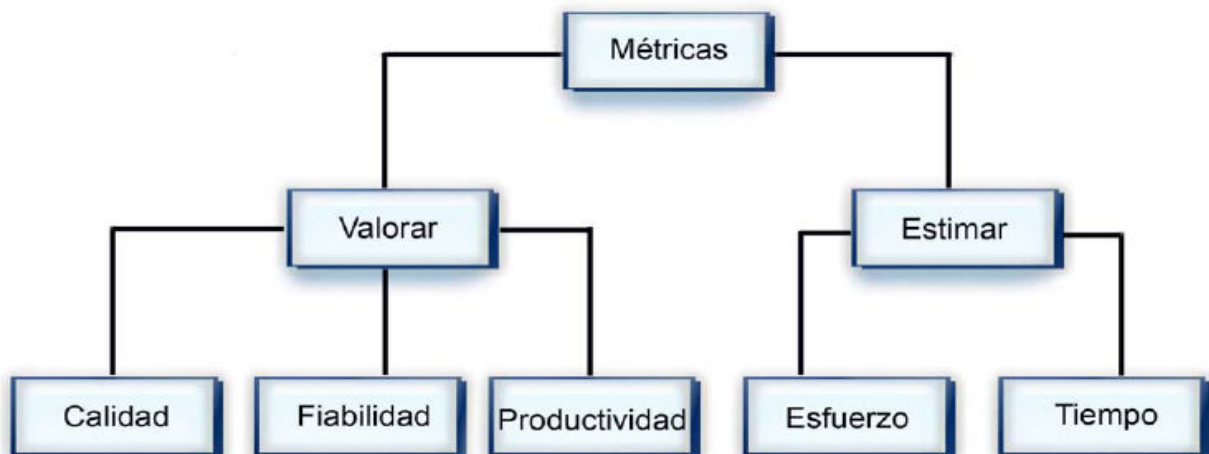


Fig 2. Objetivos de las métricas del software

1.12.1. Propiedades de las métricas.

Cientos de métricas han sido propuestas para el software pero no todas proporcionan suficiente soporte práctico para su desarrollo, pues algunas de ellas exigen mediciones demasiado complejas, de otras no se tienen los conocimientos necesarios como para atreverse a aplicarlas y otras violan las nociones básicas de lo que realmente es el software de alta calidad. Es por ello que se han definido una serie de propiedades que deben acompañar a las métricas de software para que sean efectivas, por tanto la métrica obtenida y las medidas que conducen a ello deben cumplir con las siguientes características fundamentales:

- Simple y fácil de calcular: debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no obligará a un esfuerzo o a una cantidad de tiempo inusuales.
- Empírica e intuitivamente persuasiva: 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).
- Consistente en el empleo de unidades y tamaños: el cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por 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.

Un mecanismo eficaz para la retroalimentación de la calidad: la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.

1.12.2. Métricas sobre el producto

Las métricas sobre el producto están orientadas a estimar las características del mismo antes de su desarrollo. Estas estimaciones se basan en el conocimiento que los desarrolladores adquieren a partir de datos obtenidos de proyectos anteriores. (Tecnomaestros)

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. (KAN. 2000)

En general las características a medir de los artefactos del producto son:

- **Tamaño:** Las métricas del tamaño del producto se refieren generalmente al volumen del producto desarrollado. Incluyen líneas de código (LOC), número de ficheros, páginas de la documentación.
- **Calidad:**
 - **Defectos:** indicadores de que un artefacto no funciona como ha sido especificado, o cualquier otra característica indeseable.
 - **Complejidad de una estructura o un algoritmo:** mientras mayor sea la complejidad y más difícil sea de comprender y modificar la estructura del sistema, mayor probabilidad habrá de que falle.
 - **Acoplamiento:** mediciones de cuántos elementos del sistema están interconectados y cuán extensivamente.
 - **Cohesión:** mediciones de cuán bien un elemento o un componente cumple con los requerimientos de tener un sólo y bien definido propósito.
 - **Primitividad:** el grado en el cual las operaciones o métodos de una clase pueden estar compuestos por otros de la misma clase.
- **Totalidad:** medición de la magnitud en la cual un artefacto cumple con todos los requerimientos (plan / real).
- **Rastreabilidad:** Indicadores de que los requerimientos de determinado nivel se están satisfaciendo por determinados artefactos, o que todos los artefactos tengan razón de existir.
- **Volatilidad:** el grado de cambio de un artefacto debido a defectos o a cambios en los requerimientos.
- **Esfuerzo:** medición del trabajo (Unidad de tiempo del personal) que es requerido para producir un artefacto

1.13 Métricas para la estimación.

A continuación se muestran una serie de métricas para la estimación del Software.

1.13.1 Documentación

Esta métrica da una medida de la magnitud de la documentación asociada al producto de software. Teniendo en cuenta las medidas definidas anteriormente pudieran derivarse métricas compuestas como (PRESSMAN 2002):

- Páginas de Documentación/KLOC (miles de líneas de código).
- Páginas de Documentación/PF

1.13.2 La Métrica Bang

La Métrica Bang es una métrica a partir de la cual se puede derivar una indicación del tamaño del software a implementar como consecuencia del Modelo de Análisis (MA).

Para calcular la Métrica Bang, el desarrollador de software debe evaluar primero un conjunto de primitivas (elementos del MA que no se subdividen más en el nivel de análisis) Las primitivas se determinan evaluando el MA y desarrollando cuentas para varios elementos (*Métricas en el desarrollo del Software*). (PRESSMAN 2002)

1.13.3 Tamaño medio de operación (TMedio).

Aunque las líneas de código podrían ser usadas como un indicador para el tamaño de operación, la medida LOC adolece de ciertos problemas. Por esta razón, el número de mensajes enviados por la operación proporciona una alternativa para el tamaño de operación. (PRESSMAN 2002)

1.13.4 Basados en Casos de Uso.

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema. Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas, como el Proceso Unificado de Rational (Rational Unified Process), en las cuales se propone especificar la funcionalidad de los sistemas mediante la utilización de Casos de Uso.

El método de Casos de Uso permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso.

La estimación por Puntos de Caso de Uso resulta muy efectiva para estimar el esfuerzo requerido en el desarrollo de los primeros Casos de Uso de un sistema, si se sigue una aproximación iterativa como el Proceso Unificado de Rational. En éste tipo de aproximación, los primeros Casos de Uso a desarrollar son los que ejercitan la mayor parte de la arquitectura del software y los que a su vez ayudan a mitigar los riesgos más significativos (iteraciones de Elaboración en el Proceso Unificado). (Gramajo 2005)

Puntos de función

La estimación exitosa usando puntos función se basa en varias técnicas de estimación: Top-Down, Analogía y Consejo de Expertos. La estimación Top-Down es una técnica de estimación que calcula el programa entero, costo y esfuerzo usando parámetros amplios. Los parámetros amplios y las comparaciones están basados en datos históricos usando técnicas estimativas de Analogía. El Consejo de Expertos se obtiene de expertos con experiencia en proyectos similares o experiencia en el uso de puntos función.

La comparación de proyectos con otros similares es una actividad crítica para lograr una estimación exitosa.

Una vez que el tamaño del proyecto se ha determinado en puntos función, el estimado de horas, costo y programa se puede calcular. Los cálculos se deben hacer con datos de proyectos similares como se describió anteriormente. (Inc 1995)

Los puntos de función de un sistema software se calculan teniendo en cuenta:

- Entradas al sistema
- Salidas del sistema
- Consultas
- Grupos de datos lógicos del sistema
- Grupos de datos lógicos que no son del sistema pero que el sistema usa.

COCOMO II.

Otro de los métodos que se aplican es el método COCOMO II directamente sobre los Puntos de Función sin ajustar. Éste método es el preferido en la actualidad para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir. (Inc 1995)

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo cuando se planifica una nueva actividad de desarrollo del software. Está asociado a los ciclos de vida modernos. El modelo original COCOMO ha tenido mucho éxito pero no puede emplearse con las prácticas de desarrollo software más recientes tan bien como con las prácticas tradicionales. COCOMO II apunta hacia los proyectos software de los 90 y de la primera década del 2000, y continuará evolucionando durante los próximos años.

En resumen, los objetivos de la creación del modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de tiempo y de coste del software de acuerdo con los ciclos de vida utilizados en los 90 y en la primera década del 2000.
- Desarrollar bases de datos con costes de software y herramientas de soporte para la mejora continua del modelo.
- Proporcionar un marco analítico cuantitativo y un conjunto de herramientas y técnicas para la evaluación de los efectos de la mejora tecnológica del software en costes y tiempo del ciclo de vida software.

Estos objetivos apoyan las necesidades primarias expresadas por los usuarios de la estimación de costes del software. En orden de prioridades, estas necesidades son: el apoyo de la planificación de proyectos, la previsión de personal del proyecto, la preparación del proyecto, la re-planificación, el seguimiento del proyecto, la negociación del contrato, la evaluación de la propuesta, la nivelación de recursos, exploración de conceptos, la evaluación del diseño y decisiones referentes a la oferta/demanda.

Los cuatro elementos principales de la estrategia que ha seguido COCOMO II son:

- Preservar la apertura de COCOMO original.
- Desarrollar COCOMO II de forma que sea compatible con el futuro mercado del software.
- Ajustar las entradas y salidas de los sub.-modelos de COCOMO II, al nivel de información disponible.
- Permitir que los sub. -modelos de COCOMO II se ajusten a las estrategias de proceso particulares de cada proyecto.

También todos sus interfaces están diseñadas para ser públicas, bien definidas y parametrizadas para que los pre-procesos complementarios (modelos de analogía basados en casos de otras medidas de estimación), post-procesos (planificación de proyectos y herramientas de control, modelos dinámicos de proyectos, analizadores de riesgo) y paquetes de alto nivel (paquetes de gestión de proyectos, ayudas de negociación de producto) puedan combinarse estrechamente con COCOMO II.

Para apoyar a los distintos sectores del mercado software, COCOMO II proporciona una familia de modelos de estimación de coste software cada vez más detallado y tiene en cuenta las necesidades de cada sector y el tipo de información disponible para sostener la estimación del coste software. Esta familia de modelos está compuesta por tres sub.-modelos cada uno de los cuales ofrece mayor fidelidad a medida que uno avanza en la planificación del proyecto y en el proceso de diseño. Estos tres sub.-modelos se denominan:

El modelo de Composición de Aplicaciones.

Indicado para proyectos construidos con herramientas modernas de construcción de interfaces gráficos para usuario.

El modelo de Diseño anticipado.

Este modelo puede utilizarse para obtener estimaciones aproximadas del coste de un proyecto antes de que esté determinada por completo su arquitectura. Utiliza un pequeño conjunto de drivers de coste nuevo y nuevas ecuaciones de estimación. Está basado en Punto de Función sin ajustar o KSLOC (Miles de Líneas de Código Fuente).

El modelo Post-Arquitectura.

Este es el modelo COCOMO II más detallado. Se utiliza una vez que se ha desarrollado por completo la arquitectura del proyecto. Tiene nuevos drivers de coste, nuevas reglas para el recuento de líneas y nuevas ecuaciones.(S.-Capuchino)

1.13.5. Método de análisis de puntos de función IFPUG

El objetivo principal de este método es medir el software cuantificando la funcionalidad que suministra al usuario basándose principalmente en el diseño lógico. Bajo esa perspectiva, los resultados que se quieren obtener del análisis de los Puntos de Función son:

- Medir la funcionalidad que el usuario requiere y recibe.
- Medir el desarrollo y el mantenimiento del software independientemente del de la tecnología utilizada para su implementación. (Meza, Peña et al.)

1.13.6. Método de análisis puntos de función MK II – FPA

La organización que avala este método es UKSMA (Asociación de Usuarios de la Métrica del Software del Reino Unido) y su desarrollador fue Charles Symons en el Reino Unido.

Sus principales actividades incluyen las conferencias, clases particulares, exámenes de certificación, comité de prácticas de métricas, participación en actividades internacionales del desarrollo de medición del software tales como el grupo de funcionamiento de ISO/IEC JTC1/SC7 12; es la organización clave para las asociaciones de la medida del software, ISBSG, IFPUG, y es el cuerpo controlador para el método de MkII - FPA de medida funcional del tamaño.

Los puntos de función MkII - FPA se obtienen sobre el producto de dos componentes: el tamaño de procesamiento de la información y el ajuste de complejidad técnica.

En el Tamaño del Procesamiento de la Información en vez de utilizar cinco componentes: entradas, salidas, interfaces, consultas y ficheros lógicos, el sistema se examina desde el punto de vista de las transacciones lógicas, consistiendo cada una de ellas en entrada, proceso y salida. (Meza, Peña et al.)

1.13.7. Método de análisis puntos de función FPA.

FPA es un método para determinar el tamaño funcional de un sistema o de un proyecto de información. El tamaño funcional se puede utilizar para diversos propósitos, por ejemplo presupuesto, este método realiza los siguientes pasos para determinar el tamaño de un proyecto, del sistema de información o del desarrollo del sistema:

Paso 1: Identificar las funciones del sistema que son relevantes al usuario

Paso 2: Determinar la complejidad funcional de cada función

Paso 3: Calcular los punto de función no ajustados del sistema.

Paso 4: Clasificar los requisitos generales para el sistema usando características generales del mismo.

Paso 5: Calcular los puntos de función no ajustados del sistema.

El método FPA no:

- Es una gerencia de proyecto.
- Entrega automáticamente error, proyecta libremente estimaciones proporcionando una ayuda importante en el proceso de presupuesto.
- Es un método del planeamiento del proyecto.

A manera de conclusión, puede indicarse que los métodos IFPUG, MK2, NESMA y COSMIC-FFP, son los más utilizados en el momento para medir el software. (Meza, Peña et al.)

1.13.8. Líneas de Código

La métrica de tamaño tradicional para estimar el esfuerzo de desarrollo y productividad ha sido LOC (Lines Of Code) o SLOC (Source Lines Of Code). Se han propuesto varios modelos de estimación, la mayoría de ellos son funciones de las líneas de código o de las miles de líneas de código que tendrá el software a desarrollar. Generalmente, el modelo de estimación de esfuerzo consiste de dos partes. La primera provee una base de estimación como una función del tamaño del software, y es de la siguiente forma:

E es el esfuerzo estimado en meses hombre, A , B y C son constantes y $KLOC$ es el tamaño estimado del sistema final en miles de líneas de código. La segunda parte del modelo modifica esta estimación en base a cuantificar la influencia de factores de ambiente, por ejemplo la utilización de diferentes metodologías, habilidad del equipo de desarrollo y restricciones de hardware.

La definición de $KLOC$ es importante si se quiere comparar los distintos modelos que se han propuesto en la literatura. Algunos de ellos incluyen líneas de comentarios, y otros no. Del mismo modo, la definición del esfuerzo estimado E es también importante., ya que E puede representar sólo el esfuerzo de codificación, o en el otro extremo, el esfuerzo total del análisis, diseño, codificación, test y mantención. Por estas razones, comparar estos modelos se torna complejo.

Los principales problemas de utilizar líneas de código como métrica para estimación del esfuerzo son la falta de una definición universal de línea de código, su dependencia con el lenguaje de desarrollo y la

dificultad de estimar en fases tempranas del desarrollo la cantidad de líneas que tendrá una aplicación. (Hurtado 2006)

Algunas ventajas del uso de las LOC son (E-CLASES):

- Son fáciles de calcular.
- Muchos modelos de estimación de software usan LOC o KLOC como datos de entrada.
- Existen un amplio conjunto de datos y literatura basados en LOC.

Inconvenientes del uso de las líneas de código (E-CLASES):

- Son dependientes del lenguaje de programación.
- Perjudica a los programas cortos pero bien diseñados.
- Su uso en estimación es difícil porque hay que estimar las LOC a producirse mucho antes de que se complete el análisis y el diseño.

1.14 Proceso de Pruebas.

Las actividades que se llevan a cabo dentro del proceso de pruebas son:

- Planificación de las pruebas: Para efectuar esta actividad es preciso conocer las características del proyecto y del software, de forma que se puedan definir los roles y recursos del sistema para un apropiado entorno de pruebas, estrategia a seguir, concretar aquellos requerimientos que van a ser probados, los entregables y el cronograma.
- Diseño de las pruebas: Esta actividad requiere de técnicas apropiadas para el diseño de casos y procedimientos de prueba.
- Implementación de las pruebas: Esta actividad puede o no realizarse, esto depende de que se cuente o no con herramientas adecuadas para automatizar las pruebas. Es favorable tener en

cuenta que no siempre es necesario automatizarlas, casi siempre esto se hace para las que van a ser usadas en otros ciclos de prueba y para las complejas, que al generar defectos se deben repetir para verificar la corrección de los mismos y se pierde tiempo del propio proceso.

- Ejecución de las pruebas: Una vez que han sido diseñados los casos de prueba, se procede a la ejecución de las mismas con el objetivo de detectar errores.
- Evaluación de los resultados: Se verifica la correspondencia que existe entre los resultados esperados y reales, y se obtienen estadísticas de los errores.
- Análisis de errores: Se analizan los errores encontrados dándoles una clasificación para detectar las causas más frecuentes de su aparición y se predice la fiabilidad del software. (PRESSMAN 1998)

1.14.1 Artefactos.

Los artefactos son cualquier producto de trabajo final o intermedio que se producen y se usan durante un proyecto. Estos se utilizan para capturar y transmitir información del desarrollo del Software. Un artefacto puede ser un modelo, una descripción, un documento, o el software como tal.

Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables. (Andrés 2000).

1.15 Normas de Calidad.

1.15.1 ISO/IEC ISO 12119:1994. Paquetes de Software.

Esta norma está orientada a paquetes de software. En realidad, esta no norma analiza si un programa es malo o bueno. Simplemente, certifica si el software sirve para lo que dice que sirve. Tres años más tarde de la publicación de la ISO 9126, en 1994 la misma organización considera necesario escribir una norma que especificaría los paquetes de software. Así nace la ISO-12119, que además de explicar diferentes

métodos de testeo, determina los detalles que debe respetar todo paquete de software. Algunos de ellos son:

- Documentación para el usuario de fácil comprensión, que incluya un índice y un manual de instalación.
- Que el software le informe si el programa fue correctamente instalado.
- Función de ayuda con recursos de hipertexto.
- Mensajes de error con información necesaria para su solución.
- Identificación de los archivos del sistema operativo usados por el programa.
- Capacidad de interrumpir un proceso demorado, sin que se cuelgue el equipo.
- Posibilidad de anular funciones de efectos irreversibles, como borrar.(GARCIA 1998)

1.15.2 ISO/IEC 12207:1995 Procesos del ciclo de vida del software.

Esta norma establece un marco de referencia común para los procesos del ciclo de vida del software, con una terminología bien definida a la que puede hacer referencia la industria del software. Contiene procesos, actividades y tareas para aplicar durante la adquisición de un sistema que contiene software, un producto software puro o un servicio software, y durante el suministro, desarrollo, operación y mantenimiento de productos software. El software incluye la parte software del firmware (combinación de un dispositivo hardware e instrucciones o datos de computadora que residen como software de solo lectura en el dispositivo hardware. Este software no puede modificarse fácilmente bajo el control del programa.)

Esta norma incluye también un proceso que puede emplearse para definir, controlar y mejorar los procesos del ciclo de vida del software.

Esta norma es aplicable a la adquisición de sistemas, productos y servicios software, al suministro, desarrollo, operación y mantenimiento de productos software, y a la parte software del firmware, independientemente de que sea hecho interna o externamente a una organización. Incluye también

aquellos aspectos de la definición del sistema necesarios para proporcionar el contexto de los productos y servicios software.

Procesos principales del Ciclo de Vida

Aquí se pueden definir los siguientes procesos principales del ciclo de vida:

- 1) Proceso de Adquisición.
- 2) Proceso de Suministro.
- 3) Proceso de Desarrollo.
- 4) Proceso de Operación.
- 5) Proceso de Mantenimiento.

Las actividades y tareas de un proceso principal son responsabilidad de la organización que lo inicia y ejecuta. Esta organización asegura que el proceso existe y es operativo. (C Génova 1995)

1.16 Conclusiones.

Este capítulo se centra en el estudio de los diferentes productos que se obtienen durante el desarrollo del Software, además se definieron conceptos importantes en el ámbito de la Calidad del Software, la medición, el proceso de pruebas, se analizaron además diferentes normas de calidad, entre otros elementos. Se realiza un estudio sobre las tendencias actuales a nivel mundial en el campo de la medición en los proyectos, y se analizaron los resultados arrojados por las entrevistas realizadas al proyecto de Calidad de la Facultad 3. Se hizo además un estudio de los diferentes métodos empleados para la estimación del tamaño, basado en Casos de Uso, Líneas de Código, Puntos de Función y sus variantes, entre otros. Después de haber analizado todos estos elementos se llega a la conclusión de que en la Facultad no se emplean métodos efectivos para medir el tamaño de los diferentes productos resultantes del desarrollo del Software.

CAPÍTULO II. PROPUESTA DE SOLUCIÓN.

2.1 Introducción

En el presente capítulo se realiza la descripción de los pasos a tener en cuenta para la obtención de la propuesta de los métodos de medición, dentro de ellos se definen los artefactos a medir, y las características a tener en cuenta de estos, además se definen las variables a utilizar, y se hace un estudio de los métodos utilizados actualmente para medir el Software.

2.2 ¿Por qué es importante medir el software?

Una de las razones principales del incremento masivo en el interés por la medición de software ha sido la percepción de que las métricas son necesarias para la mejora de la calidad del proceso. (MARTÍNEZ 2006)

Hay cuatro razones para medir los procesos del software, los productos y los recursos

- Caracterizar: Para comprender mejor los procesos, los productos, los recursos y los entornos y para establecer las líneas base para las comparaciones con evaluaciones futuras.
- Evaluar: Para determinar el estado con respecto al diseño. Las medidas permiten conocer cuándo los proyectos y procesos están perdiendo la pista, de modo que puedan ponerse bajo control. Además para valorar si se cumplen o no los objetivos de calidad trazados y para evaluar el impacto de la tecnología y las mejoras en los productos y procesos.
- Predecir: Para poder planificar. Los valores que se observan para algunos atributos pueden ser utilizados para predecir otros, lo que contribuye a establecer objetivos alcanzables para el coste, planificación, y calidad, de manera que se puedan aplicar los recursos apropiados, además permite analizar los riesgos y realizar intercambios diseño coste.
- Mejorar: Se mide para mejorar cuando se recoge la información cuantitativa que ayuda a identificar obstáculos, problemas de raíz, ineficiencias y otras oportunidades para mejorar la calidad del producto y el rendimiento del proceso. (PRESSMAN 1998)

Por tanto se considera que las métricas de software ayudan a los desarrolladores a valorar el trabajo desarrollado, proveen la información necesaria para la toma de decisiones técnicas, proporcionan datos objetivos que pueden ser usados en la planificación de futuros proyectos, ayudan a la evaluación de los modelos de análisis y de diseño, y a la formulación de casos de prueba, valoran la productividad de los desarrolladores , ayudan a evaluar la calidad de los productos o sistemas que se construyen así como también a entender que ocurre durante el desarrollo y el mantenimiento.

2.3 Pasos para la obtención de los métodos de medición de los productos de prueba.

2.3.1 Identificación y Decantación.

Se identifican todos los artefactos con que trabaja la organización y se hace una selección de los que se han de medir durante el proceso de prueba.

2.3.2 Clasificación.

Se clasifican los artefactos atendiendo a las características que presentan.

2.3.3 Estudio y Recopilación.

A partir de las clases en que se ordenaron los artefactos se busca información sobre los métodos existentes en el mundo para la medición de estos productos.

Si después del estudio se obtienen métodos efectivos para determinar el tamaño del artefacto se pasará directamente al punto 2.3.6.

En caso contrario se definirá un método para la obtención del tamaño del artefacto en cuestión por lo que se seguirán con los siguientes pasos.

2.3.4 Definición de características generales.

Consiste en determinar las propiedades de los artefactos seleccionados teniendo en cuenta los aspectos a revisar de cada uno.

2.3.5 Elaboración de las fórmulas.

Luego de tener definida las características generales de cada clase, es necesario tener en cuenta los aspectos particulares de cada producto dentro de estas clases para lograr una precisión más exacta cuando se mida dicho artefacto.

2.3.5.1 Obtención de variables.

A continuación se muestran los aspectos a tener en cuenta para la realización de las fórmulas:

1. Búsqueda de las variables consideradas como la medida básica de cada producto.
2. Selección de las variables constantes dentro del producto: dependientes de la bibliografía o de las definiciones del proyecto.
3. Búsqueda de las variables intermedias: son aquellas que el valor solo se puede obtener a partir del cálculo de las variables definidas anteriormente pero facilitan el entendimiento de la fórmula.

2.3.5.2 Aplicación piloto.

En este paso las fórmulas obtenidas (nuevo método) y las medidas de tamaño usadas actualmente (medidas usuales), se aplican a un conjunto de documentos y gráficas que abarcan todas clases obtenidas, registrando todos los datos resultantes del proceso.

2.3.5.3 Validación.

Después de la aplicar los métodos se hace una comparación entre lo obtenido al aplicar las medidas usuales y el nuevo método. De los mismos casos, se analizarán los resultados obtenidos de dicha comparación.

Si el nuevo método no supera al método usual o es inconsistente, se retorna al paso 2.3.5.1 y se realiza la obtención de nuevas variables y sus siguientes pasos.

Si el método cumple con las expectativas, continua con el siguiente paso de la formalización de la propuesta.

2.3.6 Formalizar la propuesta.

Inmediatamente después haber probado la propuesta y de tener la seguridad que esta cumple con los objetivos trazados, se formalizara la propuesta esta tomando un formato estándar.

Hasta este paso se tiene una propuesta consistente en un listado de métodos asociados a cada clase de artefacto objeto de prueba o revisiones.

2.3.7 Aplicar los métodos propuestos.

Luego de tener ya validada la propuesta, se puede comenzar a aplicar en los proyectos productivos y se irá recopilando información y se verán satisfactorios resultados.

La propuesta se puede ver afectada por un conjunto de factores como por ejemplo:

- Aparición de nuevas clases de artefactos.
- Cambios en las propiedades de los artefactos existentes.
- Aparece un nuevo método en la bibliografía.
- Aparición de una nueva regla que impacte las variables de algún método.

Para estos casos es necesario ejecutar los pasos desde el inicio para mejorar la propuesta.

En caso contrario la propuesta se seguirá aplicando.

2.4 Aplicación de los pasos propuestos a los productos de prueba de los proyectos productivos de la Facultad 3.

2.4.1 Identificación y Decantación.

Se hace una selección de los artefactos a medir durante la fase de prueba, para la obtención de los mismos se utiliza una tabla con todos los artefactos del expediente del proyecto, y de ahí los involucrados en el proceso de pruebas (especialistas de la dirección de calidad y miembros del equipo de pruebas de la facultad 3) marcaron los que mayormente eran revisados. De este proceso se escogieron los de mayor incidencia en procesos de revisión o pruebas, quedando así los artefactos que constituyen entregables

para la Dirección de Calidad. En la investigación se tomaron los productos de prueba que se adaptan a las características de los proyectos productivos de la Facultad 3. A continuación en la Tabla 1 se muestran 36 artefactos vigentes en el expediente de proyecto, de estos se parte para realizar la identificación.

Tabla 1: Artefactos del expediente de proyecto.

Artefactos del Expediente de Proyecto:
1. Plan de Desarrollo de Software
2. Plantillas de DCS
3. Plantillas de DCS-Ambiente de Desarrollo
4. Plantilla de DCS-Plan de Capacitación
5. Plantilla de DCS-Roles y Responsabilidades
6. Plantilla de DCS-Minuta de Reunión
7. Plantilla de DCS-Documento Visión
8. Especificación de requerimientos.
9. Especificación de CU
10. Modelo de Análisis
11. Modelo de Diseño.
12. Documento de Arquitectura.
13. Modelo Entidad Relación.

14. Modelo de Objeto.
15. Documento de Arquitectura de Información.
16. Portal Web.
17. Aplicación Web.
18. Aplicación Desktop.
19. Manual de Usuario.
20. Manual de Instalación.
21. Multimedia.
22. Glosario de Términos.
23. P1 y PA de Cursos de Capacitación.
24. Cursos de Capacitación montados en EVA.
25. Prototipos Documentados.
26. Aplicaciones en Ambientes Reales - Realidad Virtual
27. Diseño Gráfico.
28. Diccionario de Datos.
29. Diagrama de Despliegue.
30. Diagrama de Procesos-Nombre de Procesos

31. Plantilla de DCS-Evaluación de Áreas de la Organización
32. Plantilla de DCS-Modelo de CU del Sistema
33. Plantilla de DCS-Plan de Gestión de Requisitos
34. Plantilla de DCS-Diseño de Caso de Prueba
35. Código Fuente
36. Plantilla de DCS-Plan de Gestión de Requisitos

A continuación se relacionan los 22 artefactos que finalmente quedaron seleccionados como entregables para la Dirección de Calidad y de ellos se sacan solo 19 artefactos son entregados al Laboratorio de Calidad de la Facultad 3, como se muestra en la Tabla 2.

Tabla 2: Artefactos seleccionados como entregables.

Artefactos Dirección de Calidad	Proyecto de Calidad Fac.3
1. Multimedia.	NO
2. Portal Web.	NO
3. Modelo de Objeto.	NO
4. Aplicación Web.	SI
5. Aplicación Desktop.	SI
6. Diagrama de Casos de Uso.	SI

Artefactos Dirección de Calidad	Proyecto de Calidad Fac.3
7. Diagrama de Despliegue.	SI
8. Diseño Gráfico.	SI
9. Prototipos documentados.	SI
10. Modelo Entidad Relación.	SI
11. Modelo del Diseño.	SI
12. Especificación de los Casos de Uso.	SI
13. Modelo del Análisis.	SI
14. Especificación de requerimientos	SI
15. Diccionario de Datos.	SI
16. Cursos de Capacitación montados en EVA	SI
17. P1 y PA de Cursos de Capacitación.	SI
18. Documento de Arquitectura.	SI
19. Glosario de Términos.	SI
20. Manual de Instalación.	SI
21. Manual de Usuario.	SI

Artefactos Dirección de Calidad	Proyecto de Calidad Fac.3
22. Documento de Arquitectura de Información.	SI

2.4.2 Clasificación.

El proceso de clasificación de los artefactos consiste en agruparlos teniendo en cuenta la clase, estas clases pueden ser: documentos (solo textos), gráficos (artefactos UML) y código.

- **Documentos:** el contenido está constituido mayormente por texto, aunque puede contener además imágenes, tablas, diagramas, etc. Se utiliza para describir, justificar o acreditar algún hecho. Ejemplo: Plantilla de especificación de Casos de Uso.
- **Gráficos:** son aquellos que presentan gran número de representaciones de diagramas, aunque pueden contener texto. Los diagramas puede clasificarse en dinámicos o estáticos. Se utilizan para representar información de forma más sencilla y comprensible para la solución de problemas. Ejemplo: Diagrama de Despliegue (UML).
- **Código:** está representado por las líneas de código, se utiliza para medir el tamaño de un proyecto o Software. Ejemplo: Aplicaciones Web.

Tabla 3: Clasificación de los artefactos.

Artefactos	Clasificación
Aplicación Web.	Código.
Aplicación Desktop.	Código.
Diagrama de Casos de Uso.	Gráfico
Diagrama de Despliegue.	Gráfico

Artefactos	Clasificación
Diseño Gráfico.	Gráfico
Prototipos documentados.	Documentos
Modelo Entidad Relación.	Gráfico
Modelo del Diseño.	Gráfico
Especificación de los Casos de Uso.	Documentos
Modelo del Análisis.	Gráfico
Especificación de requerimientos	Documentos
Diccionario de Datos.	Documentos
Cursos de Capacitación montados en EVA	Documentos
P1 y PA de Cursos de Capacitación.	Documentos
Documento de Arquitectura.	Documentos
Glosario de Términos.	Documentos
Manual de Instalación.	Documentos
Manual de Usuario.	Documentos
Documento de Arquitectura de Información.	Documentos

2.4.3 Estudio y Recopilación.

A partir de las clasificaciones vistas anteriormente se busca información sobre los métodos existentes en el mundo para la medición de estos productos. Entre ellos están los siguientes:

Método de Expertos o Delphi.

Linston y Turoff definen la técnica Delphi como un método de estructuración de un proceso de comunicación grupal que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo.

Delphi consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes.

La capacidad de predicción de la Delphi se basa en la utilización sistemática de un juicio intuitivo emitido por un grupo de expertos. Es decir, el método Delphi procede por medio de la interrogación a expertos con la ayuda de cuestionarios sucesivos, a fin de poner de manifiesto convergencias de opiniones y deducir eventuales consensos. La encuesta se lleva a cabo de una manera anónima, actualmente es habitual realizarla haciendo uso del correo electrónico o mediante cuestionarios Web establecidos al efecto, para evitar los efectos de "líderes".

Las preguntas se refieren, por ejemplo, a las probabilidades de realización de hipótesis o de acontecimientos con relación al tema de estudio. La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los expertos consultados.

Entre las etapas principales de realización de la evaluación de una investigación a través del método de evaluación de expertos se encuentran las siguientes: (Sebastian 2005)

- Elaboración del objetivo.
- Selección de los expertos.
- Elaboración del cuestionario.

- Elección de la metodología.
- Ejecución de la metodología.
- Procesamiento de la información.

Líneas de código

El número de líneas de texto que componen el código fuente de un programa ha sido la medida más utilizada en la cuantificación del tamaño del software, entendido éste como un atributo interno de un producto. Es una medida fácil de obtener y manipular, además, ha sido considerado como factor clave de otros atributos como ocurre en el caso de la productividad. Las líneas de código, expresadas como LOC (del inglés Lines of Code), presentan, sin embargo, ciertos problemas que se ponen de manifiesto en las siguientes preguntas:

- ¿Se han de considerar los comentarios y las líneas en blanco en la contabilidad de las líneas de código de un programa?
- ¿Es equivalente el número de líneas de código contabilizado en un lenguaje de programación u otro?
- ¿Se han de considerar todas las instrucciones, o pueden obviarse definiciones de constantes y variables?
- Al igual que se hace con el código fuente ¿Puede utilizarse esta medida para otros documentos propios del desarrollo software tales como requisitos o diseño del programa?
- ¿Influye la tecnología a utilizar en la contabilidad de las líneas de código?
- Hay instrucciones que pueden necesitar más de una línea de código fuente, o por el contrario pueden existir diferentes instrucciones en una línea, ¿no sería necesario considerar el impacto de este hecho?

Hacer uso de esta medida requiere la definición de lo que se entiende por línea de código. De esta forma superar algunas ambigüedades respondiendo a varias preguntas anteriores. Una línea de código se define como: Cualquier línea de texto del programa excluyendo comentarios y líneas en blanco.

En muchos casos es interesante considerar también el número de líneas de comentarios que posee un programa ya que, aunque no son comandos necesarios para la ejecución del mismo, sí son una fuente de información muy útil que pueden influir en las posteriores modificaciones o en el mantenimiento de dicho programa.

Así las líneas de código sin considerar los comentarios se especificarán como NLOC, y aquellas que consideren los comentarios se definen como CLOC. El número de líneas totales sería:

$$LOC = CLOC + NLOC$$

Una medida de la densidad de comentarios sería:

$$d = -C LOC / LOC$$

Algunos investigadores han propuesto el *carácter* como medida alternativa a las líneas de código. Sería una medida simple de coleccionar y la conversión del número de caracteres en líneas de código sería extremadamente fácil.

$$LOC = \text{no de caracteres} / a$$

Donde *a* es un número promedio de caracteres por línea de texto.

Es fácil encontrar el acrónimo KLOC, indicando miles de líneas de código. Es una magnitud muy utilizada en grandes aplicaciones, siendo un múltiplo de LOC. Medir el tamaño del código fuente haciendo uso de las líneas de código presenta algunos inconvenientes y dudas a las que el presente trabajo se ha referido anteriormente. Estos problemas se ven agravados cuando se desea medir el tamaño de otros documentos propios de la fase de diseño o captura de los requisitos. Es fácil comprender el grave obstáculo que encierra cuantificar esta documentación si se considera que en numerosas ocasiones se componen, no sólo de líneas de texto, si no también de gráficos, diagramas de flujo, ecuaciones, símbolos y demás anotaciones propias de estas fases del desarrollo software. En muchas ocasiones se utiliza el número de páginas de documentación pero no es una medida aceptable. En resumen, las líneas de código son una medida sencilla de obtener y manipular, ampliamente utilizadas aunque con serios inconvenientes, algunos de los cuales pueden ser superados gracias a definiciones concretas y de general consenso.

Tokens:

Una alternativa a la contabilidad de las líneas de código es la contabilidad de las muestras, token, existentes en el código fuente. Un token se define como un elemento propio del lenguaje que posee significado por sí mismo (instrucciones, identificadores, operadores, constantes, delimitadores de comentarios y signos especiales). Haciendo uso de esta medida se obtendría un valor más adecuado al lenguaje de programación proporcionando una idea más precisa de la información contenida en el código fuente. La Ciencia del Software de Halstead hace uso de estas señales o tokens, que permiten conocer el tamaño de un programa, el vocabulario del mismo o su volumen. Esta última medida indica el espacio de memoria mínimo requerido para almacenar el programa. La medida propuesta por Halstead ha sido considerada como una mezcla de tamaño y esfuerzo

Funcionalidad:

En algunos casos, propios de grandes aplicaciones en las que existen miles de programas con cientos de líneas de código cada uno de ellos, se ha propuesto como unidad de medida del tamaño el módulo. Sin embargo, esta medida es de difícil aplicación en un marco de medida estricto pues no existe una clara definición de esta magnitud, es más, su uso implicaría cierta confusión entre la entidad medida, programa o conjunto de programas que constituyen una aplicación, y la medida en sí misma, módulo, entendido como programa, rutina o subrutina que forma parte de la aplicación. Como alternativa más certera existe el concepto de funcionalidad. Este atributo, propio del programa, está asociado con el concepto de funciones proporcionadas por el mismo, entendido como una colección de instrucciones que realizan una tarea determinada. El programador considera el código a través de las funciones que ha de realizar más que como un conjunto de líneas de texto o comentarios, por lo que su cuantificación sería enormemente útil al hacer coincidir un valor numérico con la apreciación subjetiva del profesional. (Hurtado 2006)

De los métodos estudiados y recopilados no se encontraron puntos de contacto con las clases propuestas, solo el método Delphi que tiene una aplicación tan general que lo hace inapropiado para el presente estudio, considerando como método efectivo para determinar el tamaño de la clase código el método de Líneas de Código, por lo que en este caso se irá directamente a formalizar la propuesta.

Para la medir el tamaño de los otros artefactos se continuará con los pasos que continúan.

2.4.4 Definición de características generales.

Se determinan las propiedades de los artefactos que se seleccionaron durante la identificación y decantación de estos, de los cuales se tiene en cuenta los aspectos a revisar de cada uno.

En el presente estudio se tuvieron en cuenta fundamentalmente:

- Las propiedades de los artefactos descritas por el lenguaje UML.
- Las recomendaciones de RUP contenidas en la literatura, plantillas y ayudas.
- Los elementos tenidos en cuenta por el equipo de calidad de la Facultad 3.
- Otras devenida de la experiencia personal de los investigadores, necesarias para obtener los artefactos con una calidad media aceptable.

2.4.4.1 Modelo Entidad Relación.

Para la medida del Modelo Entidad-Relación se debe tener en cuenta las propiedades definidas para cada elemento que componen dicho modelo, en este caso se tuvo en cuenta las propiedades definidas por la ayuda del Rational. A continuación se enumeran de acuerdo a cada elemento lo que se debe verificar de ellos en cuanto al gráfico.

Entidades:

- Que la representación gráfica sea la correcta.
- Que el nombre solo aparezca una vez.
- Verificar la clasificación en fuerte o débil.
- Si es fuerte o débil verificar que la representación de la misma este correcta.

Atributos:

- Verificar que la representación gráfica sea la correcta.
- Verificar que representen las propiedades básicas de las entidades y de las relaciones.

Relaciones:

- Verificar que la representación gráfica sea la correcta.
- Verificar que el nombre aparezca en el interior del rombo.

2.4.4.2 Diagrama de Despliegue.

Para la medida del Diagrama de Despliegue se deben tener en cuenta las propiedades definidas para cada elemento que componen dicho Diagrama, al igual que en el Modelo Entidad Relación se tuvo en cuenta las propiedades definidas por la ayuda del Rational. A continuación se enumeran de acuerdo a cada elemento lo que se debe verificar de ellos en cuanto al gráfico.

Nodos:

- Que la representación gráfica sea la correcta.
- Verificar la descripción del nodo.
- Comprobar la capacidad de almacenamiento, capacidad de memoria, o cualquier información sobre las capacidades del dispositivo.
- Verificar el número de componentes que contienen.

Componentes:

- Que la representación gráfica sea la correcta.
- Comprobar la descripción de los mismos.

Relaciones:

- Que la representación gráfica sea la correcta.

2.4.4.3 Modelo de Análisis.

Para la medida el Modelo de Análisis se deben tener en cuenta las propiedades definidas para cada elemento que componen dicho modelo, al igual que en el Modelo Entidad Relación se tuvo en cuenta las

propiedades definidas por la ayuda del Rational. A continuación se enumeran de acuerdo a cada elemento lo que se debe verificar de ellos en cuanto al gráfico.

Clases :

- Que las mismas representen conceptos.
- Que representen relaciones del dominio.
- Que la representación de las clases sea la correcta.

En el caso de las Clases, se puede encontrar tres clasificaciones que se deben tener en cuenta, de estas clasificaciones es necesario verificar que la funcionalidad que representan dentro del modelo. A continuación se muestran las clasificaciones y las funcionalidades de las mismas.

Clase Interfaz <<Interface>>:

- Recepcionar peticiones al sistema.
- Mostrar respuestas del sistema.

Clase Entidad <<Entity>>:

- Gestionar datos (información) necesaria para el sistema.
- Almacenar datos (información) persistentes del sistema.

Clase Controlador <<Controller>>:

- Procesar Información del sistema.
- Gestionar visualización de respuesta del sistema.

Relaciones:

- Verificar que la identificación de las relaciones sea la correcta.
- Verificar que la representación gráfica coincida con el tipo de relación identificada.

Atributos:

- Verificar que la identificación de los atributos sea la correcta, según los requerimientos que son necesarios recordar para que se ejecute el Caso de Uso.

Casos de Uso:

- Comprobar el nombre de los Casos de Uso.
- Comprobar el nivel de los Casos de Uso.
- Comprobar que el Caso de Uso este relacionado con al menos un actor, a menos que sea abstracto.

2.4.4.4 Modelo de Diseño.

Los elementos que conforman este modelo han sido medidos en el Modelo de Análisis. Si no se llegara a hacer Modelo de Análisis y se tiene que medir el Modelo de Diseño este se hace de la misma forma como en el Modelo de Análisis, consultar epígrafe 2.4.4.3.

2.4.4.5 Diagrama de Caso de Uso del Negocio.

Para estos diagramas es necesario tener en cuenta el número de propiedades a revisar, así como la cantidad de Casos de Uso del Negocio con los que cuenta el Software. Para este caso se tienen en cuenta las propiedades definidas por la Ayuda del Rational. A continuación se mostrarán las mismas.

Casos de Uso del Negocio:

- Comprobar el nombre del Caso de Uso.
- Comprobar que el Caso de Uso esté representando un conjunto de actividades.
- Comprobar que el Caso de Uso produce un resultado observable para ciertos actores del negocio.

Actores del Negocio:

- Comprobar que el nombre del actor exprese un rol dentro del negocio.
- Comprobar que el actor permanezca fuera de las fronteras del negocio que se investiga.

Relaciones:

- Comprobar la identificación de las relaciones.

- Comprobar que la representación gráfica coincida con el tipo de relación identificada.

2.4.4.6 Casos de Uso del Sistema.

Al igual que en los Casos de Uso del Negocio en los Casos de Uso del Sistema se tienen en cuenta las propiedades definidas por la Lista de Chequeo del Proyecto de Calidad de la Facultad 3, a continuación se enumeran las mismas:

Casos de Uso del Sistema

- Comprobar el nombre de los Casos de Uso.
- Comprobar el nivel de los Casos de Uso.
- Comprobar que el Caso de Uso este relacionado con al menos un actor, a menos que sea abstracto.
- Verificar la especificación de la Complejidad del Caso de Uso.

Actores del Sistema:

- Comprobar que se especifique que el actor inicializa el Caso de Uso.
- Comprobar que si existen dos actores interactuando con un Caso de Uso se debe generalizar en uno.

Relaciones:

- Comprobar la identificación de las relaciones.
- Comprobar que la representación gráfica coincida con el tipo de relación identificada.

2.4.4.7 Documentos.

En este caso se utiliza el formato estándar para aplicarles el método el definido para las plantillas del expediente de proyecto: Arial 11, 1,5 de interlineado, márgenes izquierdo y derecho 2,5. Esto permitirá que el método propuesto para medir estos artefactos sea independiente del formato.

2.4.5 Elaboración de las fórmulas.

Luego de tener definida las características generales de cada clase, es necesario tener en cuenta los aspectos particulares de cada producto dentro de estas clases para lograr una precisión más exacta de la medida del tamaño de este.

2.4.5.1 Obtención de las variables.

A continuación se muestran los aspectos a tener en cuenta para la realización de las fórmulas:

1. Búsqueda de las variables consideradas como la medida básica de cada producto, son aquellas que se obtienen fácilmente y caracterizan los diferentes tipos de clases, por ejemplo en el caso de los documentos puede estar dada por la cantidad de caracteres con espacio, la cantidad de caracteres sin espacio, el número de palabras, y en los gráficos por el total de cada uno de los elementos que integran el diagrama.
2. Selección de las variables constantes dentro del producto: el valor de estas variables puede variar de acuerdo a las propiedades que tengan definidas los diferentes proyectos en las listas de chequeo que se utiliza en el caso de los gráficos, y en los documentos estas variables dependerán del formato que se utilice y la definición de variables como: la cantidad de caracteres por líneas, el número de líneas por páginas, etc. Antes de definir el valor de cualquier variable se revisa la lista de chequeo, que no es más que la cantidad de propiedades que van a ser revisadas por cada elemento, entre más propiedades se definan, más grande será el tamaño del gráfico, y el formato que estará dado por el tamaño de letra, margen e interlineado.
3. Búsqueda de las variables intermedias: son aquellas que el valor solo se puede obtener a partir del cálculo de las variables definidas anteriormente, y generalmente son las variables que se utilizan para determinar la medida del tamaño de los diferentes productos de prueba, ejemplo: total de aspectos a revisar, cantidad de páginas reales.

Para ampliar mas sobre este punto se puede dirigir al epígrafe donde se proponen los métodos, 2.5

2.4.5.2 Aplicación piloto.

En este paso las fórmulas obtenidas (nuevo método) y las medidas de tamaño usadas actualmente (medidas usuales), se aplican a un conjunto de documentos y gráficas que abarcan todas clases obtenidas, registrando todos los datos resultantes del proceso.

Aquí se toma para la aplicación en el caso de los documentos una Conferencia y a este documento se le mide el tamaño como en la actualidad se hace y luego se mide con los métodos que se proponen en este trabajo.

Medidas que se utilizan actualmente para medir los documentos:

- Peso.
- Cantidad de páginas.
- Palabras.
- Cantidad de Líneas.

En el casos de los gráficos se tomaron dos ejemplos de diagramas de diseño, a estos al igual que el caso del documento se le aplicaron el nuevo método y las medidas usuales.

Las medidas usuales para el caso de los gráficos fueron:

- Cantidad de elementos totales.
- Peso.

En la mayor parte de los artefactos que son documentos se tienen gráficas o diagramas, en estos casos se le hace la medición a cada uno de las clases y la medida del artefacto será la obtenida de cada clase.

Este estudio es solo un ejemplo de una aplicación piloto, en el Capítulo 3 se amplía mejor como hacer aplicaciones de este tipo.

2.4.5.3 Validación.

Después de aplicar el nuevo método y las medidas usuales a los documentos y gráficas, la información recolectada de la comparación del resultado de esta aplicación dio como resultado que con el nuevo método se obtienen medidas más exactas que con lo obtenido con las medidas usuales.

Si por lo contrario, la información obtenida diera como resultado que el nuevo método es menos efectivo que las medias usuales se tiene que regresar al punto 2.4.5.1 a la obtención de nuevas variables y continuará con los pasos que le siguen después de ese punto.

En el Capítulo 3 se amplía como hacer las validaciones de este tipo, aunque las abordadas aquí enfocan la investigación completa y dan una visión útil del proceso.

2.4.6 Formalizar la propuesta.

Para formalizar la propuesta después de haber tenido como resultado de la validación que esta responde al problema que fue trazado, los métodos tendrán un formato estándar cuando se plantean. Este es el siguiente:

- Nombre del método.
- Medidas Básicas.
- Variables Constantes.
- Valor de las variables constantes.
- Variables Intermedias.

En algún que otro casos también se deben tomar variables auxiliares, como puede ser las variables binarias.

2.4.7 Aplicar los métodos propuestos.

Después de validados los métodos propuestos en este trabajo se puede comenzar su aplicación en los proyectos de la Facultad 3. En la medida que pase el tiempo se recopilará información la cual mostrará

una vez más que estos métodos son más seguros para determinar el tamaño del artefacto y evitará que se comentan una serie de errores en los proyectos que actualmente ocurren en estos.

Como ya se vio anteriormente en el punto 2.3.7 existen factores que pudieran obligar a que se revisen las medidas con el objetivo de su perfeccionamiento, en cuyo caso los pasos de proceso propuesto han de repetirse.

2.5 Métodos propuestos.

2.5.1 Método para la medición de código del Software.

La dimensión del código actualmente utiliza varios métodos para medirse, entre ellos: las líneas de código, los puntos de características, los puntos de función, entre estos existen variantes como los puntos de función IFPUG, FPA, MK II-FPA. A continuación se definen las variables y fórmula para obtener el tamaño del código.

Medidas Básicas:

NLT: número de líneas totales.

NLC: número de líneas comentariadas.

NLB: número de líneas en blanco.

Variables Intermedias:

TLOC: total de líneas de código.

Fórmula General:

$TLOC = NLT - (NLC + NLB).$

Análisis del Resultado:

El Total de las Líneas de Código (TLOC) es la medida que se utiliza para saber la dimensión del programa, el valor que tenga TLOC es la cantidad de líneas reales que se deben tener en cuenta cuando se utilice este valor como punto de partida para futuras estimaciones de tiempo y esfuerzo.

2.5.2 Método para la medición de los gráficos del Software.

Los gráficos constituyen una parte esencial en la Ingeniería de Software, pues los mismos son una abstracción del producto final. En cada una de las fases del desarrollo del Software se generan diferentes diagramas, los que sirven de base para las fases posteriores. A través de estas representaciones se logra en la mayoría de los casos un mejor entendimiento del usuario final hacia el trabajo realizado por parte de los desarrolladores. La medición de estos gráficos es de relevante importancia en el proceso de liberación del software, pues se logra una estimación más precisa, y esto conlleva a que se planifique mejor el tiempo y el esfuerzo de los ingenieros que intervienen en el proceso de prueba. A continuación se verán los elementos generales a tener en cuenta para medir los diagramas, además se mostrará las fórmulas para casos particulares.

Medidas Básicas:

TE: total de elementos.

Constante:

NP: número de propiedades.

Valor de la Constante:

NP= a las propiedades a revisar por cada elemento dependiente de la bibliografía que se utilice.

Variables Intermedia:

TAR: total de aspectos a revisar.

Fórmula General:

$TAR = NP_0 * TE_0 + NP_1 * TE_1 + \dots + NP_N * TE_N.$

Análisis del Resultado:

El Total de Aspectos a Revisar (TAR) es la medida que se utiliza para saber el tamaño de cada uno de los gráficos que se revisan durante el proceso de prueba, el valor de TAR dependerá del total de elementos, que serían aquellos que intervienen en cada uno de los diferentes diagramas, y el número de

propiedades, que como se define anteriormente depende de la bibliografía que se utilice para definir lo que debe tener cada elemento.

2.5.2.1. Medición del Modelo Entidad-Relación

Medidas Básicas:

TE: total de entidades.

TA: total de atributos.

TR: total de relaciones.

Constantes:

NPE: número de propiedades de las entidades.

NPA: número de propiedades de los atributos.

NPR: número de propiedades de las relaciones

Valor de las Constantes:

NPE= 4

NPA= 2

NPR= 2

Variable Intermedia:

TAR: total de aspectos a revisar.

Fórmula General:

$TAR = NPE * TE + NPA * TA + NPR * TR.$

Luego se sustituyen los valores de las propiedades:

$TAR = 4 * TE + 2 * TA + 2 * TR.$

2.5.2.2. Medición del Diagrama de Despliegue.

Medidas Básicas:

TN: total de nodos.

TC: total de componentes.

TR: total de relaciones.

Constantes:

NPN: número de propiedades de los nodos.

NPC: número de propiedades de los componentes.

NPR: número de propiedades de las relaciones.

Valor de las Constantes:

NPN = 4

NPC = 2

NPR = 1

Variable Intermedia:

TAR: total de aspectos a revisar.

Fórmula General:

$TAR = NPN * TN + NPC * TC + NPR * TR$

Luego se sustituye el valor de las propiedades:

$TAR = 4 * TN + 2 * TC + 1 * TR.$

2.5.2.3. Medición del Modelo del Análisis.

Medidas Básicas:

TA: total de atributos.

TR: total de relaciones.

TCI: total de clases interfaz.

TCC: total de clases controladoras.

TCE: total de clases entidades.

TCU: total de Casos de Uso.

Constantes:

NPA: número de propiedades de los atributos.

NPR: número de propiedades de las relaciones.

NPC: número de propiedades de las clases.

NPCI: número de propiedades de las clases interfaz.

NPCC: número de propiedades de las clases controladora.

NPCE: número de propiedades de las clases entidades.

NPCU: número de propiedades de los Casos de Uso.

Valor de las Constantes:

NPA = 1

NPR = 2

NPC = 3

NPCI= 2

NPCC= 2

NPCE= 2

NPCU= 3

Variables Intermedias:

TC: total de clases del Análisis.

TAR: total de aspectos a revisar.

Fórmula General:

$$TC = NPCI*TCI + NPCC*TCC + NPCE*TCE.$$

Luego:

$$TAR= NPC*TC + NPCU*TCU + NPA*TA + NPR*TR$$

Luego se sustituye el valor de las propiedades:

$$TAR= 3*TC + 3*TCU + 1*TA + 2*TR$$

2.5.2.4. Medición del Modelo del Diseño.

En el caso particular del Modelo del Diseño, no es necesario definir nuevamente la fórmula, pues los elementos que conforman este modelo han sido medidos en el Modelo de Análisis, por lo que definimos que el Total de Aspectos a Revisar (TAR), va a tener el mismo valor que en el caso anterior. En caso de que no se haga el Modelo de Análisis, pues la fórmula para saber el tamaño de los Aspectos a Revisar en el Diseño, sería igual que en el Análisis. Consultar en caso de duda el epígrafe 2.5.2.3.

2.5.2.5. Medición de los Diagramas de Casos de Uso del Negocio.

Medidas Básicas:

TCU: total de Casos de Uso.

TA: total de actores.

TR: total de relaciones.

Constantes:

NPCU: número de propiedades de los Casos de Uso.

NPA: número de propiedades de los actores.

NPR: número de propiedades de las relaciones.

Valor de las Constantes:

NPCU= 3

NPA= 2

NPR= 2

Variable Intermedia:

TAR: total de aspectos a revisar.

Fórmula General:

$TAR = NPCU * TCU + NPA * TA + NPR * TR$

Sustituyendo el valor de las propiedades:

$TAR = 3 * TCU + 2 * TA + 2 * TR$

2.5.2.6. Medición de los Casos de Uso del Sistema

Medidas Básicas:

TCU: total de Casos de Uso.

TA: total de actores.

TR: total de relaciones.

Constantes:

NPCU: número de propiedades de los Casos de Uso.

NPA: número de propiedades de los actores.

NPR: número de propiedades de las relaciones.

Valor de las Constantes:

NPCU= 4

NPA= 2

NPR= 2

Variable Intermedia:

TAR: total de aspectos a revisar.

Fórmula General:

$TAR = NPCU * TCU + NPA * TA + NPR * TR$

Sustituyendo el valor de las propiedades:

$TAR = 4 * TCU + 2 * TA + 2 * TR$

2.5.3 Medición de los Documentos del Software.

La documentación en el Software juega un papel importante, pues en ellos se explican todos los procesos por los cuales ha pasado el producto, como ha sido confeccionado, y los elementos que se tuvieron en cuenta para dar la solución de acuerdo a los requisitos funcionales y no funcionales recogidos por el Analista del Negocio. A continuación se relacionan las variables y propiedades a tener en cuenta:

Medida Básicas:

CCE: cantidad de caracteres con espacio.

Constantes:

NLP: número de líneas por páginas.

CCL: cantidad de caracteres por líneas.

Valor de las Constantes:

NLP= 37 líneas por páginas, este número depende del formato que se tomo de estándar.

CCL= 85 caracteres por líneas, este número es un promedio teniendo en cuenta la cantidad de líneas por páginas dependientes del formato de redacción.

Variable Binaria:

X= 0 si el redondeo es por exceso, pues al sumarle uno al número entero se tienen en cuenta los elementos restantes de la división para obtener el total de páginas.

X= 1 si el redondeo es por defecto, pues no se tienen en cuenta los elementos restantes de la división para obtener el total de páginas.

Variables intermedias:

CLT: cantidad de líneas totales

CPT: cantidad de páginas totales

Fórmula General:

$CLT = CCS/CCL$

$CPT = CLT/NLP + X$

2.6 Conclusiones.

En este capítulo se proponen los pasos necesarios para obtener los métodos de medición de tamaño de productos de prueba, además se presentaron dos de los métodos que más se utilizan en el mundo para llevar a cabo la medición del Software y su relación con la presente investigación. Se realiza la clasificación de los artefactos a medir en el proceso de prueba, y atendiendo a estas clasificaciones se hizo un estudio de las características propias de cada uno de ellos, y por último teniendo en cuenta dichas características, se formula una propuesta de los diferentes métodos de medición para determinar el tamaño de los productos de prueba ajustados las necesidades de la Facultad 3.

CAPITULO III. EVALUACIÓN DE LA PROPUESTA.

3.1. Introducción.

En la investigación, se realiza el estudio del proceso de medición, se analiza la importancia de medir los productos, se identificaron los artefactos entregables en el proceso de pruebas y los métodos más usados para la medición de los productos del desarrollo de software, principalmente en los proyectos de la Universidad. En el presente capítulo para la validación de los métodos propuestos en el Capítulo 2, se toma como herramienta de investigación el estudio de caso y se realiza una descripción de los métodos utilizados y los resultados obtenidos.

3.2. Método de Estudio de Caso.

Este permite realizar descripciones muy detalladas, a menudo personales, obteniendo los datos que tipifican a un prototipo que representa a un grupo o comunidad, bien sea de carácter social, de un objeto, de un fenómeno, mediante esto se configura una serie de resultados que de acuerdo a la experiencia del investigador, ejemplifique de una manera contundente lo que se pretende mostrar como problema y sus alternativas de solución.

Objetivo y Meta:

El estudio de caso tiene como objetivo principal aplicar los instrumentos teóricos asimilados a un caso particular o específico, que pueda atribuirse al ámbito donde labora el discente.

Estrategia:

Lo relevante es, al final proponer opciones de solución, medidas preventivas y correctivas a la estructura y procesos del objeto analizado; se pretende pues, prever, rectificar y mejorar su funcionamiento en general, haciéndolo más eficiente, productivo y de calidad.

Para tal efecto, se hará un análisis comparativo, ya sea explícito o implícito, que muestre claramente la situación actual y a su vez proyecte las perspectivas futuras de cómo funcionaría mejor el ente u objeto

estudiado, acorde con las nuevas tecnologías en la materia, de la disciplina y las disciplinas concurrentes en el entorno.(NACION 2007)

3.3. Selección de Conceptos.

Métricas de esfuerzo: se utilizan para hacer las estimaciones del proyecto. La unidad de medida de estas métricas va a estar dada por la cantidad de personas y tiempo que se necesite para revisar el producto.

El tiempo y la cantidad de personas son medidas directamente proporcionales, es decir, si aumenta el tiempo de revisión, se necesitan más personas para llevar a cabo el proceso de pruebas. El tiempo de revisión depende del tamaño de los productos, y el tamaño de los productos va a estar dado por la efectividad de las medidas.

Efectividad de las medidas: va a estar dada por la independencia de las medidas con las características del producto, que no influye en el esfuerzo para las pruebas, y el reflejo del esfuerzo, que va a estar dado por una relación proporcional con los elementos a tener en cuenta para la revisión en las pruebas.

Métodos de medición: va a estar dado por la facilidad de uso de las medidas de los productos, y este está dado por la cantidad de pasos que se necesitan para obtener la medida del tamaño de los productos del desarrollo del Software.

3.4. Selección de Variables.

En este epígrafe se hace la selección de las variables y medidas tanto las usuales como las que se propondrán en la investigación, las cuales ayudan a la obtención final de los métodos propuestos en este trabajo. En la tabla 4 se muestra la operacionalización de estas variables.

Variable Independiente = Métodos de medición.

Variable Dependiente = Desarrollo de métricas de esfuerzo.

Medidas Usuales para los documentos y gráficos:

- Cantidad de Páginas.
- Peso (KB).

Medidas Propuestas en la tesis para los documentos.

- Cantidad de páginas reales.

Medidas Propuesta en la tesis para los gráficos.

- Total de aspectos a revisar.

Tabla 4: Operacionalización de variables.

Variables	Indicadores	Sub-Indicadores	Sub-Indicadores	Unidades	Observaciones	
Métodos de medición.	Facilidad de Uso.	Posibilidad de obtención automática	Peso	Mega	Se puede obtener automáticamente	
			Cant Usuales	Pág.	Valor/10	Se puede obtener automáticamente
			Aspectos a revisar		Valor/10	No se puede obtener automáticamente
			Cant Reales	Pág.	Valor	No se puede obtener automáticamente

CAPÍTULO 3: EVALUACIÓN DE LA PROPUESTA.

Variables	Indicadores	Sub-Indicadores	Sub-Indicadores	Unidades	Observaciones
		Las medidas usuales pueden obtenerse automáticamente	Cant Caracteres	Valor/1000	Si se tiene definidas medidas usuales se le pone el valor que tiene en el documento, de lo contrario el valor es cero.
			Peso	Mega	
			Cant Usuales	Pág. Valor/10	
			Total Elementos	de Valor/10	
	Robustez	Cantidad de elementos básicos.	Tamaño del Gráfico	Valor	Si se tiene en cuenta se le pone el valor que tiene en el documento, de lo contrario se le pone cero a dicho valor.
			Aspectos a revisar	Valor/10	
			Cant Caracteres	de Valor/1000	
			Formato	Valor	
			Cant Reales	Pág. Valor	
			Cant Usuales	Pág. Valor/10	
			Peso Usual	Valor/100	
			Núm. Líneas por	Valor/10	

Variables	Indicadores	Sub-Indicadores	Sub-Indicadores	Unidades	Observaciones
			Pág.		
			Cant Caracteres por Línea	Valor/10	
		Relación entre la medida total y las medidas usuales	Cant de Caracteres	Valor/1000	Establecer la relación que existe entre la medida general del documento, y las medidas que se tienen definidas como básicas.
			Aspectos a revisar	Valor/10	
			Cant Pág. Usuales	Valor/10	
			Peso	Valor/100	
			Cant Pág. Reales	Valor	
			Total de Elementos	Valor/10	
Desarrollo de métricas de esfuerzo.	Efectividad	Independencia con el formato	Tamaño de Letra	Valor/10	Si se tiene en cuenta se le pone el valor que tiene en el documento, de lo contrario se le pone cero a dicho valor.
			Margen	Valor	
			Interlineado	Valor	

Variables	Indicadores	Sub-Indicadores	Sub-Indicadores	Unidades	Observaciones
		Reflejo del esfuerzo	Núm. de propiedades	Valor	Si se tiene en cuenta se le pone el valor que tiene en el documento, de lo contrario se le pone cero a dicho valor.
			Total de Elementos	Valor/10	

3.5. Clasificación de artefactos y construcción de casos de estudio.

En el Capítulo 2 se muestra una serie de pasos para obtener los métodos propuestos, entre unos de los pasos estaba la identificación y decantación donde como resultado se obtuvieron 19 artefactos que son entregados al Laboratorio de Calidad de la Facultad 3. Estos luego se clasificaron en 3 clases para un mayor entendimiento, estas clases fueron: documento, código y gráfico.

En la figura 3 se muestra el porciento de artefactos por cada clase.

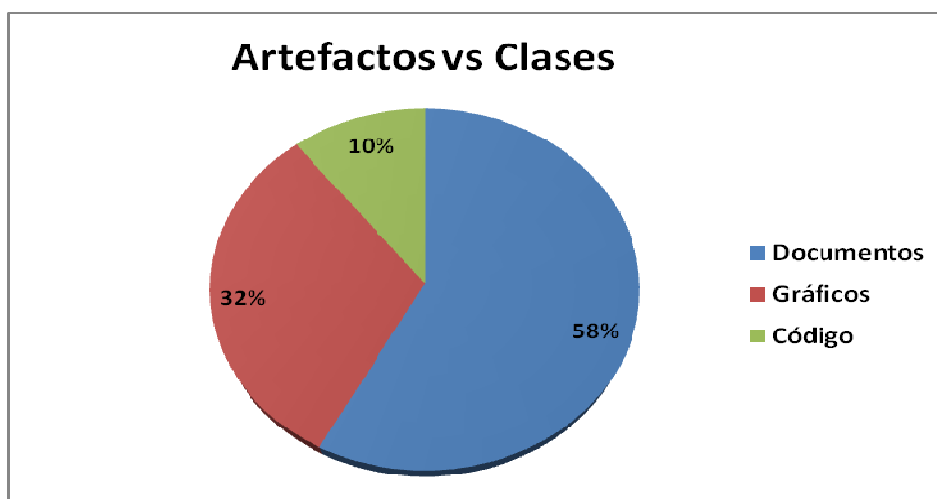


Figura 3. Porciento de artefactos por clases.

Como se puede ver en la figura 3 existe una mayor porción de documentos, prácticamente un 60%, y gráficos más de un 30%, que de código, solo un pequeño porcentaje están agrupados dentro de esta clase. En el capítulo anterior se observaron los pasos a seguir para la obtención de los métodos propuestos por los investigadores, para aplicar estos métodos se escogieron dos casos de estudio.

Tabla 4. Relación existente entre las clases y los casos.

Casos de Estudio	Características a tener en cuenta para medir.	Clase		
		Documento	Gráfico	Código
Caso 1	Formato del documento	X	--	--
Caso 2	Formato del documento	X	X	--
	Propiedades de los gráficos			

La tabla 4 ayuda a entender cuáles son los elementos a valorar en cada caso de estudio, esta aplicación ayuda a los investigadores a validar la propuesta de solución que se obtuvo en el Capítulo 2.

En el primer caso de estudio solo se analiza como documento por lo tanto las características que se tendrán en cuenta serán las del formato, relacionadas estrechamente con esta clase de artefacto o producto de pruebas. En el segundo caso se tiene en cuenta además del formato, para medir el documento como tal, las propiedades que definen los gráficos contenidos en el documento base del proceso de validación.

En el caso de la validación de método para la medición de la clase de código no se realiza ya que el método de Líneas de Código, se obtuvo en el estudio que se realizó a los métodos que se utilizan en la actualidad el cual ya está validado, y por lo que se tomó como uno más para la propuesta de solución que se hace en esta investigación.

3.6. ¿Cómo se realiza la validación de la propuesta?

Para la validación de la propuesta, como se ha explicado, se usa el método de casos de estudio siguiendo para ello un conjunto de pasos que van desde la selección de los casos, diseño de los experimentos hasta el análisis de los resultados donde se logra evidenciar, con el uso de las variables que definen el problema, la importancia de la propuesta presentada en el capítulo anterior.

3.6.1. Selección de los casos.

Para la selección de los casos de estudio donde se aplica la propuesta de los métodos, se tiene en cuenta las características más significativas de las clases que se estudian, como se explica en la tabla anterior. Estas ideas iniciales sirven de base para el diseño de los experimentos que se ve en el siguiente epígrafe.

3.6.2. Diseño de los experimentos.

En este paso se establece una estructura básica similar para cada experimento donde se refleja con claridad el objetivo del experimento, la descripción de este hasta llegar al análisis de los resultados.

➤ **Objetivo del experimento.**

En este lugar de la estructura del experimento se expresan claramente las metas que se pretenden dentro del diseño del experimento.

Tabla 5: Relación entre las variables del problema y los experimentos realizados.

Variables	Indicadores	Sub-Indicadores	Caso 1			Caso 2	
			Exp1	Exp2	Exp3	Exp1	Exp2
Métodos de	Facilidad de Uso.	Posibilidad de obtención	X	X	X	X	X

Variables	Indicadores	Sub-Indicadores	Caso 1			Caso 2	
			Exp1	Exp2	Exp3	Exp1	Exp2
medición.		automática					
		Las medidas usuales pueden obtenerse automáticamente	X	X	X	X	X
	Robustez	Cantidad de elementos básicos.	X	X	X	X	X
		Relación entre la medida total y las medidas usuales.	X	X	X	X	X
Desarrollo de métricas de esfuerzo.	Efectividad	Independencia con el formato	X	X	X	--	--
		Reflejo del esfuerzo	X	X	X	X	X

La tabla 5 detalla los elementos tenidos en cuenta en cada experimento para que se diera cumplimiento de los objetivos.

El objetivo de cada experimento es confirmar que se cumplen con alguno de los indicadores obtenidos en el epígrafe 3.4, es decir que en cada experimento se verifica si los métodos propuestos tienen, por

ejemplo: la facilidad de uso, aquí se puede ver la posibilidad de su obtención automática de las variables y si las medidas usuales pueden obtenerse automáticamente.

Otro de los objetivos del experimento es comprobar la robustez: constatando si los métodos de medición aquí teniendo en cuenta tanto las medidas usuales y las propuestas.

También se tiene que comprobar y comparar la efectividad de las medidas usuales y las propuestas, aquí teniendo en cuenta la independencia del formato de estas y el reflejo del esfuerzo.

➤ **Descripción del experimento.**

En este punto se explica como realizar el experimento con detalles y los parámetros a variar durante el mismo. De manera general durante la realización del experimento deben quedar claro tres momentos importantes:

➤ **Etapas de las medidas usuales.**

Según la investigación o la experiencia de los investigadores se escogen las medidas usuales para medir el artefacto y se aplican a cada experimento. Estas medidas se pueden observar en el epígrafe 3.4, donde se hizo la selección de estas medidas por clases.

➤ **Etapas de las medidas propuestas.**

En dependencia de la clase que sea el artefacto que se tiene en cuenta para los casos de estudio se propone la medida de este. Estas se observan en el la selección de las variables en el epígrafe 3.4.

➤ **Análisis de los resultados.**

Si en cada experimento se cumple que los métodos propuestos por este trabajo tienen facilidad de uso, son robustos y a su vez son efectivos, se puede decir que se han cumplido con los objetivos del experimento.

3.7. Descripción de los estudios de casos.

3.7.1. Caso 1.

En este caso se tiene como referencia la plantilla de Modelos de Casos de Uso del Módulo AVL de Sistema de Gestión de Emergencias de Seguridad Ciudadana (171), que describe la especificación de los Casos de Uso. El formato que presenta es: Arial 11, 1.5 de interlineado, margen izquierdo y derecho 2,5. A este caso de estudio se le harán tres experimentos en los cuales se analizará como se comportan los indicadores, es decir como varían las medidas usuales y las propuestas por este trabajo. Ver (Anexo 2).

3.7.1.1. Experimento 1.

Objetivo:

La principal tarea de la realización de este experimento es verificar que se han cumplido con los objetivos del mismo, estos objetivos se pueden ver en la tabla 5.

Descripción:

Se tiene una primera versión del documento del Caso 1, en este experimento se varía el formato al Caso 1 este tomando los siguientes valores: letra Arial 10, justificado, márgenes izquierdo y derecho 2,5, interlineado 2.0.

Análisis de resultados:

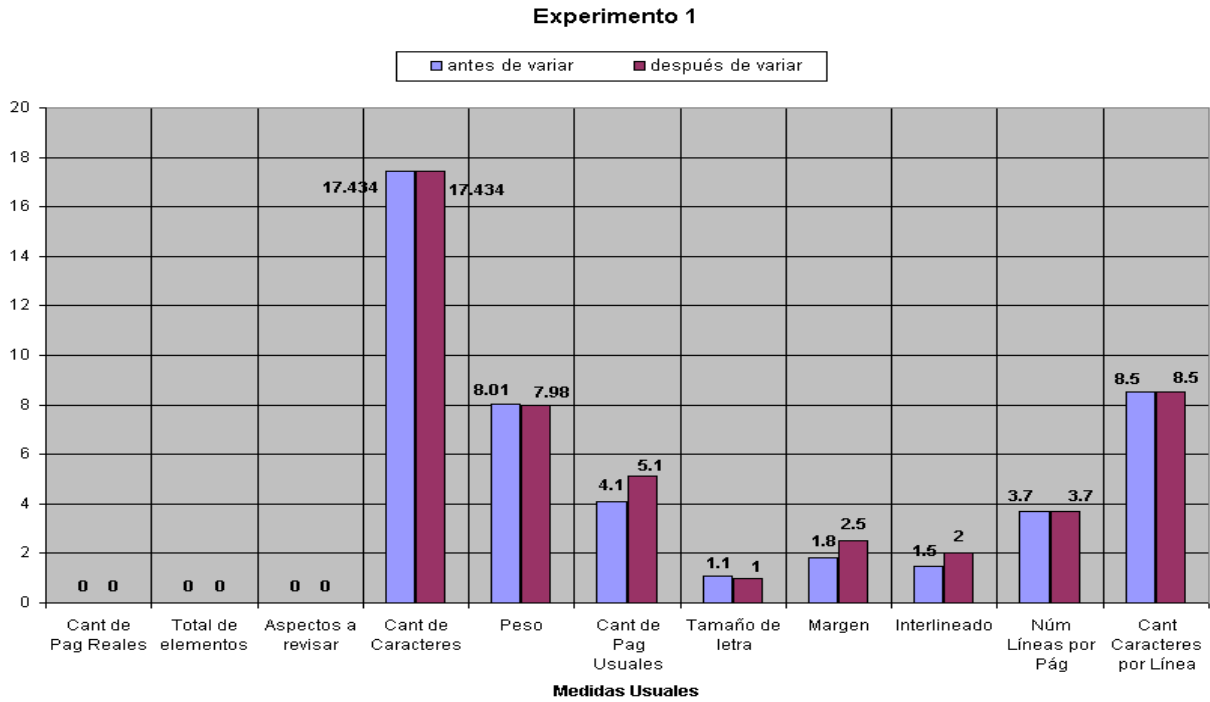


Figura 4. Experimento 1 antes y después de aplicar las variaciones a las medidas usuales

Como se puede observar en la figura 4 se puede ver como varían las medidas usuales al aplicarles las variaciones. Las variables de tamaño de letra, margen e interlineado, solo se grafican para ver los cambios que estos sufren las demás variables al estas ser cambiadas.

Peso:

En este caso disminuye pues esta es una variable que depende del formato y al variar este el peso va a variar también, además es variable que se puede obtener automáticamente.

Cantidad de páginas usuales:

Esta varía de acuerdo a los cambios que se hacen en el formato en el experimento.

Tanto el peso como la cantidad de páginas usuales son medidas básicas que se pueden obtener automáticamente.

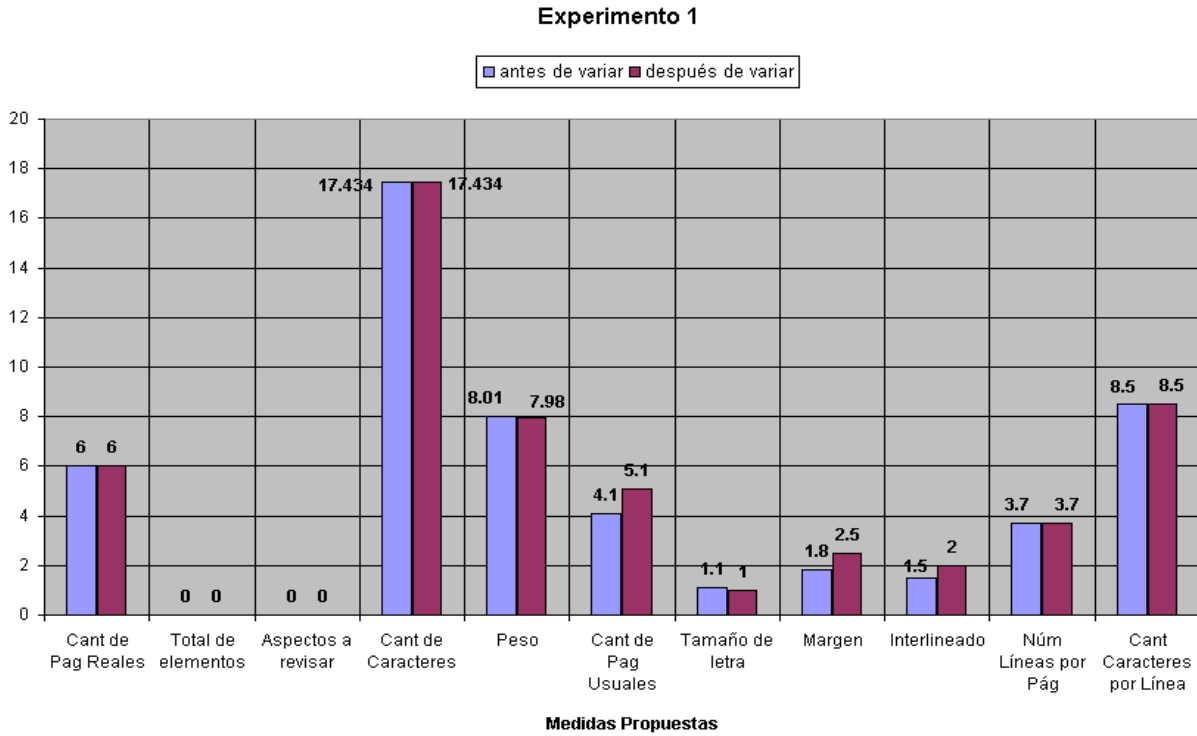


Figura 5. Experimento 1 antes y después de aplicar las variaciones a las medidas propuestas.

En la figura 5 se puede ver como las medidas propuestas en la tesis al aplicarles las variaciones de este experimento que serán en el formato del documento, estas no influye en ninguna de las variables propuestas, pues la cantidad de caracteres, el número de líneas por páginas y la cantidad de caracteres por líneas son variables independientes del tamaño de letra, margen e interlineado que se utilice. Las variables independientes son las que se utilizan para llegar a la medida del documento, en este caso es la cantidad de páginas reales y en este experimento indicador no presenta variación, las variables usuales para determinar el tamaño del documento no se tienen en cuenta dentro de las medidas propuestas, pues son variables dependientes del formato. Por todo lo antes dicho se puede decir que las Cantidad de Pág. Reales como medida básica no se puede obtener automáticamente

3.7.1.2. Experimento 2.

Objetivo:

Al igual que en el primer experimento, en este se comprueba si se cumple con sus objetivos antes planteados mostrados en la tabla 5.

Descripción:

En otra versión del documento del Caso 1 en el cual también se varía el formato el cual tendrá Arial 14, justificado, márgenes izquierdo y derecho 2,0, interlineado 1,5.

Análisis de resultados:

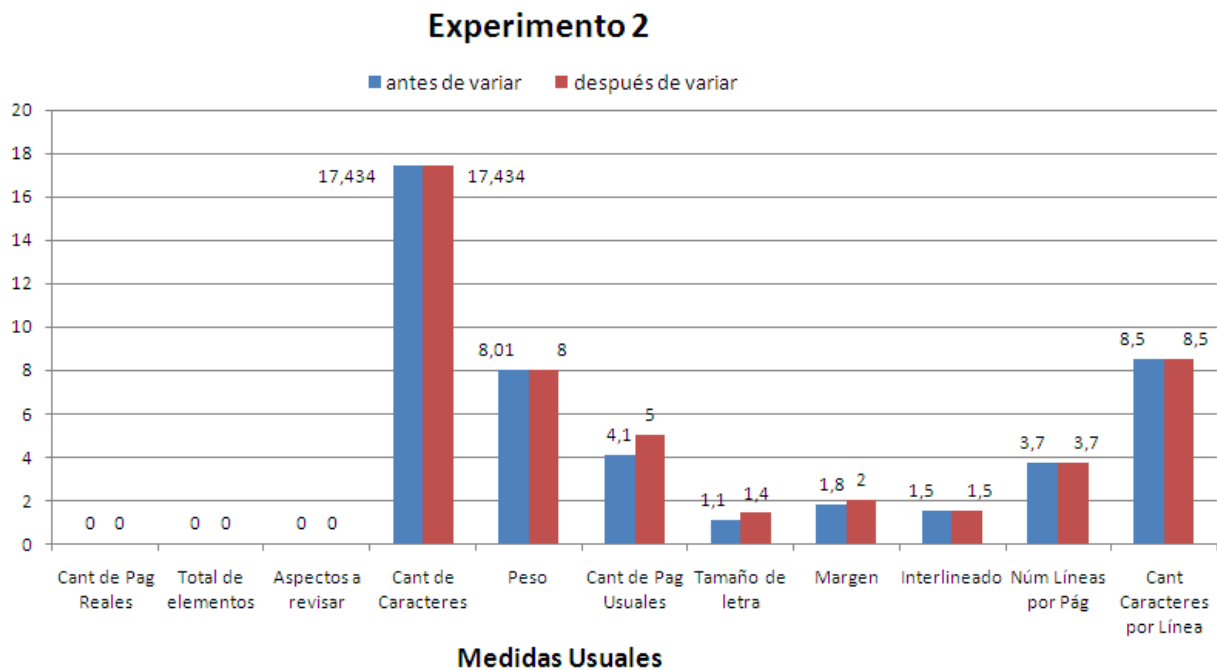


Figura 6. Experimento 2 antes y después de aplicar las variaciones a las medidas usuales.

En la gráfica que se muestra en la figura 6 se puede observar como varían las medidas usuales al aplicarle las variaciones en este experimento, tanto el peso como la cantidad de páginas usuales, pues son indicadores que dependen del formato, en este caso las variación que se le hacen al formato están dadas por: el tamaño de letra, margen e interlineado, por esto es que se muestran en la gráfica estos valores para que sea vea el cambio de estas variables y lo que ayuda en a observar las razones de la variación de las medidas usuales, las cuales a su vez son posibles de obtener automáticamente además de ser medidas básicas del documento.

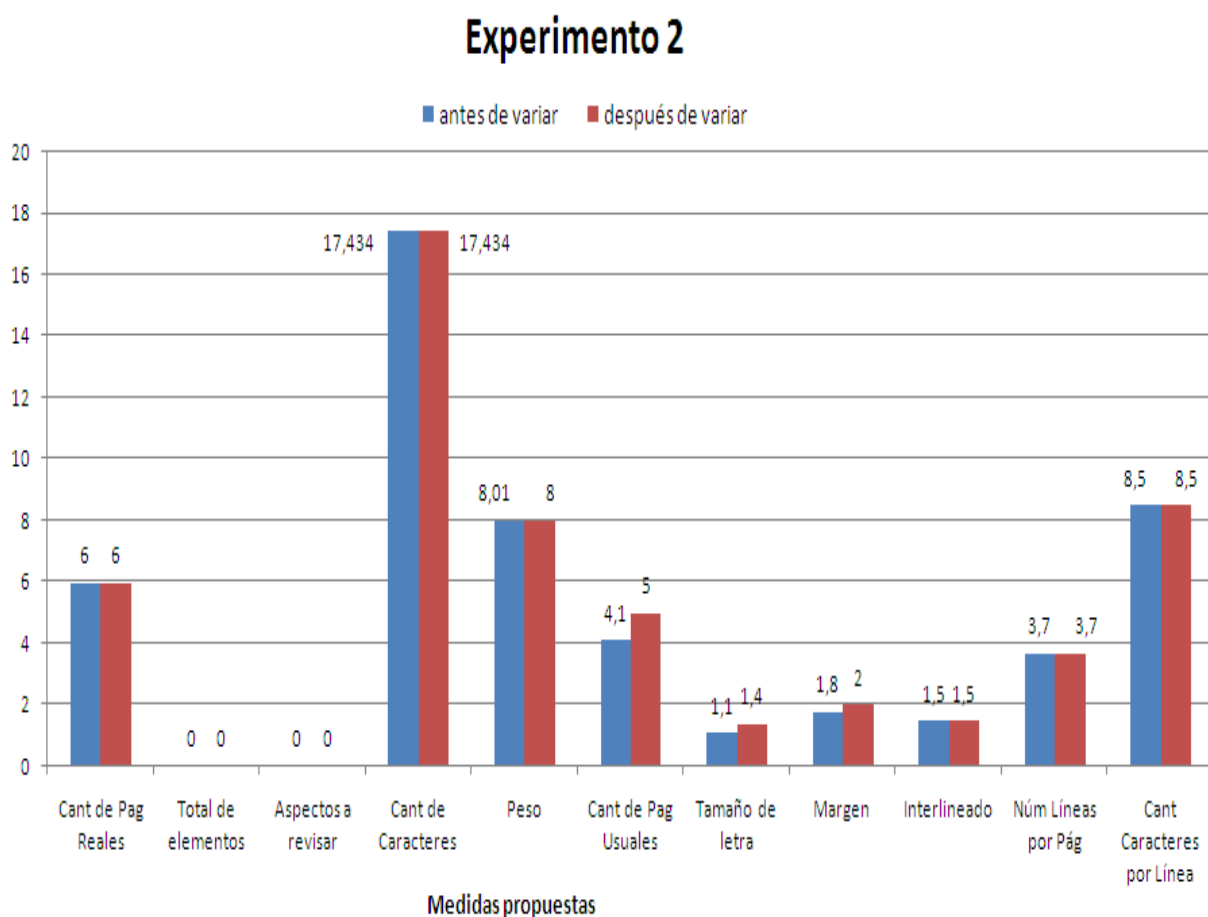


Figura 7. Experimento 2 antes y después de aplicar las variaciones a las medidas propuestas.

En la figura 7 se muestra las variaciones que ocurren cuando se aplica este experimento, en especial con las medidas propuestas por la tesis, pues la cantidad de caracteres, el número de líneas por páginas y la cantidad de caracteres por líneas son variables independientes del tamaño de letra, margen e interlineado que se utilice. Al tener estas variables estas se utilizan para llegar a la medida del documento, siendo en este caso la cantidad de páginas reales, dicho indicador en este experimento no presenta variación, las variables usuales para determinar el tamaño del documento se miden pero no se tienen en cuenta dentro de las medidas propuestas, pues son variables dependientes del formato. Por lo tanto se puede decir que las *Cantidad de Pág. Reales* como medida básica no se puede obtener automáticamente pero que si es una variable independiente del formato y que le da robustez al método, y se puede obtener con mayor eficiencia el tamaño del producto de pruebas.

Para ver el análisis del resultado general se puede remitir al epígrafe 3.8 donde se alcanza percibir que ocurre al hacer las variaciones en este experimento.

3.7.1.3. Experimento 3.

Objetivo:

En este experimento al igual que en los experimentos 1 y 2, tendrá como objetivo verificar si los métodos de medición serán fáciles de usar, robustos y efectivos para determinar el tamaño de los productos de pruebas.

Descripción:

En esta versión del Caso 1 se tiene en cuenta lo mismo que en los experimentos anteriores, el formato este tomará los valores de Arial 12, justificado, márgenes izquierdo y derecho 3, interlineado 3,0.

Análisis de resultados:

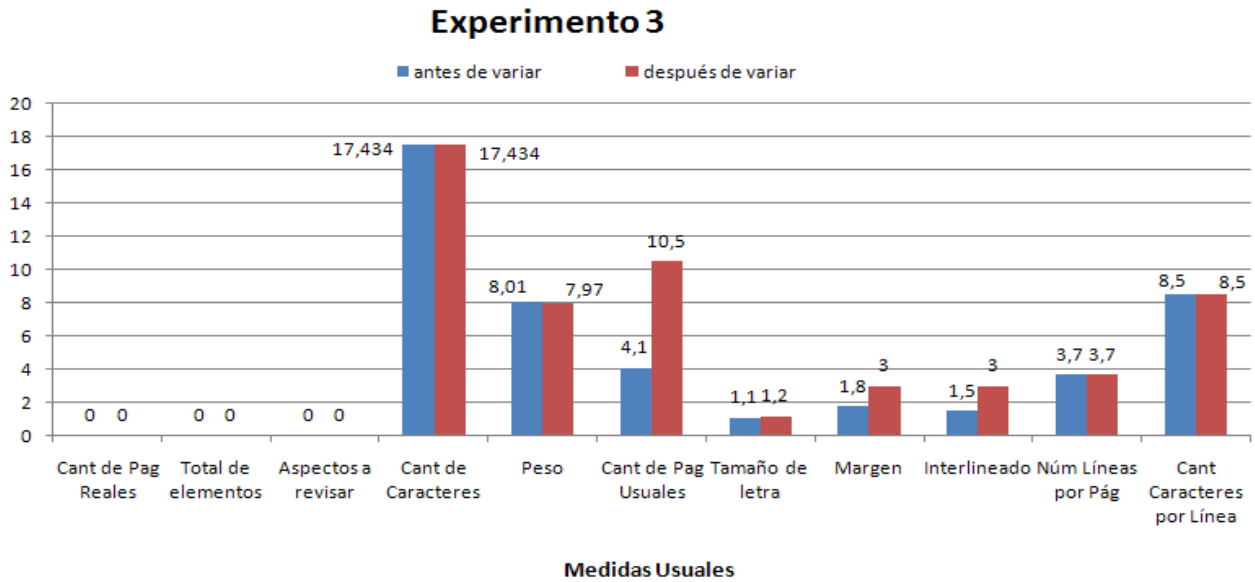


Figura 8. Experimento 2 antes y después de aplicar las variaciones a las medidas usuales

En la figura 8 se muestra la gráfica de cómo se comportan las medidas usuales en el este experimento antes de aplicarles las variaciones y después de aplicárselas. De esta gráfica se llega al siguiente resultado:

Peso:

Este indicador varía al compararlo en los 2 momentos, esto sucede debido a la variación que se le hace al formato.

Cantidad de páginas usuales:

En esta variable también se le observa como varia grandemente al hacer los cambios al formato.

De manera general en este experimento se puede decir que las medidas básicas en este son el peso y la cantidad de de páginas usuales, estas se pueden obtener de manera automática, pero por el contrario estas van depender del formato, mientras este varíe estas medidas seguirán variando por lo que no será una medida efectiva para la obtención del tamaño del artefacto.

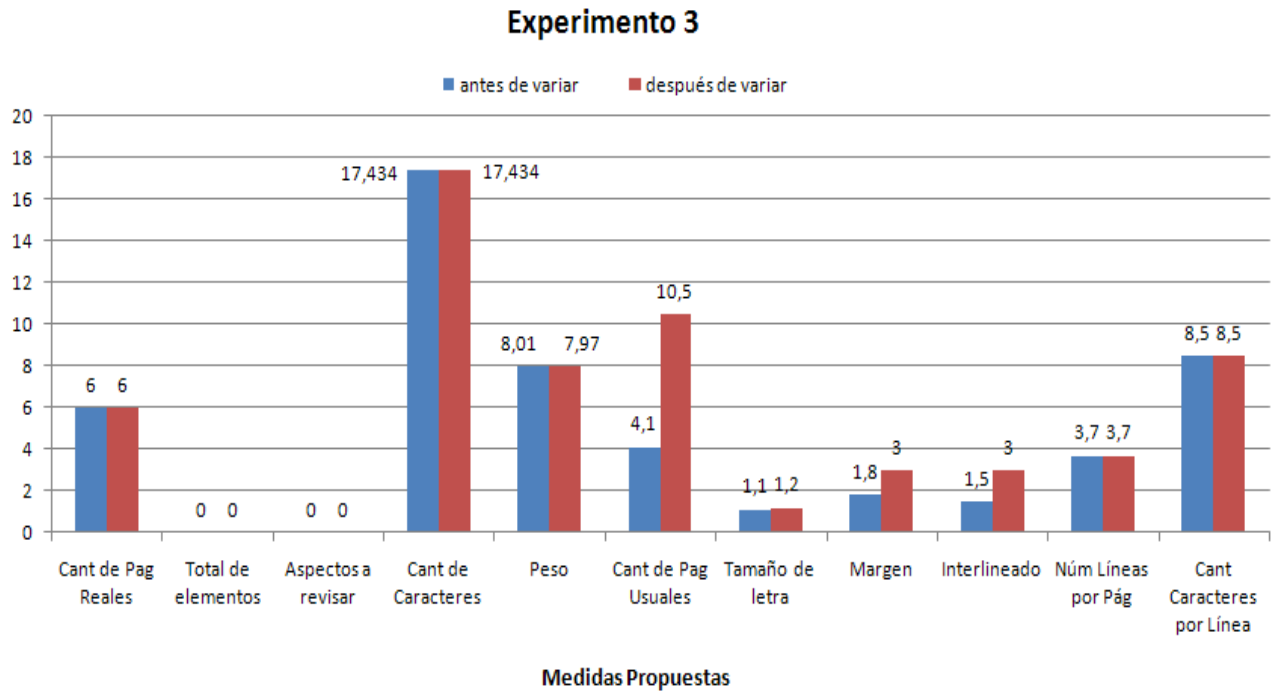


Figura 9. Experimento 2 antes y después de aplicar las variaciones a las medidas propuestas.

En la figura 9 se puede ver como las medidas propuestas al aplicarles las variaciones de este experimento que serán en el formato del documento, estas no influye en ninguna de las variables propuestas, pues la cantidad de caracteres, el número de líneas por páginas y la cantidad de caracteres por líneas son variables independientes del formato del documento. Estas variables independientes se utilizan para llegar a la medida del documento, en este caso es la cantidad de páginas reales y en este experimento indicador no presenta variación, las variables usuales para determinar el tamaño del documento no se tienen en cuenta dentro de las medidas propuestas, pues son variables dependientes del formato, pues como se puede observar en la gráfica estas varían.

Por todo lo antes dicho se puede decir que las Cantidad de Pág. Reales como medida básica no se puede obtener automáticamente ya que para su obtención se necesita de otros indicadores. También se puede decir que este indicador le va a ayudar a que el método propuesto sea más robusto y eficaz pues obtiene

una medida más exacta del tamaño del artefacto ya que esta no va a depender del formato haciéndolo más eficiente.

El análisis de los resultados obtenidos se encuentra más explícitamente en el epígrafe 3.8 donde se puede observar que ocurre al hacer las variaciones.

3.7.2. Caso 2.

En este caso se tiene como referencia la plantilla de Modelos de Casos de Uso del Módulo AVL de Sistema de Gestión de Emergencias de Seguridad Ciudadana (171), el formato que presenta es: Arial 11, 1.5 de interlineado, margen izquierdo y derecho 2,5. En este caso de estudio al igual que en el Caso 1 se harán dos experimentos para poder analizar y constatar que los métodos propuestos son efectivos, por lo que se hará una comparación de las Medidas Usuales y las propuestas, y se verán como estas se comportan al realizarle las variaciones correspondientes en cada experimento.

3.7.2.1. Experimento 1.

Objetivo:

En este experimento se analizarán y constatarán que los métodos propuestos son efectivos, fáciles de usar y robustos, por lo que se hará una comparación de las medidas usuales y las propuestas por la investigación.

Descripción:

En esta versión se tiene que las propiedades de los elementos que conforman los Casos de Uso se manejarán por las propiedades definidas en la Ayuda del Rational.

Propiedades de la Ayuda del Rational:

Casos de Uso del Sistema:

- Comprobar el nombre de los Casos de Uso.
- Comprobar el nivel de los Casos de Uso.

- Comprobar que el Caso de Uso este relacionado con al menos un actor, a menos que sea abstracto.
- Verificar la especificación de la Complejidad del Caso de Uso.

Actores del Sistema:

- Comprobar que se especifique que el actor inicializa el Caso de Uso.
- Comprobar que si existen dos actores interactuando con un Caso de Uso se debe generalizar en uno.

Relaciones:

- Comprobar la identificación de las relaciones.
- Comprobar que la representación gráfica coincida con el tipo de relación identificada.

3.7.2.2. Experimento 2.

Objetivo:

En este experimento, al igual que en el primer experimento de este caso, se comprobarán los resultados de la comparación de las medidas usuales y las propuestas al medir el tamaño del artefacto, así demostrando que se cumplen los objetivos de dicho experimento.

Descripción:

En esta versión se tiene que las propiedades de los elementos que conforman los Casos de Uso se manejarán por las propiedades definidas en la lista de chequeo del proyecto de Calidad de la Facultad 3.

Lista de Chequeo del Proyecto de Calidad:

Casos de Uso del Sistema:

- Comprobar que el nombre del Caso de Uso sea infinitivo.
- Comprobar que el nombre del Caso de Uso sea único.
- Comprobar que el nombre del Caso de Uso sea explicativo.

- Comprobar el nivel de los Casos de Uso.
- Comprobar que el Caso de Uso este relacionado con al menos un actor, a menos que sea abstracto.

Actores del Sistema:

- Comprobar que se especifique que el actor inicializa el Caso de Uso.
- Comprobar que el actor se relacione al menos con un Caso de Uso.
- Comprobar que si existen dos actores interactuando con un Caso de Uso se debe generalizar en uno.

Relaciones:

- Comprobar la identificación de las relaciones.
- Comprobar que la representación gráfica coincida con el tipo de relación identificada.

3.8. Análisis de resultado de los Estudios de Casos.

En este epígrafe se hace un análisis profundo de los resultados al realizar los experimentos en cada caso de estudio.

3.8.1. Estudio del Caso 1.

3.8.1.1. Experimento 1,2 y 3 con las medidas usuales.

A continuación, en la figura 4, se muestra un grafico de cómo se comportan las medidas usuales antes de hacer las variaciones en cada una de los experimentos realizados al caso 1 y después de realizar cada una de estos que es lo que sucedo en cada caso.

Caso 1

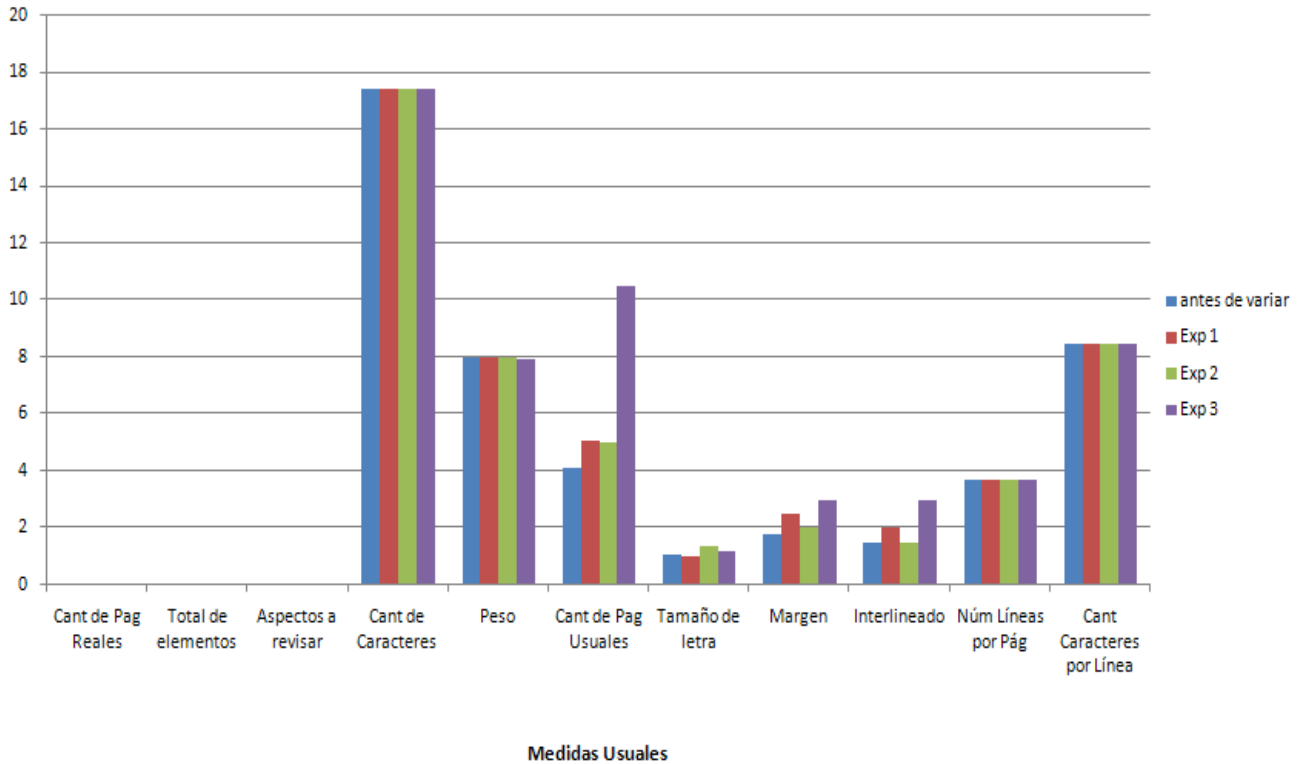


Figura 4. Comportamiento de las medidas usuales en los experimentos del caso 1

Análisis del resultado:

Cantidad de caracteres:

En la gráfica se observa que esta variable varía al hacerle los cambios que se produjeron en los tres experimentos en el formato: el cual está dado por el tamaño de letra, margen e interlineado, esto hace la utilización lógica de esta variable como base para el cálculo del tamaño.

Peso:

Esta variable se ve que al realizar los tres experimentos, por lo cual esta variable se hace dependiente del formato que el documento tenga y de este se deriva que no se pueda tomar como una variable para la

obtención del tamaño del artefacto pues a pesar que el formato varíe, el contenido siempre va a ser el mismo.

Cantidad de páginas usuales:

Esta varía de acuerdo a los cambios que se hacen en el formato en cada experimento de este caso de estudio, asiéndola dependiente de el formato por lo que esta no se puede utilizar como una variable para el cálculo del tamaño del producto de pruebas.

Las variables de tamaño de letra, margen e interlineado, solo se grafican para verificar que por los cambios que estos sufren que es lo que le ocurre a las demás variables.

3.8.1.2. Experimento 1,2 y 3 con las medidas propuestas en la tesis.

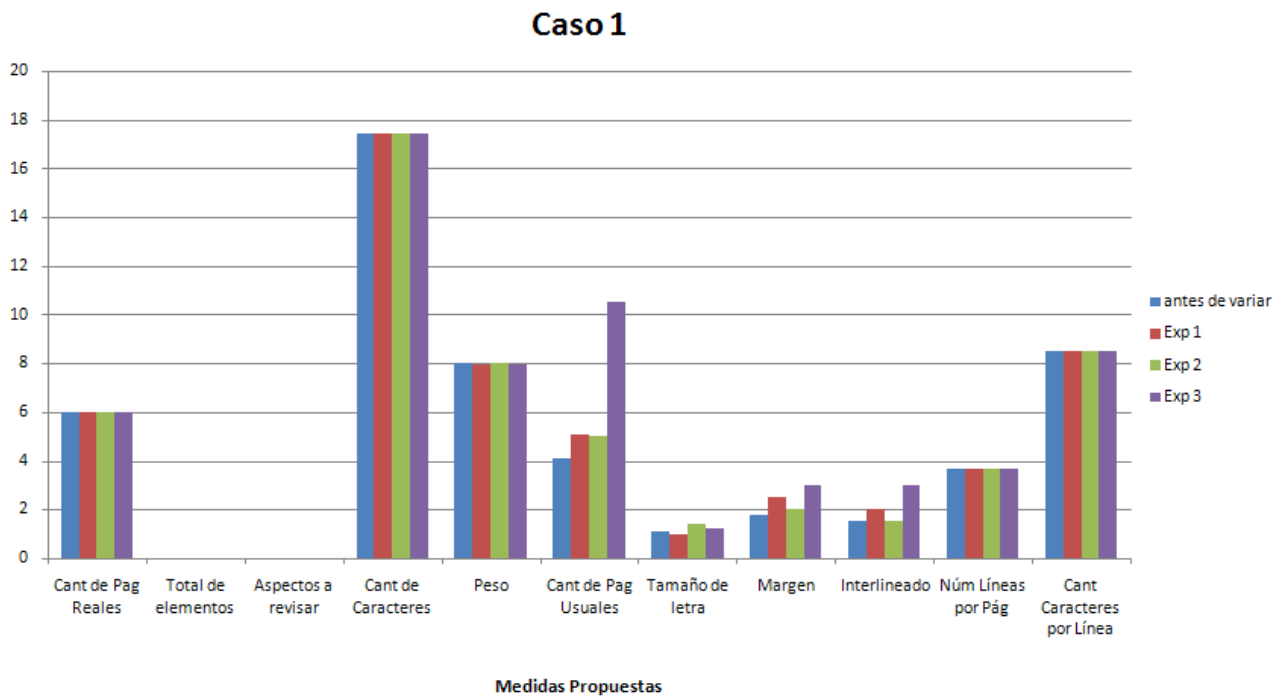


Figura 5. Comportamiento de las medidas propuestas en los experimentos del caso 1.

Análisis del resultado:

Como se refleja en la gráfica que se muestra en la figura 5, para las medidas propuestas en la tesis el formato del documento no influye en ninguno de los experimentos, pues la cantidad de caracteres, el número de líneas por páginas y la cantidad de caracteres por líneas son variables independientes del tamaño de letra, margen e interlineado que se utilice. Las variables independientes son las que se utilizan para llegar a la medida del documento, en este caso es la cantidad de páginas reales y en los experimentos representados este indicador no presenta variación, las variables usuales para determinar el tamaño del documento no se tienen en cuenta dentro de las medidas propuestas, pues son variables dependientes del formato.

3.8.1.3. Representación general de los experimentos 1,2 y 3 con las medidas usuales y las medidas propuestas en la tesis.

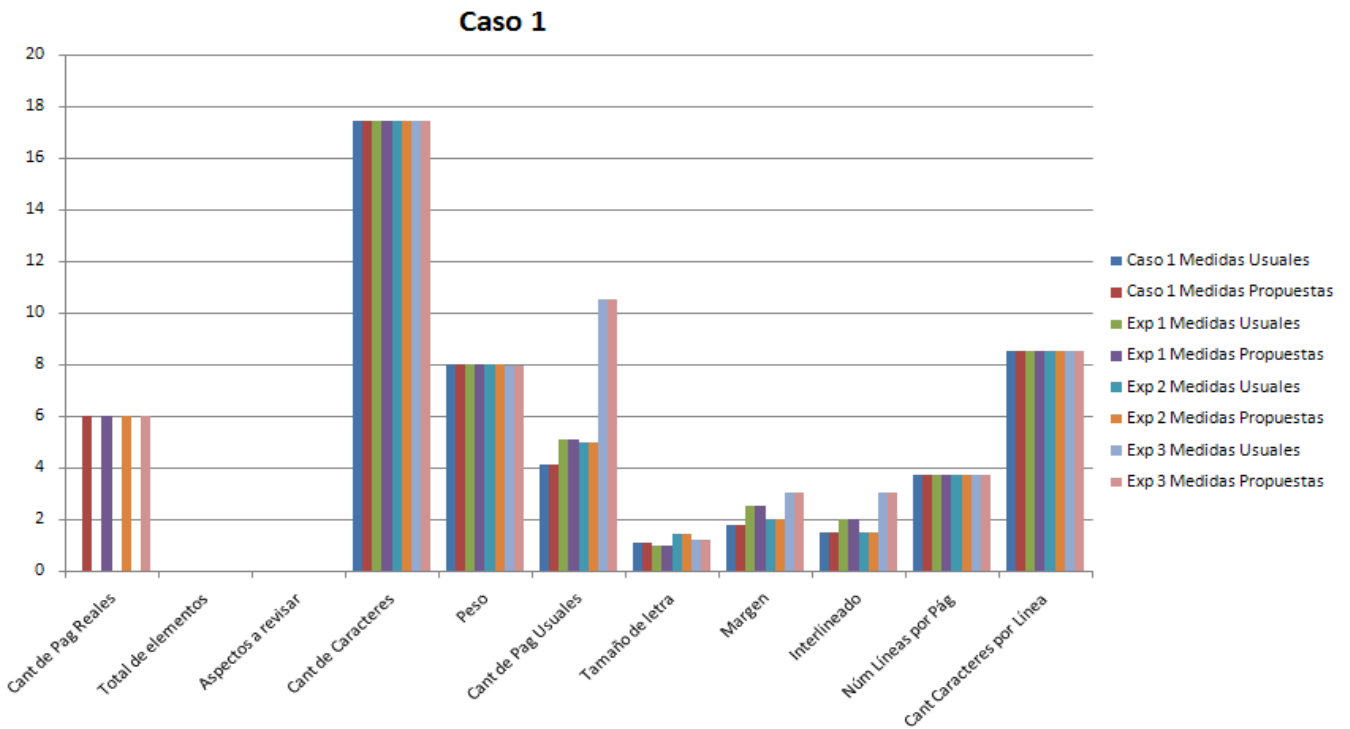


Figura 6. Comportamiento de las medidas usuales y las propuestas en los experimentos del caso 1.

En la figura 6 se muestran todas las variables, tanto las que son medidas usuales como las propuestas y su comportamiento en los experimentos 1, 2 y 3 del caso de estudio 1, el resultado de este comportamiento se analiza más detalladamente en los epígrafes 3.8.1.1 y 3.8.1.2.

Como resumen de este comportamiento se puede decir que las medidas usuales

3.8.2. Estudio del Caso 2.

3.8.2.1. Experimento 1 y 2 con las medidas usuales.

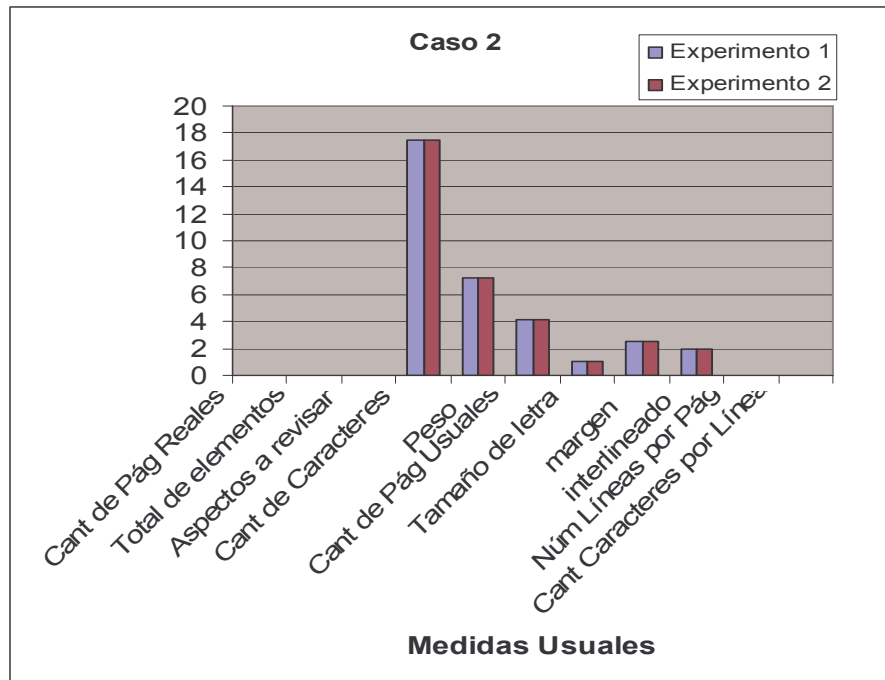


Figura 7. Comportamiento de las medidas usuales en los experimentos del caso 2

Análisis del resultado:

Después de graficar los experimentos 1 y 2 en el caso 2, para las medidas usuales, como se muestra en la figura 7, se puede observar lo siguiente:

Cantidad de caracteres:

Se observa que esta variable no tiene cambios ya que como se mencionaba anteriormente en el epígrafe 2.8.1.1, al analizar los resultados del caso de estudio 1 se determina que esta variable no va a depender de cambio que se le haga al parte del documento es decir que esta es una variable constante, además, como se puede ver, en este caso no se va a variar el formato solo se va a cambiar es la bibliografía la cual solo va a influir en las parte de los gráficos que el documento posee y se van a medir.

Peso y Cantidad de páginas usuales:

Estas variables, tanto la una como la otra, dependen del formato, en estos experimentos no ocurre variación alguna por lo cual estas van a permanecer constantes.

3.8.2.2. Experimento 1 y 2 con las medidas propuestas en la tesis.

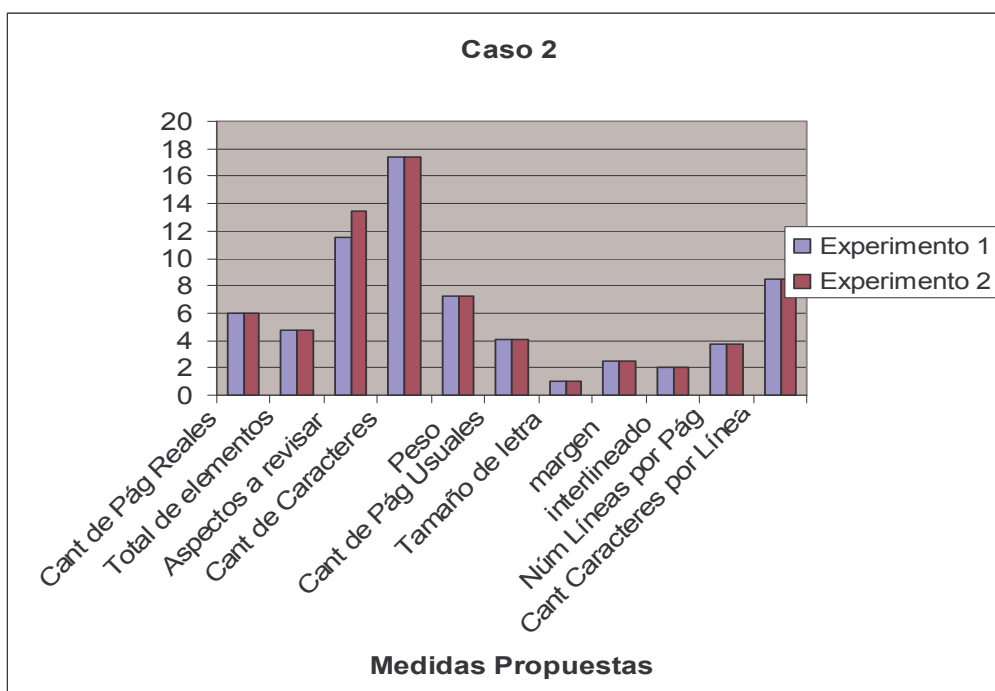


Figura 8. Comportamiento de las medidas propuestas en los experimentos del caso 1.

Análisis del resultado:

En este caso el tamaño del documento estará dado por dos indicadores:

Cantidad de páginas reales:

Esta es una variable que se obtiene de la relación de las variables independientes del formato que son: cantidad de caracteres, número de líneas por página y la cantidad de caracteres por líneas. Estas variables no van a variar en los dos experimentos por lo cual la cantidad de paginas reales se van a mantener constantes en los dos experimentos

Aspectos a revisar:

Este depende de las listas de chequeo que se utilicen para obtener las propiedades de los elementos que intervienen en los gráficos, en este caso existe variación en los aspectos a revisar debido a que se utilizan dos bibliografías diferentes para determinar las propiedades, la Ayuda del Rational, y la lista de chequeo del proyecto de Calidad de la Facultad 3.

Total de elementos a revisar:

Esta variable se va a mantener constante ya que solo va a variar las propiedades de estos elementos, por lo cual es el mismo en los dos experimentos.

Las variables usuales para determinar el tamaño del documento y de las gráficas no se tienen en cuenta dentro de las medidas propuestas, pues son variables dependientes del formato.

3.8.2.3. Representación general de los experimentos 1 y 2 con las medidas usuales y las medidas propuestas en la tesis.

En la figura 10 se muestran todas las variables, tanto las que son medidas usuales como las propuestas y su comportamiento en los experimentos 1y 2 del caso de estudio 2, el resultado de este comportamiento se analiza más detalladamente en los epígrafes 3.8.2.1 y 3.8.2.2.

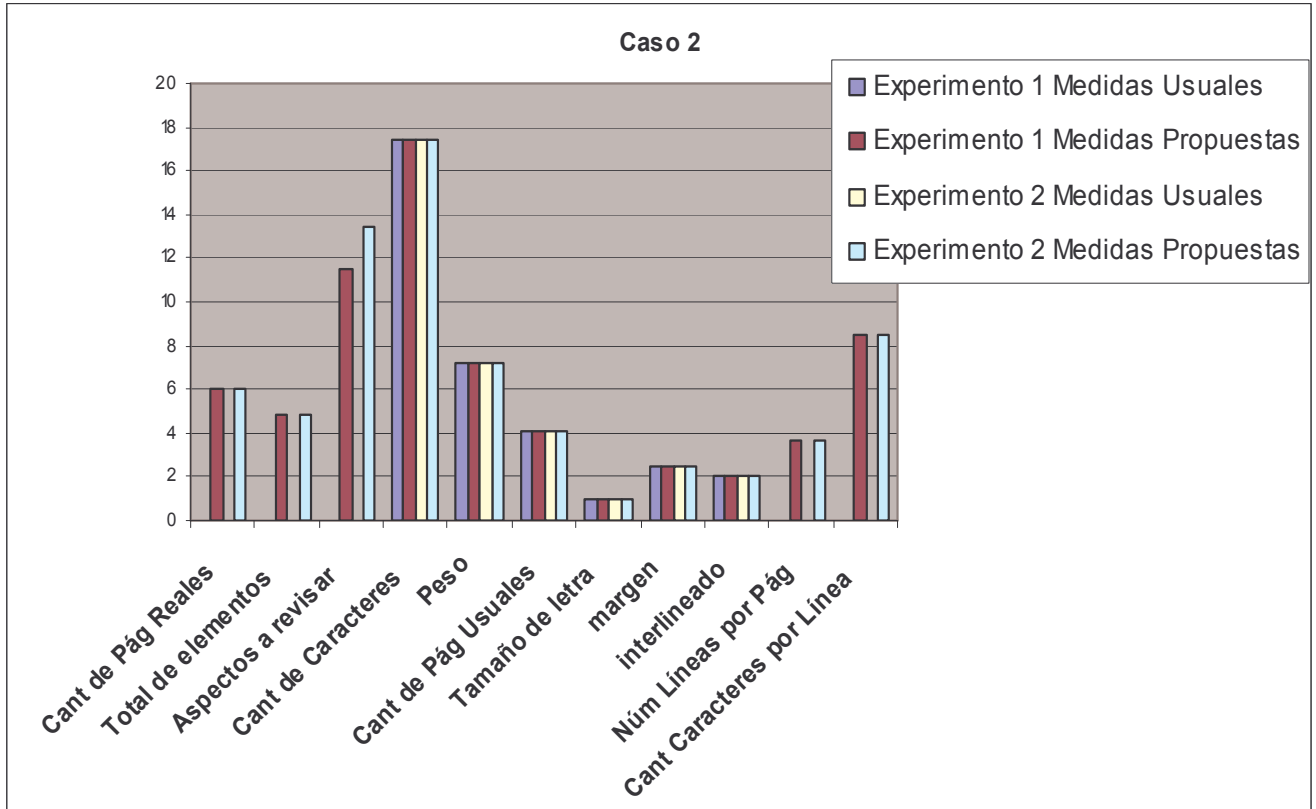


Figura 10. Comportamiento de las medidas usuales y las propuestas en los experimentos del caso 1.

3.9. Conclusiones.

La evaluación de los métodos propuestos para determinar el tamaño de los productos de prueba, fue realizada a través del Método de Casos de Estudio. Para el desarrollo de esta metodología se hizo el estudio de dos casos para los cuales fue seleccionada la plantilla de Modelos de Casos de Uso del Módulo AVL de Sistema de Gestión de Emergencias de Seguridad Ciudadana (171). Esta evaluación permite concluir que mediante los métodos propuestos se puede obtener con efectividad el tamaño de cada producto, garantía que no se puede tener con las medidas que se usan actualmente, por lo cual se contribuirá al desarrollo de las métricas de esfuerzo del Software en la Facultad 3 y en la UCI en general, cumpliéndose así el objetivo principal de la investigación.

CONCLUSIONES

Con el desarrollo del presente trabajo se da cumplimiento al principal objetivo del mismo que ha sido, elaborar métodos de medición para determinar el tamaño de los diferentes productos resultantes del desarrollo de Software ajustado a las necesidades de la Facultad 3, y a las tareas específicas propuestas en la investigación, las cuales se relacionan a continuación:

- Se selecciona y revisa la bibliografía correspondiente para actualizar los logros y limitaciones existentes sobre la medición del tamaño de un producto de pruebas, donde se analizó
- Se identificaron los principales autores y tendencias, enfatizando en los que más relación tienen con el tema.
- Se evalúa el contenido de la información obtenida.
- Se estudiaron, identificaron y clasificaron los diferentes productos de trabajo de un probador en la UCI.
- Se establece un diagnóstico de las tendencias actuales que se llevan a cabo en los proyectos sobre la medición del tamaño de un producto de software en la Universidad.
- Se aplicaron entrevistas a los directivos y especialistas en el tema.
- Se hizo un estudio de las propiedades que definen a las métricas y a los métodos, para la medición de los productos.
- Se realiza una propuesta de los métodos más efectivos para la medición del tamaño de un producto de pruebas.
- Se comprobaron y evaluaron los resultados obtenidos aplicando los métodos a un grupo de productos de prueba de estudio.

A modo de síntesis se puede decir que los métodos propuestos en la investigación permiten hacer estimaciones más exactas del tiempo y el esfuerzo que se debe utilizar para probar los productos resultantes del desarrollo del Software.

RECOMENDACIONES

- Se recomienda aplicar la propuesta de solución del presente trabajo, a todos los proyectos que realicen la fase de prueba.

BIBLIOGRAFÍA

"Definición de Medición." **Volume**, DOI:

Andrés, J. O. M. (2000). "ESTIMACIÓN DEL ESFUERZO EN EL DESARROLLO DE SOFTWARE."

Andrés, M. M. M. (2002) "Modelo Entidad-Relación." **Volume**, DOI:

C Génova (1995). "Tecnología de la información. Procesos del ciclo de vida del software."

Carrasco, O. M. F., D. G. León, et al. (1995) "Un enfoque actual sobre la calidad del software." **Volume**, DOI:

CEDEFOP (1998) "Gestion de la Calidad." **Volume**, DOI:

Dominguez, B. (2007) "Gestión de Proyectos." **Volume**, DOI:

Durán, M. R. (2007). "Mediciones Prácticas de Software y Sistemas (PSM): una propuesta para la producción de software en la UCI."

Fenton, N. E. y P., S.L (1997). Software metrics. A rigorous and practical approach, PWS Pub.

GARCÍA, F. (2007). "Proceso Software y Gestión del Conocimiento, Departamento de Tecnologías y Sistemas de Información. Escuela Superior de Informática."

GARCIA, M. (1998). "¿Quién controla la calidad de los software?"

GONZÁLEZ, D. (2001). Las Métricas de Software y su Uso en la Región. Mexico, Universidad de las Américas, Puebla.

GONZALEZ, I. (2004). Cuba se hará un sitio en el mercado del software. .

Gramajo, E., García-Martínez, R., Rossi, B., Claverie, E. y Britos, P. (2005). "COMBINACION DE ALTERNATIVAS PARA LA ESTIMACION DE PROYECTOS SOFTWARE."

Hurtado, L. L. (2006). "Modelos de Estimación".

HURTADO, L. L. (2007). La primera comunidad libre donde aprender y compartir.

- Inc, L. C. (1995). " Utilidades de los PUNTOS FUNCIÓN."
- investigativo, G. (2006). "¿Qué es la Gestión de Proyectos? ."
- Jamaica, C. d. F. d. H.-N. "Concepto de la calidad en la formacion professional."
- KAN., S. H. (2000). Metrics and Models in Software Quality Engineering.
- Kelvin, L. "Medición y Métricas del Software." **Volume**, DOI:
- Labdelaoui, H. (2005). "Métodos de Estimación de Tamaño Funcional Software Aplicación a Enfoques de desarrollo."
- MARTÍNEZ, R. D. (2006). ConfigCASE 3.0 Herramienta de apoyo a la Gestión de apoyo a la gestión de configuración. Propuesta Arquitectónica., Instituto Superior Politécnico "José Antonio Echeverría.
- Meza, N. Q., M. M. Peña, et al. "MÉTODOS DE MÉTRICAS DEL SOFTWARE ORIENTADAS A PUNTOS DE FUNCIÓN UTILIZADOS ACTUALMENTE EN EL MUNDO".
- Muñoz, C. C. (2006). "Calidad de Sistemas de Software."
- NACION, D. D. U. S. D. L. (2007). "METODOLOGÍA DEL ESTUDIO DE CASO ".
- PRESSMAN, R. S. (1993). Ingeniería del software. Un enfoque práctico. 3ª Edición.
- PRESSMAN, R. S. (1998). "Ingeniería del Software. Un Enfoque Práctico". .
- PRESSMAN, R. S. (2002). Ingeniería del Software: Un Enfoque Práctico. Quinta Edición McGraw-Hill.
- PRESSMAN, R. S. (2005). Capítulo 17: Técnicas de Prueba del Software. en: Ingeniería del Software. Quinta Edición. La Habana, Félix Varela.
- RUP (2003). Rational Unified Process.
- S.-Capuchino, A. M. M. "COCOMO II."
- Sanders, L. (2004) "Medición." **Volume**, DOI:

Sebastian, U. d. D. S. (2005). "EL MÉTODO DELPHI."

Software, D. C. d. I. d. (2007). FT Análisis y Diseño. Modelo de Diseño. Conferencia 1.

Tecnom maestros "Uso de Métricas en la Ingeniería de Software " **Volume**, DOI:

Topografía, E. d. (2005). "Medidas directas e indirectas."

UCI, D. C. d. I. Fase de Inicio. Flujo de Análisis y Diseño. Modelo de Análisis. Conferencia 6.

Vilas, A. F. (2001) "Diagrama de Despliegue." **Volume**, DOI:

Vilas, A. F. (2001) "Diagramas de Casos de Uso " **Volume**, DOI:

Zuse, H. (1998). A framework of software measurement.

ANEXOS

GLOSARIO

Ámbito del producto (Software): Identifica los datos primarios, funciones y comportamientos que caracterizan al producto, e intenta abordar estas características de una manera cuantitativa.

Artefactos: Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Calidad: 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 software desarrollado profesionalmente.

Calisoft: Centro de calidad del software de Cuba.

CLOC: Líneas de código con comentarios.

Diagrama de Casos de Uso: Es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones.

Diagrama de despliegue: Es una representación las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

Estimación: Consiste en determinar el valor de una variable desconocida a partir de otras conocidas, o de una pequeña cantidad de valores conocidos de esa misma variable.

Gestión de calidad: Es una estrategia organizativa y un método de gestión que hace participar a todos los empleados y pretende mejorar continuamente la eficacia de una organización en satisfacer el cliente.

Gestión de proyectos: Es un conjunto de técnicas, conocimientos, habilidades y herramientas encaminadas a planificar tareas que conduzcan a alcanzar los requisitos del proyecto.

KSLOC: Miles de líneas de código fuente.

LOC: Líneas de código.

Medición: Es el acto de determinar una medida.

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

Medidas directas: Son aquellas en donde el instrumento de medida mide directamente la magnitud desconocida.

Medidas indirectas: Son aquellas que se obtienen al aplicar a unas determinadas medidas directas una función matemática que relaciona la cantidad de interés con la magnitud desconocida.

Métodos: Conjunto de pasos fijados de antemano por una disciplina con el fin de alcanzar conocimientos válidos mediante instrumentos confiables.

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

Modelo del análisis: Se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas.

Modelo del diseño: Es un modelo de objetos que describe la realización de los Casos de Uso y al mismo tiempo constituye una abstracción del modelo de implementación y del código fuente, constituye una entrada esencial a las actividades de implementación y prueba.

Modelo Entidad-Relación: Está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas.

NLOC: Líneas de código sin comentarios.

Proceso de pruebas: Es donde se desarrollan todas las actividades que tienen que ver con la fase de pruebas dentro del Software como son: planificación de las pruebas, diseño de las pruebas, implementación de las pruebas, ejecución de las pruebas y evaluación de los resultados.

Producto: Cualquier Software que será construido a petición de otros.

Producto de pruebas: Son todos los entregables, generados después de finalizada la fase de Análisis & Diseño e Implementación de un proyecto, lo que permite pasar a la fase de las pruebas de liberación.

Producto de Software: Conjunto de programas de computadora, procedimientos y posiblemente documentación y datos asociados.

RUP: Proceso Unificado de Rational.

Software: Todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

Tamaño de un producto del Software: es un indicador de la amplitud y profundidad del conjunto de prestaciones que el producto incorpora.

UKSMA: Asociación de usuarios de la métrica del Software del Reino Unido.

